

1

---

2

# Spatial Capture-Recapture

3

J. Andrew Royle  
Richard B. Chandler  
Rahel Sollmann  
Beth Gardner

4

5

6

7

8

USGS Patuxent Wildlife Research Center  
North Carolina State University

9

ACADEMIC PRESS



# CONTENTS

10

11

---

12	<b>Foreword</b>	14
13	<b>Preface</b>	20
14	<b>Acknowledgments</b>	26
15	<b>I Background and Concepts</b>	1
16	<b>1 Introduction</b>	3
17	1.1 The Study of Populations by Capture-Recapture . . . . .	4
18	1.2 Lions and Tigers and Bears, Oh My: Genesis of Spatial Capture-Recapture Data . . . . .	5
19	1.2.1 Camera trapping . . . . .	5
20	1.2.2 DNA sampling . . . . .	6
21	1.2.3 Acoustic sampling . . . . .	7
22	1.2.4 Search-encounter methods . . . . .	7
23	1.3 Capture-Recapture for Modeling Encounter Probability . . . . .	8
24	1.3.1 Example: Fort Drum bear study . . . . .	9
25	1.3.2 Inadequacy of non-spatial capture-recapture . . . . .	12
26	1.4 Historical Context: A Brief Synopsis . . . . .	13
27	1.4.1 Buffering . . . . .	13
28	1.4.2 Temporary emigration . . . . .	14
29	1.5 Extension of Closed Population Models . . . . .	15
30	1.5.1 Towards spatial explicitness: Efford's formulation . . . . .	15
31	1.5.2 Abundance as the aggregation of a point process . . . . .	16
32	1.5.3 The activity center concept . . . . .	17
33	1.5.4 The state-space . . . . .	17
34	1.5.5 Abundance and density . . . . .	18
35	1.6 Characterization of SCR models . . . . .	18
36	1.7 Summary and Outlook . . . . .	19

38	<b>2 Statistical Models and SCR</b>	<b>21</b>
39	2.1 Random Variables and Probability Distributions . . . . .	22
40	2.1.1 Stochasticity in ecology . . . . .	22
41	2.1.2 Properties of probability distributions . . . . .	26
42	2.2 Common Probability Distributions . . . . .	28
43	2.2.1 The binomial distribution . . . . .	28
44	2.2.2 The Bernoulli distribution . . . . .	29
45	2.2.3 The multinomial and categorical distributions . . . . .	30
46	2.2.4 The Poisson distribution . . . . .	32
47	2.2.5 The uniform distribution . . . . .	32
48	2.2.6 Other distributions . . . . .	33
49	2.3 Statistical Inference and Parameter Estimation . . . . .	35
50	2.4 Joint, Marginal, and Conditional Distributions . . . . .	38
51	2.5 Hierarchical Models and Inference . . . . .	41
52	2.6 Characterization of SCR Models . . . . .	43
53	2.7 Summary and Outlook . . . . .	47
54	<b>3 GLMs and Bayesian Analysis</b>	<b>49</b>
55	3.1 GLMs and GLMMs . . . . .	50
56	3.2 Bayesian Analysis . . . . .	52
57	3.2.1 Bayes' rule . . . . .	52
58	3.2.2 Principles of Bayesian inference . . . . .	53
59	3.2.3 Prior distributions . . . . .	55
60	3.2.4 Posterior inference . . . . .	56
61	3.2.5 Small sample inference . . . . .	56
62	3.3 Characterizing Posterior Distributions by MCMC Simulation . . . . .	58
63	3.3.1 What goes on under the MCMC hood . . . . .	58
64	3.3.2 Rules for constructing full conditional distributions . . . . .	60
65	3.3.3 Metropolis-Hastings algorithm . . . . .	60
66	3.4 Bayesian Analysis Using the BUGS Language . . . . .	61
67	3.4.1 Linear regression in WinBUGS . . . . .	62
68	3.5 Practical Bayesian Analysis and MCMC . . . . .	64
69	3.5.1 Choice of prior distributions . . . . .	64
70	3.5.2 Convergence and so-forth . . . . .	66
71	3.5.3 Bayesian confidence intervals . . . . .	69
72	3.5.4 Estimating functions of parameters . . . . .	69
73	3.6 Poisson GLMs . . . . .	69
74	3.6.1 Example: Breeding Bird Survey data . . . . .	70
75	3.6.2 Doing it in WinBUGS . . . . .	72
76	3.6.3 Constructing your own MCMC algorithm . . . . .	73
77	3.7 Poisson GLM with Random Effects . . . . .	76
78	3.8 Binomial GLMs . . . . .	79
79	3.8.1 Binomial regression . . . . .	79

80	3.8.2 Example: waterfowl banding data . . . . .	80
81	3.9 Bayesian Model Checking and Selection . . . . .	82
82	3.9.1 Goodness-of-fit . . . . .	82
83	3.9.2 Model selection . . . . .	84
84	3.10 Summary and Outlook . . . . .	85
85	<b>4 Closed Population Models</b>	<b>87</b>
86	4.1 The Simplest Closed Population Model: Model $M_0$ . . . . .	88
87	4.1.1 The core capture-recapture assumptions . . . . .	90
88	4.1.2 Conditional likelihood . . . . .	91
89	4.2 Data Augmentation . . . . .	92
90	4.2.1 DA links occupancy models and closed population models .	92
91	4.2.2 Model $M_0$ in BUGS . . . . .	94
92	4.2.3 Formal development of data augmentation (DA) . . . . .	95
93	4.2.4 Remarks on data augmentation . . . . .	96
94	4.2.5 Example: Black bear study on Fort Drum . . . . .	98
95	4.3 Temporally Varying and Behavioral Effects . . . . .	101
96	4.4 Models with Individual Heterogeneity . . . . .	102
97	4.4.1 Analysis of model $M_h$ . . . . .	104
98	4.4.2 Analysis of the Fort Drum data with model $M_h$ . . . . .	105
99	4.4.3 Comparison with MLE . . . . .	106
100	4.5 Individual Covariate Models: Toward Spatial Capture-Recapture .	108
101	4.5.1 Example: Location of capture as a covariate . . . . .	110
102	4.5.2 Fort Drum bear study . . . . .	111
103	4.5.3 Extension of the model . . . . .	113
104	4.5.4 Invariance of density to $B$ . . . . .	116
105	4.5.5 Toward fully spatial capture-recapture models . . . . .	116
106	4.6 Distance Sampling: A Primitive SCR Model . . . . .	116
107	4.6.1 Example: Sonoran desert tortoise study . . . . .	119
108	4.7 Summary and Outlook . . . . .	121
109	<b>II Basic SCR Models</b>	<b>123</b>
110	<b>5 Fully Spatial Capture-Recapture Models</b>	<b>125</b>
111	5.1 Sampling Design and Data Structure . . . . .	126
112	5.2 The binomial observation model . . . . .	127
113	5.2.1 Definition of home range center . . . . .	129
114	5.2.2 Distance as a latent variable . . . . .	129
115	5.3 The Binomial Point Process Model . . . . .	129
116	5.3.1 The state-space of the point process . . . . .	131
117	5.3.2 Connection to model $M_h$ and distance sampling . . . . .	133
118	5.4 The Implied Model of Space Usage . . . . .	134

119	5.4.1	Bivariate normal case . . . . .	136
120	5.4.2	Empirical analysis . . . . .	136
121	5.4.3	Relevance of understanding space usage . . . . .	138
122	5.4.4	Contamination due to behavioral response . . . . .	138
123	5.5	Simulating SCR Data . . . . .	139
124	5.5.1	Formatting and manipulating real data sets . . . . .	140
125	5.6	Fitting Model SCR0 in BUGS . . . . .	141
126	5.7	Unknown N . . . . .	143
127	5.7.1	Analysis using data augmentation in WinBUGS . . . . .	145
128	5.7.2	Implied home range area . . . . .	147
129	5.7.3	Realized and expected density . . . . .	148
130	5.8	The Core SCR Assumptions . . . . .	150
131	5.9	Wolverine Camera Trapping Study . . . . .	151
132	5.9.1	Practical data organization . . . . .	151
133	5.9.2	Fitting the model in WinBUGS . . . . .	155
134	5.9.3	Summary of the wolverine analysis . . . . .	156
135	5.9.4	Wolverine space usage . . . . .	157
136	5.10	Using a Discrete Habitat Mask . . . . .	158
137	5.10.1	Evaluation of coarseness of habitat mask . . . . .	159
138	5.10.2	Analysis of the wolverine camera trapping data . . . . .	160
139	5.11	Summarizing Density and Activity Center Locations . . . . .	160
140	5.11.1	Constructing density maps . . . . .	161
141	5.11.2	Example: Wolverine density map . . . . .	164
142	5.11.3	Predicting where an individual lives . . . . .	165
143	5.12	Effective Sample Area . . . . .	165
144	5.13	Summary and Outlook . . . . .	167
145	<b>6</b>	<b>Likelihood Analysis of Spatial Capture-Recapture Models</b>	<b>173</b>
146	6.1	MLE with Known N . . . . .	174
147	6.1.1	Implementation (simulated data) . . . . .	175
148	6.2	MLE when N is Unknown . . . . .	179
149	6.2.1	Integrated likelihood under data augmentation . . . . .	182
150	6.2.2	Extensions . . . . .	183
151	6.3	Classical Model Selection and Assessment . . . . .	183
152	6.4	Likelihood Analysis of the Wolverine Camera Trapping Data . . . . .	184
153	6.4.1	Sensitivity to integration grid and state-space buffer . . . . .	185
154	6.4.2	Using a habitat mask (Restricted state-space) . . . . .	186
155	6.5	DENSITY and the R Package <b>secr</b> . . . . .	188
156	6.5.1	Encounter device types and detection models . . . . .	190
157	6.5.2	Analysis using the <b>secr</b> package . . . . .	191
158	6.5.3	Likelihood analysis in the <b>secr</b> package . . . . .	194
159	6.5.4	Multi-session models in <b>secr</b> . . . . .	195
160	6.5.5	Some additional capabilities of <b>secr</b> . . . . .	196

---

161	6.6 Summary and Outlook . . . . .	198
162	<b>7 Modeling Variation In Encounter Probability</b>	<b>201</b>
163	7.1 Encounter Probability Models . . . . .	202
164	7.1.1 Bayesian analysis with <code>bear.JAGS</code> . . . . .	204
165	7.1.2 Bayesian analysis of encounter probability models . . . . .	204
166	7.2 Modeling Covariate Effects . . . . .	206
167	7.2.1 Date and time . . . . .	208
168	7.2.2 Trap-specific covariates . . . . .	210
169	7.2.3 Behavior or trap response by individual . . . . .	210
170	7.2.4 Individual covariates . . . . .	212
171	7.3 Individual Heterogeneity . . . . .	215
172	7.3.1 Models of heterogeneity . . . . .	215
173	7.3.2 Heterogeneity induced by variation in home range size . . . . .	216
174	7.4 Likelihood Analysis in <code>secr</code> . . . . .	217
175	7.4.1 Notes for fitting standard models . . . . .	218
176	7.4.2 Sex effects . . . . .	218
177	7.4.3 Individual heterogeneity . . . . .	220
178	7.4.4 Model selection in <code>secr</code> using AIC . . . . .	220
179	7.5 Summary and Outlook . . . . .	220
180	<b>8 Model Selection and Assessment</b>	<b>223</b>
181	8.1 Model Selection by AIC . . . . .	224
182	8.1.1 AIC analysis of the wolverine data . . . . .	224
183	8.2 Bayesian Model Selection . . . . .	228
184	8.2.1 Model selection by DIC . . . . .	228
185	8.2.2 DIC analysis of the wolverine data . . . . .	229
186	8.2.3 Bayesian model averaging with indicator variables . . . . .	230
187	8.2.4 Choosing among detection functions . . . . .	234
188	8.3 Evaluating Goodness-of-Fit . . . . .	235
189	8.4 The Two Components of Model Fit . . . . .	237
190	8.4.1 Testing uniformity or spatial randomness . . . . .	237
191	8.4.2 Assessing fit of the observation model . . . . .	240
192	8.4.3 Does the SCR model fit the wolverine data? . . . . .	241
193	8.5 Quantifying Lack-of-fit and Remediation . . . . .	244
194	8.6 Summary and Outlook . . . . .	245
195	<b>9 Alternative Observation Models</b>	<b>247</b>
196	9.1 Poisson Observation Model . . . . .	248
197	9.1.1 Poisson model of space usage . . . . .	249
198	9.1.2 Poisson relationship to the Bernoulli model . . . . .	250
199	9.1.3 A cautionary note on modeling encounter frequencies . . . . .	251
200	9.1.4 Analysis of the Poisson SCR model in BUGS . . . . .	252

201	9.1.5	Simulating data and fitting the model . . . . .	253
202	9.1.6	Analysis of the wolverine study data . . . . .	255
203	9.1.7	Count detector models in the <b>secr</b> package . . . . .	256
204	9.2	Independent Multinomial Observations . . . . .	256
205	9.2.1	Multinomial resource selection models . . . . .	258
206	9.2.2	Simulating data and analysis using JAGS . . . . .	258
207	9.2.3	Multinomial relationship to the Poisson . . . . .	262
208	9.2.4	Avian mist-netting example . . . . .	262
209	9.3	Single-catch traps . . . . .	269
210	9.3.1	Inference for single-catch systems . . . . .	270
211	9.3.2	Analysis of Efford's possum trapping data . . . . .	270
212	9.4	Acoustic sampling . . . . .	273
213	9.4.1	The signal strength model . . . . .	274
214	9.4.2	Implementation in <b>secr</b> . . . . .	276
215	9.4.3	Implementation in <b>BUGS</b> . . . . .	276
216	9.4.4	Other types of acoustic data . . . . .	277
217	9.5	Summary and Outlook . . . . .	277
218	<b>10</b>	<b>Sampling Design</b>	<b>279</b>
219	10.1	General Considerations . . . . .	280
220	10.1.1	Model-based not design-based . . . . .	280
221	10.1.2	Sampling space or sampling individuals? . . . . .	281
222	10.1.3	Focal population vs. state-space . . . . .	282
223	10.2	Study design for (spatial) capture-recapture . . . . .	283
224	10.3	Trap spacing and array size relative to animal movement . . . . .	285
225	10.3.1	Black bears from Pictured Rocks National Lakeshore . . . . .	287
226	10.4	Sampling Over Large Areas . . . . .	288
227	10.5	Model-Based Spatial Design . . . . .	290
228	10.5.1	Statement of the design problem . . . . .	291
229	10.5.2	Model-based Design for SCR . . . . .	293
230	10.5.3	An Optimal Design Criterion for SCR . . . . .	294
231	10.5.4	Too much math for a Sunday afternoon . . . . .	295
232	10.5.5	Optimization of the criterion . . . . .	297
233	10.5.6	Illustration . . . . .	298
234	10.5.7	Density covariate models . . . . .	299
235	10.6	Temporal aspects of study design . . . . .	299
236	10.6.1	Total sampling duration and population closure . . . . .	300
237	10.6.2	Diagnosing and dealing with lack of closure . . . . .	301
238	10.7	Summary and Outlook . . . . .	302

---

239	<b>III Advanced SCR Models</b>	<b>305</b>
240	<b>11 Modeling Spatial Variation in Density</b>	<b>307</b>
241	11.1 Homogeneous point process revisited . . . . .	308
242	11.2 Inhomogeneous point processes . . . . .	311
243	11.3 Observed Point Processes . . . . .	314
244	11.4 Fitting inhomogeneous point process SCR models . . . . .	317
245	11.4.1 Continuous space . . . . .	317
246	11.4.2 Discrete space . . . . .	319
247	11.5 Argentina Jaguar Study . . . . .	322
248	11.6 Summary and Outlook . . . . .	325
249	<b>12 Modeling Landscape Connectivity</b>	<b>329</b>
250	12.1 Shortcomings of Euclidean Distance Models . . . . .	330
251	12.2 Least-Cost Path Distance . . . . .	331
252	12.2.1 Example of Computing Cost-weighted distance . . . . .	332
253	12.3 Simulating SCR Data using Ecological Distance . . . . .	335
254	12.4 Likelihood Analysis of Ecological Distance Models . . . . .	338
255	12.4.1 Example of SCR with least-cost path . . . . .	339
256	12.5 Bayesian Analysis . . . . .	339
257	12.6 Simulation Evaluation of the MLE . . . . .	340
258	12.7 Distance In an Irregular Patch . . . . .	341
259	12.7.1 Basic Geographic Analysis in R . . . . .	341
260	12.8 Ecological Distance and Density Covariates . . . . .	345
261	12.9 Summary and Outlook . . . . .	346
262	<b>13 Integrating Resource Selection with Spatial Capture-Recapture Models</b>	<b>349</b>
263	13.1 A Model of Space Usage . . . . .	350
264	13.1.1 A simulated example . . . . .	352
265	13.1.2 Poisson model of space use . . . . .	353
266	13.2 Integrating Capture-Recapture Data . . . . .	355
267	13.3 SW New York Black Bear Study . . . . .	356
268	13.4 Simulation Study . . . . .	358
269	13.5 Relevance and Relaxation of Assumptions . . . . .	362
270	13.6 Summary and Outlook . . . . .	362
271		
272	<b>14 Stratified Populations: Multi-session and Multi-site Data</b>	<b>365</b>
273	14.1 Stratified Data Structure . . . . .	366
274	14.2 Multinomial Abundance Models . . . . .	367
275	14.2.1 Implementation in BUGS . . . . .	368
276	14.2.2 Groups with no individuals observed . . . . .	369
277	14.2.3 The group-means model . . . . .	369

278	14.2.4 Simulating stratified capture-recapture data . . . . .	371
279	14.3 Other Approaches to Multi-Session Models . . . . .	371
280	14.4 Application to Spatial Capture-Recapture . . . . .	372
281	14.4.1 Multinomial (“multi-catch”) observations . . . . .	372
282	14.4.2 Reanalysis of the Ovenbird data . . . . .	374
283	14.5 Spatial or Temporal Dependence . . . . .	376
284	14.6 Summary and Outlook . . . . .	377
285	<b>15 Models for Search-Encounter Data</b>	<b>379</b>
286	15.1 Search-encounter designs . . . . .	380
287	15.1.1 Design 1: Fixed Search Path . . . . .	380
288	15.1.2 Design 2: Uniform search intensity . . . . .	380
289	15.2 A Model for Fixed Search Path Data . . . . .	382
290	15.2.1 Modeling total hazard to encounter . . . . .	382
291	15.2.2 Modeling movement outcomes . . . . .	384
292	15.2.3 Simulation and analysis in <b>JAGS</b> . . . . .	384
293	15.2.4 Hard plot boundaries . . . . .	388
294	15.2.5 Analysis of other protocols . . . . .	388
295	15.3 Unstructured Spatial Surveys . . . . .	389
296	15.3.1 Mountain lions in Montana . . . . .	389
297	15.3.2 Sierra National Forest Fisher Study . . . . .	390
298	15.4 Design 2: Uniform Search Intensity . . . . .	390
299	15.4.1 Alternative movement models . . . . .	392
300	15.4.2 Simulating and fitting uniform search models . . . . .	393
301	15.4.3 Movement and Dispersal in Open Populations . . . . .	395
302	15.5 Partial Information Designs . . . . .	396
303	15.6 Summary and Outlook . . . . .	396
304	<b>16 Open Population Models</b>	<b>399</b>
305	16.1 Introduction . . . . .	399
306	16.1.1 Brief overview of population dynamics . . . . .	400
307	16.1.2 Animal movement related to population demography . . . . .	401
308	16.2 Jolly-Seber Models . . . . .	402
309	16.2.1 Traditional Jolly-Seber models . . . . .	402
310	16.2.2 Data augmentation for the Jolly-Seber model . . . . .	404
311	16.2.3 Spatial Jolly-Seber models . . . . .	408
312	16.3 Cormack-Jolly-Seber models . . . . .	413
313	16.3.1 Traditional CJS models . . . . .	413
314	16.3.2 Multi-state CJS models . . . . .	415
315	16.3.3 Spatial CJS models . . . . .	420
316	16.4 Modeling movement and dispersal dynamics . . . . .	423
317	16.4.1 Thoughts on movement of American shad . . . . .	424
318	16.4.2 Modeling dispersal . . . . .	424

---

319	16.5 Summary and Outlook . . . . .	425
320	<b>IV Super-Advanced SCR Models</b>	<b>427</b>
321	<b>17 Developing Markov Chain Monte Carlo Samplers</b>	<b>429</b>
322	17.1 Why Build Your Own MCMC Algorithm? . . . . .	429
323	17.2 MCMC and Posterior Distributions . . . . .	430
324	17.3 Types of MCMC Sampling . . . . .	432
325	17.3.1 Gibbs sampling . . . . .	432
326	17.3.2 Metropolis-Hastings sampling . . . . .	436
327	17.3.3 Metropolis-within-Gibbs . . . . .	440
328	17.3.4 Rejection sampling and slice sampling . . . . .	444
329	17.4 MCMC for Closed Capture-Recapture Model $M_h$ . . . . .	445
330	17.5 MCMC Algorithm for Model SCR0 . . . . .	448
331	17.5.1 SCR model with binomial encounter process . . . . .	450
332	17.6 Looking at Model Output . . . . .	451
333	17.6.1 Markov chain time series plots . . . . .	452
334	17.6.2 Posterior density plots . . . . .	452
335	17.6.3 Serial autocorrelation and effective sample size . . . . .	454
336	17.6.4 Summary results . . . . .	455
337	17.6.5 Other useful commands . . . . .	456
338	17.7 Manipulating the State-Space . . . . .	457
339	17.8 Increasing Computational Speed . . . . .	460
340	17.8.1 Parallel computing . . . . .	461
341	17.8.2 Using C++ . . . . .	463
342	17.9 Summary and Outlook . . . . .	465
343	<b>18 Unmarked Populations</b>	<b>469</b>
344	18.1 Existing Models for Inference About Density in Unmarked Populations	470
345	18.2 Spatial Correlation in Count Data . . . . .	472
346	18.2.1 Spatial correlation as information . . . . .	472
347	18.2.2 Two types of spatial correlation . . . . .	472
348	18.3 Spatial Count Model . . . . .	474
349	18.3.1 Data . . . . .	474
350	18.3.2 Model . . . . .	474
351	18.4 How Much Correlation Is Enough? . . . . .	476
352	18.4.1 On $N$ being unknown . . . . .	476
353	18.5 Simulation Example . . . . .	477
354	18.6 The Maryland Northern Parula Study . . . . .	482
355	18.7 Improving Precision . . . . .	483
356	18.8 Extensions of the Spatial Count Model . . . . .	484
357	18.9 The Maryland Dusky Salamander Study . . . . .	485

358	18.10 Summary and Outlook . . . . .	487
359	<b>19 Spatial Mark-Resight Models</b>	<b>493</b>
360	19.1 Background . . . . .	494
361	19.1.1 Resighting techniques . . . . .	494
362	19.1.2 Types of mark-resighting data . . . . .	495
363	19.1.3 A short history of mark-resight models . . . . .	496
364	19.1.4 The random sample assumption . . . . .	498
365	19.2 Known number of marked individuals . . . . .	500
366	19.2.1 Implementing spatial mark-resight models . . . . .	501
367	19.3 Unknown number of marked individuals . . . . .	504
368	19.3.1 Canada geese in North Carolina . . . . .	505
369	19.4 Imperfect identification of marked individuals . . . . .	507
370	19.5 How much information do marked and unmarked individuals contribute? . . . . .	509
371	19.6 Incorporating telemetry data . . . . .	512
372	19.6.1 Raccoons on the Outer Banks of North Carolina . . . . .	513
373	19.7 Marked animals are not a random sample from the state-space . . . . .	516
374	19.7.1 Marked animals are uniformly distributed in a smaller area . . . . .	516
375	19.7.2 Combining marking and resighting models . . . . .	518
376	19.8 Summary and Outlook . . . . .	518
377		
378	<b>20 2012: A Spatial Capture-Recapture Odyssey</b>	<b>521</b>
379	20.1 The Growth of Spatial Capture-Recapture . . . . .	523
380	20.2 Emerging Topics . . . . .	525
381	20.2.1 Weasels . . . . .	525
382	20.2.2 Inhomogeneous Point Processes . . . . .	525
383	20.2.3 Combining data from different surveys . . . . .	525
384	20.2.4 Imperfect identification of individuals . . . . .	527
385	20.2.5 Gregarious species . . . . .	528
386	20.2.6 Design . . . . .	528
387	20.2.7 Acoustic models . . . . .	528
388	20.2.8 Single Catch Traps . . . . .	529
389	20.2.9 Model Fit and Selection . . . . .	529
390	20.2.10 Dynamics . . . . .	529
391	20.2.11 Misc. Topics . . . . .	529
392	20.3 The Future of SCR . . . . .	530
393		
394	<b>V Appendices</b>	<b>533</b>
395	20.4 WinBUGS . . . . .	535
396	20.4.1 WinBUGS through R . . . . .	535
396	20.5 OpenBUGS . . . . .	536

397	20.5.1 OpenBUGS through R . . . . .	536
398	20.6 JAGS . . . . .	536
399	20.6.1 JAGS through R . . . . .	537
400	20.7 R . . . . .	538
401	20.7.1 R packages . . . . .	538
402	<b>Bibliography</b>	<b>541</b>

---

403 Foreword

404

405 In the early 1990's, Ullas Karanth asked my advice on estimating tiger density  
406 from camera-trap data. Historic uses of camera traps had been restricted to wildlife  
407 photography and the documentation of species presence. Ullas had the innovative  
408 idea to extend these uses to inference about tiger population size, density and  
409 even survival and movement by exploiting the individual markings of tigers. I had  
410 worked on development and application of capture-recapture models, so we began  
411 a collaboration that focused on population inferences based on detection histories  
412 of marked tigers. Early on in this work, we had to consider how to deal with two  
413 problems associated with the spatial distributions of both animals and traps.



**Figure 1.** Jim Nichols (left) discussing capture-recapture with K. Ullas Karanth and Andy Royle at Patuxent Wildlife Research Center, Oct. 15, 2007.

414 The first problem was that of heterogeneous capture probabilities among animals  
415 resulting from the positions of their ranges relative to trap locations. Animals with  
416 ranges centered in the middle of a trapping array are much more likely to encounter  
417 traps and be captured than animals with range centers just outside the trapping

418 array. Ad hoc abundance estimators were available to deal with such heterogeneity,  
419 and we resolved to rely primarily on such estimators for our work.

420 Ullas was more interested in tiger density (defined loosely as animals per unit  
421 area) than in abundance, and the second problem resulted from our need to trans-  
422 late abundance estimates into estimates of density. This translation required in-  
423 ference about the total area sampled, that is the area containing animals exposed  
424 to sampling efforts. In the case of fixed sampling devices such as traps and cam-  
425 eras, the area sampled is certainly greater than the area covered by the devices  
426 themselves (e.g., as defined by the area of the convex hull around the array of  
427 devices), but how do we estimate this area? This problem had been recognized and  
428 considered since the 1930's, and ad hoc approaches to solving it included nested  
429 grids, assessment lines, trapping webs and use of movement information from either  
430 animal recaptures or radiotelemetry data. We selected an approach using distances  
431 between captures of animals.

432 We thus recognized these two problems caused by spatial distribution of animals  
433 and traps, and we selected approaches to deal with them as best we could. We were  
434 well aware of the ad hoc nature of our pragmatic solutions. In particular, we viewed  
435 the use of movement information based on recaptures to translate our abundance  
436 estimates into density estimates as the weak link in our approach to inference about  
437 density.

438 In the early 2000's, Murray Efford developed a novel approach to inference about  
439 animal density based on capture-recapture data. The manuscript on this work was  
440 rejected initially by a top ecological journal without review (an interesting comment  
441 on the response of our peer review system to innovation), but was published in  
442 Oikos in 2004. The approach was anchored in a conceptual model of the trapping  
443 process in which an animal's probability of being captured in any particular trap  
444 was a decreasing function of the distance between the animal's home range center  
445 and the trap. This assumed relationship was very similar to the key relationship  
446 on which distance sampling methods are based. Efford viewed the distribution  
447 of animal range centers as being governed by a spatial point process, and the  
448 target of estimation was the intensity of this process, equivalent to animal density  
449 in the study area. Efford (2004) initially used an ad hoc approach to inference  
450 based on inverse prediction. He later teamed with David Borchers to develop a  
451 formal likelihood approach to estimation (Borchers and Efford 2008 and subsequent  
452 papers).

453 At about the same time that Efford was formalizing his approach, yet inde-  
454 pendently of that work, Andy Royle developed a similar approach for the related  
455 problem of density estimation based on locations of captures of animals obtained  
456 during active searches of prescribed areas (as opposed to captures in traps with  
457 fixed locations). Andy approached the inference problem using explicit hierarchical  
458 models with both a process component (the spatial distribution of animal range  
459 centers and a probability distribution reflecting movement about those centers)  
460 and an observation component (non-zero capture probability for locations within

461 the surveyed area and 0 outside this area). He used the data augmentation ap-  
462 proach that he had just developed (Royle et al., 2007) to deal with animals in the  
463 population that are never captured, and he implemented the model using Markov  
464 chain Monte Carlo sampling (Royle and Young, 2008). Ullas and I asked Andy for  
465 help (Fig. 1) with inference about tiger densities, and he extended his approach  
466 to deal with fixed trap locations by modeling detection probability as a function  
467 of the distance between range center and trap, thus solving our two fundamental  
468 problems emanating from spatial distributions of animals and traps (Royle et al.,  
469 2009b,a).

470 The preceding narrative about the solution of two inference problems faced by  
471 Ullas Karanth and me was presented to motivate interest in the models that are  
472 the subject of *Spatial Capture-Recapture*. SCR models provide a formal solution  
473 to the problem of heterogeneous capture probabilities associated with locations of  
474 animal ranges relative to trap locations. They also provide a formal and direct (as  
475 opposed to ad hoc and indirect) means of estimating density, naturally defined for  
476 SCR models as number of range centers per unit area. This motivation is perhaps  
477 adequate, but it is certainly incomplete. As noted in this book's introduction, SCR  
478 models should not be viewed simply as extensions of standard capture-recapture  
479 models designed to solve specific spatial problems. Rather, SCR models represent  
480 a much more profound development, dealing explicitly with ecological processes as-  
481 sociated with animal locations and movement as well as with the spatial aspects of  
482 sampling natural populations. They provide improvements over standard capture-  
483 recapture models in our abilities to address questions about demographic state  
484 variables (density, abundance) and processes (survival, recruitment), and they pro-  
485 vide new possibilities for addressing questions about spatial organization and space  
486 use by animals.

487 As the promise of SCR models has become recognized, work on them has pro-  
488 liferated over the last five years, with substantive new developments led in part  
489 by the authors of this book, Andy Royle, Richard Chandler, Rahel Sollmann and  
490 Beth Gardner. Because of this explosive development, it is no longer possible to  
491 consult one or two key papers in order learn about SCR. Royle and colleagues  
492 recognized the need for a synthetic treatment to integrate this work and place it  
493 within a common framework. They wrote *Spatial Capture-Recapture* in order to fill  
494 this need.

495 The history of methodological development in quantitative ecology contains  
496 numerous examples of synthetic books and monographs that have been extremely  
497 influential in advancing the use of improved inference procedures. *Spatial Capture-*  
498 *Recapture* will become a part of this history, serving as a catalyst for use and  
499 further development of SCR methods. The writing style is geared to a biological  
500 readership such that this book will provide a single source for biologists interested  
501 in learning about SCR models. The statistical development is sufficiently rigorous  
502 and complete that this synthesis of existing work should serve as a springboard for  
503 statisticians interested in extensions and new developments. I believe that Spatial

504 Capture-Recapture will be an extremely important book.

505 *Spatial Capture-Recapture* is organized around four major sections (plus appendices). The first, “Background and Concepts”, provides motivation for SCRs and a history of relevant concepts and modeling. Two chapters are devoted to statistical background, one including material introducing random variables, common probability distributions, and hierarchical models. The second chapter on statistical background develops the concept of SCRs as generalized linear mixed models, with some emphasis on Bayesian inference methods for such models. Also included in this section is a chapter on standard (non-spatial) capture-recapture models for closed populations. This chapter helps motivate SCRs and introduces the idea of data augmentation as an approach to dealing with zero-inflated models for inference about abundance. The authors develop a primitive SCR model in this chapter by noting that location data for captured animals can be viewed as individual covariates.

518 The second major section, “Basic SCR Models”, begins with a complete development of SCRs as hierarchical models with observation and spatial point process components. Included is a clear discussion of space use by animals, important because any model of the detection process implies a model for space use. A chapter is devoted to likelihood analysis of SCR models including both model development and an introduction to software available for fitting models. Another chapter is devoted to various approaches to modeling variation in encounter probability. A variety of basic models is introduced, as well as approaches to modeling covariates associated with traps, time, individual capture history, and individual animals (e.g., sex, body mass, random effects models as well). The chapter on model selection and assessment does not provide an omnibus, one-size-fits-all statistic. Rather, it describes useful approaches including AIC for likelihood analyses and both DIC and the Kuo and Mallick (1998) indicator variable approach for Bayesian analyses. For assessing model adequacy, they use the Bayesian p-value approach (Gelman et al., 519 1996) applied to different components of model fit. Another chapter is devoted to the encounter process which requires attention to the nature of the detection device (e.g., can an animal be caught only once or multiple times during an occasion, do traps permit catches of multiple or only single individuals, can an individual be detected multiple times by the same device) and the kinds of data produced by these devices. The final chapter in this section deals with the important topic of study design. A fundamental design trade-off involves the competing needs to capture 520 a good number of animals (sample size) and to attain a reasonably high average capture probability, and the authors emphasize the need for designs that represent a good compromise rather than those that emphasize one component to the exclusion of the other. General recommendations about trap spacing and clustering, and 521 use of ancillary data (telemetry) are discussed as well. The material in this section 522 is extremely important in conveying the basic principles underlying SCR modeling 523 and, as such, will be the section of primary interest to many readers.

524 The next section, “Advanced SCR Models”, will be of great interest to ecolo-

547 gists, not just because of the advanced model structures presented, but because of  
548 the ecological questions that become accessible using these methods. For example,  
549 the authors show how spatial variation in density can be modeled as a function  
550 of spatial covariates associated with all locations in the state space. Similarly, the  
551 authors relax the assumption of basic SCR models that encounter probability is a  
552 function of Euclidean distance between range center and trap, and focus instead  
553 on the “least cost path” between the range center and trap. The least cost path  
554 concept is modeled by including resistance parameters related to habitat covariates,  
555 and is relevant to the ecological concepts of connectivity and variable space use.  
556 The authors note ecological interest in resource selection functions, which focus  
557 on animal use of space as a function of specific resource or habitat covariates and  
558 which are typically informed by radiotelemetry data. They present a framework  
559 for development of joint models that combine SCR and resource selection function  
560 telemetry data. In some situations, sampling is done via a search encounter process  
561 rather than using detection devices with fixed locations, and SCR models are ex-  
562 tended to deal with these. Models are developed for combining data from sampling  
563 at multiple sites or across multiple occasions. The extension of the SCR framework  
564 to models for open populations permits inference about the processes of survival,  
565 recruitment and movement. Inference about time-specific changes in space use is  
566 also directly accessible using this approach, and I anticipate a great many advances  
567 in the development and application of open population SCR models.

568 The final section, “Super-Advanced SCR Models”, includes a technical chapter  
569 on development of MCMC samplers for the primary purpose of providing increased  
570 flexibility in SCR modeling. A chapter of huge potential importance introduces  
571 SCR models for unmarked populations, relying on the spatial correlation structure  
572 of resulting count data to draw inferences about animal distribution and density.  
573 These models will see widespread use in studies employing remote detection devices  
574 (camera traps, acoustic detectors) to sample animals that do not happen to have  
575 individually recognizable visual patterns or acoustic signatures. In many sampling  
576 situations, some animals will be individually identifiable and many will not, and  
577 the authors develop mark-resight models to combine detection data from these two  
578 classes of animals. The final chapter provides a glimpse of the future by pointing  
579 to a sample of neat developments that should be possible using the conceptual  
580 framework provided by SCR models.

581 I very much like the writing style of the authors and found the book relatively  
582 easy to read (there were exceptions), with clear presentations of important ideas.  
583 Most models are illustrated nicely with actual examples and corresponding sample  
584 computer code (frequently WinBUGS).

585 In summary, I repeat my claim that *Spatial Capture-Recapture* is an extremely  
586 important and useful book. A thorough read of the section on basic SCR models  
587 provides a good understanding of exactly how these models are constructed and  
588 how they “work” in terms of underlying rationale. The two sections on advanced  
589 SCR models present a thorough account of the current state of the art written by

590 those who have largely defined this state. As an ecologist, I found myself thinking  
591 of one potential application of these models after another. These methods will free  
592 ecologists to begin to think more clearly about interesting questions concerning the  
593 statics and dynamics of space use by animals. The ability to draw inferences about  
594 distribution and density of animals based on counts of unmarked individuals using  
595 remote detection devices has the potential to revolutionize conservation monitoring  
596 programs.

597 So does *Spatial Capture-Recapture* solve the inference problems encountered by  
598 Ullas Karanth and me two decades ago? You bet. But it does so much more than  
599 that. Andy, Richard, Rahel and Beth, thanks for an exceptional contribution.

600 James D. Nichols  
601 Patuxent Wildlife Research Center  
602 March 14, 2013

605 Capture-recapture (CR) models have been around for well over a century, and  
606 in that time they have served as the primary means of estimating population size  
607 and demographic parameters in ecological research. The development of these  
608 methods has never ceased, and each year new and useful extensions are presented  
609 in ecological and statistical journals. The seemingly steady clip of development was  
610 recently punctuated with the introduction of spatial capture recapture (SCR; a.k.a.  
611 spatially explicit capture-recapture models, or SECR) models, which in our view  
612 stand to revolutionize the study of animals populations. The importance of this  
613 new class of models is rooted in the fact that they acknowledge that both ecological  
614 processes and observation processes are inherently spatial. The purpose of this book  
615 is to explain this statement, and to bring together all of the developments over the  
616 last few years while offering researchers practical options for analyzing their own  
617 data using the large and growing class of SCR models.

618 CR and SCR have been thought of mostly as ways to “estimate density” with  
619 not so much of a direct link to understanding ecological processes. So one of the  
620 things that motivated us in writing this book was to elaborate on, and develop, some  
621 ideas related to modeling ecological processes (movement, space usage, landscape  
622 connectivity) in the context of SCR models. The incorporation of spatial ecological  
623 processes is where SCR models present an important improvement over traditional,  
624 non-spatial CR models. SCR models explicitly describe exposure of individuals to  
625 sampling that results from the juxtaposition of sampling devices or traps with  
626 individuals, as well as the ecologically intuitive link between abundance and area,  
627 both of which are unaccounted for by traditional CR models. By including spatial  
628 processes, these models can be adapted and expanded to directly address many  
629 questions related to animal population and landscape ecology, wildlife management  
630 and conservation. As such, SCR models stand to revolutionize how researchers  
631 study animal populations. With such advanced tools at hand, we believe that,  
632 but for some specific situations, traditional closed population models are largely  
633 obsolete, except as a conceptual device.

634 So, while we do have a lot of material on density estimation in this book – this is  
635 problem # 1 in applied ecology – we worked hard to cover a lot more of the spatial  
636 aspect of population analysis as relevant to SCR. There are a lot of books out  
637 there that cover spatial analysis of population structure which are more theoretical  
638 or mathematical, and there are a lot of books out there that cover sampling and  
639 estimation, but that are *not* spatial. Our book bridges these two major ideas as  
640 much as is possible as of, roughly, mid-late 2012.

---

## THEMES OF THIS BOOK

641 In this book, we try to achieve a broad conceptual and methodological scope from  
 642 basic closed population models for inference about population density, movement,  
 643 space usage and resource selection, on up to open population models for inference  
 644 about vital rates such as survival and recruitment. Much of the material is a  
 645 synthesis of recent research but we also expand SCR models in a number of useful  
 646 directions, including to the development of explicit models of landscape connectivity  
 647 based on ecological or least-cost distance (Chapt. 12), use of telemetry information  
 648 to model resource selection with SCR (Chapt. 13), and to accommodate unmarked  
 649 individuals (Chapt. 18), and many other new topics that have only recently, or  
 650 not yet at all, appeared in the literature. Our intent is to provide a comprehensive  
 651 resource for ecologists interested in understanding and applying SCR models to  
 652 solve common problems faced in the study of populations. To do so, we make use  
 653 of hierarchical models (Royle and Dorazio, 2008), which allow great flexibility in  
 654 accommodating many types of capture-recapture data. We present many example  
 655 analyses, of real and simulated data using likelihood-based and Bayesian methods—  
 656 examples that readers can replicate using the code presented in the text and the  
 657 resources made available on-line and in our accompanying **R** package **scrbook**.

658 The conceptual and methodological themes of this book can be summarized as  
 659 follows:

- 660 (1) Spatial ecology: Much of ecology is about spatial variation in processes (e.g.,  
 661 density) and the mechanisms (e.g., habitat selection, movement) that determine  
 662 this variation. Temporal variation is also commonly of interest and we cover this  
 663 as well, but in less depth.
- 664 (2) Spatial observation error: Observation error is omnipotent in ecology, especially  
 665 in the study of free-ranging vertebrates, and in fact the entire 100+ year history  
 666 of capture-recapture studies have been devoted to estimating key demographic  
 667 parameters in the presence of observation error because we simply cannot observe  
 668 all the individuals that are present, and we can't know their fates even if we mark  
 669 them all. What has been missing in most of the capture-recapture methods is an  
 670 acknowledgment of the spatial context of sampling and the fact that capture (or  
 671 detection) probability will virtually always be a function of the distance between  
 672 traps and animals (or their home ranges).
- 673 (3) Hierarchical modeling: Hierarchical models (HM) are the perfect tool for mod-  
 674 eling spatial processes, especially those of the type covered in this book, where  
 675 one process (the ecological process) is conditionally related to another (the obser-  
 676 vation process). We make use of HMs throughout this book, and we do so using  
 677 both Bayesian and classical (frequentist, likelihood-based) modes of inference.  
 678 These tools allow us to mold our hypotheses into probability models which can  
 679 be used for description, testing, and prediction.
- 680 (4) Model implementation: We consider proper implementation of the models to  
 681 be very important throughout the book. We explore likelihood methods using

existing software such as the **R** package **secr** (Efford, 2011a), as well as development of custom solutions along the way. In Bayesian analyses of SCR models, we emphasize the use of the **BUGS** language for describing models. We also show readers how to devise their own MCMC algorithms for Bayesian analysis of SCR models, which can be convenient (even necessary) in some practical situations.

Altogether, these elements provide for a formulation of SCR models that will allow the reader to learn the fundamentals of standard modeling concepts and ultimately implement complex hierarchical models. We also believe that while the focus of the book is spatial capture-recapture (that is, in fact, the title), the reader will be able to apply the general principles that we cover in the introductory material (e.g., principles of Bayesian analysis) and even the advanced material (e.g., building your own MCMC algorithm) to a broad array of topics in general ecology and wildlife science. Although we aim to reach a broad audience, at times we go into details that may only be of interest to advanced practitioners who need to extend capture-recapture models to unique situations. We hope that these advanced topics will not discourage those new to these methods, but instead will allow readers to advance their own understanding and become less reliant on restrictive tools and software.

## COMPUTING

We rely heavily on data processing and analysis in the **R** programming language, which by now is something that many ecologists not only know about, but use frequently. We adopt **R** because it is free, has a large community that constantly develops code for new applications, and it gives the user flexibility in data processing and analyses. There are some great books out there, including Venables and Ripley (2002), Bolker (2008) and Zuur et al. (2009), and we encourage those new to **R** to read through the manuals that come with the software. We use a number of **R** packages in our analyses, which are described in Appendix 1, and moreover, we provide an **R** package containing the scripts and functions for all of our analyses (see below).

We also rely on the various implementations of the **BUGS** language including **WinBUGS** (Lunn et al., 2000) and **JAGS** (Plummer, 2003). Because **WinBUGS** is not in active development any more, we are transitioning to mainly using **JAGS**. Sometimes models run better or mix better in one or the other. As a side note, we don't have much experience with **OpenBUGS** (Thomas et al., 2006), but our code for **WinBUGS** should run just the same in **OpenBUGS**. The **BUGS** language provides not only a computational device for fitting models but it also emphasizes understanding of what the model is and fosters understanding of how to construct models. As our good colleague Marc Kéry wrote (Kéry, 2010, p. 30) “**BUGS** frees the modeler in you.” While we mostly use **BUGS** implementations, we do a limited amount of developing our own custom MCMC algorithms (see Chapt. 17)

721 which we find very helpful for certain problems where **BUGS/JAGS** fail, or prove  
722 to be inefficient.

723 You will find a fair amount of likelihood analysis throughout the book, and we  
724 have a chapter that provides the conceptual and technical background for how to do  
725 this, and several chapters use likelihood methods exclusively. We use the **R** package  
726 **secr** (Efford et al., 2009a) for many analyses, and we think people should use this  
727 tool because it is polished, easy to use, fairly general, has the usual **R** summary  
728 methods, and has considerable capability for doing analysis from start to finish. In  
729 some chapters we discuss models that we have to use likelihood methods for, but  
730 which are not implemented (at the time when we wrote this book) in **secr** (e.g.,  
731 Chaps. 12, 13). These provide good examples of why it is useful to understand  
732 the principles and to be able to implement these methods yourself.

733 **The R package scrbook**

734 As we were developing content for the book it became clear that it would be useful  
735 if the tools and data were available for readers to reproduce the analyses and also  
736 to modify so that they can do their own analysis. Almost every analysis we did  
737 is included as an **R** script in the **scrbook** package. The **R** package will be very  
738 dynamic, as we plan to continue to update and expand it.

739 The package is not meant to be general-purpose, flexible software for doing SCR  
740 models but, rather, a set of examples and templates illustrating how specific things  
741 are done. Code can be used by the reader to develop methods tailored to his/her  
742 situation, or possibly even more general methods. Because we use so many different  
743 software packages and computing platforms, we think it's impossible to put all of  
744 what is covered in this book into a single integrated package. The **scrbook** package  
745 is for educational purposes and not for production or consulting work.

## ORGANIZATION OF THIS BOOK

746 We expect that readers have a basic understanding of statistical models and classical  
747 inference (What is frequentist inference? What is a likelihood? Generalized linear  
748 model? Generalized linear mixed model?), Bayesian analysis (what is a prior  
749 distribution? and a posterior distribution?), and have used the **R** programming  
750 environment and maybe even the **BUGS** language. The ideal candidate for reading  
751 this book has basic knowledge of these topics; however, we do provide introductory  
752 chapters on the necessary components which we hope can serve as a brief and  
753 cursory tutorial for those who might have only limited technical knowledge, e.g.,  
754 many biologists who implement field sampling programs but do not have extensive  
755 experience analyzing data.

756 To that extent, we introduce Bayesian inference in some detail because we think  
757 readers are less likely to have had a class in that and we also wanted to produce  
758 a stand-alone product. Because we do likelihood analysis of many models, there is

759 an introduction to the relevant elements of likelihood analysis in Chapt. 6, and the  
760 implementation of SCR models in the package **secr** (Efford, 2011a). Our intent  
761 was to provide all of the material you need in one place, but naturally this led to  
762 one of the deficiencies with the book: it's a little bit long-winded, especially in the  
763 first, introductory part. This should not discourage you, and if you already have  
764 extensive background in the basics of statistical inference, you can skip straight  
765 ahead to the specifics of SCR modeling, starting with Chapt. 5.

766 In the following chapters we develop a comprehensive synthesis and extension of  
767 spatial capture-recapture models. Roughly the first third of the book is introductory  
768 material. In Chapt. 3 we provide the basic analysis tools to understand and analyze  
769 SCR models, namely generalized linear models (GLMs) with random effects, and  
770 demonstrate their analysis in **R** and **WinBUGS**. Because SCR models represent  
771 extensions of basic CR models, we cover ordinary closed population models in  
772 Chapt. 4.

773 In the 2nd section of the book, we extend capture-recapture to SCR models  
774 (Chapt. 5), and discuss a number of different conceptual and technical topics  
775 including tools for likelihood inference (Chapt. 6), analysis of model fit and model  
776 selection (Chapt. 8), and sampling design (Chapt. 10). Along with Chaps. 7 and  
777 9, this part of the book provides the basic introduction to spatial capture-recapture  
778 models and their analysis using Bayesian and likelihood methods.

779 The 3rd section of the book covers more advanced SCR models. We have a  
780 number of chapters on spatial modeling aspects related to SCR, including mod-  
781eling spatial variation in density (Chapt. 11, modeling landscape connectivity or  
782 “ecological distance” using SCR models (Chapt. 12), and modeling space usage  
783 or resource selection (Chapt 13), which includes material on integrating telemetry  
784 data into SCR models. After this there are a series of 3 chapters that involve  
785 some elements of modeling spatially or temporally stratified populations. We cover  
786 Bayesian multi-session models in Chapt. 14, what we call “search-encounter” mod-  
787els in Chapt. 15 and, finally, fully open models involving movement or population  
788 dynamics in Chapt. 16. The reason we view the search-encounter models chap-  
789 ter, Chapt. 15, as a prelude to fully open models is that these models apply to  
790 situations where we observe the animal locations “unbiased by fixed sampling lo-  
791cations” – so we get to observe clean measurements of movement outcomes. When  
792 this is possible, we can resolve parameters of explicit movement models free of those  
793 that involve encounter probability. For example, one such models has two “scale”  
794 parameters:  $\sigma$  that determines the rate of decay in encounter probability from a  
795 sampling point or line, and  $\tau$  which is the standard deviation of movements about  
796 an individuals activity center.

797 The final conceptual 4th of this book is what we call “Super-advanced SCR  
798 Models.” We include a chapter on developing your own MCMC algorithms for  
799 SCR models because many advanced models require you to do this, or can be run  
800 more efficiently than in the **BUGS** language, and we thought some readers would  
801 appreciate a practical introduction to MCMC for ecologists. Following the MCMC

802 chapter, we have a number of topics related to unmarked individuals (Chapt. 18)  
803 or partially marked populations (Chapt. 19). This last section of the book contains  
804 some research areas that we are currently developing but lays the foundation for  
805 further development of novel extensions and applications.

806 When this project was begun in 2008, the idea of producing a 550 page book  
807 would have been unimaginable – there wasn’t that much material to work with. Op-  
808 timistically, there was maybe a 250 page monograph that could have been squeezed  
809 out of the literature. But, during the project, great and new things appeared in  
810 the literature, and we developed new models and concepts ourselves, in the process  
811 of writing the book. This includes models of resource selection, landscape connec-  
812 tivity, and methods for dealing with unmarked individuals. There are at least 10  
813 chapters in the book that we couldn’t have thought about 5 years ago. We hope  
814 that the result is a timely summary and a lasting resource.

815 Acknowledgements  
816

817 The project owes a great intellectual debt to Jim Nichols, who has been an  
818 extraordinary mentor and colleague, and who generously shared his astounding in-  
819 sight into animal sampling and modeling problems, his knowledge of the literature  
820 and history of abundance and density estimation and, most importantly, his ex-  
821 tremely valuable time. He has been an extremely helpful guy on all fronts. We are  
822 honored that Jim agreed to write the Foreword to introduce the book. We thank  
823 Marc Kéry for being a great friend and colleague, and for his creativity, energy,  
824 and enthusiasm in developing new ideas and presenting workshops on hierarchical  
825 modeling in ecology.

826 **Special thanks** to the following people: (1) Kimberly Gazenski whose support  
827 was invaluable. She worked on administrative, technical and editorial aspects of  
828 the whole project. She maintained the BIBTEX database, worked on the GitHub  
829 repository that housed the LATEX and R package source trees, edited LATEX files,  
830 tested R scripts, did GIS and R programming, analysis, debugging, and graphics;  
831 (2) Our WCS Tiger program colleagues K. Ullas Karanth and Arjun Gopalaswamy  
832 for continued support and collaboration on SCR problems; (3) Sarah Converse,  
833 our PWRC colleague, for her interest in SCR models and developing a number of  
834 methodological and application papers related to multi-session models, providing  
835 feedback on draft material, and friendship; (4) Murray Efford whose seminal 2004  
836 Oikos paper first introduced spatial capture-recapture models. His R package secr  
837 is a powerful tool for analyzing spatial capture recapture data used throughout the  
838 book. Murray also answered many questions regarding secr that were helpful in  
839 developing our applications and examples.

840 We thank the following people for providing data, photographs or figures: Agustin  
841 Paviolo (jaguar data in Chapt. 11, and the cover image). Michael Wegan, Paul  
842 Curtis, and Raymond Rainbolt (black bear data from Fort Drum, NY); Audrey  
843 Magoun (wolverine data and photographs); Cat Sun and Angela Fuller (black bear  
844 data in Chapt. 13; and fisher photo Chapt. 20); Joshua Raabe and Joseph High-  
845 tower (American shad photo and data in Chap. 16); Erin Zylstra (tortoise data in  
846 Chapt. 4); Martha (Liz) Rutledge (Canada geese data and picture in 19); Craig  
847 Thompson (fisher data in Chapt. 15 and photographs in Chapt. 1); Jerrold Belant  
848 (black bear data in Chapt. 10); Kevin and April Young (FTHL photograph, Chapt.  
849 15); Theodore Simons, Allan O'Connell, Arielle Parsons, and Jessica Stocking (rac-  
850 coon data in Chapt. 19);

851 We thank the following people for reviewing one or more draft chapters and  
852 giving feedback along the way: David Borchers, Bob Dorazio, Tabitha Graves,  
853 Marc Kéry, Brett McClintock, Allan O'Connell, Krishna Pacifici, Agustin Paviolo,  
854 Brian Reich, Robin Russell, Sabrina Servany, Cat Sun, and Chris Sutherland.

855 Additionally, many colleagues and friends, as well as our families, provided us

<sup>856</sup> with encouragement and feedback throughout this project and we thank them for  
<sup>857</sup> their continued support.



858

## Part I

859

---

860

# Background and Concepts



861  
862

# 1

---

863

## INTRODUCTION

864 Space plays a vital role in virtually all ecological processes (Tilman and Kareiva,  
865 1997; Hanski, 1999; Clobert et al., 2001). The spatial arrangement of habitat can  
866 influence movement patterns during dispersal, habitat selection, and survival. The  
867 distance between an organism and its competitors and prey can influence activity  
868 patterns and foraging behavior. Further, understanding distribution and spatial  
869 variation in abundance is necessary in the conservation and management of popu-  
870 lations. The inherent spatial aspect of *sampling* populations also plays an important  
871 role in ecology as it strongly affects, and biases, how we observe population struc-  
872 ture (Seber, 1982; Buckland et al., 2001; Borchers et al., 2002; Williams et al.,  
873 2002). However, despite the central role of space and spatial processes to both  
874 understanding population dynamics and how we observe or sample populations, a  
875 coherent framework that integrates these two aspects of ecological systems has not  
876 been fully realized either conceptually or methodologically.

877 Capture-recapture methods represent perhaps the most common technique for  
878 studying animal populations, and their use is growing in popularity due to recent  
879 technological advances that provide mechanisms to study many taxa which before  
880 could not be studied efficiently, if at all. However, a major deficiency of classical  
881 capture-recapture methods is that they do not admit the spatial structure of either  
882 ecological processes that give rise to encounter history data, nor the spatial aspect  
883 of collecting these data. While many technical limitations of this lack of spatial  
884 explicitness have been recognized for decades (Dice, 1938; Hayne, 1950), it has  
885 only been very recent (Efford, 2004; Borchers, 2012) that spatially explicit capture-  
886 recapture methods – those which accommodate space – have been developed.

887 Spatial capture-recapture (SCR) methods resolve a host of technical problems  
888 that arise in applying capture-recapture methods to animal populations. However,  
889 SCR models are not merely an extension of technique. Rather, they represent a

890 much more profound development in that they make ecological processes explicit in  
891 the model – processes of density, spatial organization, movement and space-usage by  
892 individuals. The practical importance of SCR models is that they allow ecological  
893 scientists to study elements of ecological theory using individual encounter data  
894 that exhibit various biases relating to the observation mechanisms employed. At  
895 the same time, SCR models can be used, and may be the only option, for obtaining  
896 demographic data on some of the rarest and most elusive species – information  
897 which is required for effective conservation. It is this potential for advancing both  
898 applied and theoretical research that motivated us to write this book.

## 1.1 THE STUDY OF POPULATIONS BY CAPTURE-RECAPTURE

899 In the fields of conservation, management, and general applied ecology, information  
900 about abundance or density of populations and their vital rates is a basic require-  
901 ment. To that end, a huge variety of statistical methods have been devised, and  
902 as we noted already, the most well-developed are collectively known as capture-  
903 recapture (or capture-mark-recapture) methods. For example, the volumes by Otis  
904 et al. (1978), White et al. (1982), Seber (1982), Pollock et al. (1990), Borchers  
905 et al. (2002), Williams et al. (2002), and Amstrup et al. (2005) are largely syn-  
906 synthetic treatments of such methods, and contributions on modeling and estimation  
907 using capture-recapture are plentiful in the peer-reviewed ecology literature.

908 Capture-recapture techniques make use of individual *encounter history* data, by  
909 which we mean sequences of (usually) 0's and 1's denoting if an individual was  
910 encountered during sampling over a certain time period (occasion). For example,  
911 the encounter history “010” indicates that this individual was encountered only  
912 during the second of three trapping occasions. As we will see, these data contain  
913 information about encounter probability, and also abundance, and other parameters  
914 of interest in the study of populations.

915 Capture-recapture has been important in studies of animal populations for many  
916 decades, and its importance is growing dramatically in response to technological  
917 advances that improve our ability and efficiency to obtain encounter history data.  
918 Historically, such information was obtainable using methods requiring physical cap-  
919 ture of individuals. However, new methods do not require physical capture or  
920 handling of individuals. A large number of passive detection devices produce indi-  
921 vidual encounter history data including camera traps (Karanth and Nichols, 1998;  
922 O'Connell et al., 2010), acoustic recording devices (Dawson and Efford, 2009), and  
923 methods that obtain DNA samples such as hair snares for bears, scent posts for  
924 many carnivores, and related methods which allow DNA to be extracted from scat,  
925 urine or animal tissue in order to identify individuals. This book is concerned with  
926 how such data can be used to carry out inference about animal abundance or den-  
927 sity, and other parameters such as survival, recruitment, resource selection, and  
928 movement using new classes of capture-recapture models which utilize auxiliary  
929 spatial information related to the encounter process. We refer to such methods as

## **LIONS AND TIGERS AND BEARS, OH MY: GENESIS OF SPATIAL CAPTURE-RECAPTURE DATA<sup>5</sup>**

---

930 spatial capture-recapture (SCR) models<sup>1</sup>.

931 As the name implies, the primary feature of SCR models that distinguishes  
932 them from traditional CR methods is that they make use of the spatial information  
933 inherent to capture-recapture studies. Encounter histories that are associated with  
934 auxiliary information on the location of capture, are *spatial encounter histories*.  
935 This auxiliary information is informative about spatial processes including the spa-  
936 tial organization of individuals, variation in density, resource selection and space  
937 usage, and movement. As we will see, SCR models allow us to overcome critical  
938 deficiencies of non-spatial methods, and integrate ecological theory with encounter  
939 history data. As a result, this greatly expands the practical utility and scientific  
940 relevance of capture-recapture methods, and studies that produce encounter history  
941 data.

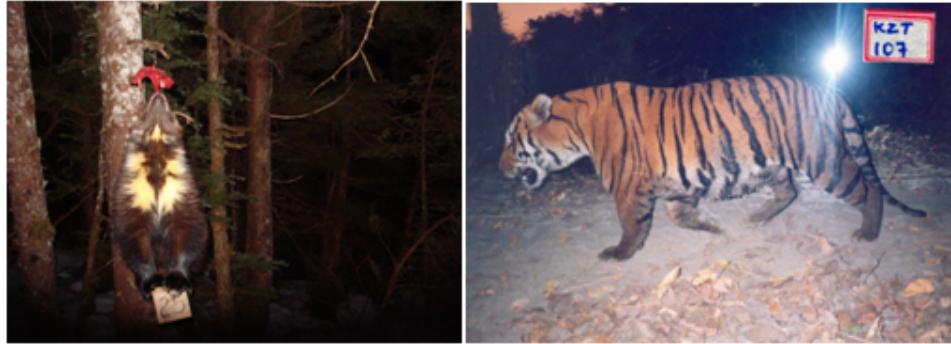
### **1.2 LIONS AND TIGERS AND BEARS, OH MY: GENESIS OF SPATIAL CAPTURE-RECAPTURE DATA**

942 A diverse number of methods and devices exist for producing individual encounter  
943 history data with auxiliary spatial information about individual locations. Histori-  
944 cally, physical “traps” have been widely used to sample animal populations. These  
945 include live traps, mist nets, pitfall traps and many other types of devices. Such  
946 devices physically retain animals until visited by a biologist, who removes the indi-  
947 vidual, marks it or otherwise molests it in some scientific fashion, and then releases  
948 it. Although these are still widely used, recent technological advances for obtain-  
949 ing encounter history data non-invasively have made it possible to study many  
950 species that were difficult if not impossible to study effectively just a few years ago.  
951 As a result, these methods have revolutionized the study of animal populations  
952 by capture-recapture methods, have inspired the development of spatially-explicit  
953 extensions of capture-recapture, and will lead to their increasing relevance in the  
954 future. We briefly review some of these here, which we consider more explicitly in  
955 later chapters of this book.

#### **956 1.2.1 Camera trapping**

957 Considerable recent work has gone into the development of camera-trapping method-  
958 ologies. For a historical overview of this method see Kays et al. (2008) and Kucera  
959 and Barrett (2011). Several recent synthetic works have been published includ-  
960 ing Nichols and Karanth (2002), and an edited volume by O’Connell et al. (2010)  
961 devoted solely to camera trapping concepts and methods. As a method for estimat-  
962 ing abundance, some of the earliest work that relates to the use of camera trapping  
963 data in capture-recapture models originates from Karanth and colleagues (Karanth,  
964 1995; Karanth and Nichols, 1998, 2000).

<sup>1</sup>In the literature the term spatially explicit capture-recapture (SECR) is also used, but we prefer the more concise term.



**Figure 1.1.** Left: Wolverine being encounter by a camera trap (Photo credit: Audrey Magoun). Right: Tiger encountered by camera trap (Photo credit: Ullas Karanth).

965 In camera trapping studies, cameras are often situated along trails or at baited  
 966 stations and individual animals are photographed and subsequently identified either  
 967 manually by a person sitting behind a computer, or sometimes now using specific  
 968 identification software. Camera trapping methods are widely used for species that  
 969 have unique stripe or spot patterns such as tigers (Karanth, 1995; Karanth and  
 970 Nichols, 1998), ocelots (*Leopardus pardalis*; (Trolle and Kéry, 2003, 2005)), leopards  
 971 (*Panthera pardus*; (Balme et al., 2010)), and many other cat species. Camera traps  
 972 are also used for other species such as wolverines (*Gulo gulo*; (Magoun et al., 2011;  
 973 Royle et al., 2011b)), and even species that are less easy to identify uniquely such as  
 974 mountain lions (*Puma concolor*, (Sollmann et al., in revision)) and coyotes (*Canis*  
 975 *latrans*, (Kelly et al., 2008)). We note that even for species that are not readily  
 976 identified by pelage patterns, it might be efficient to use camera traps in conjunction  
 977 with spatial capture-recapture models to estimate density (see Chaps. 18 and 19).

### 978 1.2.2 DNA sampling

979 DNA obtained from hair, blood or scat is now routinely used to obtain individual  
 980 identity and encounter history information about individuals (Taberlet and Bouvet,  
 981 1992; Kohn et al., 1999; Woods et al., 1999; Mills et al., 2000; Schwartz and Monfort,  
 982 2008). A common method is based on the use of “hair snares” (Fig. 1.2) which are  
 983 widely used to study bear populations (Woods et al., 1999; Garshelis and Hristienko,  
 984 2006; Kendall et al., 2009; Gardner et al., 2010b). A sample of hair is obtained as  
 985 individuals pass under or around barbed-wire (or other physical mechanism) to take  
 986 bait. Hair snares and scent sticks have also been used to sample felid populations  
 987 (García-Alaníz et al., 2010; Kéry et al., 2010) and other species. Research has  
 988 even shown that DNA information can be extracted from urine deposited in the



**Figure 1.2.** Left: Black bear in a hair snare (*Photo credit: M. Wegan*) Right: European wildcat loving on a scent stick (*Photo credit: Darius Weber*)

wild (e.g., in snow; see Valiere and Taberlet (2000)) and as a result this may prove another future data collection technique where SCR models are useful.

### 1.2.3 Acoustic sampling

Many studies of birds (Dawson and Efford, 2009), bats, and whales (Marques et al., 2009) now collect data using devices that record vocalizations. When vocalizations can be identified by individual from multiple recording devices, spatial encounter histories are produced that are amenable to the application of SCR models (Dawson and Efford, 2009; Efford et al., 2009b). Recently, these ideas have been applied to data on direction or distance to vocalizations by multiple simultaneous observers and related problems (D. Borchers, ISEC 2012 presentation).

### 1.2.4 Search-encounter methods

There are other methods which don't fall into a nice clean taxonomy of "devices". Spatial encounter histories are commonly obtained by conducting manual searches of geographic sample units such as quadrats, transects or road or trail networks. For example, DNA-based encounter histories can be obtained from scat samples located along roads or trails or by specially trained dogs (MacKay et al., 2008) searching space (Fig. 1.3). This method has been used in studies of martens, fishers (Thompson et al., 2012), lynx, coyotes, birds (Kéry et al., 2010), and many other species. A similar data structure arises from the use of standard territory or spot mapping of birds Bibby et al. (1992) or area sampling in which space is searched by observers to physically capture individuals. This is common in surveys



**Figure 1.3.** Left: A wildlife research technician for the USDA Forest Service holding a male fisher captured as part of the Kings River Fisher Project in the Sierra National Forest, California. Right: A dog handler surveying for fisher scat in the Sierra National Forest. *Photo credit: Craig Thompson.*

1010 that involve reptiles and amphibians, e.g., we might walk transects picking up box  
 1011 turtles (Hall et al., 1999), or desert tortoises (Zylstra et al., 2010), or search space  
 1012 for lizards (Royle and Young, 2008).

1013 These methods don't seem like normal capture-recapture in the sense that the  
 1014 encounter of individuals is not associated with specific trap location, but SCR  
 1015 models are equally relevant for analysis of such data as we discuss in Chapt. 15.

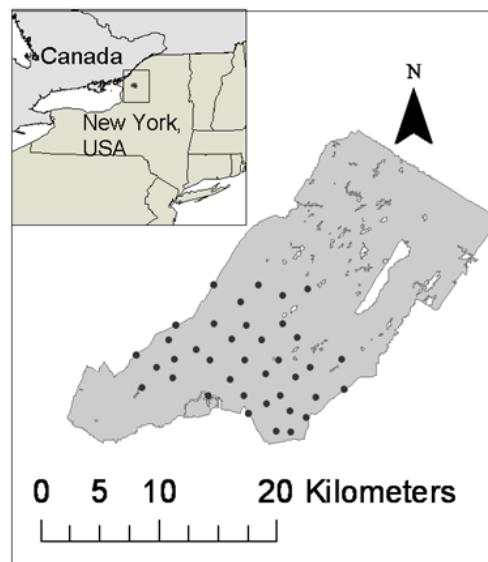
### 1.3 CAPTURE-RECAPTURE FOR MODELING ENCOUNTER PROBABILITY

1016 We briefly introduced techniques used for the study of animal populations. These  
 1017 methods produce individual encounter history data, a record of where and when  
 1018 each individual was captured. We refer to this as a *spatial encounter history*. Histor-  
 1019 ically, auxiliary spatial information has been ignored, and encounter history data  
 1020 have been *summarized* to simple “encounter or not” for the purpose of applying  
 1021 ordinary CR models. The basic problem with these ordinary (or “non-spatial”)  
 1022 capture-recapture models is they don’t have any sense of space in them, the spatial  
 1023 information is summarized out of the data set, so we aren’t able to use such mod-  
 1024 els for studying things such as movement, or resource selection, etc.*dots*. Instead,  
 1025 ordinary capture-recapture models usually resort to models of “encounter prob-

ability,” which is a nuisance parameter, seldom of any ecological relevance. We show an example here that is in keeping with the classical application of ordinary capture-recapture models.

### 1.3.1 Example: Fort Drum bear study

Here we confront the simplest possible capture-recapture problem – but one of great applied interest – estimating density from a standard capture-recapture study. We use this as a way to introduce some concepts and motivate the need for spatial capture-recapture models by confronting technical and conceptual problems that we encounter. The data come from a study to estimate black bear abundance on the Fort Drum Military Installation in upstate New York ( Wegan (2008), see also Chapt. 4 for more details). The specific data used here are encounter histories on 47 individuals obtained from an array of 38 baited “hair snares” during June and July 2006. The study area and locations of the 38 hair snares are shown in Fig. 1.4. Barbed wire traps (see Fig. 1.2) were baited and checked for hair samples each week for eight weeks. Analysis of these data appears in Gardner et al. (2009) and Gardner et al. (2010b), and we use the data in a number of analyses in later chapters.



**Figure 1.4.** Locations of hair snares on Fort Drum, New York, operated during the summer of 2006 to sample black bears.

1043     Although each bear was captured, or not, in each of the 38 hair snares, we start  
1044 by treating this data set as a standard capture-recapture data set and summarize  
1045 to an encounter history matrix with 47 rows and 8 columns with entries  $y_{ik}$ , where  
1046  $y_{ik} = 1$  if individual  $i$  was captured, at any trap, in sample occasion  $k$  and  $y_{ik} = 0$   
1047 otherwise. There is a standard closed population model, colloquially referred to  
1048 as “model  $M_0$ ” (see Chapt. 4), which assumes that encounter probability  $p$  is  
1049 constant for all individuals and sample periods. We fitted model  $M_0$  to the Fort  
1050 Drum data using traditional likelihood methods, yielding the maximum likelihood  
1051 estimate (MLE) of  $\hat{N} = 49.19$  with an asymptotic standard error (SE) of 1.9.

1052     The key issue in using such a closed population model regards how we should  
1053 interpret this estimate of  $N = 49.19$  bears. Does it represent the entire population  
1054 of Fort Drum? Certainly not – the trapping array covers less than half of Fort  
1055 Drum as we see in Fig. 1.4. So to get at the total bear population size of Fort  
1056 Drum, we would have to convert our  $\hat{N}$  to an estimate of density and extrapolate.  
1057 To get at density, then, should we assert that  $N$  applies to the southern half of  
1058 Fort Drum below some arbitrary line? Surely bears move on and off of Fort Drum  
1059 without regard to hypothetical boundaries. Without additional information there  
1060 is simply no way of converting this estimate of  $N$  to density, and hence it is really  
1061 not meaningful biologically. To resolve this problem, we will adopt the customary  
1062 approach of converting  $N$  to  $D$  by buffering the convex hull around the trap array.  
1063 The convex hull has area  $157.135 \text{ km}^2$ . We follow Bales et al. (2005) in buffering  
1064 the convex hull of the trap array by the radius of the mean female home range size.

1065     The mean female home range radius was estimated (Wegan, 2008) for this study  
1066 region to be 2.19 km, and the area of the convex hull buffered by 2.19 km is  
1067  $277.01 \text{ km}^2$ . (**R** commands to compute the convex hull, buffer it, and compute the  
1068 area are given in the **R** package **scrbook** which accompanies the book). Hence,  
1069 the estimated density here is approximately 0.178 bears/ $\text{km}^2$  using the estimated  
1070 population size obtained by model  $M_0$ . We could assert that the problem has been  
1071 solved, go home, and have a beer. But then, on the other hand, maybe we should  
1072 question the use of the estimated home range radius – after all, this is only the  
1073 female home range radius and the home ranges change for many reasons. Instead,  
1074 we may decide to rely on a buffer width based on one-half mean maximum distance  
1075 moved (MMDM) estimated from the actual hair snare data as is more customary  
1076 (Dice, 1938). In that case the buffer width is 1.19 km, and the resulting estimated  
1077 density is increased to 0.225 bears/ $\text{km}^2$  about 27 % larger. But wait – some studies  
1078 actually found the full MMDM (Parmenter et al., 2003) to be a more appropriate  
1079 measure of movement (e.g. Soisalo and Cavalcanti (2006)). So maybe we should use  
1080 the full MMDM which is 2.37 km, pretty close to the telemetry-based estimate and  
1081 therefore providing a similar estimate of density (0.171 bears/ $\text{km}^2$ ). So in trying to  
1082 decide how to buffer our trap array we have already generated 3 density estimates.  
1083 The crux of the matter is obvious: Although it is intuitive that  $N$  should scale with  
1084 area – the number of bears should go up as area increases and go down as area  
1085 decreases – in this ad hoc approach of accounting for animal movement  $N$  remains

1086 the same, no matter what area we assert was sampled. The number of bears and the  
 1087 area they live in are not formally tied together within the model, because estimating  
 1088  $N$  and estimating the area  $N$  relates to are two completely independent analytical  
 1089 steps which are unrelated to one another by a formal model.

1090 Unfortunately, our problems don't end here. In thinking about the use of model  
 1091  $M_0$ , we might naturally question some of the basic assumptions that go into that  
 1092 model. The obvious one to question is that which declares that  $p$  is constant.  
 1093 One obvious source of variation in  $p$  is variation *among individuals*. We expect  
 1094 that individuals may have more or less exposure to trapping due to their location  
 1095 relative to traps, and so we try to model this "heterogeneous" encounter probability  
 1096 phenomenon. To illustrate this phenomenon, here are the number of traps that each  
 1097 individual was encountered in:

```
1098 # traps: 1 2 3 4 5 6 9
1099 # bears: 23 13 6 2 1 1 1
```

1100 meaning, for example, 23 bears were captured in only 1 trap, and 1 bear was  
 1101 captured in 9 distinct traps. The variation in trap-encounter frequencies suggests  
 1102 quite a range in traps exposed to bears in the sampled population. Historically,  
 1103 researches try to reduce spatial heterogeneity in capture probability by placing  $> 1$   
 1104 trap per home range (Otis et al., 1978; Williams et al., 2002). This seems like a  
 1105 sensible idea but it is difficult to do in practice since you don't know where all  
 1106 the home ranges are and so we try to impose a density of traps that averages  
 1107 something  $> 1$  per home range. An alternative solution is to fit models that allow  
 1108 for individual heterogeneity in  $p$  (Karanth, 1995). Such models have the colloquial  
 1109 name of "model  $M_h$ " (Otis et al., 1978). We fitted this model (see Chapt. 4  
 1110 for details) to the Fort Drum data using each of the 3 buffer widths previously  
 1111 described (telemetry, 1/2 MMDM and MMDM), producing the estimates reported  
 1112 in Table 1.1. While we can tell by the models' AIC that  $M_h$  is clearly favored by  
 1113 more than 30 units, we might still not be entirely happy with our results. Clearly  
 1114 there is information in our data that could tell us something about the exposure  
 1115 of individual bears to the trap array – where they were captured, and how many  
 1116 times – but since space has no representation in our model, we can't make use  
 1117 of this information. Model  $M_h$  thus merely accounts for what we observe in our  
 1118 data (some bears were more frequently captured than others) rather than explicitly  
 1119 accounting for the processes that generated the data.

1120 So what are we left with? Our density estimates span a range from 0.17 to  
 1121 0.43 bears/km<sup>2</sup> depending on which estimator of  $N$  we use and what buffer strip  
 1122 we apply. Should we feel strongly about one or the other? Which buffer should  
 1123 we prefer? AIC favors model  $M_h$ , but did it adequately account for the differ-  
 1124 ences in exposure of individuals to the trap array? Are we happy with a purely  
 1125 phenomenological model for heterogeneity? It assumes that all individuals are in-  
 1126 dependent and identically distributed (*iid*) draws from some distribution, but does  
 1127 not account for the explicit mechanism of induced heterogeneity. And, further, we

have information about that (trap of capture) which model  $M_h$  ignores. And if we choose one type of buffer, how do we compare our density estimates to those from other studies that may opt for a different kind of buffer? The fact that  $N$  does not scale with  $A$ , as part of the model, renders this choice arbitrary.

**Table 1.1.** Table on estimates of density ( $D$ , bears/ $km^2$ ) for the Fort Drum data using models  $M_0$  and  $M_h$  and different buffers. Model  $M_h$  here is a logit-normal mixture (Coull and Agresti, 1999).

Model	Buffer	$\hat{D}$	SE
$M_0$	telemetry	0.178	0.178
$M_0$	MMDM	0.171	0.171
$M_0$	1/2 MMDM	0.225	0.225
$M_h$	telemetry	0.341	0.144
$M_h$	MMDM	0.327	0.138
$M_h$	1/2 MMDM	0.432	0.183

### 1.3.2 Inadequacy of non-spatial capture-recapture

The parameter  $N$  (population size) in an ordinary capture-recapture model is functionally unrelated to any notion of sample area, and so we are left taking arbitrary guesses at area, and matching it up with estimates of  $N$  from different models that do not have any explicit biological relevance. Clearly, there is not a compelling solution to be derived from this “estimate  $N$  and conjure up a buffer” approach and we are left not much wiser about bear density at Fort Drum than we were before we conducted this analysis, and certainly not confident in our assessments. Closed population models are not integrated with any ecological theory, so our  $N$  is not connected to the specific landscape in any explicit way.

The capture-recapture models that we used apply to truly closed populations – a population of goldfish in a fish bowl. Yet here we are applying them to a population of bears that inhabit a rich two-dimensional landscape of varied habitats, exposed to trapping by an irregular and sparse array of traps. It seems questionable that the same model that is completely sensible for a population of goldfish in a bowl, should also be the right model for this population of bears distributed over a broad landscape. Ordinary capture-recapture methods are distinctly non-spatial. They don’t admit spatial indexing of either sampling (the observation process) or of individuals (the ecological process). This leads immediately to a number of practical deficiencies: (1) Ordinary CR models do not provide a coherent basis for estimating density, a problem we struggled with in the black bear study. (2) Ordinary CR model and sampling methods *induce* a form of heterogeneity that can only at best be approximated by classical models of latent heterogeneity. SCR models formally accommodate heterogeneity due to the juxtaposition of individuals with the encounter devices. (3) Ordinary CR models do not accommodate trap-

1157 level covariates which exist in a large proportion of real studies; (4) Ordinary CR  
1158 models do not accommodate formal consideration of any spatial process that gives  
1159 rise to the observed data.

1160 In subsequent chapters of this book, we resolve these specific technical problems  
1161 related to density, model-based linkage of  $N$  and  $A$ , covariates, spatial variation, and  
1162 related things all within a coherent unified framework for spatial capture-recapture.

## 1.4 HISTORICAL CONTEXT: A BRIEF SYNOPSIS

1163 Spatial capture-recapture is a relatively new methodological development, at least  
1164 with regard to formal estimation and inference. However, the basic problems that  
1165 motivate the need for formal spatially-explicit models have been recognized for  
1166 decades and quite a large number of ideas have been proposed to deal with these  
1167 problems. We review some of these ideas here.

### 1168 1.4.1 Buffering

1169 The standard approach to estimating density even now is to estimate  $N$  using  
1170 conventional closed population models (Otis et al., 1978) and then try to associate  
1171 with this estimate some specific sampled area, say  $A$ , the area which is contributing  
1172 individuals to the population for which  $N$  is being estimated. The strategy is to  
1173 define  $A$  by placing a buffer of say  $W$  around the trap array or some polygon which  
1174 encloses the trap array. The historical context is succinctly stated by (O'Brien,  
1175 2011) from which we draw this description:

1176 "At its most simplistic,  $A$  may be described by a concave polygon defined by connect-  
1177 ing the outermost trap locations ( $A_{tp}$ ; Mohr (1947)). This assumes that animals do  
1178 not move from outside the bounded area to inside the area or vice versa. Unless the  
1179 study is conducted on a small island or a physical barrier is erected in the study area  
1180 to limit movement of animals, this assumption is unlikely to be true. More often, a  
1181 boundary area of width  $W$  ( $A_w$ ) is added to the area defined by the polygon  $A_{tp}$  to  
1182 reflect the area beyond the limit of the traps that potentially is contributing animals  
1183 to the abundance estimate (Otis et al., 1978). The sampled area, also known as the  
1184 effective area, is then  $A(W) = A_{tp} + A_w$ . Calculation of the buffer strip width ( $W$ )  
1185 is critical to the estimation of density and is problematic because there is no agreed  
1186 upon method of estimating  $W$ . Solutions to this problem all involve ad hoc methods  
1187 that date back to early attempts to estimate abundance and home ranges based on  
1188 trapping grids (see Hayne, 1949). Dice (1938) first drew attention to this problem  
1189 in small mammal studies and recommended using one-half the diameter of an aver-  
1190 age home range. Other solutions have included use of inter-trap distances (Blair,  
1191 1940; Burt, 1943), mean movements among traps, maximum movements among traps  
1192 (Holdenried, 1940; Hayne, 1949), nested grids (Otis et al., 1978), and assessment lines  
1193 (Smith et al., 1971)."

1194 The idea of using 1/2 mean maximum distance moved ("MMDM" Wilson and  
1195 Anderson, 1985b) to create a buffer strip seems to be the standard approach even  
1196 today, presumably justified by Dice's suggestion to use 1/2 the home range diam-  
1197 eter, with the mean over individuals of the maximum distance moved being an

estimator of home range diameter. Alternatively, some studies have used the full MMDM (e.g. Parmenter et al. (2003)), because the trap array might not provide a full coverage of the home range (home ranges near the edge should be truncated) and so 1/2 MMDM should be biased smaller than the home range radius. And, sometimes home range size is estimated by telemetry (Karanth, 1995; Bales et al., 2005). Use of MMDM summaries to estimate home range radius is usually combined with an AIC-based selection from among the closed-population models in Otis et al. (1978) which most often suggests heterogeneity in detection (model  $M_h$ ). Almost all of these early methods were motivated by studies of small mammals using classical “trapping grids” but, more recently, their popularity in the study of wildlife populations has increased with the advent of new technologies, especially related to non-invasive sampling methods such as camera trapping. In particular, the series of papers by Karanth and Nichols (Karanth, 1995; Karanth and Nichols, 1998, 2002) has led to fairly widespread adoption of these ideas.

#### 1.4.2 Temporary emigration

Another intuitively appealing idea is that by White and Shenk (2000) who discuss “correcting bias of grid trapping estimates” by recognizing that the basic problem is like random temporary emigration (Kendall et al., 1997; Chandler et al., 2011; Ivan et al., 2013a,b) where individuals flip a coin with probability  $\phi$  to determine if they are “available” to be sampled or not. White and Shenk’s idea was to estimate  $\phi$  from radio telemetry, as the proportion of time an individual spends in the study area. They obtain the estimated “super-population” size by using standard closed population models and then obtain density by  $\hat{D} = \hat{N}\hat{\phi}/A$  where  $A$  is the nominal area of the trapping array (e.g., minimum convex hull). A problem with this approach is that individuals that were radio collared represent a biased sample i.e., you fundamentally have to sample individuals randomly from the population *in proportion to their exposure to sampling* and that seems practically impossible to accomplish. In other words, “in the study area” has no precise meaning itself and is impossible to characterize in almost all capture-recapture studies. Deciding what is “in the study area” is effectively the same as choosing an arbitrary buffer which defines who is in the study area and who isn’t. That said, the temporary emigration analogy is a good heuristic for understanding SCR models and has a precise technical relevance to certain models.

Another interesting idea is that of using some summary of “average location” as an individual covariate in standard capture-recapture models. Boulanger and McLellan (2001) use distance-to-edge (DTE) as a covariate in the Huggins-Alho type of model. Ivan (2012) uses this approach in conjunction with an adjustment to the estimated  $N$  obtained by estimating the proportion of time individuals are “on the area formally covered by the grid” using radio telemetry. We do not dwell too much on these different variations but we do note that the use of DTE as an individual covariate amounts to some kind of intermediate model between simple

1239 closed population models and fully spatial capture-recapture models, which we  
1240 address directly in Chapt. 4.

1241 While these procedures are all heuristically appealing, they are also essentially  
1242 ad hoc in the sense that the underlying model remains unspecified or at least im-  
1243 precisely characterized and so there is little or no basis for modifying, extending  
1244 or generalizing the methods. These methods are distinctly *not* model-based pro-  
1245 cedures. Despite this, there seems to be an enormous amount of literature developing,  
1246 evaluating and “validating” these literally dozens of heuristic ideas that solve spe-  
1247 cific problems, as well as various related tweaks and tunings of them and really it  
1248 hasn’t led to any substantive breakthroughs that are sufficiently general or theo-  
1249 retically rigorous.

## 1.5 EXTENSION OF CLOSED POPULATION MODELS

1250 The deficiency with classical closed population models is that they have no spatial  
1251 context.  $N$  is just an integer parameter that applies equally well to estimating the  
1252 number of unique words in a book, the size of some population that exists in a  
1253 computer, or a bucket full of goldfish. The question of *where* the  $N$  items belong  
1254 is central both to interpretation of data and estimates from all capture-recapture  
1255 studies and, in fact, to the construction of spatial capture-recapture models con-  
1256 sidered in this book. Surely it must matter whether the  $N$  items exist as words in  
1257 a book, or goldfish in a bowl, or tigers in a patch of forest! That classical closed  
1258 population models have no spatial context leads to a number of conceptual and  
1259 methodological problems or limitations as we have encountered previously. More  
1260 important, ecologists seldom care only about  $N$  – space is often central to objec-  
1261 tives of many population studies – movement, space usage, resource selection, how  
1262 individuals are distributed in space and in response to explicit factors related to  
1263 landuse or habitat. Because space is central to so many real problems, this is proba-  
1264 bly the number 1 reason that many ecologists don’t bother with capture-recapture.  
1265 They haven’t seen capture-recapture methods as being able to solve their problems.  
1266 Thus, the essential problem is that classical closed population models are too sim-  
1267 ple – they ignore the spatial attribution of traps and encounter events, movement  
1268 and variability in exposure of individuals to trap proximity. These problems can be  
1269 addressed formally by the development of more general capture-recapture models.

### 1270 1.5.1 Towards spatial explicitness: Efford’s formulation

1271 The solution to the various issues that arise in the application of ordinary capture-  
1272 recapture models is to extend the closed population model so that  $N$  becomes  
1273 spatially explicit. Efford (2004) was the first to formalize an explicit model for  
1274 spatial capture-recapture problems in the context of trapping arrays. He adopted  
1275 a Poisson point process model to describe the distribution of individuals and essen-  
1276 tially a distance sampling formulation of the observation model which describes the

1277 probability of detection as a function of individual location, regarded as a latent  
1278 variable governed by the point process model. While earlier (and contemporary)  
1279 methods of estimating density from trap arrays have been ad hoc in the sense of  
1280 lacking a formal description of the spatial model, Efford achieved a formalization  
1281 of the model, describing explicit mechanisms governing the spatial distribution of  
1282 individuals and how they are encountered by traps, but adopted a more or less  
1283 ad hoc framework for inference under that spatial model using a simulation based  
1284 method known as inverse prediction (Gopalanayam, 2012).

1285 Recently, there has been a flurry of effort devoted to formalizing inference un-  
1286 der this model-based framework for the analysis of spatial capture-recapture data  
1287 (Borchers and Efford, 2008; Royle and Gardner, 2011; Borchers, 2012; Gopalanayam,  
1288 2012). There are two distinct lines of work which adopt the model-based formula-  
1289 tion in terms of the underlying point process but differ primarily by the manner in  
1290 which inference is achieved. One approach (Borchers and Efford, 2008) uses classi-  
1291 cal inference based on likelihood (see Chapt. 6), and the other (Royle and Young,  
1292 2008) adopts a Bayesian framework for inference (Chapts. 5 and 17).

1293 **1.5.2 Abundance as the aggregation of a point process**

1294 Spatial point process models represent a major methodological theme in spatial  
1295 statistics (Cressie, 1991) and they are widely applied as models for many ecological  
1296 phenomena (Stoyan and Penttinen, 2000; Illian et al., 2008). Point process models  
1297 apply to situations in which the random variable in question represents the locations  
1298 of events or objects: trees in a forest, weeds in a field, bird nests, etc. . . As such,  
1299 it seems natural to describe the organization of individuals in space using point  
1300 process models. SCR models represent the extension of ordinary capture-recapture  
1301 by augmenting the model with a point process to describe individual locations.

1302 Specifically, let  $s_i; i = 1, 2, \dots, N$  be the locations of all individuals in the popu-  
1303 lation. One of the key features of SCR models is that the point locations are latent,  
1304 or unobserved, and we only obtain imperfect information about the point locations  
1305 by observing individuals at trap or observation locations. Thus, the realized loca-  
1306 tions of individuals represent a type of “thinned” point process, where the thinning  
1307 mechanism is not random but, rather, biased by the observation mechanism. It is  
1308 also natural to think about the observed point process as some kind of a compound  
1309 or aggregate point process with a set of “parent” nodes being the locations of in-  
1310 dividual home ranges or their centroids, and the observed locations as “offspring”  
1311 - i.e., a Poisson cluster process (PCP). In that context, density estimation in SCR  
1312 models is analogous to estimating the number of parents of a Poisson cluster process  
1313 (Chandler and Royle, 2013).

1314 Most of the recent developments in modeling and inference from spatial en-  
1315 counter history data, including most methods discussed in this book, are predicated  
1316 on the view that individuals are organized in space according to a relatively simple  
1317 point process model. More specifically, we assume that the collection of individ-

1318 ual activity centers are independent and identically distributed random variables  
 1319 distributed uniformly over some region. This is consistent with the assumption  
 1320 that the activity centers represent the realization of a Poisson point process or, if  
 1321 the total number of activity centers fixed, then this is usually referred to as a  
 1322 binomial point process.

### 1323 **1.5.3 The activity center concept**

1324 In the context of SCR models, and because most animals we study by capture-  
 1325 recapture are not sessile, there is not a unique and precise mathematical definition  
 1326 of the point locations  $\mathbf{s}$ . Rather, we imagine these to be the centroid of individ-  
 1327 uals home ranges, or the centroid of an individual's activities during the time of  
 1328 sampling, or even it's average location measured with error (e.g., from a long series  
 1329 of telemetry measurements). In general, this point is unknown for any individual  
 1330 but if we could track an individual over time and take many observations then we  
 1331 could perhaps get a good idea of where that point is. We'll think of the collection  
 1332 of these points as defining the spatial distribution of individuals in the population.

1333 We use the terms home range or activity center interchangeably. The term  
 1334 "home range center" suggests that models are only relevant to animals that exhibit  
 1335 behavior of establishing home ranges or territories, or central place foragers, and  
 1336 since not all species do that, perhaps the construction of SCR models based on this  
 1337 idea is flawed. However, the notion of a home range center is just a conceptual  
 1338 device and we don't view this concept as being strictly consistent with classical  
 1339 notions of animal territories. Rather our view is that a home range or territory  
 1340 is inherently dynamic, temporally, and thus it is a transient quantity - where the  
 1341 animal lived during the period of study, a concept that is completely analogous to  
 1342 the more conventional notion of utilization distributions. Therefore, whether or not  
 1343 individuals of a species establish home ranges is irrelevant because, once a precise  
 1344 time period is defined, this defines a distinct region of space that an individual must  
 1345 have occupied.

### 1346 **1.5.4 The state-space**

1347 Once we introduce the collection of activity centers,  $\mathbf{s}_i; i = 1, 2, \dots, N$ , then the  
 1348 question "what are the possible values of  $\mathbf{s}$ ?" needs to be addressed because the  
 1349 individual  $\mathbf{s}_i$  are *unknown*. As a technical matter, we will regard them as random  
 1350 effects and in order to apply standard methods of statistical inference we need to  
 1351 provide a distribution for these random effects. In the context of the point process  
 1352 model, the possible values of the point locations referred to as the "state-space" of  
 1353 the point process and this is some region or set of points which we will denote by  
 1354  $\mathcal{S}$ . This is analogous to what is sometimes called the *observation window* for  $\mathbf{s}$  in  
 1355 the point process literature. The region  $\mathcal{S}$  serves as a prior distribution for  $\mathbf{s}_i$  (or,  
 1356 equivalently, the random effects distribution). In animal studies, as a description

1357 of where individuals that could be captured are located, it includes our study area,  
 1358 and should accommodate all individuals that could have been captured in the study  
 1359 area. In the practical application of SCR models, in most cases estimates of density  
 1360 will be relatively insensitive to choice of state-space which we discuss further in  
 1361 Chapt. 5 and elsewhere.

1362 **1.5.5 Abundance and density**

1363 When the underlying point process is well-defined, including a precise definition  
 1364 of the state-space, this in turn induces a precise definition of the parameter  $N$ ,  
 1365 “population size”, as the number of individual activity centers located within the  
 1366 prescribed state-space, and its direct linkage to density,  $D$ . That is, if  $A(\mathcal{S})$  is the  
 1367 area of the state-space then

$$D = \frac{N}{A(\mathcal{S})}.$$

1368 A deficiency with some classical methods of “adjustment” is they attempted to  
 1369 prescribe something like a state-space - a “sampled area” - except absent any pre-  
 1370 cise linkage of individuals with the state-space. SCR models formalize the linkage  
 1371 between individuals and space and, in doing so, provide an explicit definition of  $N$   
 1372 associated with a well-defined spatial region, and hence density. That is, the pro-  
 1373 vide a model in which  $N$  scales, as part of the model, with the size of the prescribed  
 1374 state-space. In a sense, the whole idea of SCR models is that by defining a point  
 1375 process and its state-space  $\mathcal{S}$ , this gives context and meaning to  $N$  which can be  
 1376 estimated directly for that specific state-space. Thus, it is fixing  $\mathcal{S}$  that resolves  
 1377 the problem of “unknown area” that we have previously discussed.

## 1.6 CHARACTERIZATION OF SCR MODELS

1378 Formulation of capture-recapture models conditional on the latent point process is  
 1379 the critical and unifying element of *all* SCR models. However, SCR models differ  
 1380 in how the underlying process model is formulated, and its complexity. Most of the  
 1381 development and application of SCR models has focused on their use to estimate  
 1382 density and touting the fact that they resolve certain specific technical problems  
 1383 related to the use of ordinary capture-recapture models. This is achieved with a sim-  
 1384 ple process model being a basic point process of independently distributed points.  
 1385 At the same time, there are models of CR data that focus exclusively on *movement*  
 1386 modeling, or models with explicit dynamics (Ovaskainen, 2004; Ovaskainen et al.,  
 1387 2008). Conceptually, these are akin to spatial versions of so-called Cormack-Jolly-  
 1388 Seber (CJS) models in the traditional capture-recapture literature, except they  
 1389 involve explicit mathematical models of movement based on diffusion or Brownian  
 1390 motion. Finally, there are now a very small number of papers that focus on *both*  
 1391 movement and density simultaneously (Royle and Young, 2008; Royle et al., 2011a;

1392 Royle and Chandler, 2012) or population dynamics and density (Gardner et al.,  
1393 2010b).

1394 A key thing is that these models, whether focused just on density, or just on  
1395 movement, or both, are similar models in terms of the underlying concepts, the  
1396 latent structure, and the observation model. They differ primarily in terms of the  
1397 ecological focus. Understanding movement is an important topic in ecology, but  
1398 models that strictly focus on movement will be limited by two practical consider-  
1399 ations: (1) most capture-recapture data e.g., by camera trapping or whatever,  
1400 produces only a few observations of each individual (between 1-5 would be typi-  
1401 cal). So there is not too much information about complex movement models. (2)  
1402 Typically people have an interest in density of individuals and therefore we need  
1403 models that can be extrapolated from the sample to the unobserved part of the  
1404 population. That said, there are clearly some cases where more elaborate move-  
1405 ment models should come into play. If one has some telemetry data in addition to  
1406 SCR then there is additional information on fine-scale movements that should be  
1407 useful.

## 1.7 SUMMARY AND OUTLOOK

1408 Spatial capture-recapture models are an extension of traditional capture-recapture  
1409 models to accommodate the spatial organization of both individuals in a population  
1410 and the observation mechanism (e.g., locations of traps). They resolve problems  
1411 which have been recognized historically and for which various ad hoc solutions  
1412 have been suggested: heterogeneity in encounter probability due to the spatial  
1413 organization of individuals relative to traps, the need to model trap-level effects  
1414 on encounter, and that a well-defined sample area does not exist in most studies,  
1415 and thus estimates of  $N$  using ordinary capture-recapture models cannot be related  
1416 directly to density.

1417 As we have shown already, SCR models are not simply an extension of a tech-  
1418 nique to resolve certain technical problems. Rather, they provide a coherent, flex-  
1419 ible framework for making ecological processes explicit in models of individual  
1420 encounter history data, and for studying animal populations processes such as individ-  
1421 ual movement, resource selection, space usage, population dynamics, and density.  
1422 Historically, researchers studied these questions independently, using ostensibly un-  
1423 related study designs and statistical procedures. For example, resource selection  
1424 function (RSF) models for resource selection, state-space models for movement,  
1425 density using closed capture-recapture methods, and population dynamics with  
1426 various “open” capture-recapture models. SCR can bring all of these problems  
1427 together into a single unified framework for modeling and inference. Most impor-  
1428 tantly, spatial capture-recapture models promise the ability to integrate explicit  
1429 ecological theories directly into the models so that we can directly test hypoth-  
1430 eses about either space usage (e.g., Chapt. 13), landscape connectivity (Chapt.  
1431 12), movement, or spatial distribution (Chapt. 11). We imagine that, in the near

1432 future, SCR models will include point process models that allow for interactions  
1433 among individuals such as inhibition or clustering (Reich et al., 2012). In the  
1434 following chapters we develop a comprehensive synthesis and extension of spatial  
1435 capture-recapture models as they presently exist, and we suggest areas of future  
1436 development and needed research.

1437  
1438

---

1439

## 2

# STATISTICAL MODELS AND SCR

1440 In the previous chapter we described the basics of capture-recapture methods and  
1441 the advantages that spatial models have over traditional non-spatial models. We  
1442 avoided statistical terminology like the plague so that we could focus on a few key  
1443 concepts. Although it is critical to understand the non-technical motivation for this  
1444 broad class of models, it is impossible to fully appreciate them, and apply them to  
1445 real data, without a solid grasp of the fundamentals of statistical inference.

1446 In this chapter, we present a brief overview of the basic statistical principals that  
1447 are referenced throughout the remainder of this book. Emphasis is placed on the  
1448 definition of a random variable, the common probability distributions used to model  
1449 random variables, and how hierarchical models can be used to describe conditionally  
1450 related random variables. For some readers, this material will be familiar, perhaps  
1451 even elementary, and thus you may want to skip to the next chapter. However, our  
1452 experience is that many basic statistics courses taken by ecologists do not emphasize  
1453 the important subjects covered in this chapter. Instead, there seems to be much  
1454 attention paid to minor details such as computing the number of degrees of freedom  
1455 in various  $F$ -tests, which, although useful in some contexts, do not provide the basis  
1456 for drawing conclusions from data and evaluating scientific hypotheses.

1457 The material in the beginning of this chapter is explained in numerous other  
1458 texts. Technical treatments that emphasize ecological problems are given by Williams  
1459 et al. (2002), Royle and Dorazio (2008) and Link and Barker (2010), to name just  
1460 a few. A very accessible introduction to some of the topics covered in this chapter  
1461 is presented in Chapt. 3 of MacKenzie et al. (2006). With all these resources, one  
1462 might wonder why we bother rehashing these concepts here. Our motivation is  
1463 two-fold: first, we wish to develop this material using examples relevant to spatial  
1464 capture-recapture, and second, we find that most introductory texts are not accom-  
1465 panied by code that can be helpful to the novice. We therefore attempt to present

1466 simple **R** code throughout this chapter so that those who struggle with equations  
1467 and mathematical notation can learn by doing. As mentioned in the Preface, we  
1468 rely on **R** because it provides tremendous flexibility for analyzing data and because  
1469 it is free. We do not, however, try to explain how to use **R** because there are so  
1470 many good references already, including Venables and Ripley (2002); Bolker (2008);  
1471 Venables et al. (2012).

1472 After covering some basic concepts of hierarchical modeling, we end the chapter  
1473 by describing spatial capture-recapture models using hierarchical modeling nota-  
1474 tion. This makes the concepts outlined in the previous chapter more precise, and  
1475 it highlights the fact that SCR models include explicit models for the ecological  
1476 processes of interest (e.g. spatial variation in density) and the observation process,  
1477 which describes how individuals are encountered.

## 2.1 RANDOM VARIABLES AND PROBABILITY DISTRIBUTIONS

### 1478 2.1.1 Stochasticity in ecology

1479 Few ecological processes can be described using purely deterministic models, and  
1480 thus we need a formal method for drawing conclusions from data while acknowl-  
1481 edging the stochastic nature of ecological systems. This is the role of statistical  
1482 inference, which is founded on the laws of probability. For our purposes, it suffices  
1483 to be familiar with a small number of concepts from probability theory—the most  
1484 important of which is the concept of a random variable, say  $X$ . A random variable  
1485 is a variable whose realized value is the outcome of some stochastic process. To  
1486 be more precise, a random variable is characterized by a function that describes  
1487 the probability of observing the value  $x$ . This probability function can be written  
1488  $\Pr(X = x|\theta)$  where  $\theta$  is a parameter, or set of parameters of the function. If  $x$  is  
1489 discrete, e.g. binary or integer, then we call the probability function a probability  
1490 mass function (pmf). If  $x$  is continuous, the function is called a probability density  
1491 function (pdf).

1492 To clarify the concept of a random variable, let  $X$  be the number of American  
1493 shad (*Alosa sapidissima*) caught after  $K = 20$  casts at the shad hole on Deerfield  
1494 River in Massachusetts. Suppose that we had a good day and caught  $x = 7$  fish.  
1495 If there were no random variation at play, we would say that the probability of  
1496 catching a fish, which we will call  $p$ , is  $p = 7/20 = 0.35$ , and we would always  
1497 expect to catch 7 shad after 20 casts. In other words, our deterministic model is  
1498  $x = 0.35 \times K$ . In reality, however, we can be pretty sure that this deterministic  
1499 model would not be very good. Even if we knew for certain that  $p \equiv 0.35$ , we would  
1500 expect some variation in the number of fish caught on repeated fishing outings.  
1501 To describe this variation, we need a model that acknowledges uncertainty (i.e.,  
1502 stochasticity), and specifically we need a model that describes the probability of  
1503 catching  $x$  fish given  $K$  and  $p$ ,  $\Pr(X = x|K, p)$ . Since  $x$  is discrete, not continuous,  
1504 we need a pmf. Before contemplating which pmf is most appropriate in this case,

1505 we need to first mention a few issues related to notation.

1506 Statisticians make things easier for themselves, and more complicated for ev-  
 1507 eryone else, by using different notation for probability distributions. Sometimes  
 1508 you will see  $\Pr(X = x|K, p)$  expressed as  $f(X|K, p)$  or  $f(X; K, p)$  or  $p(X|K, p)$  or  
 1509  $\pi(X|K, p)$  or  $\mathbb{P}(X|K, p)$  or  $[X|K, p]$  or even just  $[X]!$  Just remember that these  
 1510 expressions all have the same meaning—they are all probability distributions that  
 1511 tell us the probability of observing any possible realization of the random variable  
 1512  $X$ . In this book, we will almost always use bracket notation (the last two examples  
 1513 above) to represent arbitrary probability distributions. Hence, from here on out,  
 1514 when you see  $[X|K, p]$ , just remember that this is equivalent to the more traditional  
 1515 expression  $\Pr(X = x|K, p)$ . In addition, from here on, to achieve a more concise  
 1516 presentation, we will no longer use uppercase letters to denote random variables  
 1517 and lowercase letters for realized values. Rather, we will define a random vari-  
 1518 able by some symbol ( $x$ ,  $N$ , etc...) and let the context determine whether we are  
 1519 talking about the random variable itself, or realized values of it. In some limited  
 1520 cases, we will want upper- and lower-case letters to represent different variables.  
 1521 For example, we will often let  $N$  denote population size and  $n$  denote the number  
 1522 of individuals actually detected.

1523 When we wish to be specific about a probability distribution, we will do so in  
 1524 one of two ways, one mathematically precise and one symbolic. Before explaining  
 1525 these two options, let's choose a specific distribution as a model for the data in our  
 1526 example. In this case, the natural choice for  $[x|K, p]$  is the binomial distribution,  
 1527 the mathematically precise representation of which is

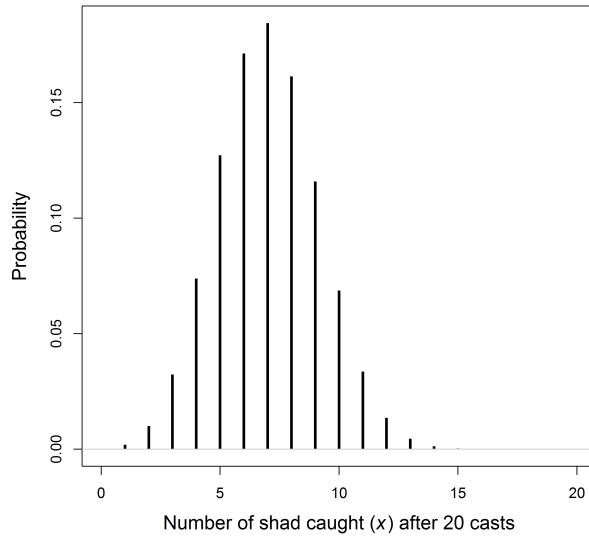
$$[x|K, p] = \binom{x}{K} p^x (1-p)^{K-x}. \quad (2.1.1)$$

1528 The right-hand side of this equation is the binomial pmf (described in more detail  
 1529 in Sec. 2.2), and plugging in values for the parameters  $K$ , and  $p$  will return the  
 1530 probability of observing any realized value of the random variable  $x$ . This is precise,  
 1531 but it is also cumbersome to write repetitively, and it may make the eyes glaze over  
 1532 when seen too often. Thus, we will often simplify Eq. 2.1.1 using the symbolic  
 1533 notation:

$$x \sim \text{Binomial}(K, p) \quad (2.1.2)$$

1534 The “ $\sim$ ” symbol is meant to represent a stochastic relationship, and can be read  
 1535 “is distributed as.” Another reason for using this notation is that it resembles the  
 1536 syntax of the **BUGS** language, which we will frequently use to conduct Bayesian  
 1537 inference.

1538 Note that once we choose a probability distribution, we have chosen a model. In  
 1539 our example, we have specified our model as  $x \sim \text{Binomial}(K, p)$ , and because we  
 1540 are assuming that the parameters are known, we can make probability statements  
 1541 about future outcomes. Continuing with our fish example, we might want to know  
 1542 the probability of catching  $x = 7$  again after  $K = 20$  casts on a future fishing



**Figure 2.1.** The binomial probability mass function with  $N = 20$  and  $p = 0.35$ .

1543 outing, assuming that we know  $p = 0.35$ . Evaluating the binomial pmf returns a  
 1544 probability of approximately 0.18, as show using this bit of **R** code:

```
1545 > dbinom(7, 20, 0.35)
1546 [1] 0.1844012
```

1547 By definition, the pmf allows us to evaluate the probability of observing any  $x$  given  
 1548  $K = 20$  and  $p = 0.35$ , thus the distribution of the random variable can be visualized  
 1549 by evaluating it for all values of  $x$  that have non-negligible probabilities, as can be  
 1550 easily done in **R**:

```
1551 plot(0:20, dbinom(0:20, 20, 0.35), type="h", ylab="Probability",
1552       xlab="Number of shad caught (X)")
```

1553 the result of which is shown in Fig. 2.1 with some extra details.

1554 The purpose of this little example is to show that once we specify a model for the  
 1555 random variable(s) being studied, we can begin drawing conclusions, i.e. making  
 1556 inferences, about the processes of interest, even in the face of uncertainty. Prob-  
 1557 ability distributions are essential to this process, and thus we need to understand  
 1558 them in more depth.

**Table 2.1.** Common probability density functions (pdfs) and probability mass functions (pmfs) used throughout this book.

Distribution	Notation	pmf or pmf	Support	Mean $\mathbb{E}(x)$	Variance $\text{Var}(x)$
Discrete random variables					
Poisson	$x \sim \text{Pois}(\lambda)$	$\exp(-\lambda)\lambda^x/x!$	$x \in \{0, 1, \dots\}$	$\lambda$	$\lambda$
Bernoulli	$x \sim \text{Bern}(p)$	$p^x(1-p)^{1-x}$	$x \in \{0, 1\}$	$p$	$p(1-p)$
Binomial	$x \sim \text{Bin}(N, p)$	$\binom{N}{x} p^x (1-p)^{N-x}$	$x \in \{0, 1, \dots, N\}$	$Np$	$Np(1-p)$
Multinomial	$\mathbf{x} \sim \text{Multinom}(N, \boldsymbol{\pi})$	$\binom{N}{x_1 \dots x_k} \pi_1^{x_1} \dots \pi_k^{x_k}$	$x_k \in \{0, 1, \dots, N\}$	$N\pi_k$	$N\pi_k(1 - \pi_k)$
Continuous random variables					
Normal	$x \sim \text{N}(\mu, \sigma^2)$	$\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$	$x \in [-\infty, \infty]$	$\mu$	$\sigma^2$
Uniform	$x \sim \text{Unif}(a, b)$	$\frac{1}{b-a}$	$x \in [a, b]$	$(a+b)/2$	$(b-a)^2/12$
Beta	$x \sim \text{Beta}(a, b)$	$\frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1} (1-x)^{b-1}$	$x \in [0, 1]$	$a/(a+b)$	$\frac{ab}{(a+b)^2(a+b+1)}$
Gamma	$x \sim \text{Gamma}(a, b)$	$\frac{b^a}{\Gamma(a)} x^{a-1} \exp(-bx)$	$x \in [0, \infty]$	$a/b$	$a/b^2$
Multivariate Normal	$\mathbf{x} \sim \text{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	$(2\pi)^{-k/2}  \boldsymbol{\Sigma} ^{-1/2} \exp(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})' \times \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}))$	$x_k \in [-\infty, \infty]$	$\boldsymbol{\mu}$	$\boldsymbol{\Sigma}$

1559 **2.1.2 Properties of probability distributions**

1560 A pdf or a pmf is a function like any other function in the sense that it has one  
 1561 or more arguments whose values determine the result of the function. However,  
 1562 probability functions have a few properties that distinguish them from other func-  
 1563 tions. The first is that the function must be non-negative for all possible values of  
 1564 the random variable, i.e.  $[x] \geq 0$ . The second requirement is that the integral of  
 1565 a pdf must be unity,  $\int_{-\infty}^{\infty} [x] dx = 1$ , and similarly for a pmf, the summation over  
 1566 all possible values is unity,  $\sum_x [x] = 1$ . The following **R** code demonstrates this for  
 1567 the normal and binomial distributions:

```
1568 > integrate(dnorm, -Inf, Inf, mean=0, sd=1)$value
1569 [1] 1
1570 > sum(dbinom(0:5, size=5, p=0.1))
1571 [1] 1
```

1572 This requirement is important to remember when one develops a non-standard  
 1573 probability distribution. For example, in Chapt. 11 and 13, we work with resource  
 1574 selection functions whose probability density function is not one that is pre-defined  
 1575 in software packages such as **R** or **BUGS**.

1576 Another feature of probability distributions is that they can be used to compute  
 1577 important summaries of random variables. The two most important summaries  
 1578 are the expected value,  $\mathbb{E}(x)$ , and the variance  $\text{Var}(x)$ . The expected value, or  
 1579 mean, can be thought of as the average of a very large sample from the specified  
 1580 distribution. For example, one way of approximating the expected values of a  
 1581 binomial distribution with  $K = 20$  trials and  $p = 0.35$  can be implemented in  
 1582 **R** using:

```
1583 > mean(rbinom(10000, 20, 0.3))
1584 [1] 6.9865
```

1585 For most probability distributions used in this book, the expected values are known  
 1586 exactly, as shown in Table 2.1, and thus we don't need to resort to such Monte Carlo  
 1587 approximations. For instance, the expected value of the binomial distribution is  
 1588 exactly  $\mathbb{E}(x) = Kp = 20 \times 0.35 = 7$ . In this case, it happens to take an integer  
 1589 value, but this is not a necessary condition, even for discrete random variables.

1590 A more formal definition of an expected value is the average of all possible  
 1591 values of the random variable, weighted by their probabilities. For continuous  
 1592 random variables, this weighted average is found by integration:

$$\mathbb{E}(x) = \int_{-\infty}^{\infty} x \times [x] dx. \quad (2.1.3)$$

1593 For example, if  $[x]$  is normally distributed with mean 3 and unit variance, we could  
 1594 find the expected value using the following code.

---

```
1595 > integrate(function(x) x*dnorm(x, 3, 1), -Inf, Inf)
1596 3 with absolute error < 0.00033
```

1597 Of course, the mean *is* the expected value of the normal distribution, so we didn't  
 1598 need to compute the integral but, the point is, that Eq. 2.1.3 is generic. For  
 1599 discrete random variables, the expected value is found by summation rather than  
 1600 integration:

$$\mathbb{E}(x) = \sum_x x \times [x] \quad (2.1.4)$$

1601 where the summation is over all possible values of  $x$ . Earlier we approximated the  
 1602 expected value of the binomial distribution with  $K = 20$  trials and  $p = 0.35$  by  
 1603 taking a Monte Carlo average. Eq. 2.1.4 let's us find the exact answer, using this  
 1604 bit of R code:

```
1605 > sum(dbinom(0:100, 20, 0.35)*0:100)
1606 [1] 7
```

1607 This is great. But of what use is it? One very important concept to understand is  
 1608 that when we fit models, we are often modeling changes in the expected value of  
 1609 some random variable. For example, in Poisson regression, we model the expected  
 1610 value of the random variable, which may be a function of environmental variables.

1611 The ability to model the expected value of a random variable gets us very far,  
 1612 but we also need a model for the variance of the random variable. The variance  
 1613 describes the amount of variation around the expected value. Specifically,  $\text{Var}(x) =$   
 1614  $\mathbb{E}((x - \mathbb{E}(x))^2)$ . Clearly, if the variance is zero, the variable is not random as  
 1615 there is no uncertainty in its outcome. For some distributions, notably the normal  
 1616 distribution, the variance is a parameter to be estimated. Thus, in ordinary linear  
 1617 regression, we estimate both the expected value  $\mu = \mathbb{E}(x)$ , which may be a function  
 1618 of covariates, and the variance  $\sigma^2$ , or similarly the residual standard error  $\sigma$ . For  
 1619 other distributions, the variance is not an explicit parameter to be estimated, and  
 1620 instead, the mean to variance ratio is fixed. In the case of the Poisson distribution,  
 1621 the mean is equal to the variance,  $\mathbb{E}(x) = \text{Var}(x) = \lambda$ . A similar situation is true  
 1622 for the binomial distribution—the variance is determined by the two parameters  $K$   
 1623 and  $p$ ,  $\text{Var}(x) = Kp(1-p)$ . In our earlier example with  $K = 20$  and  $p = 0.35$ , the  
 1624 variance is 4.55. Toying around with these ideas using random number generators  
 1625 may be helpful. Here is some code to illustrate some of these basic concepts:

```
1626 > 20*0.35*(1-0.35)                      # Exact variance, Var(x)
1627 [1] 4.55
1628 > x <- rbinom(100000, 20, 0.35)
1629 > mean((x-mean(x))^2)                   # Monte Carlo approximation
1630 [1] 4.545525
```

## 2.2 COMMON PROBABILITY DISTRIBUTIONS

1631 We got a little ahead of ourselves in the previous sections by using the binomial  
 1632 and Poisson distributions without describing them in detail. A solid understanding  
 1633 of the binomial, Poisson, multinomial, uniform, and normal (or Gaussian) distri-  
 1634 butions is absolutely essential throughout the remainder of the book. We will  
 1635 occasionally make use of other distributions such as the beta, log-normal, gamma,  
 1636 Dirichlet, etc... that can be helpful when modeling capture-recapture data, but  
 1637 these distributions can be readily understood once you are comfortable with the  
 1638 more commonly used distributions described in this section.

1639 **2.2.1 The binomial distribution**

1640 The binomial distribution plays a critical role in ecology. It is used for purposes  
 1641 as diverse as modeling count data, survival probability, occurrence probability, and  
 1642 capture probability, just to name a few. To describe the properties of the binomial  
 1643 distribution, and related distributions, we will introduce a new example. Suppose  
 1644 we are conducting a bird survey at a site in which  $N = 10$  chestnut-sided warblers  
 1645 (*Setophaga pensylvanica*) occur, and each of these individuals has a detection prob-  
 1646 ability of  $p = 0.5$ . The binomial distribution is the natural choice for describing  
 1647 the number of individuals that we would expect to detect ( $n$ ) in this situation, and  
 1648 using our notation, we can write the model as:  $n \sim \text{Binomial}(10, 0.5)$ . When  $p < 1$ ,  
 1649 we can expect that we will observe a different number of warblers on each of  $K$   
 1650 replicate survey occasions. To see this, we simulate data under this simple model  
 1651 with  $K = 3$ .

```
1652 > n <- rbinom(3, size=10, prob=0.5) # Generate 3 binomial outcomes
1653 > n                               # Display the 3 values
1654 [1] 6 4 8
```

1655 The vector of counts will typically differ each time you issue this command; however,  
 1656 we know the probability of observing any value of  $n_k$  because it is defined by the  
 1657 binomial pmf. As we demonstrated earlier, in R this probability can be found using  
 1658 the `dbinom` function. For example, the probability of observing  $n_k = 5$  is given by:

```
1659 > dbinom(5, 10, 0.5)
```

1660 This simply evaluates the function shown in Table 2.1. We could do the same more  
 1661 transparently, but less efficiently, using any of the following:

```
1662 > n <- 5; N <- 10; p <- 0.5
1663 > factorial(N)/(factorial(n)*factorial(N-n))*p^n*(1-p)^(N-n)
1664 > exp(lgamma(N+1) - (lgamma(n+1) + lgamma(N-n+1)))*p^n*(1-p)^(N-n)
1665 > choose(N, n)*p^n*(1-p)^(N-n)
```

1666 Note that the last three lines of code differ only in how they compute the binomial  
 1667 coefficient  $\binom{N}{n}$ , which is the number of different ways we could observe  $n = 5$  of  
 1668 the  $N = 10$  chestnut-sided warblers at the site. The binomial coefficient, which is  
 1669 read “N choose n” is defined as

$$\binom{N}{n} = \frac{N!}{n!(N-n)!}. \quad (2.2.1)$$

1670 Now that we know how to simulate binomial data and compute the probabilities  
 1671 of observing any particular outcome  $n$ , conditional on the parameters  $N$  and  
 1672  $p$ , we can contemplate the relevance of the binomial distribution in spatial capture-  
 1673 recapture models. One important application of the binomial distribution is as a  
 1674 model encounter frequencies. Indeed, one of the most important encounter models  
 1675 in SCR will be referred to as the “binomial encounter model”, in which the number  
 1676 of times individual  $i$  is captured at “trap”  $j$  after  $K$  survey occasions is modeled as  
 1677  $y_{ij} \sim \text{Binomial}(K, p_{ij})$ . Here,  $p_{ij}$  is the encounter probability determined, in part,  
 1678 by the distance between an animal’s activity center and the trap location. This  
 1679 binomial encounter model is described in detail in Sec. 7.1. Another important application  
 1680 of the binomial distribution is as a prior for the population size parameter  
 1681 in Bayesian analyses, as is discussed in Chapt. 4.

### 1682 2.2.2 The Bernoulli distribution

1683 Above, we showed 3 alternatives to `dbinom` for evaluating the binomial pmf. These  
 1684 three commands differed only in how they computed the binomial coefficient, which  
 1685 we needed because of the numerous ways in which we could observe  $n = 5$  given  
 1686  $N = 10$ . To conceptualize this, let  $y_i$  be a binary variable indicating if individual  $i$   
 1687 was detected or not. Hence, given that 5 individuals were detected, the vector of  
 1688 individual detections could be something like  $\mathbf{y} = (0, 0, 1, 1, 1, 1, 0, 0, 0)$ , indicating  
 1689 that we detected individuals 3-7 but not 1-2 or 8-10. For  $N = 10$  and  $n = 5$ ,  
 1690 the binomial coefficient tells us that there are 252 possible vectors  $\mathbf{y}$  with 5 ones.  
 1691 However, when  $N \equiv 1$ , this term drops from the pmf and the result is the pmf for  
 1692 the Bernoulli distribution. That is, the Bernoulli distribution is simply the binomial  
 1693 distribution when  $N \equiv 1$ . Alternatively, we could say that the binomial distribution  
 1694 is the outcome of  $N$  iid Bernoulli trials. We use the standard abbreviation “iid”  
 1695 to mean *independent, identically distributed*.

1696 The utility of the Bernoulli distribution is evident when we imagine that not all  
 1697 of the chestnut-sided warblers have the same detection probability. Thus, if some  
 1698 individuals can be detected with probability 0.3 and others have a 0.7 detection  
 1699 probability, then the model  $n \sim \text{Binomial}(N, p)$  is no longer an accurate description  
 1700 of system since  $p$  is no longer constant for all individuals.

To properly account for variation in  $p$ , we could redefine our model for the

counts of chestnut-sided warblers as

$$\begin{aligned} y_{ik} &\sim \text{Bernoulli}(p_i) \\ n_k &= \sum_{i=1}^N y_{ik} \end{aligned} \tag{2.2.2}$$

1701 This states that individual  $i$  is detected with probability  $p_i$ , and the observed count  
 1702 is the sum of the  $N$  Bernoulli outcomes.

1703 An important point is that the individual-specific data  $y_{ik}$  can only be observed  
 1704 if the individuals are uniquely distinguishable, such as when they are marked by  
 1705 biologists with color bands. In such cases, the Bernoulli distribution allows us  
 1706 to model variation in detection probability among individuals and thus would be  
 1707 preferable to the binomial distribution, which assumes that each of the  $N$  indi-  
 1708 viduals have the same  $p$ . For this reason, the Bernoulli distribution, as simple as  
 1709 it is, is of paramount importance in capture-recapture models, including spatial  
 1710 capture-recapture models in which there is virtually always substantial and impor-  
 1711 tant variation in capture probability among individuals. Indeed, it could be said  
 1712 that the Bernoulli model is the canonical model in capture-recapture studies, and  
 1713 most of the different flavors of capture-recapture models differ primarily in how  $p_i$   
 1714 is specified.

1715 The Bernoulli pmf is given by  $p^n(1-p)^{1-n}$  and hence we do not need canned  
 1716 functions to facilitate its evaluation. Of course, if you wanted to, you could always  
 1717 use `dbinom` with the `size` argument set to 1. For example, `dbinom(1, 1, 0.3)`  
 1718 returns the Bernoulli probability of observing  $n = 1$  given  $p = 0.3$ .

### 1719 2.2.3 The multinomial and categorical distributions

1720 The binomial distribution is used when we are accumulating a binary response—  
 1721 that is, one in which there are two possible categories such as success/failure or  
 1722 captured/not-captured. The multinomial distribution is a multivariate extension  
 1723 of the binomial used when there are  $G > 2$  categories. The multinomial distribution  
 1724 can be thought of as a model for placing  $N$  items in the  $G$  categories, which are  
 1725 also called bins or cells. Each bin has its own probability  $\pi_g$  and these probabilities  
 1726 must sum to one. In ecology,  $N$  is often population size or the number of individuals  
 1727 detected, but the definition of the  $G$  bins varies among applications. For example,  
 1728 in distance sampling, when the distance data are aggregated into intervals, the  
 1729 bins are the distance intervals, and the cell probabilities are functions of detection  
 1730 probability in each interval (Royle et al., 2004).

1731 The multinomial distribution is widely used to model data from traditional,  
 1732 non-spatial capture-recapture studies. Earlier we let  $y_{ik}$  denote a binary random  
 1733 variable indicating if warbler  $i$  was detected on survey  $k$ . The vector of observations  
 1734 for an individual,  $\mathbf{y}_i$ , is often referred to as the individual's "encounter history".

1735 The number of possible encounter histories depends on  $K$ , the number of survey  
 1736 occasions. Specifically, there are  $2^K$  possible encounter histories<sup>1</sup>. If we tabulate the  
 1737 number of individuals with each encounter history, the frequencies can be modeled  
 1738 using the multinomial distribution.

1739 Going back to our chestnut-sided warbler example, suppose the 10 individuals  
 1740 are marked and we make  $K = 2$  visits to the site such that there are  $2^K = 4$  pos-  
 1741 sible encounter histories: (11, 10, 01, 00), where, for example, “10” is the encounter  
 1742 history for an individual detected on the first visit but not the second. If  $p = 1$ ,  
 1743 then the encounter history for each of the 10 individuals must be “11”. That is, we  
 1744 would detect each individual on both occasions. In this case, we the data would be:  
 1745  $\mathbf{h} = (10, 0, 0, 0)$ , which indicates that all 10 warblers had the first encounter history.  
 1746 The corresponding cell probabilities would be  $\boldsymbol{\pi} = (1, 0, 0, 0)$ . What about the sit-  
 1747 uation where  $p < 1$ , e.g.  $p = 0.3$ ? In this case, the probability of observing the  
 1748 capture history “11” (detected on both occasions) is  $p \times p = 0.3 \times 0.3 = 0.09$ . The  
 1749 probability of observing “10” is  $p \times (1 - p) = 0.21$ . Following this logic, the vector  
 1750 of cell probabilities is  $\boldsymbol{\pi} = (0.09, 0.21, 0.21, 0.49)$ . We can simulate data under this  
 1751 model as follows:

```
1752 > caphist.probs <- c("11"=0.09, "10"=0.21, "01"=0.21, "00"=0.49)
1753 > drop(rmultinom(1, 10, caphist.probs))
1754 11 10 01 00
1755 0 3 2 5
```

1756 The result of our simulation is that zero individuals were observed with the capture  
 1757 history “11” and 5 individuals were observed with the capture history “00”. The  
 1758 other 5 individuals were observed one out of the two occasions. This is not such a  
 1759 surprising outcome given  $p = 0.3$ .

1760 As in non-spatial capture-recapture studies, the multinomial distribution turns  
 1761 out to be very important in spatial capture-recapture studies. However,  $N$  is not  
 1762 defined as population size. Rather, we use the multinomial distribution when an  
 1763 individual can only be captured in a single trap during an occasion. Thus  $N = 1$   
 1764 and the cell probabilities are the probabilities of being captured in each trap. A  
 1765 thorough discussion of this point can be found in Chapt. 9. Another application  
 1766 of the multinomial distribution in SCR models is discussed in Chapt. 11 where we  
 1767 discuss how to model the probability that an individual’s activity center is located  
 1768 in one of the cells of a raster defining the spatial region of interest.

1769 Just as the Bernoulli distribution is the elemental form of the binomial distri-  
 1770 bution (being the case  $N = 1$ ), the categorical distribution is essentially equivalent  
 1771 to the multinomial distribution with size parameter  $N \equiv 1$ . The only difference is  
 1772 that, rather than returning a vector with a single element equal to 1, it returns the  
 1773 element *location* where the 1 occurs. For example, if  $\mathbf{y} = (0, 0, 1, 0)$  is an outcome

<sup>1</sup>When  $N$  is unknown, we can never observe the “all-0” encounter history, corresponding to an individual that is not detected, and thus the number of “observable” encounter histories is  $2^K - 1$

of a multinomial distribution with  $N = 1$ , then the categorical outcome would be 3 because the 1 is located in third position in the vector. Thus, in spatial capture-recapture models, we might use either the multinomial distribution with  $N = 1$  or the categorical distribution. The various **BUGS** engines describe the categorical distribution by the declaration `dcat` and, in **R**, we can simulate categorical outcomes using the function `sample` or as so:

```
> which(rmultinom(1, 1, c(0.1, 0.7, 0.2)) == 1)
[1] 2
```

## 2.2.4 The Poisson distribution

The Poisson distribution is the canonical model for count data in ecology. More generally, the Poisson distribution is a model for random variables taking on non-negative, integer values. Although it is a simple model having just one parameter,  $\lambda = \mathbb{E}(x) = \text{Var}(x)$ , its applications are highly diverse, including as a model of spatial variation in abundance or as a model for the frequency of behaviors over time. Just as logistic regression is the standard generalized linear model (GLM) used to model binary data, Poisson regression is the default GLM for modeling count data and variation in  $\lambda$ .

The Poisson distribution is related to both the binomial and multinomial distributions, and the following three bits of trivia are occasionally worth knowing. First, it is the limit of the binomial distribution as  $N \rightarrow \infty$  and  $p \rightarrow 0$ , which means that for high values of  $N$  and low values of  $p$ ,  $\text{Poisson}(N \times p)$  is approximately equal to  $\text{Binomial}(N, p)$ . Second, if  $\{n_1 \sim \text{Poisson}(\lambda_1), \dots, n_K \sim \text{Poisson}(\lambda_K)\}$  then the vector of counts is multinomial,  $\{n_1, \dots, n_K\} \sim \text{Multinomial}(\sum_k n_k, \{\frac{\lambda_1}{\sum_k \lambda_k}, \dots, \frac{\lambda_K}{\sum_k \lambda_k}\})$ . Third, the sum of two Poisson random variables  $x_1 \sim \text{Poisson}(\lambda_1)$  and  $x_2 \sim \text{Poisson}(\lambda_2)$  is also Poisson:  $x_1 + x_2 \sim \text{Poisson}(\lambda_1 + \lambda_2)$ .

The Poisson distribution has two important uses in spatial capture-recapture models: (1) as a prior distribution for the population size parameter  $N$ , and (2) as a model for the frequency of captures in a trap. In the first context, the Poisson prior for  $N$  results in a Poisson point process for the location of the  $N$  activity centers in the region of interest. This topic is discussed in Chapt. 5 and Chapt 11. The second use of the Poisson distribution in spatial capture-recapture is to describe data from sampling methods in which an individual can be detected multiple times at a trap during a single occasion. For example, in camera trapping studies we might obtain multiple pictures of the same individual at a trap during a single sampling occasion. Thus,  $\lambda$  in this case would be defined as the expected number of detections or captures per occasion.

## 2.2.5 The uniform distribution

The lowly uniform distribution is a continuous distribution whose only two parameters are the lower and upper bounds that restrict the possible values of the

random variable  $x$ . These bounds are almost always known, so there is typically nothing to estimate. Nonetheless, the uniform distribution is one of the most widely used distributions, especially among Bayesians who frequently use it to as a “non-informative” prior distribution for a parameter. For example, if we have a capture probability parameter  $p$  that we wish to estimate, but we have no prior knowledge of what value it may take in the range  $[0,1]$ , we will often use the prior  $p \sim \text{Uniform}(0,1)$ . This states that  $p$  is equally likely to take on any value between zero and one. Prior distributions are described in more detail in the next chapter.

Another common usage of the uniform distribution is as a prior for the coordinates of points in the real plane, i.e. in two-dimensional space. Such a use of the uniform distribution implies that a point process is “homogeneous”, meaning that the location of one point does not affect the location of another point and that the expected density of points is constant throughout the region. Thus, to simulate a realization from a homogeneous Poisson point process in the unit square  $[0, 1] \times [0, 1]$ , we could use the following **R** code:

```
1828 D <- 100      # points per unit area
1829 A <- 1        # Area of unit square
1830 N <- rpois(1, D*A)
1831 plot(s <- cbind(runif(N), runif(N)))
```

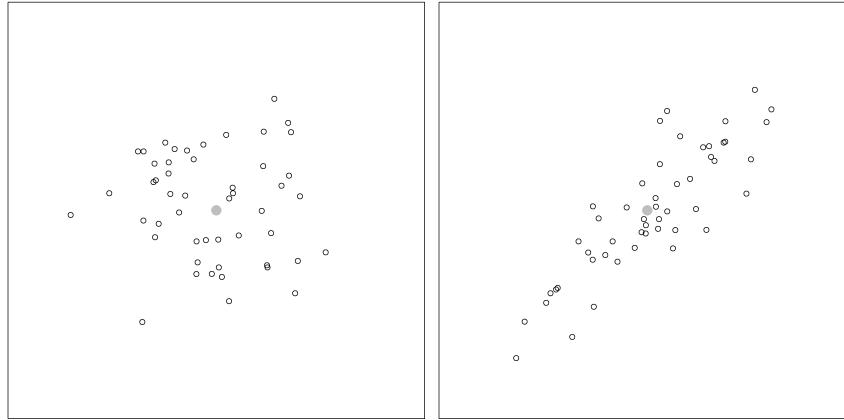
where  $\mathbf{s}$  is a matrix of coordinates with  $N$  rows and 2 columns. We will often represent the uniform point process using the following notation:

$$\mathbf{s} \sim \text{Uniform}(\mathcal{S}) \quad (2.2.3)$$

where  $\mathcal{S}$  is some specific unit of space called the state-space of the random variable  $\mathbf{s}$ . It would be more correct to somehow distinguish this two-dimensional uniform distribution for the univariate one. That is, it might be more clear to use notation such as  $\mathbf{s} \sim \text{Uniform}_2(\mathcal{S})$  instead, but this is somewhat cumbersome, so we will opt for the former expression.

### 2.2.6 Other distributions

The other continuous distributions that are regularly encountered in SCR models are primarily used as priors in Bayesian analyses, and thus we will avoid a lengthy discussion of their properties. The normal distribution, also called the Gaussian distribution, is perhaps the most widely recognized and applied probability model in statistics, but it plays only a minor role in SCR models other than as a model for signal strength in acoustic SCR models (Efford et al., 2009b; Dawson and Efford, 2009), and see Sec. 9.4. Nonetheless, it is the canonical prior for any continuous random variable with infinite support, and thus it is often used as a prior when applying Bayesian methods. One common usage is as a prior for the  $\beta$  coefficients of a linear model defining some parameter as a function of covariates (usually on



**Figure 2.2.** Two realized point patterns from the bivariate normal distribution.

a transformed scale). An example, including a cautionary note, is provided in Sec. 3.5.1. Be aware that although the normal distribution is typically parameterized in terms of the variance parameter  $\sigma^2$ , in the **BUGS** language, the inverse of the variance, or precision, is used instead,  $\tau = 1/\sigma^2$ . In **R**, the `dnorm` function requires the standard deviation  $\sigma$ , rather than the variance  $\sigma^2$ .

The bivariate normal distribution is a generalization of the normal distribution and a special case of the multivariate normal distribution whose pdf is shown in Table 2.1. The bivariate normal distribution is used to model two (possibly) dependent continuous variables whose symmetric variance-covariance matrix is denoted  $\Sigma$ . In SCR models, we most often use this model as a rudimentary description of movement outcomes about a home range center. If there is no correlation, then the model reduces to two independent normal draws along the coordinate axes. The following code generates bivariate normal outcomes with no correlation ( $\rho = 0$ ), as well as outcomes in which the correlation is  $\rho = 0.9$ .

```

1864 library(mvtnorm)
1865 set.seed(3)
1866 mu <- c(0,0)
1867 Sigma <- matrix(c(1, .9, .9, 1), 2, 2)
1868 X1 <- cbind(rnorm(50, mu[1], Sigma[1,1]), # No correlation (rho=0)
1869             rnorm(50, mu[2], Sigma[2,2]))
1870 X2 <- rmvnorm(50, mu, Sigma)           # rho=0.9

```

Fig. 2.2 shows the simulated points.

Several of the parameters in capture-recapture models do not have infinite support, but instead are probabilities restricted to the range  $[0, 1]$ , or are positive

1874 valued living between zero and  $\infty$ . The beta distribution is the standard prior  
 1875 used for probabilities because it can be used to express either a lack of knowledge  
 1876 or very precise knowledge about a parameter. For example, a Beta(1, 1) distribu-  
 1877 tion is equivalent to a Uniform(0, 1) distribution. However, unlike the uniform  
 1878 distribution, the beta distribution can be used as an informative prior; for exam-  
 1879 ple if published estimates of detection probability exist we can choose parameters  
 1880 of the beta distribution to reflect that. To gain some familiarity with the beta  
 1881 distribution, execute the following R commands:

```
1882 curve(dbeta(x, 1, 1), col="black", ylim=c(0,5))
1883 curve(dbeta(x, 10, 10), col="blue", add=TRUE)
1884 curve(dbeta(x, 10, 20), col="darkgreen", add=TRUE)
```

1885 Other parameters in SCR models are continuous but positive-valued and can be  
 1886 modeled using the gamma distribution. As with the beta distribution, the gamma  
 1887 distribution is typically favored over the uniform distribution when one is interested  
 1888 in using an informative prior. It is also frequently used as a vague prior for the  
 1889 inverse of variance parameters, but it is wise to compare this prior to a uniform to  
 1890 assess its influence on the posterior.

### 2.3 STATISTICAL INFERENCE AND PARAMETER ESTIMATION

1891 If the parameters of a statistical model were known with absolute certainty, then it  
 1892 would be possible to use pdfs and pmfs to make direct probability statements about  
 1893 unknowns such as future outcomes. However, we almost never know the actual  
 1894 values of parameters, and instead we have to estimate them from observations  
 1895 (i.e., data). Our inferences must then acknowledge the uncertainty associated with  
 1896 our imperfect knowledge of the parameters. Doing so is most often accomplished  
 1897 using one of two approaches: classical (frequentist) inference or Bayesian inference.  
 1898 These two modes of inference regard the uncertainty about parameters in entirely  
 1899 different ways. In the next chapter, we will review some of the important concepts  
 1900 in Bayesian inference, so here, we will focus on the frequentist perspective.

1901 Suppose we count oak trees at  $J$  sites, and the resulting data  $\{y_1, \dots, y_J\}$  can  
 1902 be assumed to be *iid* outcomes from some distribution, such as the Poisson with  
 1903 unknown parameter  $\lambda$ . We want to estimate this parameter. In classical inference,  
 1904 the only uncertainty about  $\lambda$  is that attributable to sampling. For instance, we can  
 1905 imagine repeatedly sampling the population (sites in this example) and obtaining  
 1906 sample-specific estimates of  $\lambda$ . Typically, we entertain the idea that there are an  
 1907 infinite number of possible samples and so we could obtain an infinite number of  
 1908 estimates:  $\{\hat{\lambda}_1, \hat{\lambda}_2, \dots, \hat{\lambda}_\infty\}$ . If these estimates are produced using the method  
 1909 of maximum likelihood, and as  $n$  tends to infinity, the distribution of estimates,  
 1910 called the sampling distribution, will be normally distributed with  $E(\hat{\lambda}) = \lambda$ . The  
 1911 standard deviation of the sampling distribution is called the standard error, which  
 1912 can also be estimated as part of the maximum likelihood procedure. Of course, we

1913 almost always have just a single sample of data, and hence a single  $\hat{\lambda}$  and a single  
 1914 estimate of the standard error. However, under the assumption of a normally  
 1915 distributed sampling distribution, we can construct a confidence interval that will  
 1916 include the true value of  $\lambda$  with coverage probability  $1 - \alpha$ , where  $\alpha$  is a prescribed  
 1917 value like 0.05. An important point is that there is no uncertainty associated with  
 1918 the actual parameter—it is regarded as a fixed value, and hence probability is only  
 1919 used to characterize the estimator via its sampling distribution.

1920 Maximum likelihood is heuristically a method of finding the most “likely” value  
 1921 of  $\lambda$ , given the observed data, and of characterizing the variance of the sampling dis-  
 1922 tribution. Of course, it also applies to cases where the observations are multivariate,  
 1923 or the probability distribution is a function of multiple parameters. Endless num-  
 1924 bers of textbooks and online resources are available for those interested in a detailed  
 1925 explanation of maximum likelihood. For our purposes, we wish to keep it simple  
 1926 and focus on *how* to do it. The first step is to define the likelihood function, which  
 1927 is the joint distribution of the data regarded as a function of the parameter(s). If  
 1928 the joint distribution of the observations is denoted by  $[y_1, y_2, \dots, y_n | \lambda]$ , we usually  
 1929 denote the likelihood by flipping the arguments:  $\mathcal{L}(\lambda | \mathbf{y}) = [\lambda | y_1, y_2, \dots, y_n]$ .

1930 If the observations are *iid*, the likelihood simplifies to

$$\mathcal{L}(\lambda | \mathbf{y}) = \prod_{i=1}^n [y_i | \lambda]. \quad (2.3.1)$$

1931 where  $[y_i | \lambda]$  is a probability distribution, like those discussed in the previous sec-  
 1932 tions. For example, if  $y_i$  is Poisson distributed, then  $[y_i | \lambda] = \text{Poisson}(\lambda) = \frac{\lambda^{y_i} e^{-\lambda}}{y_i!}$ .  
 1933 Although likelihoods are typically shown on the natural scale, we almost always  
 1934 maximize the logarithm of the likelihood to avoid computational problems that  
 1935 arise when multiplying very small probabilities. Thus, we rewrite Eq. 2.3.1 as

$$\ell(\lambda | \mathbf{y}) = \sum_{i=1}^n \log(f(y_i | \lambda)) \quad (2.3.2)$$

1936 Here is some simple **R** code to simulate independent Poisson outcomes and esti-  
 1937 mate  $\lambda$  (as though we did not know it) using the method of maximum likelihood.  
 1938 Actually, we will minimize the negative log-likelihood because it is equivalent and  
 1939 is the default for **R**’s optimizers like `optim` and `nls`.

```
1940 > lambda <- 3          # Actual parameter value
1941 > y1 <- rpois(100, lambda)    # Realized values (data)
1942 > negLogLike1 <- function(par) -sum(dpois(y1, par, log=TRUE))
1943 > starting.value <- c('lambda'=1)
1944 > optim(starting.value, negLogLike1)$par # MLE
1945   lambda
1946 3.039844
```

1947 Explicitly maximizing the likelihood, numerically, isn't actually necessary here be-  
 1948 cause the MLE of  $\lambda$  is given by the mean of the observations. A more interesting  
 1949 example is when there are covariates of  $\lambda$ . For example, suppose  $\lambda$  is a function of  
 1950 elevation and vegetation height according to:  $\log(\lambda_i) = \beta_0 + \beta_1 ELEV_i + \beta_2 VEGHT_i$ .  
 1951 This is a standard Poisson regression problem, with likelihood:

$$\mathcal{L}(\boldsymbol{\beta}|\mathbf{y}) = \prod_i \text{Poisson}(y_i|\lambda_i) \quad (2.3.3)$$

1952 This likelihood is almost identical to the previous one except that  $\lambda$  is now a  
 1953 function, and so we need to estimate the parameters of the function, i.e. the  $\beta$ 's.  
 1954 Some code to fit this model to simulated data is shown here:

```
1955 > nsites <- 100
1956 > elevation <- rnorm(100)
1957 > veght <- rnorm(100)
1958 > beta0 <- 1
1959 > beta1 <- -1
1960 > beta2 <- 0
1961 > lambda <- exp(beta0 + beta1*elevation + beta2*vegght)
1962 > y2 <- rpois(nsites, lambda)
1963 > negLogLike2 <- function(pars) {
1964   +   beta0 <- pars[1]
1965   +   beta1 <- pars[2]
1966   +   beta2 <- pars[3]
1967   +   lambda <- exp(beta0 + beta1*elevation + beta2*vegght)
1968   +   -sum(dpois(y2, lambda, log=TRUE))
1969   +
1970 > starting.values <- c('beta0'=0, 'beta1'=0, 'beta2'=0)
1971 > optim(starting.values, negLogLike2)$par
1972   beta0      beta1      beta2
1973   0.98457756 -1.03025173 -0.01218292
```

1974 We see that the maximum likelihood estimates (MLEs) are very close to the true  
 1975 parameter values.

In these examples, the parameters we estimated are called fixed effects by frequentists. Fixed effects are parameters that are not regarded as being random variables. A random effect, in contrast, is a parameter that can be regarded as the outcome of a random variable. For instance, we could entertain the idea that the intercept of our GLM differs among locations, and that its actual value is an outcome of a normal distribution with parameters  $\mu$  and  $\sigma^2$ . In this case,  $\beta_i$  would

be a random effect, and our model could be written:

$$\begin{aligned}y_i &\sim \text{Poisson}(\lambda_i) \\ \log(\lambda_i) &= \beta_0 + \beta_1 \text{ELEV}_i + \beta_2 \text{VEGHT}_i \\ \beta_i &\sim \text{Normal}(\mu, \sigma^2)\end{aligned}$$

1976 This is an example of a mixed effects model or a hierarchical model. How do we  
 1977 estimate the parameters of a model that includes random effects? Earlier the like-  
 1978 lihood function was written as the product of probabilities determined by a single  
 1979 pmf or pdf,  $[y|\lambda]$ , but now we have an additional random variable, and we are forced  
 1980 to think about conditional relationships, because  $y$  depends upon  $\beta_i$  and  $\beta_i$  depends  
 1981 upon other parameters, specifically  $\mu$  and  $\sigma^2$ . This type of conditional dependence  
 1982 among parameters is the essence of hierarchical models, and statistical analysis  
 1983 of hierarchical models requires that we discuss joint distributions, marginal distri-  
 1984 butions and conditional distributions. These concepts will be used extensively in  
 1985 Chapt. 6 where we demonstrate how to estimate parameters of hierarchical models  
 1986 using maximum likelihood.

## 2.4 JOINT, MARGINAL, AND CONDITIONAL DISTRIBUTIONS

1987 So far we have restricted our attention to situations in which we wish to make  
 1988 inference about a single random variable. However, in ecology, we often are inter-  
 1989 ested in multiple random variables and how they are related. Let  $Y$  be a random  
 1990 variable that may or may not be independent of  $X$  (here again we will distinguish  
 1991 between random variables and realized values for conceptual clarity). Inference  
 1992 about these two random variables can be made using the joint, marginal, or condi-  
 1993 tional distributions—or, we may make use of all of them depending on the question  
 1994 being asked. In the case of discrete random variables, the joint distribution is the  
 1995 probability that  $X$  takes on the value  $x$  and that  $Y$  takes on the value  $y$ , which  
 1996 is written  $[X = x, Y = y]$ . To clarify this concept, let's go back to our original  
 1997 example where  $X$  was the number of fish caught after 20 casts, which we said  
 1998 was an *iid* binomial random variable. Now, let's suppose that  $X$  depends on the  
 1999 random variable  $Y$ , which is the number of other fisherman at the hole. Specifi-  
 2000 cally, let's say that the probability of catching a fish  $p$  is related to  $Y$  according  
 2001 to  $\text{logit}(p) = -0.6 + -2y$ . Furthermore, let's make the intuitive assumption that  
 2002 the number of fishermen at the hole is a Poisson random variable with mean 0.6,  
 2003 i.e.  $Y \sim \text{Poisson}(0.6)$ . Our model is now fully specified, and so we can answer the  
 2004 question: “what is the probability of catching  $x$  fish and of there being  $y$  fishermen  
 2005 at the hole”. This joint distribution is given by the product of the binomial pmf  
 2006 (with  $p$  determined by  $y$ ) and the Poisson pmf with  $\lambda = 0.6$ . The following R code  
 2007 creates the joint distribution.

```
2008 > X <- 0:20 # All possible values of X
2009 > Y <- 0:10 # All possible values of Y
2010 > lambda <- 0.6
```

```

2011 > p <- plogis(-0.62 + -2*Y) # p as function of Y
2012 > round(p,2)
2013 [1] 0.35 0.07 0.01 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
2014 > joint <- matrix(NA, length(X), length(Y))
2015 > rownames(joint) <- paste("X=", X, sep="")
2016 > colnames(joint) <- paste("Y=", Y, sep="")
2017 >
2018 > # Joint distribution [X,Y]
2019 > for(i in 1:length(Y)) {
2020 +   joint[,i] <- dbinom(X, 20, p[i]) * dpois(Y[i], lambda)
2021 + }
2022 > round(joint,2)
2023   Y=0  Y=1  Y=2  Y=3  Y=4  Y=5  Y=6  Y=7  Y=8  Y=9  Y=10
2024 X=0  0.00 0.08 0.08 0.02  0  0  0  0  0  0  0
2025 X=1  0.00 0.12 0.02 0.00  0  0  0  0  0  0  0
2026 X=2  0.01 0.08 0.00 0.00  0  0  0  0  0  0  0
2027 X=3  0.02 0.04 0.00 0.00  0  0  0  0  0  0  0
2028 X=4  0.04 0.01 0.00 0.00  0  0  0  0  0  0  0
2029 X=5  0.07 0.00 0.00 0.00  0  0  0  0  0  0  0
2030 X=6  0.09 0.00 0.00 0.00  0  0  0  0  0  0  0
2031 X=7  0.10 0.00 0.00 0.00  0  0  0  0  0  0  0
2032 X=8  0.09 0.00 0.00 0.00  0  0  0  0  0  0  0
2033 X=9  0.06 0.00 0.00 0.00  0  0  0  0  0  0  0
2034 X=10 0.04 0.00 0.00 0.00  0  0  0  0  0  0  0
2035 X=11 0.02 0.00 0.00 0.00  0  0  0  0  0  0  0
2036 X=12 0.01 0.00 0.00 0.00  0  0  0  0  0  0  0
2037 X=13 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
2038 X=14 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
2039 X=15 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
2040 X=16 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
2041 X=17 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
2042 X=18 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
2043 X=19 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
2044 X=20 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0

```

2045 This matrix tells us the probability of all possible combinations of  $x$  and  $y$ , and  
 2046 we see that the most likely value is  $(X = 1, Y = 1)$ , i.e. we will catch 1 fish and  
 2047 there will be 1 other fisherman. This matrix also demonstrates the law of total  
 2048 probability, which dictates that the sum of these probabilities must equal 1.

Perhaps most fisherman don't care about joint distributions, but a question that might be asked is "what is the probability of catching 1 fish today?" We know that this depends on the number of fisherman, but we don't know how many will show up today, so this is a different question than "what is most likely value of  $X$  and  $Y$ ". This brings us to the marginal distribution, which is defined by

$$[X] = \sum_Y [X, Y] \quad [Y] = \sum_X [Y, X]$$

for discrete random variables, and

$$[X] = \int_{-\infty}^{\infty} [X, Y] dY \quad [Y] = \int_{-\infty}^{\infty} [Y, X] dX$$

for continuous random variables. The key idea here is that to get the marginal distribution of  $X$ , we have to contemplate all possible values of  $Y$ . Computing marginal distributions is a key step in maximizing likelihoods involving random effects, as will be demonstrated in Chapt.6. Here is some **R** code to compute the marginal distribution of  $X$ , i.e. the probability of catching  $X = x$  fish:

```
2054 > margX <- rowSums(joint)
2055 > round(margX, 2)
2056   X=0  X=1  X=2  X=3  X=4  X=5  X=6  X=7  X=8  X=9  X=10  X=11  X=12  X=13  X=14
2057 0.18 0.14 0.09 0.05 0.05 0.07 0.09 0.10 0.09 0.06 0.04 0.02 0.01 0.00 0.00
2058 X=15  X=16  X=17  X=18  X=19  X=20
2059 0.00 0.00 0.00 0.00 0.00 0.00
```

Bad news—the most likely value is  $X = 0$ . However, the chances of catching 1 fish is pretty similar.

The last type of question we can ask about these two random variables relates to their conditional distributions. The conditional probability distribution is the distribution of one variable, given a realized value of the other. In the case of two discrete random variables, the conditional distribution may be written as  $[X = x|Y = y]$ , i.e. the probability of  $X$  taking on the value  $x$  given the realized value of  $Y$  being  $y$ . For simplicity, we will write this as  $[X|Y]$ . Conditional distributions are defined as follows:

$$[X|Y] = \frac{[X, Y]}{[Y]} \quad [Y|X] = \frac{[X, Y]}{[X]}.$$

That is, the conditional distribution of  $X$  given  $Y$  is the joint distribution divided by the marginal distribution of  $Y$ .

```
2062 > XgivenY <- joint/matrix(margY, nrow(joint), ncol(joint), byrow=TRUE)
2063 > round(XgivenY, 2)
2064   Y=0  Y=1  Y=2  Y=3  Y=4  Y=5  Y=6  Y=7  Y=8  Y=9  Y=10
2065   X=0  0.00 0.25 0.82 0.97  1  1  1  1  1  1  1
2066   X=1  0.00 0.36 0.16 0.03  0  0  0  0  0  0  0
2067   X=2  0.01 0.25 0.02 0.00  0  0  0  0  0  0  0
2068   X=3  0.03 0.11 0.00 0.00  0  0  0  0  0  0  0
2069   X=4  0.07 0.03 0.00 0.00  0  0  0  0  0  0  0
2070   X=5  0.13 0.01 0.00 0.00  0  0  0  0  0  0  0
2071   X=6  0.17 0.00 0.00 0.00  0  0  0  0  0  0  0
2072   X=7  0.18 0.00 0.00 0.00  0  0  0  0  0  0  0
```

---

2075	X=8	0.16	0.00	0.00	0.00	0	0	0	0	0	0	0
2076	X=9	0.12	0.00	0.00	0.00	0	0	0	0	0	0	0
2077	X=10	0.07	0.00	0.00	0.00	0	0	0	0	0	0	0
2078	X=11	0.03	0.00	0.00	0.00	0	0	0	0	0	0	0
2079	X=12	0.01	0.00	0.00	0.00	0	0	0	0	0	0	0
2080	X=13	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
2081	X=14	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
2082	X=15	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
2083	X=16	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
2084	X=17	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
2085	X=18	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
2086	X=19	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
2087	X=20	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0

2088 Note that we have 11 probability distributions for  $X$ , one for each possible value of  
 2089  $Y$ , and each pmf sums to unity as it should. Note also that if you show up at the  
 2090 hole and there are  $> 2$  fisherman, your chance of catching a fish is very low. Go  
 2091 home. These concepts are explained in more detail in other texts such as Casella  
 2092 and Berger (2002), Royle and Dorazio (2008), and Link and Barker (2010), but  
 2093 hopefully, the code shown here complements the equations and makes it easier for  
 2094 non-statisticians to understand these concepts.

The last point we wish to make in the section is that this simple example *is* a hierarchical model, and we can put the pieces together using the following notation:

$$Y \sim \text{Poisson}(0.6) \quad (2.4.1)$$

$$\text{logit}(p) = -0.6 + -2Y \quad (2.4.2)$$

$$X|Y \sim \text{Binomial}(20, p) \quad (2.4.3)$$

2095 From here on out, when you see such notation, you should immediately grasp  
 2096 the fact that  $Y$  is a random variable independent of  $X$ , but  $X$  depends upon  
 2097  $Y$  through  $p$ . Now you have the tools to make probability statements about the  
 2098 random variables in this system. The one caveat faced in reality is that we typically  
 2099 do not know the values of the parameters, and instead we have to estimate them.  
 2100 Maximum likelihood methods for hierarchical models are covered in Chapt. 6.

## 2.5 HIERARCHICAL MODELS AND INFERENCE

2101 The term hierarchical modeling (or hierarchical model) has become something of  
 2102 a buzzword over the last decade with hundreds of papers published in ecological  
 2103 journals using that term. So then, what exactly is a hierarchical model, anyhow?  
 2104 Obviously, this term stems from the root “hierarchy” which means:

2105 **Definition:** *hierarchy* (noun) – a series of ordered groupings of people or things  
 2106 within a system;

2107 In the case of a hierarchical model (hierarchical being the adjective form of hi-  
 2108 erarchy), the “things” are probability distributions, and they are ordered according  
 2109 to their conditional probability structure. Thus, a hierarchical model is *an ordered*  
 2110 *series of models, ordered by their conditional probability structure.*

2111 A canonical hierarchical model in ecology is this elemental model of species  
 2112 occurrence or distribution (MacKenzie et al., 2002; Tyre et al., 2003; Kéry, 2011):

$$y_i|z_i \sim \text{Binomial}(K, z_i p)$$

$$z_i \sim \text{Bernoulli}(\psi)$$

2114 where  $y_i$  = observation of presence/absence at a site  $i$  and  $z_i$  = occurrence status  
 2115 ( $z_i = 1$  if a species occurs at site  $i$  and  $z_i = 0$  if not). Note that if  $p = 1$ , then we  
 2116 would perfectly observe  $z$  and the model would no longer be hierarchical—it would  
 2117 be a simple logistic regression model. Note also that this hierarchical model has an  
 2118 important conceptual distinction between other types of classical multi-level models  
 2119 such as repeated measures on subjects, in that  $z_i$  is an actual state of nature. In  
 2120 that sense,  $z$  is a random variable that is the outcome of a “real” process. Royle  
 2121 and Dorazio (2008) used the term *explicit* hierarchical model to describe this type of  
 2122 model to distinguish from hierarchical models (*implicit* hierarchical models) where  
 2123 the latent variables don’t correspond to an actual state of nature—but rather just  
 2124 soak up variation that is unmodeled by explicit elements of the model. At best,  
 2125 latent variables in such models are surrogates for something of ecological relevance  
 2126 (“time effects”, “space effects” etc.).

2127 With these examples, we expand on our definition of a hierarchical model as we  
 2128 will use it in this book:

2129 **Definition: Hierarchical Model:** A model with explicit component models that de-  
 2130 scribe variation in the data due to (spatial/temporal) variation in *ecological process*,  
 2131 and due to *imperfect observation* of the process.

2132 Most models considered in this book describe the encounter of individuals con-  
 2133 ditional on the “activity center” of the individual, which is a latent variable (i.e.,  
 2134 unobserved random effect). The definition of an activity center will be context-  
 2135 dependent as discussed in Chapt. 5, but often it can be thought of as an individual’s  
 2136 home range center. The collection of these latent variables represents the outcome  
 2137 of an ecological process describing how individuals distribute themselves over the  
 2138 landscape. Moreover, how individuals are encountered in traps is, in some cases,  
 2139 the result of a model governing movement. As such, these models are examples of  
 2140 hierarchical models that contain formal model components representing both eco-  
 2141 logical process and also the observation of that process. That is, they are explicit  
 2142 hierarchical models (Royle and Dorazio, 2008) as opposed to implicit hierarchical  
 2143 models.

## 2.6 CHARACTERIZATION OF SCR MODELS

2144 For the purposes of this book, an SCR model is any “individual encounter model”  
 2145 (not just “capture-recapture”!) where auxiliary spatial information is also obtained.  
 2146 To be more precise we could as well use the term “spatial capture and/or recap-  
 2147 ture” but that is slightly unwieldy and, besides, it also abbreviates to SCR. The  
 2148 class of SCR models includes traditional capture-recapture models with auxiliary  
 2149 spatial information and even some models that do not even require “recapture”  
 2150 (e.g., distance sampling). There is even a class of models (Chapt. 18) which don’t  
 2151 require capture or unique identification of individuals.

2152 Conceptually, SCR models involve a collection of random variables,  $\mathbf{s}$ ,  $\mathbf{u}$  and  
 2153  $y$  where  $\mathbf{s}$  is the activity center, or home range center,  $\mathbf{u}$  is the location of the  
 2154 individual at the time of sampling, which we may think of as a realization from some  
 2155 movement model, and  $y$  is the “response variable”—what the observer records. For  
 2156 example,  $y = 1$  means “detected” and  $y = 0$  means “not detected”, but many other  
 2157 types of responses are possible (Chapt 9). A broad class of models for estimating  
 2158 density are unified by a hierarchical model involving explicit models for animal  
 2159 activity centers  $\mathbf{s}$ , movement outcomes  $\mathbf{u}$ , and encounter data  $y$ . In some cases, we  
 2160 don’t observe  $y$  but rather summaries of  $y$ , say  $n(y)$ , yet it might be convenient  
 2161 in such cases to retain an explicit focus on  $y$  in terms of model construction. We  
 2162 thus introduce a sequence of models—a hierarchical model—to relate these random  
 2163 variables, which can be written as

$$[n(y)|y][y|\mathbf{u}][\mathbf{u}|\mathbf{s}][\mathbf{s}]. \quad (2.6.1)$$

2164 Every model we talk about in this book has a subset of these components although  
 2165 we never fit the full model because we have not encountered a situation requiring  
 2166 that we do so. However, a detailed description of this model and its various com-  
 2167 ponents is the subject of this book, and we will not pretend to condense hundreds  
 2168 of pages of material into the next few paragraphs. However, we give a cursory  
 2169 overview here to whet the appetite and provide some indication of where we are  
 2170 going. Don’t worry if some of this material doesn’t sink in just yet—we will walk  
 2171 through it slowly in the subsequent chapters.

2172 Let’s begin with the model  $[\mathbf{s}]$  that describes the distribution of the activity  
 2173 centers of each animal in the spatial region  $\mathcal{S}$  (the state-space as we called it previ-  
 2174 ously). As will be explained in Chapt. 5 and Chapt. 11,  $[\mathbf{s}]$  defines a spatial point  
 2175 process, which may be inhomogeneous if there exists spatial variation in density, or  
 2176 it may be homogeneous if density is constant throughout  $\mathcal{S}$ . In the later case, we can  
 2177 write  $[\mathbf{s}] = \text{Uniform}(\mathcal{S})$ , which is to say that the  $N$  activity centers are uniformly  
 2178 distributed in the polygon  $\mathcal{S}$ . A point process is also a model for the number of indi-  
 2179 viduals in the population  $N$ . So we could write  $[\mathbf{s}|\mu]$  where  $\mu$  is an intensity param-  
 2180 eter defined as the number of points per unit area. In other words,  $\mu$  is population  
 2181 density, and we often model population size as either  $N \sim \text{Poisson}(\mu A(\mathcal{S}))$ , where  
 2182  $A(\mathcal{S})$  is the area of the state-space; or,  $N \sim \text{Binomial}(M, \psi)$  where  $\psi = \mu A(\mathcal{S})/M$

2183 and  $M$  is some large integer used simply as a convenience measure when conducting  
 2184 Bayesian analysis. As it turns out, there is very little practical difference in the  
 2185 Poisson prior versus a binomial models for  $N$  (Chapt. 11).

2186 The model  $[\mathbf{u}|\mathbf{s}]$  describes the locations of animals conditional on their activity  
 2187 center. In the original formulation of SCR models (Efford, 2004), this model com-  
 2188 ponent was intentionally ignored. Indeed when movement is not of direct interest,  
 2189 or when  $\mathbf{s}$  is defined in a way not related to a home range center, it may be prefer-  
 2190 able to ignore this model component (Borchers, 2012). In other cases, we might use  
 2191 an explicit model, such as the bivariate normal model (Royle and Young, 2008).

2192 The third component of the model,  $[y|\mathbf{u}]$ , describes how the observed data—the  
 2193 so-called capture-histories—arise conditional on the locations of animals. However,  
 2194 as mentioned previously, most SCR models do not contain a movement model, and  
 2195 thus, we typically entertain the model  $[y|\mathbf{s}]$  instead of  $[y|\mathbf{u}]$ . This encounter model  
 2196 generally has at least two parameters, say  $p_0$  and  $\sigma$ , describing the probability of  
 2197 capturing or detecting an individual given the distance between  $\mathbf{s}$  and the trap.  
 2198 The most basic model is often called the half-normal model, although we typically  
 2199 refer to it as the Gaussian model since, in two-dimensional space, it is the kernel  
 2200 of a bivariate normal distribution. The model is  $p_{ij} = p_0 \exp(-\|\mathbf{x}_j - \mathbf{s}_i\|/(2\sigma^2))$   
 2201 where  $p_0$  is the capture probability when the activity center occurs at the trap  
 2202 location  $\mathbf{x}_j$ , and  $\sigma$  is a spatial scale parameter determining how rapidly capture  
 2203 probability declines with distance. One common design leads to the model  $[y_{ij}|\mathbf{s}_i] =$   
 2204 Bernoulli( $p_{ij}$ ). Chapt. 5 and Chapt. 9 describe many other possible encounter  
 2205 models.

2206 When individuals are marked by biologists or have natural markings permit-  
 2207 ting individual recognition,  $y_{ij}$  is the observed data. However, some or all of the  
 2208 individuals cannot be uniquely identified, then we cannot record this individual-  
 2209 specific encounter history data. Instead, the data might be simply the number of  
 2210 detections at a trap or perhaps binary detection/non-detection data at each trap on  
 2211 each survey occasion. We call this reduced information data  $n(y)$ , and Chapt. 18  
 2212 and Chapt. 19 describe models for  $[n(y)|y]$  that still allow for density estimation.  
 2213 The basic strategy is to view  $y$  as “missing data” and to use the spatial correlation  
 2214 in the counts, or other sources of information, to provide information about these  
 2215 latent encounter histories.

2216 Eq. 2.6.1 is a compact description of the the basic components of a SCR model,  
 but it is also rather vague. The previous four paragraphs added enough extra detail  
 so that we can now describe a specific SCR model. Perhaps the simplest SCR model  
 is this:

$$\begin{aligned} N &\sim \text{Poisson}(\mu A(\mathcal{S})) \\ \mathbf{s}_i &\sim \text{Uniform}(\mathcal{S}) \\ y_{ijk}|\mathbf{s}_i &\sim \text{Bernoulli}(p(\|\mathbf{x}_j - \mathbf{s}_i\|)) \end{aligned} \tag{2.6.2}$$

2216 These “assumptions” are statistical statements of three basic hypotheses that (1)

2217 population size  $N$  is Poisson distributed (2) activity centers are uniformly dis-  
 2218 tributed in two-dimensional space, and (3) capture probability is a function of the  
 2219 distance between the activity and the trap. Each of these model components can  
 2220 be modified as needed to match specific hypotheses, study designs, and data struc-  
 2221 tures. For example, spatial variation in abundance or density can be easily modeled  
 2222 as a function of habitat covariates (Chapt. 11).

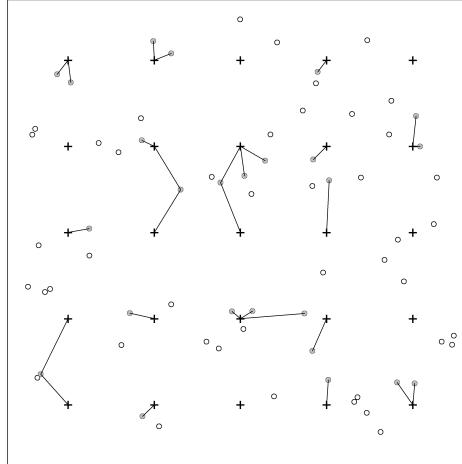
2223 We realize that many the model description in Eq. 2.6.2 may not be self-evident  
 2224 to some ecologists. However, it is absolutely essential that one can understand  
 2225 such a model description—not just for being able to read this book, but also for  
 2226 understanding any statistical model in ecology. One of the best ways of familiarizing  
 2227 oneself with this notation is to translate it into **R** code that simulates outcomes  
 2228 from the model. The following code is an example.

```
2229 set.seed(36372)
2230 Area <- 1 # area of state-space (unit square)
2231 x <- cbind(rep(seq(.1,.9,.2), each=5), # trap locations
2232             rep(seq(.1,.9,.2), times=5))
2233 p0 <- 0.3 # baseline capture probability
2234 sigma <- 0.05 # Gaussian scale parameter
2235 mu <- 50 # population density
2236 N <- rpois(1, mu*Area) # population size
2237 s <- cbind(runif(N, 0, 1), # activity centers in unit square
2238             runif(N, 0, 1))
2239 K <- 5
2240 y <- matrix(NA, N, nrow(x)) # capture data
2241 for(i in 1:N) {
2242   d.ij <- sqrt((x[,1] - s[i,1])^2 + # distance between x and s[i]
2243                 (x[,2] - s[i,2])^2)
2244   p.ij <- p0*exp(-d.ij^2 / (2*sigma^2)) # capture probability
2245   y[i,] <- rbinom(nrow(x), K, p.ij) # capture history for animal i
2246 }
```

2247 Fig. 2.3 shows the results of this simulation from a basic, yet very useful, SCR  
 2248 model.

2249 Having briefly explained each of the model components in Eq. 2.6.1, and having  
 2250 shown how a subset of these components results in a basic SCR model, we can  
 2251 now discuss other relevant arrangements. Examples include: (1) Classical distance  
 2252 sampling (Buckland et al., 2001; Borchers et al., 2002), (2) Spatial capture-recapture  
 2253 models with fixed arrays of traps (Efford, 2004; Borchers and Efford, 2008; Royle  
 2254 et al., 2009a,b; Gardner et al., 2010a; Royle et al., 2011b), and (3) Search-encounter  
 2255 models (Royle and Young, 2008; Royle et al., 2011a). We will now elaborate on  
 2256 some of these distinctions.

2257 1. **Distance sampling.** The last 2 stages of the hierarchy are confounded  
 2258 (implicitly) and so analysis is based on the model  $[y|\mathbf{u}][\mathbf{u}]$ . The “process  
 2259 model” is that of “uniformity”:  $\mathbf{u} \sim \text{Uniform}(\mathcal{S})$ .



**Figure 2.3.** Population of  $N = 69$  home-range centers ( $s$ , circles) and 25 trap locations ( $x$ , crosses). Lines connect activity centers to the traps where the individuals were detected. As in many SCR models, movement outcomes ( $u$ ) are ignored.

2260     2. **Spatial capture-recapture model with a fixed array of traps.** SCR  
 2261     models appear to have little in common with distance sampling because ob-  
 2262     servations are made only at a pre-defined set of discrete locations—where  
 2263     traps are placed. However, the models are closely related in terms of our  
 2264     hierarchical representation above. In SCR models based on fixed arrays, we  
 2265     cannot estimate both  $\Pr(y = 1|u)$  and  $\Pr(u|s)$ —the probability that an in-  
 2266     dividual “moves to  $u$ ” cannot be separated from the probability that it is  
 2267     detected given that it moves to  $u$ , because of the fact that the observation  
 2268     locations are fixed by design. Formally, such SCR models confound  $[y|u]$   
 2269     with  $[u|s]$  so that the observation model arises as:

$$[y|s] = \int_u [y|u][u|s]du$$

2270     This confounding happens because SCR sampling is spatially biased—restricted  
 2271     to a fixed pre-determined set of locations. Conversely, distance sampling  
 2272     confounds  $[u|s][s]$  because, essentially, there is only a single realization of the  
 2273     encounter process. It is probably reasonable to assume that  $\Pr(y = 1|u) = 1$   
 2274     or at least it is locally constant for most devices (e.g., cameras, etc..), and  
 2275     thus the detection model will have the interpretation in terms of movement  
 2276     (see Chapt. 13 and 12).

2277     3. **Search-encounter models.** What we call “search-encounter” models (Royle

2278 and Young, 2008; Royle et al., 2011a) are kind of a hybrid model combining  
2279 features of SCR models and features of distance sampling. Like distance  
2280 sampling they allow for encounters in continuous space which provide di-  
2281 rect observations from  $[\mathbf{u}|\mathbf{s}]$ . Thus, the hierarchical model is fully identified.  
2282 These models are described in Chapt. chapt.search-encounter.

## 2.7 SUMMARY AND OUTLOOK

2283 Spatial capture-recapture models are hierarchical models, and hierarchical models  
2284 are models of multiple random variables that are conditionally related. It is there-  
2285 fore important that the basic rules of modeling random variables are understood,  
2286 and we hope that this chapter has made some of the basic concepts accessible to  
2287 ecologists with rudimentary background in statistics. If some of this material still  
2288 seems difficult to grasp, we recommend working with the provided **R** code, which  
2289 is perhaps the best way of making the equations more tangible.

2290 In some respects, it is possible to understand the jist of SCR without knowing  
2291 anything about marginal and conditional relationships. One can always fit models  
2292 using canned software and interpret the output without understanding the guts of  
2293 the model or the details of the estimation process. For some applied ecologists,  
2294 this may be perfectly fine, and this book is meant to be useful for both statistical  
2295 novices and ecologists with more advanced quantitative skills. In most chapters, we  
2296 begin with a basic conceptual discussion, then we explain the technical details that  
2297 require an understanding of the concepts in this chapter, and finally we end with  
2298 one or more worked examples. For those not interested in the technical details,  
2299 we recommend focusing on the chapter introductions and the examples. However,  
2300 taking the time to understand the concepts presented in this chapter can only  
2301 increase one's ability to tackle the unique and complex problems that often present  
2302 themselves when modeling spatial and temporal aspects of population dynamics.



2303  
2304

---

# 3

2305

## GLMS AND BAYESIAN ANALYSIS

2306 A major theme of this book is that spatial capture-recapture models are, for the  
2307 most part, just generalized linear models (GLMs) wherein the covariate, distance  
2308 between trap and home range center, is partially or fully unobserved – and therefore  
2309 regarded as a random effect. Outside of capture-recapture, such models are usually  
2310 referred to as generalized linear mixed models (GLMMs) and, therefore, SCR mod-  
2311 els can be thought of as a specialized type of GLMM. Naturally then, we should  
2312 consider analysis of these slightly simpler models in order to gain some experience  
2313 and, hopefully, develop a better understanding of spatial capture-recapture models.

2314 In this chapter, we consider classes of GL(M)Ms – Poisson and binomial (i.e.,  
2315 logistic regression) models – that will prove to be enormously useful in the analysis  
2316 of capture-recapture models of all kinds. Many readers are likely familiar with these  
2317 models already because they are among the most useful models in ecology and,  
2318 as such, have received considerable attention in many introductory and advanced  
2319 texts. We focus on them here in order to introduce the readers to the analysis of  
2320 such models in **R** and **WinBUGS** or **JAGS**, which we will translate directly to  
2321 the analysis of SCR models in subsequent chapters.

2322 Bayesian analysis is convenient for analyzing GL(M)Ms because it allows us to  
2323 work directly with the conditional model – i.e., the model that is conditional on the  
2324 random effects, using computational methods known as Markov chain Monte Carlo  
2325 (MCMC). Learning how to do Bayesian analysis of GLMs and GLMMs using the  
2326 **BUGS** language is, in part, the purpose of this chapter. We focus here on the use of  
2327 **WinBUGS** because it is the most popular “**BUGS** engine”. However, later in the  
2328 book we transition to another popular **BUGS** engine known as **JAGS** (Plummer,  
2329 2009) which stands for *Just Another Gibbs Sampler*. For most of our purposes, the  
2330 specification of models in either platform is the same, but **JAGS** is under active  
2331 development at the present time while **WinBUGS** no longer is, having transitioned

2332 to **OpenBUGS** (Lunn et al., 2009) which is still in active development. While we  
 2333 use **BUGS** of one sort or another to do the Bayesian computations, we organize and  
 2334 summarize our data and execute **WinBUGS** or **JAGS** from within **R** using the  
 2335 packages **R2WinBUGS** (Sturtz et al., 2005), **R2jags** (Su and Yajima, 2011) or **rjags**  
 2336 (Plummer, 2009). Kéry (2010), and Kéry and Schaub (2012) provide excellent  
 2337 and accessible introductions to the basics of Bayesian analysis and GL(M)Ms using  
 2338 **WinBUGS**. We don't want to be too redundant with those books and so we avoid  
 2339 a detailed treatment of Bayesian methodology and software usage - instead just  
 2340 providing a cursory overview so that we can move on and attack the problems  
 2341 we're most interested in related to spatial capture-recapture. In addition, there are  
 2342 a number of texts that provide general introductions to Bayesian analysis, MCMC,  
 2343 and their applications in ecology including McCarthy (2007), Kéry (2010), Link  
 2344 and Barker (2010), and King et al. (2008).

2345 While this chapter is about Bayesian analysis of GL(M)Ms, such models are  
 2346 routinely analyzed using likelihood methods too. Later in this book (Chapt. 6), we  
 2347 will use likelihood methods to analyze SCR models but, for now, we concentrate on  
 2348 providing a basic introduction to Bayesian analysis because that is the approach  
 2349 we will use in a majority of cases in later chapters.

### 3.1 GLMS AND GLMMS

2350 We have asserted already that SCR models work out most of the time to be variations  
 2351 of GL(M)Ms. You might therefore ask: What are these GLM and GLMM  
 2352 models, anyhow? These models are covered extensively in many very good applied  
 2353 statistics books and we refer the reader elsewhere for a detailed introduction. The  
 2354 classical references for GLMs are Nelder and Wedderburn (1972) and McCullagh  
 2355 and Nelder (1989). In addition, we think Kéry (2010), Kéry and Schaub (2012),  
 2356 and Zuur et al. (2009) are all accessible treatments. Here, we'll give the 1 minute  
 2357 treatment of GL(M)Ms, not trying to be complete but rather only to preserve a  
 2358 coherent organization to the book.

2359 The GLM is an extension of standard linear models allowing the response variable  
 2360 to have some distribution from the exponential family of distributions. This  
 2361 includes the normal distribution but also others such as the Poisson, binomial,  
 2362 gamma, exponential, and many more. In addition, GLMs allow the response variable  
 2363 to be related to the predictor variables (i.e., covariates) using a link function,  
 2364 which is usually nonlinear. The GLM consists of three components:

- 2365 1. A probability distribution for the dependent (or response) variable  $y$ , from the  
 2366 exponential family of probability distributions.
- 2367 2. A "linear predictor"  $\eta = \beta_0 + x\beta_1$ , where  $x$  is a predictor variable (i.e., a covariate).
- 2368 3. A link function  $g$  that relates the expected value of  $y$ ,  $\mathbb{E}(y)$ , to the linear predictor,  
 2369  $\mathbb{E}(y) = \mu = g^{-1}(\eta)$ . Therefore  $g(\mathbb{E}(y)) = \eta = \beta_0 + x\beta_1$ .

2371 A key aspect of GLMs is that  $g(\mathbb{E}(y))$  is assumed to be a linear function of the  
 2372 predictor variable(s), here  $x$ , with unknown parameters, here  $\beta_0$  and  $\beta_1$ , to be  
 2373 estimated. In standard GLMs, the variance of  $y$  is a function  $V$  of the mean of  $y$ :  
 2374  $\text{Var}(y) = V(\mu)$  (see below for examples). As an example, a Poisson GLM posits  
 2375 that  $y \sim \text{Poisson}(\lambda)$  with  $\mathbb{E}(y) = \lambda$  and usually the model for the mean is specified  
 2376 using the *log link function* by

$$\log(\lambda_i) = \beta_0 + \beta_1 x_i$$

2377 The variance function is  $V(y_i) = \lambda_i$ . To see how a Poisson GLM works, use the **R**  
 2378 code below to simulate some data and then estimate the parameters:

```
2379 > set.seed(13)
2380 > n <- 100          # set sample size
2381 > beta0 <- -2        # set intercept term
2382 > beta1 <- 1.5       # set coefficient
2383 > x <- rnorm(n, 0,1) # generate a predictor variable, x
2384
2385 > linpred <- beta0 + beta1*x # calculate linear predictor of E(y)
2386 > y <- rpois(n, exp(linpred)) # generate observations from model
```

2387 The **R** function `glm()` fits a GLM to the data we just generated and returns estimates of  
 2388  $\beta_0$  and  $\beta_1$ , which we see are fairly close to the data generating values above:

```
2389 > glm(y ~ 1 + x, family='poisson')      # the fit model
```

2390 This produces the output:

```
2391 Call: glm(formula = y ~ 1 + x, family = "poisson")
2392
2393 Coefficients:
2394 (Intercept)      x
2395     -2.007      1.446
2396
2397 [... some output deleted ...]
```

2398 In this summary output, the maximum likelihood estimates (MLEs) of the regression  
 2399 parameters  $\beta_0$  and  $\beta_1$  are labeled “Coefficients.” We see that these are not too different  
 2400 from the data-generating values (-2 and 1.5, respectively).

2401 The binomial GLM posits that  $y_i \sim \text{Binomial}(K, p)$  where  $K$  is the fixed sample size  
 2402 parameter and  $\mathbb{E}(y_i) = K \times p_i$ . Usually the model for the mean is specified using the *logit*  
 2403 *link function* according to

$$\text{logit}(p_i) = \beta_0 + \beta_1 x_i$$

2404 Where  $\text{logit}(p) = \log(p/(1-p))$ . The inverse-logit function, consequently, is  $\text{logit}^{-1}(p) =$   
 2405  $\exp(p)/(1 + \exp(p))$ .

2406 A GLMM is the extension of GLMs to accommodate “random effects”. Often this  
 2407 involves adding a normal random effect to the linear predictor. One simple example is  
 2408 using a random intercept,  $\alpha$ :

$$\log(\lambda_i) = \alpha_i + \beta_1 x_i$$

2409 where

$$\alpha_i \sim \text{Normal}(\mu, \sigma^2)$$

2410 Many other probability distributions and formulations of the linear predictor might be  
 2411 considered. GLMMs are enormously useful in ecological modeling applications for mod-  
 2412 eling variation due to subjects, observers, spatial or temporal stratification, clustering,  
 2413 and dependence that arises from any kind of group structure and, of course, because SCR  
 2414 models prove to be a type of GLM with a random effect, but one that does not enter the  
 2415 mean linearly.

### 3.2 BAYESIAN ANALYSIS

2416 Bayesian analysis is less familiar to many ecological researchers because they are often  
 2417 educated only in the classical statistical paradigm of frequentist inference. But advances  
 2418 in technology and increasing exposure to the benefits of Bayesian analysis are fast mak-  
 2419 ing Bayesians out of people or at least making Bayesian analysis an acceptable, general  
 2420 alternative to classical, frequentist inference.

2421 Conceptually, the main thing about Bayesian inference is that it uses probability  
 2422 directly to characterize uncertainty about things we don't know. "Things", in this case,  
 2423 are parameters of models and, just as it is natural to characterize uncertain outcomes of  
 2424 stochastic processes using probability, it seems natural also to characterize information  
 2425 about unknown parameters using probability. At least this seems natural to us and, we  
 2426 think, most ecologists either explicitly adopt that view or tend to fall into that point  
 2427 of view naturally. Conversely, frequentists use probability in many different ways, but  
 2428 never to characterize uncertainty about parameters<sup>1</sup>. Instead, frequentists use probability  
 2429 to characterize the behavior of *procedures* such as estimators or confidence intervals (see  
 2430 below). It is surprising that people readily adopt a philosophy of statistical inference in  
 2431 which the things you don't know (i.e., parameters) should *not* be regarded as random  
 2432 variables, so that, as a consequence, one cannot use probability to characterize one's state  
 2433 of knowledge about them.

#### 2434 3.2.1 Bayes' rule

2435 As its name suggests, Bayesian analysis makes use of Bayes' rule in order to make direct  
 2436 probability statements about model parameters. Given two random variables  $z$  and  $y$ ,  
 2437 Bayes' rule relates the two conditional probability distributions  $[z|y]$  and  $[y|z]$  by the  
 2438 relationship:

$$[z|y] = [y|z][z]/[y]. \quad (3.2.1)$$

2439 Bayes' rule itself is a mathematical fact and there is no debate in the statistical community  
 2440 as to its validity and relevance to many problems. Generally speaking, these distributions  
 2441 are characterized as follows:  $[y|z]$  is the conditional probability distribution of  $y$  given  $z$ ,  
 2442  $[z]$  is the marginal distribution of  $z$  and  $[y]$  is the marginal distribution of  $y$ . In the context  
 2443 of Bayesian inference we usually associate specific meanings in which  $[y|z]$  is thought of  
 2444 as "the likelihood",  $[z]$  as the "prior" and so on. We leave this for later because here the  
 2445 focus is on this expression of Bayes' rule as a basic fact of probability.

---

<sup>1</sup>To hear this will be shocking to some readers perhaps.

As an example of a simple application of Bayes' rule, consider the problem of determining species presence at a sample location based on imperfect survey information. Let  $z$  be a binary random variable that denotes species presence ( $z = 1$ ) or absence ( $z = 0$ ), let  $\Pr(z = 1) = \psi$  where  $\psi$  is usually called occurrence probability, "occupancy" (MacKenzie et al., 2002) or "prevalence". Let  $y$  be the *observed* presence ( $y = 1$ ) or absence ( $y = 0$ ) (or, strictly speaking, detection and non-detection), and let  $p$  be the probability that a species is detected in a single survey at a site given that it is present. Thus,  $\Pr(y = 1|z = 1) = p$ . The interpretation of this is that, if the species is present, we will only observe it with probability  $p$ . In addition, we assume here that  $\Pr(y = 1|z = 0) = 0$ . That is, the species cannot be detected if it is not present which is a conventional view adopted in most biological sampling problems (but see Royle and Link (2006)). If we survey a site  $K$  times but never detect the species, then this clearly does not imply that the species is not present ( $z = 0$ ) at this site but that we failed to observe it. Rather, our degree of belief in  $z = 0$  should be made with a probabilistic statement, namely the conditional probability  $\Pr(z = 1|y_1 = 0, \dots, y_K = 0)$ . If the  $K$  surveys are independent so that we might regard  $y_k$  as *iid* Bernoulli trials, then the total number of detections, say  $y$ , is Binomial with probability  $p$ , and we can use Bayes' rule to compute the probability that the species is present given that it is not detected in  $K$  samples, i.e.,  $\Pr(z = 1|y_1 = 0, \dots, y_K = 0)$ . In words, the expression we seek is:

$$\Pr(\text{present}|\text{not detected}) = \frac{\Pr(\text{not detected}|\text{present})\Pr(\text{present})}{\Pr(\text{not detected})}$$

Mathematically, this is

$$\begin{aligned}\Pr(z = 1|y = 0) &= \frac{\Pr(y = 0|z = 1)\Pr(z = 1)}{\Pr(y = 0)} \\ &= \frac{(1 - p)^K \psi}{(1 - p)^K \psi + (1 - \psi)}.\end{aligned}$$

The denominator here, the probability of not detecting the species, is composed of two parts: (1) not observing the species given that it is present (this occurs with probability  $(1 - p)^K \psi$ ) and (2) the species is not present (this occurs with probability  $1 - \psi$ ). To apply this result, suppose that  $K = 2$  surveys are done at a wetland for a species of frog, and the species is not detected there. Suppose further that  $\psi = 0.8$  and  $p = 0.5$  are obtained from a prior study. Then the probability that the species is present at this site, even though it was not detected, is  $(1 - 0.5)^2 \times 0.8 / ((1 - 0.5)^2 \times 0.8 + (1 - 0.8)) = 0.5$ . That is, there is a 50/50 chance that the site is occupied despite the fact that the species wasn't observed there.

In summary, Bayes' rule provides a simple linkage between the conditional probabilities  $[y|z]$  and  $[z|y]$ , which is useful whenever we need to deduce one from the other.

### 3.2.2 Principles of Bayesian inference

Bayes' rule as a basic fact of probability is not disputed. What is controversial to some is the scope and manner in which Bayes' rule is applied by Bayesian analysts. Bayesian analysts assert that Bayes' rule is relevant, in general, to all statistical problems by regarding

2481 all unknown quantities of a model as realizations of random variables – this includes data,  
 2482 latent variables, and also parameters. Classical (non-Bayesian) analysts sometimes object  
 2483 to regarding parameters as outcomes of random variables. Classically, parameters are  
 2484 thought of as “fixed but unknown” (using the terminology of classical statistics). Indeed,  
 2485 a common misunderstanding on the distinction between Bayesian and frequentist infer-  
 2486 ence goes something like this “in frequentist inference parameters are fixed but unknown  
 2487 but in a Bayesian analysis parameters are random.” At best this is a sad caricature of the  
 2488 distinction and at worst it is downright wrong. In Bayesian analysis the parameters are  
 2489 also unknown and, in fact, there is a single data-generating value of each parameter, and  
 2490 so they are also fixed. The difference is that the fixed but unknown values are regarded  
 2491 as having been generated from some probability distribution. Specification of that prob-  
 2492 ability distribution is necessary to carry out Bayesian analysis, but it is not required in  
 2493 classical frequentist inference.

2494 To see the general relevance of Bayes’ rule in the context of statistical inference, let  $y$   
 2495 denote observations - i.e., data - and let  $[y|\theta]$  be the observation model (often colloquially  
 2496 referred to as the “likelihood”). Suppose  $\theta$  is a parameter of interest having (prior)  
 2497 probability distribution  $[\theta]$  (also simply referred to as the prior). These are combined to  
 2498 obtain the posterior distribution using Bayes’ rule, which is:

$$[\theta|y] = [y|\theta][\theta]/[y]$$

2499 Asserting the general relevance of Bayes’ rule to all statistical problems, we can conclude  
 2500 that the two main features of Bayesian inference are that: (1) parameters,  $\theta$ , are regarded  
 2501 as realizations of a random variable and, as a result, (2) inference is based on the prob-  
 2502 ability distribution of the parameters given the data,  $[\theta|y]$ , which is called the posterior  
 2503 distribution. This is the result of using Bayes’ rule to combine the “likelihood” and the  
 2504 prior distribution. The key concept is regarding parameters as realizations of a random  
 2505 variable because, once you admit this conceptual view, this leads directly to the posterior  
 2506 distribution, a very natural quantity upon which to base inference about things we don’t  
 2507 know - including parameters of statistical models. In particular,  $[\theta|y]$  is a probability  
 2508 distribution for  $\theta$  and therefore we can make direct probability statements to characterize  
 2509 uncertainty about  $\theta$ .

2510 The denominator of our invocation of Bayes’ rule,  $[y]$ , is the marginal distribution of  
 2511 the data  $y$ . We note without further remark right now that, in many practical problems,  
 2512 this can be an enormous pain to compute. The main reason that the Bayesian paradigm  
 2513 has become so popular in the last 20 years or so is because methods have been developed  
 2514 for characterizing the posterior distribution that do not require that we possess a math-  
 2515 ematical understanding of  $[y]$ . This means we never have to compute it or know what it  
 2516 looks like, or know anything specific about it.

2517 While we can understand the conceptual basis of Bayesian inference merely by under-  
 2518 standing Bayes’ rule – that’s really all there is to it – it is not so easy to understand the  
 2519 basis of classical frequentist inference. What is mostly coherent in frequentist inference is  
 2520 the manner in which procedures are evaluated – the performance of a given procedure is  
 2521 evaluated by “averaging over” hypothetical realizations of  $y$ , regarding the *estimator* as a  
 2522 random variable. For example, if  $\hat{\theta}$  is an estimator of  $\theta$  then the frequentist is interested  
 2523 in  $E_y(\hat{\theta}|y)$  which is used to characterize bias. If the expected value of  $\hat{\theta}$ , when averaged  
 2524 over realizations of  $y$ , is equal to  $\theta$ , then  $\hat{\theta}$  is unbiased.

2525 The view of parameters as being random variables allows Bayesians to use probability  
2526 to make direct probability statements about parameters. Frequentist inference procedures  
2527 do not permit direct probability statements to be made about parameter values. Instead,  
2528 the view of parameters as fixed constants and estimators as random variables leads to  
2529 interpretations that are not so straightforward. For example confidence intervals having  
2530 the interpretation “95% probability that the interval contains the true value” and p-values  
2531 being “the probability of observing an outcome of the test statistic as extreme or more  
2532 than the one observed.” These are far from intuitive interpretations to most people.  
2533 Moreover, this is conceptually problematic to some because we will never get to observe  
2534 the hypothetical realizations that characterize the performance of our procedure.

2535 While we do tend to favor Bayesian inference for the conceptual simplicity (parameters  
2536 are random, posterior inference), we mostly advocate for a pragmatic non-partisan  
2537 approach to inference because, frankly, some of the frequentist methods are actually very  
2538 convenient in certain situations, and will generally yield very similar inferences about  
2539 parameters, as we will see in later chapters.

### 2540 3.2.3 Prior distributions

2541 The prior distribution  $[\theta]$  is an important feature of Bayesian inference. As a conceptual  
2542 matter, the prior distribution characterizes “prior beliefs” or “prior information” about  
2543 a parameter. Indeed, an oft-touted benefit of Bayesian analysis is the ease with which  
2544 prior information can be included in an analysis. However, more commonly, the prior  
2545 is chosen to express a lack of prior information, even if previous studies have been done  
2546 and even if the investigator does in fact know quite a bit about a parameter. This is  
2547 because the manner in which prior information is embodied in a prior (and the amount  
2548 of information) is usually very subjective and thus the result can wind up being very  
2549 contentious; e.g., different investigators might report different results based on subjective  
2550 assessments of prior information. Thus it is usually better to “let the data speak” and  
2551 use priors that reflect absence of information beyond the data set being analyzed. An  
2552 example for an uninformative prior is a Uniform(0, 1) for a probability, or a Uniform( $-\infty$ ,  
2553  $\infty$ ) (also called a “flat” or “improper” prior) for an unbounded continuous parameter.  
2554 Alternatively, people use “diffuse priors”; these contain some information, but (ideally)  
2555 not enough to exert meaningful influence on the posterior. An example for a diffuse prior  
2556 could be a normal distribution with a large standard deviation.

2557 But still the need occasionally arises to embody prior information or beliefs about a  
2558 parameter formally into the estimation scheme. In SCR models we often have a parameter  
2559 that is closely linked to “home range size” and thus auxiliary information on the home  
2560 range size of a species can be used as prior information, which may improve parameter  
2561 estimation (e.g., see Chandler and Royle (2013); also Chapt. 18).

2562 At times the situation arises where a prior can inadvertently impose substantial effect  
2563 on the posterior of a parameter, and that is not desirable. For example, we use data  
2564 augmentation to deal with the fact that the population size  $N$  is an unknown parameter  
2565 (Royle et al., 2007) which is equivalent to imposing a  $\text{Binomial}(M, \psi)$  prior on  $N$  for some  
2566 integer  $M$  (see Sec. 4.2). One has to take care to make sure that  $M$  is sufficiently large so  
2567 as to not affect the posterior distribution on  $N$  (see Fig. 17.6, and also Kéry and Schaub  
2568 (2012, Ch. 5)). Another situation that we have to be careful of is that prior distributions

2569 are *not* invariant to transformation of the parameter, and therefore neither are posterior  
 2570 distributions (Link and Barker, 2010, Sec. 6.2.1). Thus, a prior that is ostensibly non-  
 2571 informative on one scale, may be very informative on another scale. For example, if we  
 2572 have a flat prior on  $\text{logit}(p)$  for some probability parameter  $p$ , this is very different from  
 2573 having a Uniform(0,1) prior on  $p$ . We show an example where this makes a difference in  
 2574 Chapt. 5. Nonetheless, it is always possible to assess the influence of prior choice, and  
 2575 it is often the case (with sufficient data and a structurally identifiable model) that the  
 2576 influence of priors is negligible.

2577 **3.2.4 Posterior inference**

2578 In Bayesian inference, we are not focusing on estimating a single point or interval but  
 2579 rather on characterizing a whole distribution – the posterior distribution – from which  
 2580 one can report any summary of interest. A point estimate might be the posterior mean,  
 2581 median, mode, etc.. In many applications in this book, we will compute 95% Bayesian  
 2582 confidence intervals using the 2.5% and 97.5% quantiles of the posterior distribution. For  
 2583 such intervals, it is correct to say  $\Pr(L < \theta < U) = 0.95$ . That is, “the probability that  $\theta$   
 2584 lies between  $L$  and  $U$  is 0.95”.

2585 As an example, suppose we conducted a Bayesian analysis to estimate detection prob-  
 2586 ability ( $p$ ) of some species at a study site, and we obtained a posterior distribution of  
 2587 beta(20,10) for the parameter  $p$ . The following R commands demonstrate how we make  
 2588 inferences based upon summaries of the posterior distribution:

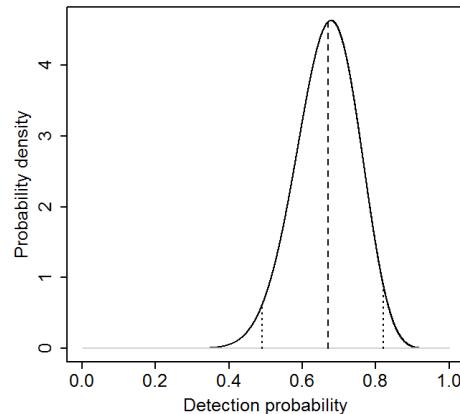
```
2589 > post.median <- qbeta(0.5, 20, 10)
2590 [1] 0.6704151
2591
2592 > post.95ci <- qbeta(c(0.025, 0.975), 20, 10)
2593 [1] 0.4916766 0.8206164
```

2594 Thus, we can state that there is a 95% probability that  $\theta$  lies between 0.49 and 0.82. Fig.  
 2595 3.1 shows the posterior along with the summary statistics. It is not a subtle thing that  
 2596 such statements cannot be made using frequentist methods, although people tend to say  
 2597 it anyway and not really understand why it is wrong or even that it is wrong.

2598 **3.2.5 Small sample inference**

2599 The posterior distribution is an exhaustive summary of the state-of-knowledge about an  
 2600 unknown quantity. It is *the* posterior distribution - not an estimate of that thing. It is  
 2601 also not, usually, an approximation except to within Monte Carlo error (in cases where  
 2602 we use simulation to calculate it, see Sec. 3.5.2). One of the great virtues of Bayesian  
 2603 analysis which is not widely appreciated is that posterior inference is not “asymptotic”,  
 2604 which is to say, valid in a limiting sense as the sample size tends to infinity. Rather,  
 2605 posterior inference is valid for *any* sample size and, in particular, *the* sample size on-hand.  
 2606 Conversely, almost all frequentist procedures are based on asymptotic approximations to  
 2607 the procedure which is being employed.

2608 There seems to be a prevailing view in statistical ecology that classical likelihood-based  
 2609 procedures are virtuous because of the availability of simple formulas and procedures for



**Figure 3.1.** Probability density plot of a hypothetical posterior distribution of  $\text{beta}(20,10)$ ; dashed lines indicate mean and upper and lower 95% interval

carrying out inference, such as calculating standard errors, doing model selection by Akaike information criterion (AIC), and assessing goodness-of-fit. In large samples, this may be an important practical benefit, but the theoretical validity of these procedures cannot be asserted in most situations involving small samples. This is not a minor issue because it is typical in many wildlife sampling problems – especially in surveys of carnivores or rare/endangered species – to wind up with a small, sometimes extremely small, data set, that is nevertheless extremely valuable (Foster and Harmsen, 2012). For examples: A recent paper (Hawkins and Racey, 2005) on the fossa (*Cryptoprocta ferox*), estimated an adult density of 0.18 adults per sq. km based on a sample size of 20 animals captured over 3 years. Sepúlveda et al. (2007) estimated density of the endangered southern river otter (*Lontra provocax*) based on 12 individuals captured over 3 years, Gardner et al. (2010a) estimated density from a study of the Pampas cat (*Leopardus colocolo*), a species for which very little is known, based on only 22 captured individuals over a two year study period, Trolle and Kéry (2005) reported only 9 individual ocelots captured and Jackson et al. (2006) captured 6 individual snow leopards (*Panthera uncia*) using camera trapping. Thus, almost all likelihood-based analysis of data on rare and/or secretive carnivores necessarily and flagrantly violate one of Le Cam's Basic Principles: "If you need to use asymptotic arguments, do not forget to let your number of observations tend to infinity" (Le Cam, 1990).

The biologist thus faces a dilemma with such data. On one hand, these data sets, and the resulting inference, are often criticized as being poor and unreliable. Or, even worse<sup>2</sup>, "the data set is so small, this is a poor analysis." On the other hand, such data

<sup>2</sup>Actual quote from a referee

2632 may be all that is available for species that are extraordinarily important for conservation  
2633 and management. The Bayesian framework for inference provides a valid, rigorous, and  
2634 flexible framework that is theoretically justifiable in arbitrary sample sizes. This is not to  
2635 say that one will obtain precise estimates of density or other parameters, just that your  
2636 inference is coherent and justifiable from a conceptual and technical statistical point of  
2637 view. That is, for example when we estimate the density  $D$  of some animal population,  
2638 we report the posterior probability  $\Pr(D|data)$  which is easily interpretable and just what  
2639 it is advertised to be and we don't need to do a simulation study to evaluate how well  
2640 the reported  $\Pr(D|data)$  deviates from the "true"  $\Pr(D|data)$  because they are the same  
2641 quantity.

### 3.3 CHARACTERIZING POSTERIOR DISTRIBUTIONS BY MCMC SIMULATION

2642 In practice, it is not really feasible to ever compute the marginal probability distribution  
2643 [ $y$ ], the denominator resulting from application of Bayes' rule (Eq. 3.2.1). For decades  
2644 (even centuries!) this impeded the adoption of Bayesian methods by practitioners. Or,  
2645 the few Bayesian analyses done were based on asymptotic normal approximations to the  
2646 posterior distribution. While this was useful from a theoretical and technical standpoint  
2647 and, practically, it allowed people to make the probability statements that they naturally  
2648 would like to make, it was kind of a bad joke around the Bayesian water-cooler to, on  
2649 one hand, criticize classical statistics for being, essentially, completely ad hoc in their  
2650 approach to things but then, on the other hand, have to devise various approximations to  
2651 what they were trying to characterize. The advent of Markov chain Monte Carlo (MCMC)  
2652 methods has made it easier to calculate posterior distributions for just about any problem  
2653 to sufficient levels of precision.

2654 Broadly speaking, MCMC is a class of methods for drawing random samples (i.e.,  
2655 simulating from or just "sampling") from the target posterior distribution. Thus, even  
2656 though we might not recognize the posterior as a named distribution or be able to analyze  
2657 its features analytically, e.g., devise mathematical expressions for the mean and variance,  
2658 we can use these MCMC methods to obtain a large sample from the posterior and then  
2659 use that sample to characterize features of the posterior. What we do with the sample  
2660 depends on our intentions – typically we obtain the mean or median for use as a point  
2661 estimate, and take a confidence interval based on Monte Carlo estimates of the quantiles.

#### 2662 3.3.1 What goes on under the MCMC hood

2663 We will develop and apply MCMC methods in some detail for spatial capture-recapture  
2664 models in Chapt. 17. Here we provide a simple illustration of some basic ideas related to  
2665 the practice of MCMC.

2666 A type of MCMC method relevant to most problems is Gibbs sampling (Geman and  
2667 Geman, 1984) which we address in more detail in Chapt. 17. Gibbs sampling involves iter-  
2668 ative simulation from the "full conditional" distributions (also called conditional posterior  
2669 distributions). The full conditional distribution for an unknown quantity is the conditional  
2670 distribution of that quantity given every other random variable in the model - the data  
2671 and all other parameters (see Sec. 3.3.2 for rules of how to construct full conditionals).

2672 For example, for a normal regression model <sup>3</sup> with  $y \sim \text{Normal}(\beta_0 + \beta_1(x - \bar{x}), \sigma^2)$  where  
 2673 lets say  $\sigma^2$  is known, the full conditionals are, using “bracket notation”,

$$[\beta_0|y, \beta_1]$$

2674 and

$$[\beta_1|y, \beta_0].$$

2675 We might use our knowledge of probability to identify these mathematically. In particular,  
 2676 by Bayes' Rule,  $[\beta_0|y, \beta_1] = [y|\beta_0, \beta_1][\beta_0|\beta_1]/[y|\beta_1]$  and similarly for  $[\beta_1|y, \beta_0]$ . For  
 2677 example, if we have priors for  $[\beta_0] = \text{Normal}(\mu_{\beta_0}, \sigma_{\beta_0}^2)$  and  $[\beta_1] = \text{Normal}(\mu_{\beta_1}, \sigma_{\beta_1}^2)$  then  
 2678 some algebra reveals that

$$[\beta_0|y, \beta_1] = \text{Normal}(w\bar{y} + (1-w)\mu_{\beta_0}, (\tau n + \tau_{\beta_0})^{-1}) \quad (3.3.1)$$

2679 where  $\tau = 1/\sigma^2$  and  $\tau_{\beta_0} = 1/\sigma_{\beta_0}^2$  (the inverse of the variance is sometimes called *precision*),  
 2680 and  $w = \tau n / (\tau n + \tau_{\beta_0})$ . We see in this case that the posterior mean is a *precision-weighted*  
 2681 sum of the sample mean  $\bar{y}$  and the prior mean  $\mu_{\beta_0}$ , and the posterior *precision* is the  
 2682 sum of the precision of the likelihood and that of the prior. These results are typical of  
 2683 many classes of problems. In particular, note that as the prior precision tends to 0, i.e.,  
 2684  $\tau_{\beta_0} \rightarrow 0$ , then the posterior of  $\beta_0$  tends to  $\text{Normal}(\bar{y}, \sigma^2/n)$ . We recognize the variance of  
 2685 this distribution as that of the variance of the sampling distribution of  $\bar{y}$  and its mean is  
 2686 in fact the MLE of  $\beta_0$  for this model. The conditional posterior of  $\beta_1$  has a very similar  
 2687 form:

$$[\beta_1|y, \beta_0] = \text{Normal}\left(\frac{\tau(\sum_i y_i(x_i - \bar{x})) + \tau_{\beta_1}\mu_{\beta_1}}{\tau \sum_i (x_i - \bar{x})^2 + \tau_{\beta_1}}, (\tau \sum_i (x_i - \bar{x})^2 + \tau_{\beta_1})^{-2}\right) \quad (3.3.2)$$

2688 which might look slightly unfamiliar, but note that if  $\tau_{\beta_1} = 0$ , then the mean of this  
 2689 distribution is the familiar  $\hat{\beta}_1$ , and the variance is, in fact, the sampling variance of  $\hat{\beta}_1$ .  
 2690 The MCMC algorithm for this model has us simulate in succession, repeatedly, from  
 2691 those two distributions. See Gelman et al. (2004) for more examples of Gibbs sampling  
 2692 for the normal model, and we also provide another example in Chapt. 17. A conceptual  
 2693 representation of the MCMC algorithm for this simple model is therefore:

**Algorithm:** Gibbs Sampling for linear regression

```

0. Initialize  $\beta_0$  and  $\beta_1$ 
Repeat {
  1. Draw a new value of  $\beta_0$  from Eq. 3.3.1
  2. Draw a new value of  $\beta_1$  from Eq. 3.3.2
}

```

2695 As we just saw for this simple “normal-normal” model, it is sometimes possible to  
 2696 specify the full conditional distributions analytically. In general, when certain so-called  
 2697 conjugate prior distributions are used, which have an analytic form that, in a statistical

---

<sup>3</sup>We center the independent variable here so that things look more familiar in the result

2698 sense, “matches” the likelihood, then the form of the full conditional distributions is also  
 2699 similar to that of the observation model. In this normal-normal case, the normal distribu-  
 2700 tion for the mean parameters is the conjugate prior for the normal observation model, and  
 2701 thus the full-conditional distributions are also normal. This is convenient because, in such  
 2702 cases, we can simulate directly from them using standard methods (or **R** functions). But,  
 2703 in practice, we don’t really ever need to know such things because most of the time we  
 2704 can get by using a simple algorithm, called the Metropolis-Hastings (henceforth “MH”)  
 2705 algorithm, to obtain samples from these full conditional distributions without having to  
 2706 recognize them as specific, named, distributions. This gives us enormous freedom in devel-  
 2707 oping models and analyzing them without having to resolve them mathematically because  
 2708 to implement the MH algorithm we need only identify the full conditional distribution up  
 2709 to a constant of proportionality, that being the marginal distribution in the denominator  
 2710 (e.g.,  $[y|\beta_1]$  above).

2711 We will talk about the Metropolis-Hastings algorithm shortly, and we will use it ex-  
 2712 tensively in the analysis of SCR models (e.g., Chapt. 17).

### 2713 3.3.2 Rules for constructing full conditional distributions

2714 The basic strategy for constructing full-conditional distributions for devising MCMC al-  
 2715 gorithms can be reduced conceptually to a couple of basic steps summarized as follows:

- 2716   **(step 1)** Identify all stochastic components of the model and collect their probability  
   distributions;
- 2718   **(step 2)** Express the full conditional in question as proportional to the product of all  
   probability distributions identified in step 1;
- 2720   **(step 3)** Remove the ones that don’t have the focal parameter in them.
- 2721   **(step 4)** Do some algebra on the result in order to identify the resulting probability  
   distribution function (pdf) or mass function (pmf).

2723 Of the 4 steps, the last of those is the main step that requires quite a bit of statistical  
 2724 experience and intuition because various algebraic tricks can be used to reshape the mess  
 2725 into something recognizable – i.e., a standard, named distribution. But step 4 is not  
 2726 necessary if we decide instead to use the Metropolis-Hastings algorithm as described below.

2727 In the context of our simple linear regression model that we’ve been working with,  
 2728 to characterize  $[\beta_0|y, \beta_1]$  we first apply step 1 and identify the model components as:  
 2729  $[y|\beta_0, \beta_1]$ , with prior distributions  $[\beta_0]$  and  $[\beta_1]$ . Step 2 has us write  $[\beta_0|y, \beta_1] \propto [y|\beta_0, \beta_1][\beta_0][\beta_1]$ .  
 2730 Step 3: We note that  $[\beta_1]$  is not a function of  $\beta_0$  and therefore we remove it to obtain  
 2731  $[\beta_0|y, \beta_1] \propto [y|\beta_0, \beta_1][\beta_0]$ . Similarly, applying step 2 and 3 for  $\beta_1$  we obtain  $[\beta_1|y, \beta_0] \propto$   
 2732  $[y|\beta_0, \beta_1][\beta_1]$ . We apply step 4 and manipulate these algebraically to arrive at the re-  
 2733 sult (which we provided in Eqs. 3.3.1 and 3.3.2) or, alternatively, we can sample them  
 2734 indirectly using the Metropolis-Hastings algorithm, which we discuss now.

### 2735 3.3.3 Metropolis-Hastings algorithm

2736 The Metropolis-Hastings (MH) algorithm is a completely generic method for sampling  
 2737 from any distribution, say  $[\theta]$ . In our applications,  $[\theta]$  will typically be the full conditional  
 2738 distribution of  $\theta$ . While we sometimes use Gibbs sampling, we seldom use “pure” Gibbs

2739 sampling because full conditionals do not always take the form of known distributions we  
 2740 can sample from directly. In such cases, we use MH to sample from the full conditional  
 2741 distributions. When the MH algorithm is used to sample from full conditional distributions  
 2742 of a Gibbs sampler the resulting hybrid algorithm is called *Metropolis-within-Gibbs*. In  
 2743 Sec. 3.6.3 we will construct such an algorithm for a simple class of models. We discuss  
 2744 both the Gibbs and the MH algorithm, as well as their hybrid in more depth in Chapt.  
 2745 17.

2746 The MH algorithm generates candidate values for the parameter(s) we want to estimate  
 2747 from some proposal or candidate-generating distribution that may be conditional on the  
 2748 current value of the parameter, denoted by  $h(\theta^*|\theta^{t-1})$ . Here,  $\theta^*$  is the *candidate* or  
 2749 proposed value and  $\theta^{t-1}$  is the value of  $\theta$  at the previous time step, i.e., at iteration  $t - 1$   
 2750 of the MCMC algorithm. The proposed value is accepted with probability

$$r = \frac{[\theta^*]h(\theta^{t-1}|\theta^*)}{[\theta^{t-1}]h(\theta^*|\theta^{t-1})}$$

2751 which is called the MH acceptance probability. This ratio can sometimes be  $> 1$  in which  
 2752 case we set it equal to 1. It is useful to note that  $h()$  can be any probability distribution.

2753 In the context of using the MH algorithm to do MCMC (in which case the target  
 2754 distribution is a full-conditional or posterior distribution), an important fact is, no matter  
 2755 the choice of  $h()$ , we can compute the MH acceptance probability directly because the  
 2756 marginal distribution of  $y$  cancels from both the numerator and denominator of  $r$ . This  
 2757 is the magic of the MH algorithm.

### 3.4 BAYESIAN ANALYSIS USING THE BUGS LANGUAGE

2758 We won't be too concerned with devising our own MCMC algorithms for every analysis,  
 2759 although we will do that a few times for fun. More often, we will rely on the freely available  
 2760 software package **WinBUGS** or **JAGS** for doing this. We will always execute these  
 2761 **BUGS** engines from within **R** using the **R2WinBUGS** (Sturtz et al., 2005) or, for **JAGS**,  
 2762 the **R2jags** (Su and Yajima, 2011) or **rjags** (Plummer, 2009) packages. **WinBUGS** and  
 2763 **JAGS** are MCMC black boxes that take a pseudo-code description (i.e., written in the  
 2764 **BUGS** language) of all of the relevant stochastic and deterministic elements of a model  
 2765 and generate an MCMC algorithm for that model. But you never get to see the algorithm.  
 2766 Instead, **WinBUGS/JAGS** will run the algorithm and return the Markov chain output  
 2767 - the posterior samples of model parameters.

2768 The great thing about using the **BUGS** language is that it forces you to become  
 2769 intimate with your statistical model - you have to write each element of the model down,  
 2770 admit (explicitly) all of the various assumptions, understand what the actual probability  
 2771 assumptions are and how data relate to latent variables and data and latent variables  
 2772 relate to parameters, and how parameters relate to one another.

2773 While we normally use **WinBUGS**, we note that **OpenBUGS** is the current active  
 2774 development tree of the **BUGS** project. See Kéry (2010) and Kéry and Schaub (2012,  
 2775 especially Appendix 1) for more on practical analysis in **WinBUGS**. Those books should  
 2776 be consulted for a more comprehensive introduction to using **WinBUGS**. Recently we  
 2777 have migrated many of our analyses to **JAGS** (Plummer, 2009), which we adopt later in

2778 the book. You can refer to Hobbs (2011) for an ecological introduction to **JAGS**. Next,  
 2779 we provide an example of a Bayesian analysis using **WinBUGS**.

2780 **3.4.1 Linear regression in WinBUGS**

2781 We provide a brief introductory example of a normal regression model using a small  
 2782 simulated data set. The following commands are executed from within your **R** workspace.  
 2783 First, simulate a covariate  $x$  and observations  $y$  having prescribed intercept, slope and  
 2784 variance:

```
2785 > x <- rnorm(10)
2786 > mu <- -3.2 + 1.5*x
2787 > y <- rnorm(10, mu, sd=4)
```

2788 The **BUGS** model specification for a normal regression model is written within **R** as  
 2789 a character string input to the command `cat()` and then dumped to a text file named  
 2790 `normal.txt`:

```
2791 > cat("
2792   model{
2793     for (i in 1:10){
2794       y[i] ~ dnorm(mu[i],tau)      # the likelihood
2795       mu[i] <- beta0 + beta1*x[i]  # the linear predictor
2796     }
2797     beta0 ~ dnorm(0,.01)          # prior distributions
2798     beta1 ~ dnorm(0,.01)
2799     sigma ~ dunif(0,100)
2800     tau <- 1/(sigma*sigma)      # tau is the precision
2801   }                                # and a derived parameter
2802 ",file="normal.txt")
```

2803 Alternatively, you can write the model specifications directly within a text file and save it  
 2804 in your current working directory, but we do not usually take that approach in this book.

2805 The **BUGS** dialects<sup>4</sup> parameterize the normal distribution in terms of the mean and  
 2806 inverse-variance, called the precision. Thus, `dnorm(0,.01)` implies a variance of 100.  
 2807 We typically use diffuse normal priors for mean parameters,  $\beta_0$  and  $\beta_1$  in this case, but  
 2808 sometimes we might use uniform priors with suitable bounds  $-B$  and  $+B$ . Also, we  
 2809 typically use a Uniform( $0, B$ ) prior on standard deviation parameters (Gelman, 2006).  
 2810 But sometimes we might use a gamma prior on the precision parameter  $\tau$ . In a **BUGS**  
 2811 model file, every variable referenced in the model description has to be either data, which  
 2812 will be input (see below), a random variable which must have a probability distribution  
 2813 associated with it using the tilde character “~” (a.k.a. “twiddle”) or it has to be a derived  
 2814 parameter connected to variables and data using an assignment arrow: “<-”.

2815 To fit the model, we need to describe various data objects to **WinBUGS**. In particular,  
 2816 we create an **R** list object called `data` which are the data objects identified in the **BUGS**  
 2817 model file. In the example, the data consist of two objects which exist as  $y$  and  $x$  in the

---

<sup>4</sup>We use this to mean **WinBUGS**, **OpenBUGS** and **JAGS**

2818    **R** workspace and also in the **WinBUGS** model definition. We also create an **R** function  
 2819    that produces a list of starting values, **inits**, that get sent to **WinBUGS**. In general,  
 2820    starting values are optional. We recommend to always provide reasonable starting values  
 2821    where possible, both for structural parameters and also random effects<sup>5</sup>. Finally, we  
 2822    identify the names of the parameters (labeled correspondingly in the **WinBUGS** model  
 2823    specification) that we want **WinBUGS** to save the MCMC output for. In this example,  
 2824    we will “monitor” the parameters  $\beta_0$ ,  $\beta_1$ ,  $\sigma$  and  $\tau$ . **WinBUGS** is executed using the  
 2825    **R** command **bugs()**. We set the option **debug=TRUE** if we want the **WinBUGS** GUI to  
 2826    stay open (useful for analyzing MCMC output and looking at the **WinBUGS** error log).  
 2827    Also, we set **working.dir=getwd()** so that **WinBUGS** output files and the log file are  
 2828    saved in the current **R** working directory (note that sometimes you will need to specify the  
 2829    place where you installed **WinBUGS** within the **bugs()** call, using the **bugs.directory**  
 2830    argument). All of these activities together look like this:

```
2831 > library(R2WinBUGS)      # "load" the R2WinBUGS package
2832 > data <- list( y=y, x=x)
2833 > inits <- function()
2834 > list ( beta1=rnorm(1),beta0=rnorm(1),sigma=runif(1,0,2) )
2835 > parameters <- c("beta0","beta1","sigma","tau")
2836 > out <- bugs(data, inits, parameters, "normal.txt", n.thin=1, n.chains=2,
2837   n.burnin=2000, n.iter=6000, debug=TRUE,working.dir=getwd())
```

2838    Note that the previously created objects defining data, initial values and parameters to  
 2839    monitor are passed to the function **bugs()**. In addition, various other things are declared:  
 2840    The number of parallel Markov chains (**n.chains**), the thinning rate (**n.thin**), the number  
 2841    of burn-in iterations (**n.burnin**) and the total number of iterations (**n.iter**). To develop  
 2842    a detailed understanding of the various parameters and settings used for MCMC, consult  
 2843    a basic reference such as Kéry (2010). We also come back to these issues in the following  
 2844    section (3.5) and in Chapt. 17. A common question is “how should my data be formatted?”  
 2845    That depends on how you describe the model in the **BUGS** language, and how your data  
 2846    are input into **R**. There is no unique way to describe any particular model and so you have  
 2847    some flexibility. We talk about data format further in the context of capture-recapture  
 2848    models and SCR models in Chapt. 5 and elsewhere.

2849    You should execute all of the commands given above and then close the **WinBUGS**  
 2850    GUI, and the data will be read back into **R** (or specify **debug=FALSE** in the **bugs()** call).  
 2851    We don’t want to give instructions on how to navigate and use the GUI – but you can  
 2852    fire up **WinBUGS** and read the help files, or see Chapt. 4 from Kéry (2010) for a brief  
 2853    introduction. The **print** command applied to the object **out** prints some basic summary  
 2854    output (this is slightly edited):

```
2855 > print(out,digits=2)
2856 Inference for Bugs model at "normal.txt", fit using WinBUGS,
2857 2 chains, each with 6000 iterations (first 2000 discarded)
```

---

<sup>5</sup>While **WinBUGS** is reasonably robust to a wide range of more or less plausible starting values, **JAGS** is a lot more sensitive and especially with more complex models you might actually have to spend some time thinking about how to specify good starting values to get the model running (Appendix 1); we will come back to this issue when we use **JAGS**

```

2858 n.sims = 8000 iterations saved
2859      mean   sd 2.5% 25% 50% 75% 97.5% Rhat n.eff
2860 beta0    -6.62 1.64 -9.77 -7.63 -6.64 -5.63 -3.29     1  4200
2861 beta1     0.81 1.20 -1.63  0.09  0.80  1.54  3.24     1  5100
2862 sigma     4.99 1.56  2.93  3.92  4.66  5.70  8.85     1  8000
2863 tau       0.05 0.03  0.01  0.03  0.05  0.07  0.12     1  8000
2864 deviance 58.72 3.21 55.06 56.35 57.85 60.26 67.15     1  6200
2865
2866 For each parameter, n.eff is a crude measure of effective sample size,
2867 and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
2868
2869 DIC info (using the rule, pD = Dbar-Dhat)
2870 pD = 2.5 and DIC = 61.3

```

2871 In the **WinBUGS** output you see a column called “Rhat”, as well as one called  
2872 “n.eff”. These are convergence diagnostics (the  $\hat{R}$  or Brooks-Gelman-Rubin statistic  
2873 and the effective sample size) and we will discuss those in the following section, 3.5.2.  
2874 DIC is the deviance information criterion (Spiegelhalter et al. (2002), see section 3.9)  
2875 which some people use in a manner similar to AIC although it is recognized to have some  
2876 problems in hierarchical models (Millar, 2009). We consider use of DIC in the context of  
2877 SCR models in Chapt. 8.

### 3.5 PRACTICAL BAYESIAN ANALYSIS AND MCMC

2878 The mere execution of a Bayesian analysis using the **BUGS** language, as demonstrated  
2879 with the linear regression example, is fairly straight forward. There are, however, a number  
2880 of really important practical issues to be considered in any Bayesian analysis and we cover  
2881 some of these briefly here before we move on to implementing slightly more complex  
2882 GL(M)Ms in a Bayesian framework.

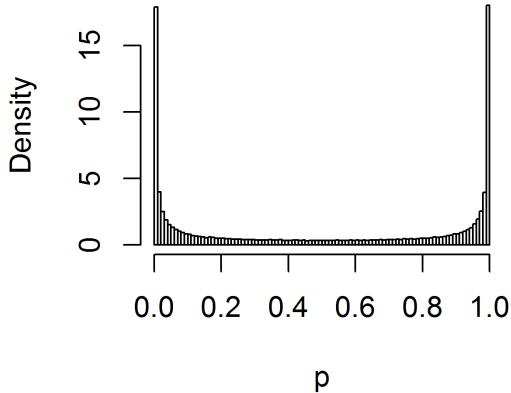
2883 **3.5.1 Choice of prior distributions**

2884 Bayesian analysis requires that we choose prior distributions for all of the structural pa-  
2885 rameters of the model (we use the term structural parameter to mean all parameters that  
2886 aren’t customary thought of as latent variables). We will strive to use priors that are  
2887 meant to express little or no prior information - default or customary “non-informative”  
2888 or diffuse priors. This will be  $\text{Uniform}(a, b)$  priors for parameters that have a natural  
2889 bounded support and, for parameters that live on the real line we use either (1) diffuse  
2890 normal priors, as we did in the linear regression example above; (2) improper uniform  
2891 priors which have unbounded support, e.g.,  $[\theta] \propto 1$ , or (3) sometimes even a bounded  
2892  $\text{Uniform}(a, b)$  prior, if that greatly improves the performance of **WinBUGS** or other  
2893 software doing the MCMC for us. In **WinBUGS** a prior with low precision,  $\tau$ , where  
2894  $\tau = 1/\sigma^2$ , such as  $\text{Normal}(0, .01)$  will typically be used. Of course  $\tau = 0.01$  ( $\sigma^2 = 100$ )  
2895 might be very informative for a regression parameter depending on its magnitude and  
2896 scaling of  $x$ . Therefore, we recommend that predictor variables (covariates) *always* be  
2897 standardized to have mean 0 and variance 1.

2898     **Lack of invariance of priors to transformation.** Clearly there are a lot of choices  
 2899 for ostensibly non-informative priors, and the degree of non-informativeness depends on  
 2900 the parameterization. For example, a natural non-informative prior for the intercept of a  
 2901 logistic regression

$$\text{logit}(p_i) = \beta_0 + \beta_1 x_i$$

2902 would be a very diffuse normal prior,  $[\beta_0] = \text{Normal}(0, \text{Large})$  or even  $\beta_0 \sim \text{Uniform}(-\text{Large}, \text{Large})$ .  
 2903 However, we might also use a prior on the parameter  $p_0 = \text{logit}^{-1}(\beta_0)$ , which is  $\Pr(y=1)$   
 2904 for the value  $x=0$ . Since  $p_0$  is a probability a natural choice is  $p_0 \sim \text{Uniform}(0, 1)$ . These  
 2905 priors are very different in their implications. For example, if we choose the normal prior  
 2906 for  $\beta_0$  with variance  $\text{Large} = 5^2$  and look at the implied prior for  $p_0$  we have the result  
 shown in Fig. 3.2 which looks nothing like a  $\text{Uniform}(0, 1)$  prior. These two priors can



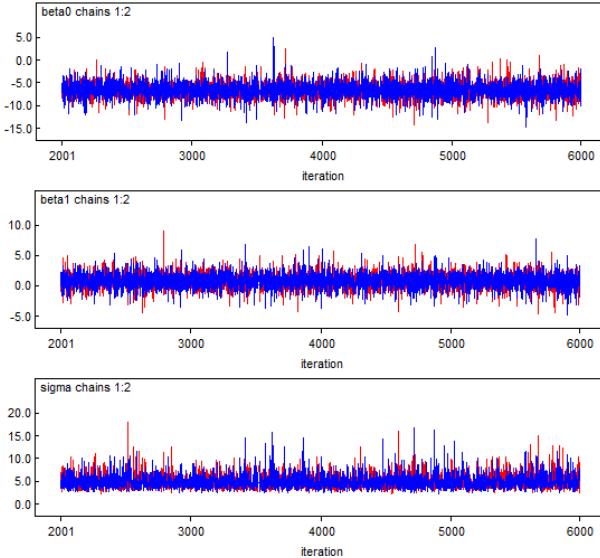
**Figure 3.2.** Implied prior for  $p_0 = \exp(\beta_0)/(1 + \exp(\beta_0))$  if  $\beta_0 \sim \text{Normal}(0, 5^2)$ .

2907 affect results (see Sec. 4.4.2 for an illustration of this for a real data set), yet they are  
 2908 both sensible non-informative priors. Despite this, it is often the case that priors will have  
 2909 little or no impact on the results. Choice of priors and parameterization is very much  
 2910 problem-specific and often largely subjective. Moreover, it also affects the behavior of  
 2911 MCMC algorithms and therefore the analyst needs to pay some attention to this issue  
 2912 and possibly try different things out. Most standard Bayesian analysis books address  
 2913 issues related to specification and effect of prior distribution choice in some depth. Some  
 2914 good references include Kass and Wasserman (1996), Gelman (2006) and Link and Barker  
 2915 (2010).

2917 **3.5.2 Convergence and so-forth**

2918 Once we have carried out an analysis by MCMC, there are many other practical issues  
 2919 that we have to confront. One characteristic of MCMC sampling is that Markov chains  
 2920 take some time to converge to their stationary distribution - in our case the posterior  
 2921 distribution for some parameter given data,  $[\theta|y]$ . Only when the Markov chain has  
 2922 reached its stationary distribution, the generated samples can be used to characterize the  
 2923 posterior distribution. Thus, one of the most important issues we need to address is “have  
 2924 the chains converged?” Since we do not know what the stationary posterior distribution  
 2925 of our Markov chain should look like (this is the whole point of doing an MCMC analysis),  
 2926 we effectively have no means to assess whether or not it has truly converged to this desired  
 2927 distribution. Most MCMC algorithms only guarantee that, eventually, the samples being  
 2928 generated will be from the target posterior distribution, but no-one can tell us how long  
 2929 this will take. Also, you only know the part of your posterior distribution that the Markov  
 2930 chain has explored so far – for all you know the chain could be stuck in a local maximum,  
 2931 while other maxima remain completely undiscovered. Acknowledging that there is truly  
 2932 nothing we can do to ever prove convergence of our MCMC chains, there are several things  
 2933 we can do to increase the degree of confidence we have about the convergence of our chains.  
 2934 Some problems are easily detected using simple plots, such as a time-series plot, where  
 2935 parameter values of each MCMC iteration are plotted against the number of iterations.  
 2936 Fig. 3.3 shows the time series plots for the three parameters –  $\beta_0$ ,  $\beta_1$  and  $\sigma$  – from our  
 2937 linear regression example, taken from the **WinBUGS** GUI before closing it to return to  
**R**.

2938 Typically a period of transience is observed in the early part of the MCMC algorithm,  
 2939 and this is usually discarded as the “burn-in” period. In our linear regression example,  
 2940 within the `bugs()` call we set the burn-in period as 2000 iterations so these are auto-  
 2941 matically removed by **WinBUGS** and are not part of the output (but Fig. 3.6 shows a  
 2942 time-series plot that starts at iteration 0 with a clearly visible burn-in period). The quick  
 2943 diagnostic to whether convergence has been achieved is that your Markov chains look  
 2944 “grassy” – this seems a reasonable statement for the plots in Fig. 3.3. Another way to  
 2945 check convergence is to update the parameters some more and see if the posterior changes.  
 2946 If the chains have converged to the posterior, the posterior mean, confidence intervals, and  
 2947 other summaries should be relatively static as we continue to run the algorithm. Yet an-  
 2948 other option, and one generally implemented in **WinBUGS**, is to run several Markov  
 2949 chains and to start them off at different initial values that are over-dispersed relative to  
 2950 the posterior distribution. Such initial values help to explore different areas of the param-  
 2951 eter space simultaneously; if, after a while, all chains oscillate around the same average  
 2952 value, chances are good that they indeed converged to the posterior distribution. Gelman  
 2953 and Rubin came up with the so-called “R-hat” statistic ( $\hat{R}$ ) or Brooks-Gelman-Rubin  
 2954 statistic that essentially compares within-chain and between-chain variance to check for  
 2955 convergence of multiple chains (Gelman et al., 1996). The R-hat statistic should be close  
 2956 to 1 if the Markov chains have converged and sufficient posterior samples have been ob-  
 2957 tained. For the linear regression example, we ran two parallel chains (also specified in the  
 2958 `bugs()` call) and **WinBUGS** returns the  $\hat{R}$  statistic for us as part of the summary model  
 2959 output. If you look back to Sec. 3.4.1 you see that  $\hat{R} = 1$  for all parameters of the linear  
 2960 model. In practice,  $\hat{R} \leq 1.2$  may be good enough for some problems. For some models you  
 2961 can’t actually realize a low  $\hat{R}$ . E.g., if the posterior is a discrete mixture of distributions



**Figure 3.3.** Time-series plots for parameters from a linear regression run in **WinBUGS** using two parallel Markov chains.

then you can be misled into thinking that your Markov chains have not converged when in fact the chains are just jumping back and forth in the posterior state-space. This happens in some of indicator variable model selection discussed in Chapt. 8. Often, when there is little information about a parameter in the data, or when parameters are on the boundary of the parameter space, convergence will appear to be poor also. These kinds of situations are normally ok and you need to think really hard about the context of the model and the problem before you conclude that your MCMC algorithm is ill-behaved.

Some models exhibit “poor mixing” of the Markov chains (or “slow convergence”) in which case the samples might well be from the posterior (i.e., the Markov chains have converged to the proper stationary distribution) but simply mix or move around the posterior rather slowly. Poor mixing can happen for many reasons – when parameters are highly correlated (even confounded), or barely identified from the data, or the algorithms are very terrible and probably other reasons as well.

Slow mixing equates to high autocorrelation in the Markov chain - the successive draws are highly correlated, and thus we need to run the MCMC algorithm much longer to get an effective sample size that is sufficient for estimation, or to reduce the MC error (see below) to a tolerable level. A strategy often used to reduce autocorrelation is “thinning”, where only every  $m^{th}$  value of the Markov chain output is kept. However, thinning is necessarily inefficient from the stand point of inference - you can always get more precise posterior estimates by using all of the MCMC output regardless of the level of autocorrelation

(MacEachern and Berliner, 1994; Link and Eaton, 2011). Practical considerations might necessitate thinning, even though it is statistically inefficient. For example, in models with many parameters or other unknowns being tabulated, the output files might be enormous and unwieldy to work with. In such cases, thinning is perfectly reasonable. In many cases, how well the Markov chains mix is strongly influenced by parameterization, standardization of covariates, and the prior distributions being used. Some things work better than others, and the investigator should experiment with different settings and remain calm when things don't work out perfectly.

**Is the posterior sample large enough?** The subsequent samples generated from a Markov chain are not *independent* samples from the posterior distribution, due to the correlation among samples introduced by the Markov process<sup>6</sup> and the sample size has to be adjusted to account for the autocorrelation in subsequent samples (see Chapt. 8 in Robert and Casella (2010) for more details). This adjusted sample size is referred to as the effective sample size. Checking the degree of autocorrelation in your Markov chains and estimating the effective sample size your chain has generated should be part of evaluating your model output. **WinBUGS** will automatically return the effective sample size for all monitored parameters, as we saw in our linear regression example (the “n.eff” column of the summary output). If you find that your supposedly long Markov chain has only generated a very short effective sample, you should consider a longer run. What exactly constitutes a reasonable effective sample size is hard to say. A more palpable measure of whether you've run your chain for enough iterations is the time-series or Monte Carlo error - the “noise” introduced into your samples by the stochastic MCMC process. The MC error is printed by default in summaries produced in the **WinBUGS** GUI, which can be reproduced in **R** using `bugs.log('log.txt')$stats` (note that “log.txt” refers to a model log file that **WinBUGS** automatically creates in the working directory; it is overwritten with every new model you run unless you save it under a different name).

```
3009 > bugs.log('log.txt')$stats
3100 $stats
3101      mean      sd   Mcerror    2.5%   median   97.5% start sample
3102 beta0    -6.64700 1.60300 0.0179400 -9.7140 -6.70800 -3.2730 2001 8000
3103 beta1     0.82100 1.19000 0.0116800 -1.4900  0.82560  3.1800 2001 8000
3104 deviance  58.66000 3.08800 0.0506800 55.0700 57.93000 66.8400 2001 8000
3105 sigma      4.96800 1.52300 0.0248300  2.9350  4.68100  8.7410 2001 8000
3106 tau        0.05074 0.02677 0.0003651  0.0131  0.04564  0.1162 2001 8000
```

When using **JAGS** the `summary` command will automatically produce the MC error (which is called “Time-series SE” in **JAGS**). You want the MC error to be smallish relative to the magnitude of the parameter and what smallish means will depend on the purpose of the analysis. For a preliminary analysis you might settle for a few percent whereas for a final analysis then certainly less than 1% is called for. You can run your MCMC algorithm as long as it takes to achieve that. A consequence of the MC error is that even for the exact same model, results will usually be slightly different. Thus, as a good rule of thumb, you should avoid reporting MCMC results to more than 2 or 3 significant digits!

---

<sup>6</sup>In case you are not familiar with Markov chains, for  $T$  random samples  $\theta^{(1)}, \dots, \theta^{(T)}$  from a Markov chain the distribution of  $\theta^{(t)}$  depends only on the immediately preceding value,  $\theta^{(t-1)}$ .

3025 **3.5.3 Bayesian confidence intervals**

3026 The 95% Bayesian confidence interval based on percentiles of the posterior is not a unique  
 3027 interval - there are many of them. The so-called “highest posterior density” (HPD) inter-  
 3028 val is an alternative, defined as the narrowest interval that contains *at least* 95% of the  
 3029 posterior mass. As a result (of the *at least* clause), for discrete parameters, the 95% HPD  
 3030 is not often exactly 95% but usually slightly more conservative than nominal.

3031 **3.5.4 Estimating functions of parameters**

3032 A benefit of analysis by MCMC is that we can seamlessly estimate functions of parameters  
 3033 by simply tabulating the desired function of the simulated posterior draws. For example,  
 3034 if  $\theta$  is the parameter of interest and let  $\theta^{(i)}$  for  $i = 1, 2, \dots, M$  be the posterior samples  
 3035 of  $\theta$ . Let  $\eta = \exp(\theta)$ , then a posterior sample of  $\eta$  can be obtained simply by computing  
 3036  $\exp(\theta^{(i)})$  for  $i = 1, 2, \dots, M$ . Almost all SCR models in this book involve at least 1 derived  
 3037 parameter. For example, density  $D$  is a derived parameter, being a function of population  
 3038 size  $N$  and the area  $A$  of the underlying state-space of the point process (see Chapt. 5).

3039 **Example: Finding the optimum value of a covariate.** As another example of  
 3040 estimating functions of model parameters, suppose that the normal regression model from  
 3041 Sec. 3.4.1 had a quadratic response function of the form

$$\mathbb{E}(y_i) = \beta_0 + \beta_1 x_i + \beta_2 x_i^2.$$

3042 Then the optimum value of  $x$ , i.e., that corresponding to the optimal expected response,  
 3043 can be found by setting the derivative of this function to 0 and solving for  $x$ . We find that

$$df/dx = \beta_1 + 2 * \beta_2 x = 0$$

3044 yields that  $x_{opt} = -\beta_1/(2 * \beta_2)$ . We can just take our posterior draws for  $\beta_1$  and  $\beta_2$   
 3045 and obtain a posterior sample of  $x_{opt}$  by this simple calculation applied to the posterior  
 3046 output. As an exercise, take the normal model above and simulate a quadratic response  
 3047 and then describe the posterior distribution of  $x_{opt}$ .

## 3.6 POISSON GLMS

3048 The Poisson GLM (also known as “Poisson regression”) is probably the most relevant  
 3049 and important class of models in all of ecology. The basic model assumes observations  
 3050  $y_i; i = 1, 2, \dots, n$  follow a Poisson distribution with mean  $\lambda$  which we write

$$y_i \sim \text{Poisson}(\lambda)$$

3051 Commonly  $y_i$  is a count of animals or plants at some point in space (“site”)  $i$ , and  $\lambda$   
 3052 might vary over sites as well. For example,  $i$  might index point count locations in a  
 3053 forest, survey route centers, or sample quadrats, or similar, and we are interested in how  
 3054  $\lambda$  depends on site characteristics such as habitat. If covariates are available it is typical to  
 3055 model them as linear effects on the log mean. If  $x_i$  is some measured covariate associated  
 3056 with observation  $i$ , then,

$$\log(x_i) = \beta_0 + \beta_1 x_i$$

3057 While we only specify the mean of the Poisson model directly, the Poisson model (and  
 3058 all GLMs) has a “built-in” variance which is directly related to the mean. In this case,  
 3059  $\text{Var}(y) = \mathbb{E}(y) = \lambda$ . Thus the model accommodates a linear increase in variance with the  
 3060 mean.

### 3061 3.6.1 Example: Breeding Bird Survey data

3062 As an example we consider a classical situation in ecology where counts of an organism  
 3063 are made at a collection of spatial locations. In this particular example, we have  
 3064 mourning dove (*Zenaida macroura*) counts made along North American Breeding Bird  
 3065 Survey (BBS) routes in Pennsylvania, USA. A route consists of 50 stops separated by  
 3066 0.5 miles. For the purposes here we are defining  $y_i$  = route total count and the sample  
 3067 location will be marked by the center point of the BBS route. The survey is run annually  
 3068 and the data set we analyze is 1966-1998. BBS data can be obtained online at  
 3069 <http://www.pwrc.usgs.gov/bbs/>, but the particular chunk of data we will be using here  
 3070 is also included in the **scrbook** package (**data(bbsdata)**). We will make use of the whole  
 3071 data set shortly but for now we’re going to focus on a specific year of counts (1990) for  
 3072 the sake of building a simple model. In 1990 there were 77 active routes; this data set  
 3073 contains rows which index the unique route, column 1 is the route ID, columns 2-3 are  
 3074 the route coordinates (longitude/latitude), column 4 is a habitat covariate “forest cover”  
 3075 (standardized, see below) and the remaining columns are the yearly counts. Years for  
 3076 which a survey was not conducted on a route are coded as “NA” in the data matrix. We  
 3077 imagine that this will be a typical format for many ecological studies, perhaps with more  
 3078 columns representing covariates. To read in the data and display the first few elements of  
 3079 the data frame containing the counts, do this:

```
3080 > data(bbsdata)           #  loads data frame 'bbs'  

3081 > bbsdata$counts[1:2,1:6]  

3082  

3083      X     lon     lat   habitat X66 X67  

3084 1 72002 -80.445 41.501 -0.3871372 NA 24  

3085 2 72003 -80.347 41.214 -1.0171629 NA NA
```

3086 It is useful to display the spatial pattern in the observed counts. For that we use a  
 3087 spatial dot plot – where we plot the coordinates of the observations and mark the color  
 3088 of the plotting symbol based on the magnitude of the count. We have a special plotting  
 3089 function for that which is called **spatial.plot()** and it is available with the supplemental  
 3090 **R** package **scrbook**. Actually, what we want to do here is plot the log-counts (+1 of  
 3091 course) which (Fig. 3.4) display a notable pattern that could be related to something.  
 3092 The **R** commands for obtaining this figure are:

```
3093 > library(scrbook)  

3094 > data(bbsdata)  

3095 > library(maps)  

3096  

3097 > y <- bbsdata$counts[, "X90"] # Pick year 1990  

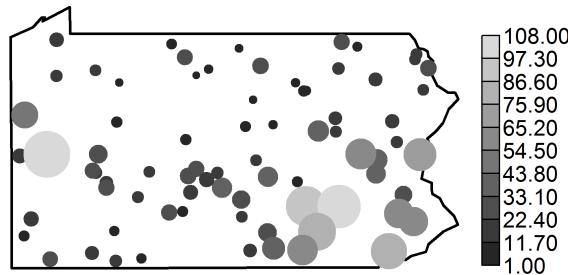
3098 > notna <- !is.na(y)
```

```

3099 > y <- y[notna]
3100 > locs <- bbsdata$counts[notna,c("lon","lat")]
3101 > sz <- y/max(y)
3102
3103 > par(mar=c(3,3,3,6))
3104 > plot(locs,pch=" ",axes=FALSE,xlim=range(locs[,1])+c(-.3,+.3),
3105   ylim=c(range(locs[,2]) + c(-.6,.6)), xlab=" ",ylab=" ")
3106 > map('state', regions='pennsylvania', add=TRUE, lwd=2)
3107 > spatial.plot(bbsdata$counts[notna,2:3], y, cx=1+sz*6, add=TRUE)

```

3108 We can ponder the potential effects that might lead to dove counts being high - corn  
 3109 fields, telephone wires, barn roofs along with misidentification of pigeons, these could all  
 3110 correlate reasonably well with the observed count of mourning doves. Unfortunately we  
 don't have any of that information. However, we do have a measure of forest cover (pro-



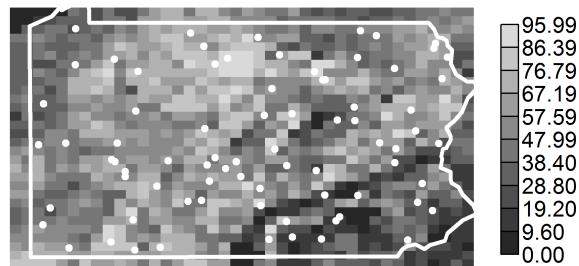
**Figure 3.4.** Mourning dove counts along North American Breeding Bird Survey routes in Pennsylvania (year = 1990). Plot symbol shading and circle size is proportional to raw count.

3111 vided in the data frame `bbsdata$habitat`) which can be plotted using the `spatial.plot`  
 3112 function with the following R commands  
 3113

```

3114 > habdata <- bbsdata$habitat
3115 > map('state',regions="penn",lwd=2)
3116 > I <- matrix(NA, nrow=30, ncol=40)
3117 > I <- matrix(habdata[,"dfor"], ncol=40, byrow=FALSE)
3118 > ux <- unique(habdata[,2])

```



**Figure 3.5.** Forest cover (percent deciduous) in Pennsylvania. BBS route locations are shown by white dots.

```

3119 > uy <- sort(unique(habdata[,3]))
3120
3121 > par(mar=c(3,3,3,6))
3122 > plot(locs,pch=" ", axes=FALSE, xlim=range(locs[,1])+c(-.3,+.3),
3123   ylim=c(range(locs[,2]) + c(-.6,.6)), xlab=" ",ylab=" ")
3124 > image(ux,uy,rot(I), add=TRUE, col=gray(seq(3,17,,10)/20) )
3125 > map('state', regions='pennsylvania', add=TRUE, lwd=3, col="white")
3126 > image.scale(I, col=gray(seq(3,17,,10)/20) )
3127 > points(locs,pch=20, col="white")

```

3128 The result appears in Fig. 3.5. We see a prominent pattern that indicates high forest  
 3129 coverage in the central part of the state and low forest cover in the SE. Inspecting the  
 3130 previous figure of the raw counts suggests a relationship between counts and forest cover  
 3131 which is perhaps not surprising.

### 3132 3.6.2 Doing it in WinBUGS

3133 Here we demonstrate how to fit a Poisson GLM in **WinBUGS** using the covariate  $x_i =$   
 3134 forest cover along BBS route  $i$ . It is advisable that  $x_i$  be standardized in most cases as  
 3135 this will improve mixing of the Markov chains. We have pre-standardized the forest cover  
 3136 covariate for the BBS route locations, and so we don't have to worry about that here. To  
 3137 read the BBS data into **R** and get things set up for **WinBUGS** we issue the following  
 3138 commands:

```

3139 > library(scrbook)
3140 > data(bbsdata)
3141
3142 > y <- bbsdata$counts[, "X90"] # Pick year 1990
3143 > notna <- !is.na(y)
3144 > y <- y[notna]
3145
3146 ## Forest cover already standardized here:
3147 > habitat <- bbsdata$counts[notna, "habitat"]
3148 > M <- length(y)
3149
3150 > library(R2WinBUGS) # Load R2WinBUGS
3151 > data <- list (y=y, M=M, habitat=habitat) # Bundle data for WinBUGS

```

3152 Now we write out the Poisson model specification in **WinBUGS** pseudo-code, provide  
 3153 initial values, identify parameters to be monitored and then execute **WinBUGS**:

```

3154 > cat("
3155 model{
3156   for (i in 1:M){
3157     y[i] ~ dpois(lam[i])
3158     log(lam[i]) <- beta0+beta1*habitat[i]
3159   }
3160   beta0 ~ dunif(-5,5)
3161   beta1 ~ dunif(-5,5)
3162 }
3163 ",file="PoissonGLM.txt")

3164 > inits <- function() list ( beta0=rnorm(1),beta1=rnorm(1) )
3165 > parameters <- c("beta0","beta1")
3166 > out <- bugs(data, inits, parameters, "PoissonGLM.txt", n.thin=2,n.chains=2,
3167   n.burnin=2000,n.iter=6000,debug=TRUE,working.dir=getwd())

```

3168 The **WinBUGS** output can be viewed in **R** using the `print` command:

```

3169 print(out,digits=2)
3170 Inference for Bugs model at "PoissonGLM.txt", fit using WinBUGS,
3171 2 chains, each with 6000 iterations (first 2000 discarded), n.thin = 2
3172 n.sims = 4000 iterations saved
3173      mean    sd    2.5%    25%    50%    75%   97.5% Rhat n.eff
3174 beta0     3.15  0.02    3.10    3.13    3.15    3.17    3.20     1  4000
3175 beta1    -0.50  0.02   -0.54   -0.51   -0.50   -0.48   -0.46     1  4000
3176 deviance 1116.56 1.95 1115.00 1115.00 1116.00 1117.00 1122.00     1  4000

```

### 3177 3.6.3 Constructing your own MCMC algorithm

3178 At this point it might be helpful to suffer through an example building a custom MCMC  
 3179 algorithm. Here, we develop an MCMC algorithm for the Poisson regression model, using

3180 a Metropolis-within-Gibbs sampling framework. Building MCMC algorithms is covered in  
 3181 more detail in Chapt. 17 where you can also find step-by-step instructions for Metropolis-  
 3182 within-Gibbs samplers, should the following section move through all this material too  
 3183 quickly.

3184 We will assume that the two parameters,  $\beta_0$  and  $\beta_1$ , have diffuse normal priors, say  
 3185  $[\beta_0] = \text{Normal}(0, 100)$  and  $[\beta_1] = \text{Normal}(0, 100)$  where each has *standard deviation* 100  
 3186 (recall that **WinBUGS** parameterizes the normal in terms of  $1/\sigma^2$ ). We need to assem-  
 3187 ble the relevant elements of the model which are these two prior distributions and the  
 3188 likelihood  $[\mathbf{y}|\beta_0, \beta_1] = \prod_i [y_i|\beta_0, \beta_1]$  which is, mathematically, the product of the Poisson  
 3189 pmf evaluated at each  $y_i$ , given particular values of  $\beta_0$  and  $\beta_1$ . Next, we need to identify  
 3190 the full conditionals  $[\beta_0|\beta_1, \mathbf{y}]$  and  $[\beta_1|\beta_0, \mathbf{y}]$ . We use the all-purpose rule for constructing  
 3191 full conditionals (section 3.3.2) to discover that:

$$[\beta_0|\beta_1, \mathbf{y}] \propto \left\{ \prod_i [y_i|\beta_0, \beta_1] \right\} [\beta_0]$$

3192 Mathematically, the full conditional is of the form

$$[\beta_0|\beta_1, \mathbf{y}] \propto \left\{ \prod_i \exp(-\exp(\beta_0 + \beta_1 x_i)) \exp(\beta_0 + \beta_1 x_i)^{y_i} \right\} \exp\left(-\frac{\beta_0^2}{2 * 100}\right)$$

3193 which you can program as an **R** function with arguments  $\beta_0$ ,  $\beta_1$  and  $\mathbf{y}$  without difficulty.

3194 The full-conditional for  $\beta_1$  is:

$$[\beta_1|\beta_0, \mathbf{y}] \propto \left\{ \prod_i [y_i|\beta_0, \beta_1] \right\} [\beta_1]$$

3195 which has a similar mathematical representation except the prior is expressed in terms  
 3196 of  $\beta_1$  instead of  $\beta_0$ . Remember, we could replace the “ $\propto$ ” with “=” if we put  $[y|\beta_1]$  or  
 3197  $[y|\beta_0]$  in the denominator. But, in general,  $[y|\beta_0]$  or  $[y|\beta_1]$  will be quite a pain to compute  
 3198 and, more importantly, it is a constant as far as the operative parameters ( $\beta_0$  or  $\beta_1$ ,  
 3199 respectively) are concerned. Therefore, the MH acceptance probability will be the ratio  
 3200 of the full-conditional evaluated at a candidate draw to that evaluated at the current  
 3201 value, and so the denominator required to change  $\propto$  to  $=$  winds up canceling from the  
 3202 MH acceptance probability.

3203 Here we will use the so-called random walk candidate generator, which is a Normal  
 3204 proposal distribution, so that, for example,  $\beta_0^* \sim \text{Normal}(\beta_0^t, \delta)$  where  $\delta$  is the standard-  
 3205 deviation of the proposal distribution, which is just a tuning parameter that is set by  
 3206 the user and adjusted to achieve efficient mixing of chains (see Sec. 17.3.2). We remark  
 3207 also that calculations are often done on the log-scale to preserve numerical integrity of  
 3208 things when quantities evaluate to small or large numbers, so keep in mind, for example,  
 3209  $a * b = \exp(\log(a) + \log(b))$  for two positive numbers  $a$  and  $b$ . The “Metropolis within  
 3210 Gibbs” algorithm for a Poisson regression turns out to be remarkably simple and is given  
 3211 in Panel 3.1. It is also part of the **scrbook** package and you can run 1000 iterations of it  
 3212 by calling `PoisGLMBBS(y=y, habitat=habitat, niter=1000)` (note that  $y$  = point count  
 3213 data and `habitat` = forest cover have to be defined in your **R** workspace as shown in the  
 3214 previous analysis of these data).

```

> set.seed(2013)      # So we all get the same result

> out <- matrix(NA,nrow=1000,ncol=2)    # Matrix to store the output
> beta0 <- -1                         # Starting values
> beta1 <- -.8

# Begin the MCMC loop ; do 1000 iterations
> for(i in 1:1000){

  # Update the beta0 parameter
  lambda <- exp(beta0+beta1*habitat)
  lik.curr <- sum(log(dpois(y,lambda)))
  prior.curr <- log(dnorm(beta0,0,100))
  beta0.cand <- rnorm(1,beta0,.05)        # generate candidate
  lambda.cand <- exp(beta0.cand + beta1*habitat)
  lik.cand <- sum(log(dpois(y,lambda.cand)))
  prior.cand <- log(dnorm(beta0.cand,0,100))
  mhratio <- exp(lik.cand +prior.cand - lik.curr-prior.curr)
  if(runif(1)< mhratio)
    beta0 <- beta0.cand

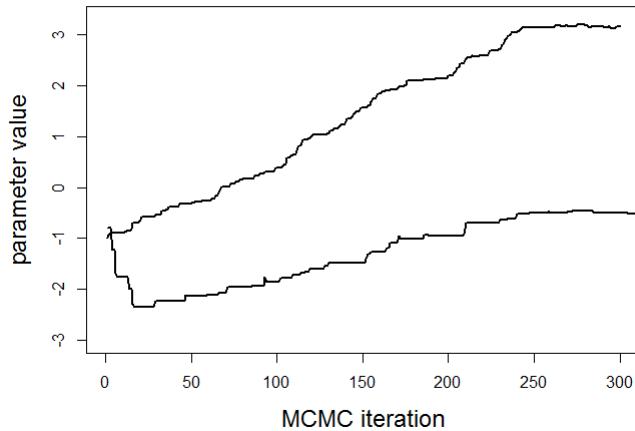
  # update the beta1 parameter
  lik.curr <- sum(log(dpois(y,exp(beta0+beta1*habitat))))
  prior.curr <- log(dnorm(beta1,0,100))
  beta1.cand <- rnorm(1,beta1,.25)
  lambda.cand <- exp(beta0+beta1.cand*habitat)
  lik.cand <- sum(log(dpois(y,lambda.cand)))
  prior.cand <- log(dnorm(beta1.cand,0,100))
  mhratio <- exp(lik.cand + prior.cand - lik.curr - prior.curr)
  if(runif(1)< mhratio)
    beta1 <- beta1.cand

  out[i,] <- c(beta0,beta1)            # save the current values
}

> plot(out[,1],ylim=c(-1.5,3.3),type="l",lwd=2,ylab="parameter value",
       xlab="MCMC iteration")
> lines(out[,2],lwd=2,col="red")

```

Panel 3.1: **R** code to run a Metropolis sampler on a simple Poisson regression model.



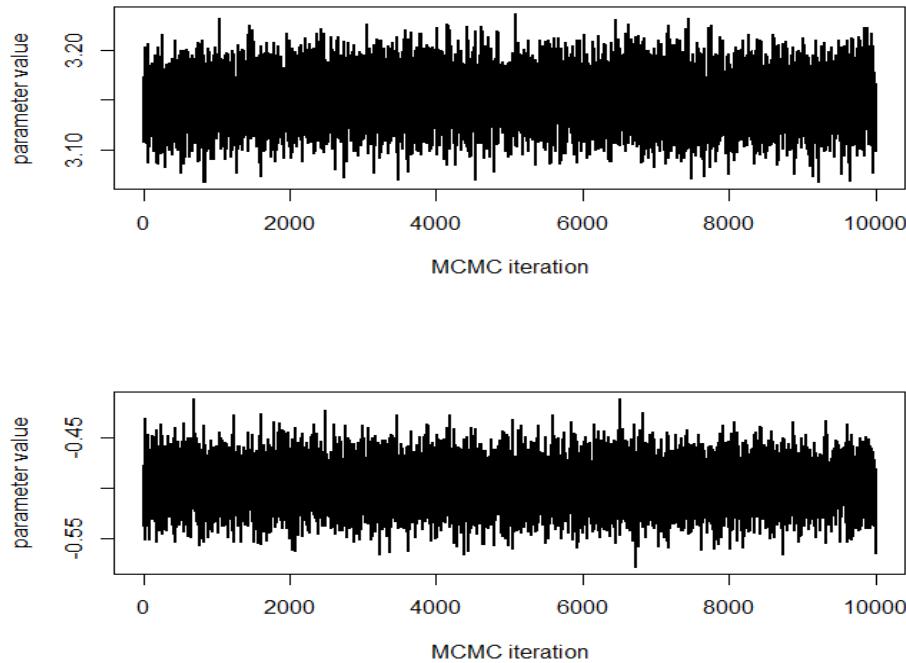
**Figure 3.6.** First 300 MCMC iterations for the Poisson GLM model parameters  $\beta_0$  (top) and  $\beta_1$  (bottom) using a Metropolis-Hastings tuning parameter of  $\delta = 0.05$ .

3215     The first 300 iterations of the MCMC history of each parameter are shown in Fig. 3.6.  
 3216     These chains are not very appealing but a couple of things are evident: We see that the  
 3217     burn-in takes about 250 iterations and that after that chains seem to mix reasonably well,  
 3218     although this is not so clear given the scale of the y-axis, which we have chosen to get  
 3219     both variables on the same graph. We generated 10,000 posterior samples, discarding the  
 3220     first 500 as burn-in, and the result is shown in Fig. 3.7, this time on separate panels for  
 3221     each parameter. The “grassy” look of the MCMC history is diagnostic of Markov chains  
 3222     that are well-mixing and we would generally be very satisfied with results that look like  
 3223     this.

3224     Note that we used a specific set of starting values for these simulations. It should be  
 3225     clear that starting values closer to the mass of the posterior distribution might cause burn-  
 3226     in to occur faster. Note also that we have used a different prior than in our **WinBUGS**  
 3227     model specification given previously. We encourage you to evaluate whether this seems to  
 3228     affect the result.

### 3.7 POISSON GLM WITH RANDOM EFFECTS

3229     In most of this book, we will be dealing with random effects in GLM-like models – similar  
 3230     to what are usually referred to as generalized linear mixed models (GLMMs). We provide  
 3231     a brief introduction of such a model by way of example, extending our Poisson regression  
 3232     model to include a random effect.



**Figure 3.7.** Nice grassy plots of 10,000 MCMC iterations for the Poisson GLM model parameters  $\beta_0$  (top) and  $\beta_1$  (bottom) using a Metropolis-Hastings tuning parameter of  $\delta = 0.05$ .

3233        **The Log-Normal mixture:** The classical situation involves a GLM with a normally  
 3234        distributed random effect that is additive on the linear predictor. For the Poisson case,  
 3235        we have:

$$\log(\lambda_i) = \beta_0 + \beta_1 x_i + \eta_i$$

3236        where  $\eta_i \sim \text{Normal}(0, \sigma^2)$ . In this context,  $\eta$  could represent an error term capturing  
 3237        variation in  $\lambda_i$  not accounted for by the covariates, or overdispersion. It is really amazingly  
 3238        simple to express this model in the **BUGS** language and have **WinBUGS** (or **JAGS**,  
 3239        etc..) draw samples from the posterior distribution. The code for analysis of the BBS  
 3240        dove counts is given as follows:

```
3241 > library(scrbook)
3242 ### Grab the BBS Data as before
3243 > data(bbsdata)
3244 ### Set random seed so that results are repeatable
```

**Table 3.1.** Posterior summaries for Poisson GLMM containing a normal random effect and a habitat effect for mourning dove counts across BBS routes in PA, 1990. Model was fit using WinBUGS, 2 chains, each with 5000 iterations (first 1000 discarded), n.thin = 2 n.sims = 4000 iterations saved.

Parameter	Mean	SD	2.5%	25%	50%	75%	97.5%	Rhat	n.eff
$\beta_0$	2.98	0.08	2.82	2.93	2.98	3.03	3.12	1.00	1400
$\beta_1$	-0.53	0.07	-0.68	-0.58	-0.53	-0.49	-0.38	1.01	350
$\sigma$	0.60	0.06	0.49	0.56	0.59	0.64	0.73	1.00	2000
$\tau$	2.88	0.57	1.88	2.47	2.86	3.24	4.12	1.00	2000
deviance	445.94	12.18	424.00	437.40	445.20	453.90	471.50	1.00	4000

```

3245 > set.seed(2013)
3246 ### Dump the BUGS model into a file
3247 > cat("
3248 model{
3249   for (i in 1:M){ # Observation model, linear predictor, etc..
3250     y[i] ~ dpois(lam[i])
3251     log(lam[i]) <- beta0+ beta1*habitat[i] + eta[i]
3252     frog[i] <- beta1*habitat[i] + eta[i]
3253     eta[i] ~ dnorm(0,tau)
3254   }
3255   # Prior distributions:
3256   beta0 ~ dunif(-5,5)
3257   beta1 ~ dunif(-5,5)
3258   sigma ~ dunif(0,10)
3259   tau <- 1/(sigma*sigma)
3260 }
3261 ",file="model.txt")

3262 > data <- list ("y","M","habitat") # Define the data
3263 > inits <- function() # inits and parameters
3264   list ( beta0=rnorm(1), beta1=rnorm(1), sigma=runif(1,0,4))
3265 > parameters <- c("beta0","beta1","sigma","tau")
3266
3267 > library(R2WinBUGS) # Load and run R2WinBUGS
3268 > out <- bugs (data, inits, parameters, "model.txt", n.thin=2,n.chains=2,
3269   n.burnin=1000, n.iter=5000, debug=TRUE)

```

3270 This produces the posterior summary statistics given in Table 3.1. One thing we notice  
3271 is that the posterior standard deviations of the regression parameters are much higher,  
3272 a result of the extra-Poisson variation allowed for by this model. We would also notice  
3273 much less precise predictions of hypothetical new observations.

### 3.8 BINOMIAL GLMS

3274 Another extremely important class of models in ecology are binomial models. We use  
 3275 binomial models for count data whenever the observations are counts or frequencies and  
 3276 it is natural to condition on a “sample size”, say  $K$ , the maximum frequency possible in  
 3277 a sample. The random variable,  $y \leq K$ , is then the frequency of occurrences out of  $K$   
 3278 “trials”. The parameter of the binomial models is  $p$ , often called “success probability”  
 3279 which is related to the expected value of  $y$  by  $\mathbb{E}(y) = pK$ . Usually we are interested  
 3280 in modeling covariates that affect the parameter  $p$ , and such models are called binomial  
 3281 GLMs, binomial regression models or logistic regression, although logistic regression re-  
 3282 ally only applies when the logistic link is used to model the relationship between  $p$  and  
 3283 covariates (see below).

3284 One of the most typical binomial GLMs occurs when the sample size equals 1 and  
 3285 the outcome,  $y$ , is “presence” ( $y = 1$ ) or “absence” ( $y = 0$ ) of a species. In this case,  $y$   
 3286 has a Bernoulli distribution. This is a classical species distribution modeling situation. A  
 3287 special situation occurs when presence/absence is observed with error (MacKenzie et al.,  
 3288 2002; Tyre et al., 2003). In that case,  $K > 1$  samples are usually needed for effective  
 3289 estimation of model parameters.

3290 In standard binomial regression problems the sample size is fixed by design but in-  
 3291 teresting models also arise when the sample size is itself a random variable. These are  
 3292 the  $N$ -mixture models (Royle, 2004b; Kéry et al., 2005; Royle and Dorazio, 2008; Kéry,  
 3293 2010) and related models (in this case,  $N$  being the sample size, which we labeled  $K$   
 3294 above)<sup>7</sup>. Another situation in which the binomial sample size is “fixed” is closed popula-  
 3295 tion capture-recapture models in which a population of individuals is sampled  $K$  times.  
 3296 The number of times each individual is encountered is a binomial outcome with parameter  
 3297 (encounter probability)  $p$ , based on a sample of size  $K$ . In addition, the total number of  
 3298 unique individuals observed,  $n$ , is also a binomial random variable based on population  
 3299 size  $N$ . We consider such models in Chapt. 4.

#### 3300 3.8.1 Binomial regression

3301 In binomial models, covariates are modeled on a suitable transformation (the link function)  
 3302 of the binomial success probability,  $p$ . Let  $x_i$  denote some measured covariate for sample  
 3303 unit  $i$  and let  $p_i$  be the success probability for unit or subject  $i$ . The standard choice is the  
 3304 logit link function (3.1) but there are many other possible link functions. We sometimes use  
 3305 the complementary log-log (= “cloglog”) link function in ecological applications because  
 3306 it is natural in some cases when the response should scale in relation to area or effort  
 3307 (Royle and Dorazio, 2008, p. 150). As an example, the “probability of observing a count  
 3308 greater than 0” under a Poisson model is  $\Pr(y > 0) = 1 - \exp(-\lambda)$ . In that case, for the  
 3309  $i^{th}$  observation,

$$\text{cloglog}(p_i) = \log(-\log(1 - p_i)) = \log(\lambda_i)$$

3310 so that if you have covariates in your linear predictor for  $\mathbb{E}(y)$  under a Poisson model then  
 3311 they are linear on the complementary log-log link of  $p$ . In models of species occurrence

---

<sup>7</sup>Some of the jargon is actually a little bit confusing here because the binomial index is customarily referred to as “sample size” but in the context of  $N$ -mixture models  $N$  is actually the “population size”

3312 it seems natural to view occupancy as being derived from local abundance  $N$  (Royle  
 3313 and Nichols, 2003; Royle and Dorazio, 2006; Dorazio, 2007). Therefore, models of local  
 3314 abundance in which  $N_i \sim \text{Poisson}(A_i \lambda_i)$  for a habitat patch of area  $A_i$  implies a model  
 3315 for occupancy  $\psi_i$  of the form

$$\text{cloglog}(\psi_i) = \log(A_i) + \log(\lambda_i).$$

3316 We will use the cloglog link in some analyses of SCR models in Chapt. 5 and elsewhere.

3317 **3.8.2 Example: waterfowl banding data**

3318 The standard binomial modeling problem in ecology is that of modeling species distri-  
 3319 butions, where  $K = 1$  and the outcome is occurrence ( $y = 1$ ) or not ( $y = 0$ ) of some  
 3320 species. Such examples abound in books (e.g., Royle and Dorazio (2008, ch. 3); Kéry  
 3321 (2010, ch. 21); Kéry and Schaub (2012, ch. 13)) and in the literature. Therefore, instead,  
 3322 we will consider an example involving band returns of waterfowl in the upper great plains  
 3323 including some Canadian provinces, which were analyzed by Royle and Dubovsky (2001).

3324 For these data,  $y_{it}$  is the number of mallard (*Anas platyrhynchos*) bands recovered out  
 3325 of  $B_{it}$  birds banded at some location  $s_i$  in year  $t$ . In this case  $B_{it}$  is fixed. Thinking about  
 3326 recovery rate as being proportional to harvest rate, we use these data to explore geographic  
 3327 gradients in recovery rate resulting from variability in harvest pressure experienced by  
 3328 different populations. As such, we fit a basic binomial GLM with a linear response to  
 3329 geographic coordinates (including an interaction term). Here we provide the part of the  
 3330 script for creating the model and fitting the model in **WinBUGS**. There are few structural  
 3331 differences between this model and the Poisson GLM fitted previously. The main things  
 3332 are due to the data structure (we have a matrix here instead of a vector) and otherwise  
 3333 we change the distributional assumption to binomial (specified with **dbin**) and then use  
 3334 the **logit** function to relate the parameter  $p_{it}$  to the covariates.

3335 **Dummy variables in BUGS:** In the mallard example, we model the band recovery  
 3336 probability  $p_{it}$  not only as a linear function (on the logit scale) of geographic location, but  
 3337 also allow for variation in  $p_{it}$  with year,  $t$ ;  $t = 1, 2, \dots, T$ . In this particular example there  
 3338 are  $T = 5$  years of data and we could describe the full mallard model with a formula in  
 3339 terms of “dummy variables.” Dummy variables are binary variables, one variable for each  
 3340 level of the categorical variable they describe, such that variable for level  $t$  takes on the  
 3341 value 1 if the observation belongs with level  $t$  and 0 otherwise. So, the mallard model in  
 3342 terms of dummy variables for “year” looks like this:

$$y_{it} \sim \text{Binomial}(p_{it}, B_{it})$$

$$\text{logit}(p_{it}) = \beta_0 + \beta_1 x_{2,it} + \beta_2 x_{3,it} + \beta_3 x_{4,it} + \beta_4 x_{5,it} + \beta_5 \text{Lat}_i + \beta_6 \text{Lon}_i + \beta_7 \text{Lat}_i \text{Lon}_i$$

3343 Here,  $x_2$  to  $x_5$  are the dummy variable vectors of length  $T$  that take on the value of 1  
 3344 when  $t$  corresponds to the respective year and 0 otherwise;  $\beta_0$  is the common intercept  
 3345 term and corresponds to  $t = 1$ ;  $\beta_1 - \beta_4$  describe the difference in  $p_{it}$  for each  $t$  relative to  
 3346  $t = 1$ .

3347 There is a more concise way of implementing such a model with a categorical covariate  
 3348 in **BUGS**, namely, by using indexing instead of dummy variables<sup>8</sup>. Essentially, instead of  
 3349 estimating the difference in  $p$  relative to category 1, we estimate a separate intercept term  
 3350 for each category, so that we have 5 different  $\beta_0$  parameters indexed by  $t$ . This reduces  
 3351 the linear predictor to:

$$\text{logit}(p_{it}) = \beta_{0t} + \beta_5 \text{Lat}_i + \beta_6 \text{Lon}_i + \beta_7 \text{Lat}_i \text{Lon}_i$$

3352 The model can be implemented in the **BUGS** language for the mallard banding data  
 3353 using the following **R** script, provided in the **scrbook** package (see `help(mallard)`):

```
3354 > library(scrbook)
3355 > data(mallard)      # Load mallard data
3356
3357 > cat("
3358 model{
3359   for(t in 1:5){
3360     for (i in 1:nobs){
3361       y[i,t] ~ dbin(p[i,t], B[i,t])
3362       pl[i,t] <- beta0[t]+beta1*X[i,1]+beta2*X[i,2]+beta3*X[i,1]*X[i,2]
3363       p[i,t] <- exp(pl[i,t])/(1+exp(pl[i,t]))
3364     }
3365   }
3366   beta1 ~ dnorm(0,.001)
3367   beta2 ~ dnorm(0,.001)
3368   beta3 ~ dnorm(0,.001)
3369   for(t in 1:5){
3370     beta0[t] ~ dnorm(0,.001)
3371   }
3372 }
3373 ",file="BinomialGLM.txt")

3374 > library(R2WinBUGS)
3375 > data <- list(B=mallard$bandings, y=mallard$recoveries,
3376   X=mallard$locs, nobs=nrow(mallard$locs))
3377 > inits <- function(){ list(beta0=rnorm(5),beta1=0,beta2=0,beta3=0) }
3378 > parms <- c('beta0','beta1','beta2','beta3')
3379 > out <- bugs(data, inits, parms,"BinomialGLM.txt", n.chains=3,
3380   n.iter=2000, n.burnin=1000, n.thin=2, debug=TRUE)
```

3381 Look at the posterior summaries of model parameters in Table 3.2. The basic result  
 3382 suggests a negative east-west gradient and a positive south to north gradient of band  
 3383 recovery probabilities, but no interaction. A map of the response surface is shown in Fig.  
 3384 3.8.

---

<sup>8</sup>Actually, in some cases a model may mix or converge better depending on whether you choose a dummy variable or an indexing description of it, although they are structurally equivalent (Kéry, 2010)

**Table 3.2.** Posterior summaries for the binomial GLM of mallard band recovery rate. Model contains year-specific intercepts ( $\beta_{0t}$ ) and a linear response surface with interaction. Model was fit using **WinBUGS**, and posterior summaries are based on 3 chains, each with 2000 iterations (first 1000 discarded), n.thin = 2 n.sims = 1500 iterations saved.

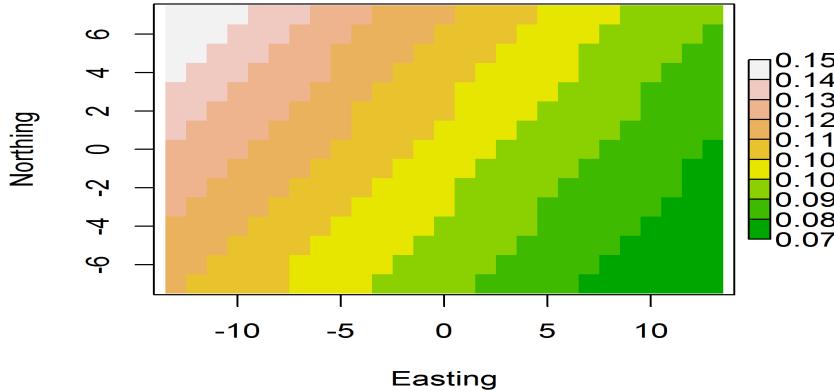
Parameter	Mean	SD	2.5%	50%	97.5%	Rhat	n.eff
$\beta_0[1]$	-2.346	0.036	-2.417	-2.346	-2.277	1.001	1500
$\beta_0[2]$	-2.356	0.032	-2.420	-2.356	-2.292	1.001	1500
$\beta_0[3]$	-2.220	0.035	-2.291	-2.219	-2.153	1.001	1500
$\beta_0[4]$	-2.144	0.039	-2.225	-2.143	-2.068	1.000	1500
$\beta_0[5]$	-1.925	0.034	-1.990	-1.924	-1.856	1.004	570
$\beta_1$	-0.023	0.003	-0.028	-0.023	-0.018	1.001	1500
$\beta_2$	0.020	0.006	0.009	0.020	0.031	1.001	1500
$\beta_3$	0.000	0.001	-0.002	0.000	0.002	1.001	1500
deviance	1716.001	4.091	1710.000	1715.000	1726.000	1.001	1500

### 3.9 BAYESIAN MODEL CHECKING AND SELECTION

3385 In general terms, model checking – or assessing the adequacy of the model – and model  
 3386 selection are quite thorny issues and, despite contrary and, sometimes, strongly held belief  
 3387 among practitioners, there are not really definitive, general solutions to either problem.  
 3388 We're against dogma on these issues and think people need to be open-minded about  
 3389 such things and recognize that models can be useful whether or not they pass certain  
 3390 statistical tests. Some models are intrinsically better than others because they make more  
 3391 biological sense or foster understanding or achieve some objective that some bootstrap or  
 3392 other goodness-of-fit test can't decide for you. That said, it gives you some confidence if  
 3393 your model seems adequate in a purely statistical sense. We provide a very brief overview  
 3394 of concepts here, but provide more detailed coverage in Chapt. 8. See also coverage of  
 3395 these topics in Kéry (2010) and Link and Barker (2010) for specific context related to  
 3396 Bayesian model checking and selection.

#### 3397 3.9.1 Goodness-of-fit

3398 Goodness-of-fit testing is an important element of any analysis because our model re-  
 3399 presents a general set of hypotheses about the ecological and observation processes that  
 3400 generated our data. Thus, if our model “fits” in some statistical or scientific sense, then  
 3401 we believe it to be consistent with the hypotheses that went into the model. More for-  
 3402 mally, we would conclude that the data are *not inconsistent* with the hypotheses, or that  
 3403 the model appears adequate. If we have enough data, then of course we will reject any  
 3404 set of statistical hypotheses. Conversely, we can always come up with a model that fits  
 3405 by making the model extremely complex. Despite this paradox, it seems to us that sim-  
 3406 ple models that you can understand should usually be preferred even if they don't fit,  
 3407 for example if they embody essential mechanisms central to our understanding of things,  
 3408 or if we think that some contributing factors to lack-of-fit are minor or irrelevant to the  
 3409 scientific context and intended use of the model. In other words, models can be useful



**Figure 3.8.** Predicted recovery rates of mallard bands in the upper great plains of North America. Note the negative gradient from the NW to the SE.

irrespective of whether they fit according to some formal statistical test of fit. Yet the tension is there to obtain fitting models, and this comes naturally at the expense of models that can be easily interpreted and studied and effectively used. Unfortunately, conducting a goodness-of-fit test is not always so easy to do. And, moreover, it is never really easy (or especially convenient) to decide if your goodness-of-fit test is worth anything. It might have 0 power! Despite this, we recommend attempting to assess model fit in real applications, as a general rule, and we provide some basic guidance here and some more specific to SCR models in Chapt. 8.

To evaluate goodness-of-fit in Bayesian analyses, we will most often use the Bayesian p-value (Gelman et al., 1996). The basic idea is to define a fit statistic or “discrepancy measure” and compare the posterior distribution of that statistic to the posterior predictive distribution of that statistic for hypothetical perfect data sets for which the model is known to be correct. For example, with count frequency data, a standard measure of fit is the sum of squares of the “Pearson residuals”,

$$D(y_i, \theta) = \frac{(y_i - \mathbb{E}(y_i))}{\sqrt{\text{Var}(y_i)}}$$

The fit statistic based on the squared residuals computed from the observations is

$$T(\mathbf{y}, \theta) = \sum_i D(y_i, \theta)^2$$

which can be computed at each iteration of a MCMC algorithm given the current values of parameters that determine the response distribution. At the same time (i.e., at each

3427 MCMC iteration), the equivalent statistic is computed for a “new” data set, say  $\mathbf{y}^{new}$ ,  
 3428 simulated using the current parameter values. From the new data set, we compute the  
 3429 same fit statistic:

$$T(\mathbf{y}^{new}, \theta) = \sum_i D(y_i^{new}, \theta)^2$$

3430 and the Bayesian p-value is simply the posterior probability  $\text{Pr}(T(\mathbf{y}^{new}) > T(\mathbf{y}))$  which  
 3431 should be close to 0.50 for a good model – one that “fits” in the sense that the observed  
 3432 data set is consistent with realizations simulated under the model being fitted to the  
 3433 observed data. In practice we judge “close to 0.50” as being “not too close to 0 or 1” and,  
 3434 as always, closeness is somewhat subjective. We’re happy with anything  $> .1$  and  $< .9$   
 3435 but might settle for  $> .05$  and  $< 0.95$ . Another useful fit statistic is the Freeman-Tukey  
 3436 statistic, in which

$$D(\mathbf{y}, \theta) = \sum_i (\sqrt{y_i} - \sqrt{\mathbb{E}(y_i)})^2$$

3437 (Brooks et al., 2000), where  $y_i$  is the observed value of observation  $i$  and  $\mathbb{E}(y_i)$  its expected  
 3438 value. In contrast to a Chi-square discrepancy, the Freeman-Tukey statistic removes the  
 3439 need to pool cells with small expected values. In summary, you can see that the Bayesian  
 3440 p-value is easy to compute, and it is widely used as a result.

### 3441 3.9.2 Model selection

3442 In ecology, scientific hypotheses are often manifest as different models or parameters of  
 3443 a model, and so evaluating the importance of different models is fundamental to many  
 3444 ecological studies. For Bayesian model selection we typically use three different methods:  
 3445 First is, let’s say, common sense. If a variable should plausibly be relevant to explaining  
 3446 the data-generating processes, and it has posterior mass concentrated away from 0, then it  
 3447 seems like it should be regarded as important - that is, it is “significant.” This approach  
 3448 seems to have fallen out of favor in ecology over the last 10 or 15 years but in many  
 3449 situations it is a reasonable thing to do.

3450 For regression problems we sometimes use the indicator variable method of Kuo and  
 3451 Mallick (1998), in which we introduce a set of binary variables  $w_k$  for variable  $k$ , and  
 3452 express the model as, e.g., for a single covariate model:

$$\mathbb{E}(y_i) = \beta_0 + w_1 \beta_1 x_i$$

3453 where  $w_1$  is given a Bernoulli prior distribution with some prescribed probability. E.g.,  
 3454  $w_1 \sim \text{Bernoulli}(0.50)$  to provide a prior probability of 0.50 that variable  $x$  should be an  
 3455 element of the linear predictor. The posterior probability of the event  $w_1 = 1$  is a gage of  
 3456 the importance of the variable  $x$ . i.e., high values of  $\text{Pr}(w_1 = 1)$  indicate stronger evidence  
 3457 to support that “ $x$  is in the model” whereas values of  $\text{Pr}(w_1 = 1)$  close to 0 suggest that  
 3458  $x$  is less important. Expansion of the model to include the binary variable  $w_1$  defines a  
 3459 set of 2 distinct models for which we can directly compute the posterior probabilities for,  
 3460 merely by tallying up the posterior frequency of  $w_1$ . See Royle and Dorazio (2008, Chapt.  
 3461 3) for an example in the context of logistic regression.

3462 This approach seems to even work sometimes with fairly complex hierarchical models  
 3463 of a certain form. E.g., Royle (2008) applied it to a random effects model to evaluate the  
 3464 importance of the random effect component of the model. The main problem, which is

3465 really a general problem in Bayesian model selection, is that its effectiveness and results  
 3466 will typically be highly sensitive to the prior distribution on the structural parameters  
 3467 (e.g., see Royle and Dorazio (2008, table 3.6)). The reason for this is obvious: If  $w_1 = 0$   
 3468 for the current iteration of the MCMC algorithm, so that  $\beta$  is sampled from the prior  
 3469 distribution, and the prior distribution is very diffuse, then extreme values of  $\beta$  are likely.  
 3470 Consequently, when the current value of  $\beta$  is far away from the mass of the posterior when  
 3471  $w_1 = 1$ , then the Markov chain may only jump from  $w_1 = 0$  to  $w_1 = 1$  infrequently. One  
 3472 seemingly reasonable solution to this problem is to fit the full model to obtain posterior  
 3473 distributions for all parameters, and then use those as prior distributions in a “model  
 3474 selection” run of the MCMC algorithm (Aitkin, 1991). This seems preferable to more-or-  
 3475 less arbitrary restriction of the prior support to improve the performance of the MCMC  
 3476 algorithm.

3477 A third method that we advocate is subject-matter context. It seems that there are  
 3478 some situations – some models – where one should not have to do model selection because a  
 3479 specific model may be necessitated by the biological context of the problem, thus rendering  
 3480 a formal hypothesis test pointless (Johnson, 1999). Certain aspects of SCR models are  
 3481 such an example. In SCR models, we will see that “spatial location” of individuals is  
 3482 an element of the model. The simpler, reduced, model is an ordinary capture-recapture  
 3483 model which is not spatially explicit (i.e., Chapt. 4), but it seems silly and pointless to  
 3484 think about actually using the reduced model even if we could concoct some statistical  
 3485 test to refute the more complex model. The simpler model is manifestly wrong but, more  
 3486 importantly, not even a plausible data-generating model! Other examples are when effort,  
 3487 area or sample rate is used as a covariate. One might prefer to have such things in models  
 3488 regardless of whether or not they pass some statistical litmus test.

3489 Many problems can be approached using one of these methods. In later chapters  
 3490 (especially Chapt. 8) we will address model selection in specific contexts and we hope  
 3491 those will prove useful for a majority of the situations you might encounter.

### 3.10 SUMMARY AND OUTLOOK

3492 GLMs and GLMMs are the most useful statistical methods in all of ecology. The prin-  
 3493 ciples and procedures underlying these methods are relevant to nearly all modeling and  
 3494 analysis problems in every branch of ecology. Therefore, understanding how to analyze  
 3495 these models is an essential skill for the quantitative ecologist to possess. If you under-  
 3496 stand and can conduct classical likelihood and Bayesian analysis of Poisson and binomial  
 3497 GL(M)Ms, then you will be successful analyzing and understanding more complex classes  
 3498 of models that arise. We will see shortly that spatial capture-recapture models are a  
 3499 type of GL(M)M and thus having a basic understanding of the conceptual origins and  
 3500 formulation of GL(M)Ms and their analysis is extremely useful.

3501 We note that GL(M)Ms are routinely analyzed by likelihood methods but we have  
 3502 focused on Bayesian analysis here in order to develop the tools that are less familiar  
 3503 to most ecologists, and that we will apply in much of the remainder of the book. In  
 3504 particular, Bayesian analysis of models with random effects is relatively straightforward  
 3505 because the models are easy to analyze conditional on the random effect, using MCMC.  
 3506 Thus, we will often analyze SCR models in later chapters by MCMC, explicitly adopting a  
 3507 Bayesian inference framework. In that regard, the various **BUGS** engines (**WinBUGS**,

3508   **OpenBUGS, JAGS**; see also Appendix 1) are enormously useful because they provide  
3509   an accessible platform for carrying out analyses by MCMC by just describing the model,  
3510   and not having to worry about how to actually build MCMC algorithms. That said, the  
3511   **BUGS** language is more important than just to the extent that it enables one to do  
3512   MCMC - it is useful as a modeling tool because it fosters understanding, in the sense  
3513   that it forces you to become intimate with your model. You have to think about and  
3514   write down all of the probability assumptions, and the relationships between observations  
3515   and latent variables and parameters in a way that is ecologically sensible and statistically  
3516   coherent. Because of this, it focuses your thinking on *model construction*, as M. Kéry says  
3517   in his **WinBUGS** book (Kéry, 2010), “**WinBUGS** frees the modeler in you.”

3518   While we have emphasized Bayesian analysis in this chapter, and make primary use of  
3519   it through the book, we will provide an introduction to likelihood analysis in Chapt. 6  
3520   and use those methods also from time to time. Before getting to that, however, it will be  
3521   useful to talk about more basic, conventional closed population capture-recapture models  
3522   and such models are the topic of the next chapter.

3523  
3524

# 4

3525

## CLOSED POPULATION MODELS

3526 In this chapter we introduce ordinary *non-spatial* capture-recapture (CR) models for es-  
3527 timating population size in closed populations. A closed population is one whose size,  $N$ ,  
3528 does not change during the study. Two forms of closure are often discussed: demographic  
3529 closure, meaning that no births or deaths occur, and geographic closure, which states  
3530 that no individuals move onto or off of the sampled area during the study. Although few  
3531 populations are actually closed except during very short time intervals, closed population  
3532 CR models serve as the basis for the development of the rest of the models presented in  
3533 this book, including the models for open populations discussed in Chapt. 16.

3534 We begin with the most basic capture-recapture model, colloquially referred to as  
3535 “model  $M_0$ ” (Otis et al., 1978), in which encounter probability is strictly constant in all  
3536 respects (across individuals, and replicates). This allows us to highlight the basic structure  
3537 of closed population models as binomial GLMs. We then consider some important exten-  
3538 sions of ordinary closed population models that accommodate various types of “individual  
3539 effects” — either in the form of explicit, observed covariates (sex, age, body mass) or  
3540 unstructured “heterogeneity” in the form of an individual random effect, which represent  
3541 unobserved or unmeasured covariates. A special type of individual covariate models is dis-  
3542 tance sampling, which could be thought of as the most primitive spatial capture-recapture  
3543 model. All of these different types of closed population models are closely related to bi-  
3544 nomial (or logistic) regression-type models. In fact, when  $N$  is known, they are precisely  
3545 logistic regression models.

3546 We emphasize Bayesian analysis of capture-recapture models and we accomplish this  
3547 using a method related to classical “data augmentation” from the statistics literature (e.g.,  
3548 Tanner and Wong, 1987). This is a general concept in statistics but, in the context of  
3549 capture-recapture models where  $N$  is unknown, it has a consistent implementation across  
3550 classes of capture-recapture models and one that is really convenient from the standpoint  
3551 of doing MCMC (Royle et al., 2007; Royle and Dorazio, 2012). We use data augmentation  
3552 throughout this book and thus emphasize its conceptual and technical origins and demon-  
3553 strate applications to closed population models. We refer the reader to Kéry and Schaub  
3554 (2012, ch. 6) for an accessible and complementary development of Bayesian analysis of

3555 ordinary, i.e., nonspatial closed population models.

#### 4.1 THE SIMPLEST CLOSED POPULATION MODEL: MODEL $M_0$

3556 To start looking at the simplest capture-recapture model, let's suppose there exists a pop-  
 3557 ulation of  $N$  individuals which we subject to repeated sampling, say over  $K$  "occasions",  
 3558 such as trap nights, where individuals are captured, marked, released, and subsequently  
 3559 recaptured. We suppose that individual encounter histories are obtained, and these are of  
 3560 the form of a sequence of 0's and 1's indicating capture ( $y = 1$ ) or not ( $y = 0$ ) during any  
 3561 sampling occasion. As an example, suppose  $K = 5$  sampling occasions, then an individual  
 3562 captured during occasion 2 and 3 but not otherwise would have an encounter history of  
 3563 the form  $\mathbf{y} = (0, 1, 1, 0, 0)$ . Thus, the observation  $\mathbf{y}_i$  for each individual ( $i = 1, 2, \dots, N$ )  
 3564 is a vector having elements denoted by  $y_{ik}$  for  $k = 1, 2, \dots, K$ . Usually this is organized  
 3565 as a row of a matrix with elements  $y_{ik}$ , see Table 4.1. Except where noted explicitly,  
 3566 we suppose that observations are independent within individuals and among individuals.  
 3567 Formally, this allows us to say that  $y_{ik}$  are independent and identically distributed ("iid")  
 3568 Bernoulli random variables and we may write  $y_{ik} \sim \text{Bernoulli}(p)$ . Consequently, for this  
 3569 very simple model in which  $p$  is constant (i.e., there are no individual or temporal co-  
 3570 variates that affect  $p$ ) the original binary detection variables can be aggregated into the  
 3571 total number of encounters for each individual<sup>1</sup>,  $y_{i\cdot} = \sum_k y_{ik}$ , and the observation model  
 3572 changes from a Bernoulli distribution to a binomial distribution based on a sample of size  
 3573  $K$ . That is

$$y_i = \sum_k y_{ik} \sim \text{Binomial}(p, K)$$

3574 for every individual in the population  $i = 1, 2, \dots, N$ , where  $N$  is the number of individuals  
 3575 in the population (i.e., population size).

3576 We emphasize the central importance of the basic Bernoulli encounter model – an  
 3577 individual is either encountered in a sample, or not – which forms the cornerstone of  
 3578 almost all of classical capture-recapture models, including many spatial capture-recapture  
 3579 models discussed in this book.

3580 Evidently, the basic capture-recapture model is a simplistic version of a logistic-  
 3581 regression model with only an intercept term ( $\text{logit}(p) = \text{constant}$ ). To say that all  
 3582 capture-recapture models are just logistic regressions is a slight over-simplification. In  
 3583 fact, we are proceeding here as if we knew  $N$ . In practice we don't, of course, and esti-  
 3584 mating  $N$  is actually the central objective. But, by proceeding as if  $N$  were known, we  
 3585 can specify a simple model and then deal with the fact that  $N$  is unknown using standard  
 3586 methods that you are already familiar with (i.e., GLMs - see Chapt. 3).

3587 Assuming individuals in the population are encountered independently, the joint prob-  
 3588 ability distribution of the observations is the product of  $N$  binomials

$$\Pr(y_1, \dots, y_N | p) = \prod_{i=1}^N \text{Binomial}(y_i | K, p). \quad (4.1.1)$$

3589 We emphasize that this expression is conditional on  $N$ , in which case we get to observe  
 3590 the  $y_i = 0$  observations and the resulting data are just iid binomial counts. Because this

<sup>1</sup>We use the common "dot notation" to denote having summed over one or more indices of a variable.  $y_{i\cdot} = \sum_j y_{ij}$ ,  $y_{\cdot\cdot} = \sum_i \sum_j y_{ij}$ , etc..

**Table 4.1.** A toy capture-recapture data set with  $n = 6$  observed individuals and  $K = 5$  sample occasions. Under a model with constant encounter probability, the binary detection history data can be summarized in the detection frequency (the total number of detections,  $y_i$ ), which is shown in the right-most column.

indiv $i$	Sample occasion					$y_i$
	1	2	3	4	5	
1	1	0	0	1	0	2
2	0	1	0	0	1	2
3	1	0	0	1	0	2
4	1	0	1	0	1	3
5	0	1	0	0	0	1
$n = 6$	1	0	0	0	0	1

3591 is a binomial regression model of the variety described in Chapt. 3, fitting this model  
 3592 using a **BUGS** engine poses no difficulty.

3593 Equation 4.1.1 can be simplified even further if we reformat the observations as en-  
 3594 counter frequencies. Specifically, let  $n_k$  denote the number of individuals captured exactly  
 3595  $k$  times after  $K$  survey occasions,  $n_k = \sum_{i=1}^N I(y_i = k)$  where  $I()$  is the indicator func-  
 3596 tion evaluating to 1 if its argument is true and 0 otherwise. For sake of illustration, we  
 3597 converted the data from Table 4.1 to this format (Table 4.2). What is important to note  
 3598 is that if we know  $N$ , then we know  $n_0$ , i.e. the number of individuals not captured. In  
 3599 this case, an alternative and equivalent expression to Eq. 4.1.1 is

$$\Pr(y_1, \dots, y_N | p) = \prod_{k=0}^K \pi_k^{n_k} \quad (4.1.2)$$

3600 where  $\pi_k = \Pr(y = k)$  under the binomial model with parameter  $p$  and sample size  $K$ .  
 The essential problem in capture-recapture, however, is that  $N$  is *not* known because the

**Table 4.2.** Data from Table 4.1 reformatted as capture frequencies. Since  $N$  is unknown, the number of individuals not captured ( $n_0$ ) is also unknown.

Number of individuals captured $k$ times ( $n_k$ )	$k$					
	0	1	2	3	4	5
$N - 6$	6	2	3	1	0	0

3601 number of uncaptured individuals ( $n_0$ ) is unknown. Consequently, the observed capture  
 3602 frequencies  $n_k$  are no longer independent because  $n_0$  is a function of the other frequencies,  
 3603  $n_0 = N - \sum_{k=1}^K n_k$ . Hence, their joint distribution is multinomial (e.g., see Illian et al.  
 3604 2008, p. 61):

$$n_0, n_1, \dots, n_K \sim \text{Multinomial}(N, \pi_0, \pi_1, \dots, \pi_K) \quad (4.1.3)$$

3606 We gave a general overview of the multinomial distribution in Sec. 2.2. The multino-  
 3607 mial distribution is the standard model for discrete responses that can fall into a fixed  
 3608 number ( $K + 1$  in this case) of possible categories. In the context of capture-recapture,

3609 the multinomial posits a population of  $N$  individuals with  $K + 1$  possible outcomes de-  
 3610 fined by the possible encounter frequencies: encountered  $y = 1, 2, \dots, K$  times or not  
 3611 encountered at all. These possible outcomes occur with probabilities  $\pi_k$ , which we refer  
 3612 to as “cell probabilities” or in the specific context of capture-recapture, encounter history  
 3613 probabilities.

3614 To fit the model in which  $N$  is *unknown*, we can regard  $n_0$  as a parameter and maximize  
 3615 the multinomial likelihood directly. Direct likelihood analysis of the multinomial model is  
 3616 straightforward, but that is not always sufficiently useful in practice because we seldom  
 3617 are concerned with models for the aggregated encounter history frequencies, which entail  
 3618 that capture probabilities are the same for all individuals. In many instances, including  
 3619 for spatial capture-recapture (SCR) models, we require a formulation of the model that  
 3620 can accommodate individual-level covariates to account for differences in detection among  
 3621 individuals, which we address subsequently in this chapter, and also in Chapt. 7.

3622 **4.1.1 The core capture-recapture assumptions**

3623 This basic capture-recapture model – model  $M_0$  – comes with it a host of specific biological  
 3624 and statistical assumptions. In addition to the basic assumption of population closure,  
 3625 Otis et al. (1978) list the following:

- 3626 1. animals do not lose their marks during the experiment,  
 3627 2. all marks are correctly noted and recorded at each trapping occasion, and  
 3628 3. each animal has a constant and equal probability of capture on each trapping oc-  
 3629 casion.

3630 The remainder of their classic work is dedicated to relaxing assumption 3. While assump-  
 3631 tions 1 and 2 are undoubtedly necessary for inference from basic CR methods to be valid,  
 3632 and while they are also assumed by most of the models we present in the following chap-  
 3633 ters, we refrain from repeatedly making such statements. Our opinion is that all model  
 3634 assumptions are apparent when a model is clearly specified, and it is both redundant and  
 3635 impossible to list all the things not allowed by the model. For example, closed population  
 3636 models also assume that other sources of error do not occur, but it is not necessary to  
 3637 enumerate each possibility. Rather, it is necessary to make clear statements such as

$$y_i \stackrel{iid}{\sim} \text{Bernoulli}(p) \quad \text{for } i = 1, \dots, N.$$

3638 This simple model description carries a tremendous amount of information, and it leaves  
 3639 very little left to say with respect to assumptions. Although we will not always show  
 3640 the *iid* symbol, it will be assumed unless otherwise noted, and this assumption is critical  
 3641 for valid inference. It implies that the encounter of one individual does not affect the  
 3642 encounter of another individual, and encounter does not affect future encounter. Under  
 3643 this assumption, it is easy to write down the likelihood of the parameters and obtain  
 3644 parameter estimates; however, whether or not it is true depends upon biological and  
 3645 sampling issues. If this assumption is deemed false, the model can be discarded in favor  
 3646 of a more realistic alternative. However, once we have settled on our model, statistical  
 3647 inference proceeds by assuming the model is truth—not an approximation to truth—but  
 3648 actual truth.

3649 In spite of the fact that we assume that all models are truth, but we acknowledge that  
 3650 all models are wrong due to their assumptions, assumptions should not be viewed as a  
 3651 necessary evil. In fact, one way to view assumptions is as embodiments of our ecological  
 3652 hypotheses. If we make these assumptions too complex or too specific, then we will never  
 3653 be able to study general phenomena that hold true across space and time. Furthermore,  
 3654 in practice, we will rarely have enough data to estimate the parameters of highly complex  
 3655 models.

#### 3656 4.1.2 Conditional likelihood

3657 We saw that the closed population model is a simple logistic regression model if  $N$  is known  
 3658 and, when  $N$  is unknown, the model is multinomial with index or sample size parameter  
 3659  $N$ . This multinomial model, being conditional on  $N$ , is sometimes referred to as the “joint  
 3660 likelihood” the “full likelihood” or the “unconditional likelihood” (sometimes “model” in  
 3661 place of “likelihood”) (Sanathanan, 1972; Borchers et al., 2002). This formulation differs  
 3662 from the so-called “conditional likelihood” approach in which the likelihood of the observed  
 3663 encounter histories is devised conditional on the event that an individual is captured at  
 3664 least once. To construct this likelihood, we have to recognize that individuals appear  
 3665 or not in the sample based on the value of the random variable  $y_i$ , that is, if and only  
 3666 if  $y_i > 0$ . The observation model is therefore based on  $\Pr(y|y > 0)$ . For the simple  
 3667 case of model  $M_0$ , the resulting conditional distribution is a “zero truncated” binomial  
 3668 distribution which accounts for the fact that we cannot observe the value  $y = 0$  in the data  
 3669 set. Both the conditional and unconditional models are legitimate modes of analysis in  
 3670 all capture-recapture types of studies. They provide equally valid descriptions of the data  
 3671 and, for many practical purposes provide equivalent inferences, at least in large sample  
 3672 sizes (Sanathanan, 1972).

3673 In this book we emphasize Bayesian analysis of capture-recapture models using data  
 3674 augmentation (described in Sec. 4.2 below), which produces yet a third distinct formu-  
 3675 lation of capture-recapture models based on the zero-*inflated* binomial distribution that  
 3676 we describe in the next section. Thus, there are 3 distinct formulations of the model – or  
 3677 modes of analysis – for analyzing all capture-recapture models based on the (1) binomial  
 3678 model for the joint or unconditional specification; (2) zero-truncated binomial that arises  
 3679 “conditional on  $n$ ”; and (3) the zero-inflated binomial that arises under data augmen-  
 3680 tation. Each formulation has distinct model parameters (shown in Table 4.3 for model  
 3681  $M_0$ ).

**Table 4.3.** Modes of analysis of capture-recapture models. Closed population models can be analyzed using the joint or “full likelihood” which contains  $N$  as an explicit parameter, the conditional likelihood which does not involve  $N$ , or by data augmentation which replaces  $N$  with  $\psi$ . Each approach yields a distinct likelihood.

Mode of analysis	parameters in model	statistical model
Joint likelihood	$p, N$	multinomial with index $N$
Conditional likelihood	$p$	zero-truncated binomial
Data augmentation	$p, \psi$	zero-inflated binomial

## 4.2 DATA AUGMENTATION

3682 We consider a method of analyzing closed population models using parameter-expanded  
 3683 data augmentation (PX-DA), which we abbreviate to “data augmentation” or DA, which  
 3684 is useful for Bayesian analysis and, in particular, analysis of models using the various  
 3685 **BUGS** engines and other Bayesian model fitting software. Data augmentation is a general  
 3686 statistical concept that is widely used in statistics in many different settings. The classical  
 3687 reference is Tanner and Wong (1987), but see also Liu and Wu (1999). Data augmentation  
 3688 can be adapted to provide a very generic framework for Bayesian analysis of capture-  
 3689 recapture models with unknown  $N$ . This idea was introduced for closed populations by  
 3690 Royle et al. (2007), and has subsequently been applied to a number of different contexts  
 3691 including individual covariate models (Royle, 2009b), open population models (Royle and  
 3692 Dorazio, 2008, 2012; Gardner et al., 2010a), spatial capture-recapture models (Royle and  
 3693 Young, 2008; Royle et al., 2009a; Gardner et al., 2009), and many others. Kéry and Schaub  
 3694 (2012, Chaps. 6 and 10) provide a good introduction to data augmentation in the context  
 3695 of closed and open population models.

3696 Conceptually, the technique of data augmentation represents a reparameterization  
 3697 of the “complete data” model – i.e., that conditional on  $N$ . The reparameterization  
 3698 is achieved by embedding this data set into a larger data set having  $M > N$  “rows”  
 3699 (individuals) and re-expressing the model conditional on  $M$  instead of  $N$ . The great thing  
 3700 about data augmentation is that we do not need to know  $N$  for this reparameterization.  
 3701 Although this has a whiff of arbitrariness or even outright ad hockery to it, in the choice  
 3702 of  $M$ , it is always possible, in practice, to choose  $M$  pretty easily for a given problem and  
 3703 context and results will be insensitive to choice of  $M^2$ . Then, under data augmentation,  
 3704 analysis is focused on the “augmented data set.” That is, we analyze the bigger data set -  
 3705 the one having  $M$  rows - with an appropriate model that accounts for the augmentation.  
 3706 This is achieved by a Bernoulli sampling process that determines whether an individual  
 3707 in  $M$  is also a member of  $N$ . Inference is focused directly on estimating the proportion  
 3708  $\psi = E[N]/M$ , instead of directly on  $N$ , where  $\psi$  is the “data augmentation parameter.”

### 3709 4.2.1 DA links occupancy models and closed population models

3710 There is a close correspondence between so-called “occupancy” models and closed popu-  
 3711 lation models (see Royle and Dorazio, 2008, Sec. 5.6). In occupancy models (MacKenzie  
 3712 et al., 2002; Tyre et al., 2003) the sampling situation is that  $M$  sites, or patches, are sam-  
 3713 pled multiple times to assess whether a species occurs at the sites. This yields encounter  
 3714 data such as that illustrated in the left panel of Table 4.4. The important problem is that  
 3715 a species may occur at a site, but go undetected, yielding an all-zero encounter history for  
 3716 the site, which in the case of occupancy studies, are *observed*. However, some of the zero  
 3717 vectors will typically correspond to sites where the species in fact *does* occur. Thus, while  
 3718 the zeros are observed, there are too many of them and, in a sense, the inference problem  
 3719 is to partition the zeros into “structural” (fixed) and “sampling” (or stochastic) zeros,  
 3720 where the former are associated with unoccupied sites and the latter with occupied sites  
 3721 where the species went undetected. More formally, inference is focused on the parameter  
 3722  $\psi$ , the probability that a site is occupied.

---

<sup>2</sup>Unless the data set is sufficiently small that parameters are weakly identified

3723 In contrast to occupancy studies, in classical closed population studies, we observe a  
 3724 data set as in the middle panel of Table 4.4 where *no* zeros are observed. The inference  
 3725 problem is, essentially, to estimate how many sampling zeros there are – or should be – in  
 3726 a “complete” data set. This objective (how many sampling zeros?) is precisely the same  
 3727 for both types of problems if an upper limit  $M$  is specified for the closed population model.  
 3728 The only distinction being that, in occupancy models,  $M$  is set by design (i.e., the number  
 3729 of sites in the sample), whereas a natural choice of  $M$  for capture-recapture models may  
 3730 not be obvious. However, the choice of  $M$  induces a uniform prior for  $N$  on the integers  
 3731  $[0, M]$  (Royle et al., 2007). Then, one can analyze capture-recapture models by adding  
 3732  $M - n$  all-zero encounter histories to the data set and regarding the augmented data  
 3733 set, essentially, as a site-occupancy data set, where the occupancy or data augmentation  
 3734 parameter ( $\psi$ ) takes the place of the abundance parameter ( $N$ ).

3735 Thus, the heuristic motivation of data augmentation is to fix the size of the data  
 3736 set by adding *too many* all-zero encounter histories to create the data set shown in the  
 3737 right panel of Table 4.4, and then analyze the augmented data set using an occupancy  
 3738 type model which includes both “unoccupied sites” (in capture-recapture, augmented  
 3739 individuals that are not members of the real population that was sampled) as well as  
 3740 “occupied sites” (in capture-recapture, individuals that are members of the population  
 3741 but that were undetected by sampling) at which detections did not occur. We call these  
 3742  $M - n$  all-zero histories “potential individuals” because they exist to be recruited (in a  
 3743 non-biological sense) into the population, for example during an analysis by MCMC.

3744 To analyze the augmented data set, we recognize that it is a zero-inflated version of  
 3745 the known- $N$  data set. That is, some of the augmented all-zero rows are sampling zeros  
 3746 (corresponding to actual individuals that were missed) and some are “structural” zeros,  
 3747 which do not correspond to individuals in the population. For a basic closed-population  
 3748 model, the resulting likelihood under data augmentation - that is, for the data set of size  
 3749  $M$  – is a simple zero-inflated binomial likelihood. The zero-inflated binomial model can be  
 3750 described “hierarchically”, by introducing a set of binary latent variables,  $z_1, z_2, \dots, z_M$ ,  
 3751 to indicate whether each individual  $i$  is ( $z_i = 1$ ) or is not ( $z_i = 0$ ) a member of the  
 3752 population of  $N$  individuals exposed to sampling. We assume that  $z_i \sim \text{Bernoulli}(\psi)$   
 3753 where  $\psi$  is the probability that an individual in the data set of size  $M$  is a member of  
 3754 the sampled population – in the sense that  $1 - \psi$  is the probability of a “structural zero”  
 3755 in the augmented data set. The zero-inflated binomial model which arises under data  
 3756 augmentation can be formally expressed by the following set of assumptions (we include  
 3757 typical priors for a Bayesian analysis):

$$\begin{aligned} y_i | z_i = 1 &\sim \text{Binomial}(K, p) \\ y_i | z_i = 0 &\sim I(y = 0) \\ z_i &\stackrel{iid}{\sim} \text{Bernoulli}(\psi) \\ \psi &\sim \text{Uniform}(0, 1) \\ p &\sim \text{Uniform}(0, 1) \end{aligned}$$

3758 for  $i = 1, \dots, M$ , where  $I(y = 0)$  is a point mass at  $y = 0$ . It is sometimes convenient to  
 3759 express the conditional-on- $z$  observation model concisely in just one step:

$$y_i | z_i \sim \text{Binomial}(K, z_i p)$$

3760 and we understand this to mean, if  $z_i = 0$ , then  $y_i$  is necessarily 0 because its success  
 3761 probability is  $z_i p = 0$ .

3762 Note that, under data augmentation,  $N$  is no longer an explicit parameter of this  
 3763 model. In its place, we estimate  $\psi$  and functions of the latent variables  $z$ . In particular,  
 3764 under the assumptions of the zero-inflated model,  $z_i \stackrel{iid}{\sim} \text{Bernoulli}(\psi)$ ; therefore,  $N$  is a  
 3765 function of these latent variables:

$$N = \sum_{i=1}^M z_i.$$

3766 Further, we note that the latent  $z_i$  parameters *can be* removed from the model by inte-  
 3767 gration, in which case the joint probability of the data is

$$\Pr(y_1, \dots, y_M | p, \psi) = \prod_{i=1}^M (\psi * \text{Binomial}(y_i | K, p) + I(y_i = 0)(1 - \psi)) \quad (4.2.1)$$

3768 Interpreted as a likelihood, we can directly maximize this expression to obtain the MLEs of  
 3769 the structural parameters  $\psi$  and  $p$  or those of other more complex models (e.g., see Royle,  
 3770 2006). We could estimate these parameters and then use them to obtain an estimator of  
 3771  $N$  using the so-called “Best unbiased predictor” (see Royle and Dorazio, 2012). Normally,  
 3772 however, we will analyze the model in its “conditional-on- $z$ ” form using methods of MCMC  
 3773 either in the **BUGS** engines or using our own MCMC algorithms (see Chapt. 17).

#### 3774 4.2.2 Model $M_0$ in **BUGS**

3775 It is helpful to understand data augmentation by seeing what its effect is on implementing  
 3776 model  $M_0$ . For this model, in which we can aggregate the encounter data to individual-  
 3777 specific encounter frequencies, the augmented data are given by the vector of frequencies  
 3778  $(y_1, \dots, y_n, 0, 0, \dots, 0)$  where the augmented values of  $y = 0$  represent the encounter fre-  
 3779 quency for potential individuals  $y_{n+1}, \dots, y_M$ . The zero-inflated model of the augmented  
 3780 data combines the model of the latent variables,  $z_i \sim \text{Bernoulli}(\psi)$ . The **BUGS** model  
 3781 description of the closed population model  $M_0$  is shown in Panel 4.1. The last line of the  
 3782 model specification provides the expression for computing  $N$  from the data augmentation  
 3783 variables  $z_i$ . Note that, to improve readability of code snippets (especially of large ones),  
 3784 we will sometimes deviate from our standard notation a bit. In this case we use **nind**  
 3785 for  $n$  (the number of encountered individuals), and  $M = nind + nz$  is the total size of the  
 3786 augmented data set. In other cases we might also use **nocc** in place of  $K$  and **ntraps**  
 3787 in place of  $J$ . We find that word definitions make code easier to understand, especially  
 3788 without having to read surrounding text.

3789 Specification of a more general model in terms of the individual encounter observations  
 3790  $y_{ik}$  is not much more difficult than for the individual encounter frequencies. We define  
 3791 the observation model by a double loop and change the indexing of quantities accordingly,  
 3792 i.e.,

```
3793 for(i in 1:(nind+nz)){
  3794   z[i] ~ dbern(psi)
  3795   for(k in 1:K){
    3796     mu[i,k] <- z[i]*p
```

**Table 4.4.** Hypothetical occupancy data set (left), capture-recapture data in standard form (center), and capture-recapture data augmented with all-zero capture histories (right).

site	Occupancy data			Capture-recapture				Augmented C-R			
	k=1	k=2	k=3	ind	k=1	k=2	k=3	ind	k=1	k=2	k=3
1	0	1	0	1	0	1	0	1	0	1	0
2	1	0	1	2	1	0	1	2	1	0	1
3	0	1	0	3	0	1	0	3	1	0	1
4	1	0	1	4	1	0	1	4	1	0	1
5	0	1	1	5	0	1	1	5	1	0	1
.	0	1	1	.	0	1	1	.	0	1	1
.	1	1	1	.	1	1	1	.	0	1	1
.	1	1	1	.	1	1	1	.	1	1	1
1	1	1	.	1	1	1	.	1	1	1	1
n	1	1	1	n	1	1	1	n	1	1	1
.	0	0	0	.	0	0	0	.	0	0	0
.	0	0	0	.	0	0	0	.	0	0	0
0	0	0	.	0	0	0	0	0	0	0	0
0	0	0	.	0	0	0	0	0	0	0	0
0	0	0	.	0	0	0	N	0	0	0	0
.	0	0	0	.	0	0	0	.	0	0	0
.	0	0	0	.	0	0	0	.	0	0	0
M	0	0	0	.	0	0	0	.	0	0	0
				.	.	.	.	.	.	.	.
				.	.	.	.	M	0	0	0

```

3797     y[i,k] ~ dbin(mu[i,k],1)
3798   }
3799 }
```

3800 In this manner, it is straightforward to incorporate covariates on  $p$  for both individuals  
3801 and sampling occasions (see discussion of this below and also Chapt. 7) as well as to devise  
3802 other extensions of the model, including models for open populations (see Chapt. 16).

### 3803 4.2.3 Formal development of data augmentation (DA)

3804 Use of parameter-expanded data augmentation (PX-DA), or DA for short, for solving  
3805 inference problems with unknown  $N$  can be justified as originating from the choice of a  
3806 uniform prior on  $N$ . The Uniform(0,  $M$ ) prior for  $N$  is innocuous in the sense that the  
3807 posterior associated with this prior is equal to the likelihood for sufficiently large  $M$ . One  
3808 way of inducing the Uniform(0,  $M$ ) prior on  $N$  is by assuming the following hierarchical  
3809 prior:

$$\begin{aligned}
N &\sim \text{Binomial}(M, \psi) \\
\psi &\sim \text{Uniform}(0, 1).
\end{aligned} \tag{4.2.2}$$

```

model{
  p ~ dunif(0,1)
  psi ~ dunif(0,1)

  # nind = number of individuals captured at least once
  # nz = number of uncaptured individuals added for DA
  for(i in 1:(nind+nz)){
    z[i] ~ dbern(psi)
    mu[i] <- z[i]*p
    y[i] ~ dbin(mu[i],K)
  }

  N<-sum(z[1:(nind+nz)])
}

```

---

Panel 4.1: Model  $M_0$  under data augmentation. Here  $y$ ,  $K$ ,  $nind$  and  $nz$  are provided as data. The population size,  $N$ , is computed as a function of the data augmentation variables  $z$ .

3810 The model assumptions, specifically the multinomial model (Eq. 4.1.3) and Eq. 4.2.2, may  
 3811 be combined to yield a reparameterization of the conventional model that is appropriate  
 3812 for the augmented data set of known size  $M$ :

$$(n_1, n_2, \dots, n_K) \sim \text{Multinomial}(M, \psi\pi_1, \psi\pi_2, \dots, \psi\pi_K) \quad (4.2.3)$$

3813 This expression arises by removing  $N$  from Eq. 4.1.3 by integrating over the binomial  
 3814 prior distribution for  $N$ . Thus, the models we analyze under data augmentation arise  
 3815 formally by removing the parameter  $N$  from the ordinary closed-population model, which  
 3816 is conditional on  $N$ , by integrating over a binomial prior distribution for  $N$ .

3817 Note that the  $M - n$  unobserved individuals in the augmented data set have probability  
 3818  $\psi\pi(0) + (1 - \psi)$ , indicating that these unobserved individuals are a mixture of individuals  
 3819 that are sampling zeros ( $\psi\pi_0$ ), and belong to the population of size  $N$ , and others that  
 3820 are “structural zeros” (occurring in the augmented data set with probability  $1 - \psi$ ). In  
 3821 Eq. 4.2.3,  $N$  has been eliminated as a formal parameter of the model by marginalization  
 3822 (integration) and replaced with the new parameter  $\psi$ , the data augmentation parameter.  
 3823 However, the full likelihood containing both  $N$  and  $\psi$  can also be analyzed (see Royle  
 3824 et al., 2007).

#### 3825 4.2.4 Remarks on data augmentation

3826 Data augmentation may seem like a strange and mysterious black-box, and likely it is un-  
 3827 familiar to most people, even to many of those with substantial experience with capture-

recapture models. However, it really is just a formal reparameterization of capture-recapture models in which  $N$  is marginalized out of the ordinary (conditional-on- $N$ ) model (by summation over a binomial prior). As a result, we could refer to the resulting model as the “binomial-integrated likelihood” to reflect that an estimator could be obtained from the ordinary likelihood, integrated over a binomial prior. Other such “integrated likelihood” models are sensible. For example, we could place a Poisson prior on  $N$  with mean  $\Lambda$  and marginalize  $N$  over the Poisson prior. This produces a likelihood in which  $\Lambda$  replaces  $N$ , instead of  $\psi$  replacing  $N$ . We note that this type of marginalization (over a Poisson prior) is done by the **R** package **secr** for analysis of spatial capture-recapture models (see Sec. 6.5.3).

We emphasize the motivation for data augmentation being that it produces a data set of fixed size, so that the parameter dimension in any capture-recapture model is also fixed. As a result, MCMC is a relatively simple proposition using standard Gibbs Sampling. And, in particular, capture-recapture models become trivial to implement in **BUGS**. Consider the simplest context—analyzing model  $M_0$  using the occupancy-type model. In this case, DA converts model  $M_0$  to a basic occupancy model, and the parameters  $p$  and  $\psi$  have known full-conditional distributions (in fact, beta distributions) that can be sampled from directly. Furthermore, the data augmentation variables, i.e., the collection of  $z$ 's, can be sampled from Bernoulli full conditionals. MCMC is not much more difficult for complicated models—sometimes the hyperparameters need to be sampled using a Metropolis-Hastings step (e.g., Chapt. 17), but nothing more sophisticated than that is required.

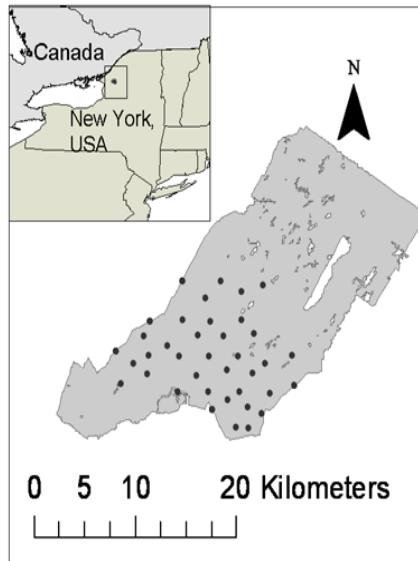
Potential sensitivity of parameter estimates to  $M$  (especially of  $N$ ) might be cause for some concern. The guiding principle is that it should be chosen large enough so that the posterior for  $N$  is not truncated, but it should not be too large due to the increased computational burden. It seems likely that the properties of the Markov chains should be affected by  $M$  and so some optimal choice of  $M$  might exist (Gopalaswamy, 2012). Formal analysis of this is needed.

There are other approaches to analyzing models with unknown  $N$ , using reversible jump MCMC (RJMCMC) or other so-called “trans-dimensional” (TD) algorithms (King and Brooks, 2001; Durban and Elston, 2005; King et al., 2008; Schofield and Barker, 2008; Wright et al., 2009). What distinguishes DA from RJMCMC and related TD methods is that DA is used to create a distinctly new model that is unconditional on  $N$  and we (usually) analyze the unconditional model. The various TD/RJMCMC approaches seek to analyze the conditional-on- $N$  model in which the dimension of the parameter space is a function of  $N$ , and will therefore typically vary at each iteration of the MCMC algorithm. TD/RJMCMC approaches might appear to have the advantage that one can model  $N$  explicitly or consider alternative priors for  $N$ . However, despite that  $N$  is removed as an explicit parameter in DA, it is possible to develop hierarchical models that involve structure on  $N$  (Converse and Royle, 2012; Royle et al., 2012c; Royle and Converse, in review) which we consider in Chapt. 14. Furthermore, data augmentation is often easier to implement than RJMCMC, and the details of the DA implementation are the same for all capture-recapture problems.

---

**4.2.5 Example: Black bear study on Fort Drum**

To illustrate the analysis of model  $M_0$  using data augmentation, we use a data set collected at Fort Drum Military Installation in upstate New York by P.D. Curtis and M.T Wegan of Cornell University and their colleagues at the Fort Drum Military Installation. These data have been analyzed in various forms by Wegan (2008); Gardner et al. (2009) and Gardner et al. (2010b). The specific data used here are encounter histories on 47 individuals obtained from an array of 38 baited “hair snares” (Fig. 4.1) during June and July 2006. Barbed wire traps were baited and checked for hair samples each week for eight weeks, thus we distinguished  $K = 8$  weekly sample intervals. The data are provided in the **R** package **scrbook**, can be loaded by typing `data(beardata)` at the **R** prompt, and the analysis can be set up and run as follows (see `?beardata` for the commands to do the analysis). Here, the data were augmented with 128 all-zero encounter histories, resulting in a total sample size of  $M = 175$ .



**Figure 4.1.** Fort Drum Black bear study area and the 38 baited hair snare locations operated for 8 weeks during June and July, 2006.

```

3884 > library(scrbook)
3885 > data(beardata)           # load the bear data and extract components
3886 > trapmat <- beardata$trapmat
3887 > nind <- dim(beardata$bearArray)[1]
3888 > K <- dim(beardata$bearArray)[3]
3889 > ntraps <- dim(beardata$bearArray)[2]
3890

```

```

3891 > M <- 175
3892 > nz <- M-nind
3893 > Yaug <- array(0, dim=c(M,ntraps,K))
3894
3895 > Yaug[1:nind,,] <- beardata$bearArray
3896 > y <- apply(Yaug,c(1,3),sum) # summarize by ind x rep
3897 > y[y>1] <- 1 # toss out multiple encounters per occasion
3898 # b/c traditional CR models ignore space

```

3899     The raw data object, `beardata$bearArray` is a 3-dimensional array  $nind \times ntraps \times K$   
 3900 of individual encounter events (i.e.,  $y_{ijk} = 1$  if individual  $i$  was encountered in trap  $j$  during  
 3901 occasion  $k$ , and 0 otherwise). For fitting model  $M_0$  (or  $M_h$ , see below), it is sufficient to  
 3902 reduce the data to individual encounter frequencies which we have re-labeled “y” above.  
 3903 The **BUGS** model file along with commands to fit the model are as follows:

```

3904 > set.seed(2013) # to obtain the same results each time
3905 > library(R2WinBUGS) # load R2WinBUGS, set-up:
3906 > data0 <- list(y=y, M=M, K=K) # data ....
3907 > params0 <- c('psi','p','N') # parameters ....
3908 > zst <- c(rep(1,nind),rbinom(M-nind, 1, .5)) # inits ....
3909 > inits <- function(){ list(z=zst, psi=runif(1), p=runif(1)) }
3910
3911 > cat("
3912 model{
3913
3914   psi ~ dunif(0, 1)
3915   p ~ dunif(0,1)
3916
3917   for (i in 1:M){
3918     z[i] ~ dbern(psi)
3919     for(k in 1:K){
3920       tmp[i,k] <- p*z[i]
3921       y[i,k] ~ dbin(tmp[i,k],1)
3922     }
3923   }
3924   N<-sum(z[1:M])
3925 }
3926 ",file="modelM0.txt")
3927
3928 ## Run the model:
3929 > fit0 <- bugs(data0, inits, params0, model.file="modelM0.txt",n.chains=3,
3930   n.iter=2000, n.burnin=1000, n.thin=1,debug=TRUE,working.directory=getwd())

```

3931     This produces the following posterior summary statistics:

```

3932 > print(fit0,digits=2)
3933 Inference for Bugs model at "modelM0.txt", fit using WinBUGS,

```

---

```

3934 3 chains, each with 2000 iterations (first 1000 discarded)
3935 n.sims = 3000 iterations saved
3936      mean    sd   2.5%   25%   50%   75% 97.5% Rhat n.eff
3937 psi     0.29  0.04  0.22  0.26  0.29  0.31  0.36    1 3000
3938 p      0.30  0.03  0.25  0.28  0.30  0.32  0.35    1 3000
3939 N      49.94 1.99 47.00 48.00 50.00 51.00 54.00    1 3000
3940 deviance 489.05 11.28 471.00 480.45 488.80 495.40 513.70    1 3000
3941
3942 [... some output deleted ...]

3943 WinBUGS did well in choosing an MCMC algorithm for this model – we have  $\hat{R} = 1$ 
3944 for each parameter, and an effective sample size of 3000, equal to the total number of
3945 posterior samples3. We see that the posterior mean of  $N$  under this model is 49.94 and
3946 a 95% posterior interval is (48, 54). We revisit these data later in the context of more
3947 complex models.
```

3948 In order to obtain an estimate of density,  $D$ , we need an area to associate with the  
 3949 estimate of  $N$ , and in Chapt. 1 we already went through a number of commonly used  
 3950 procedures to conjure up such an area, including buffering the trap array by the home range  
 3951 radius, often estimated by the mean maximum distance moved (MMDM) (Parmenter  
 3952 et al., 2003), 1/2 MMDM (Dice, 1938) or directly from telemetry data (Wallace et al.,  
 3953 2003). Typically, the trap array is defined by the convex hull around the trap locations,  
 3954 and this is what we applied a buffer to. We computed the buffer by using a telemetry-based  
 3955 estimate of the mean female home range radius (2.19 km) (Bales et al., 2005) instead of  
 3956 using an estimate based on our relatively more sparse recapture data. For the Fort Drum  
 3957 study, the convex hull has an area of 157.135 km<sup>2</sup>, and the buffered convex hull has an  
 3958 area of 277.011 km<sup>2</sup>. To create this we used functions contained in the **R** package **rgeos**  
 3959 and created a utility function **bcharea** which is in our **R** package **scrbook**. The commands  
 3960 are as follows:

```

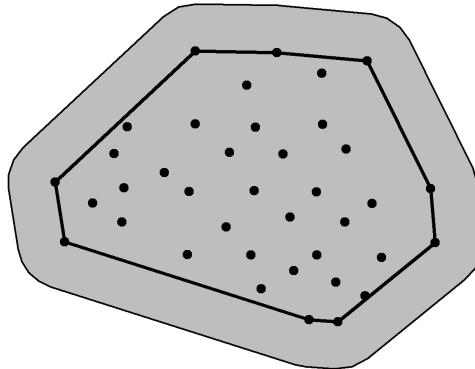
3961 > library(rgeos)
3962
3963 > bcharea <- function(buff,traplocs){
3964   p1 <- Polygon(rbind(traplocs,traplocs[1,]))
3965   p2 <- Polygons(list(p1=p1),ID=1)
3966   p3 <- SpatialPolygons(list(p2=p2))
3967   p1ch <- gConvexHull(p3)
3968   bp1 <- (gBuffer(p1ch, width=buff))
3969   plot(bp1, col='gray')
3970   plot(p1ch, border='black', lwd=2, add=TRUE)
3971   gArea(bp1)
3972 }
3973
3974 > bcharea(2.19,traplocs=trapmat)
```

3975 The resulting buffered convex hull is shown in Fig. 4.2.

3976 To conjure up a density estimate under model  $M_0$ , we compute the appropriate pos-  
 3977 terior summary of the ratio of  $N$  and the prescribed area (277.011 km<sup>2</sup>):

---

<sup>3</sup>This is even a little suspicious....



**Figure 4.2.** Convex hull of the bear hair snare array at Fort Drum, NY, buffered by mean female home range radius (2.19 km).

```

3978 > summary(fit0$sims.list$N/277.011)
3979   Min. 1st Qu. Median Mean 3rd Qu. Max.
3980 0.1697 0.1733 0.1805 0.1803 0.1841 0.2130
3981
3982 > quantile(fit0$sims.list$N/277.011,c(0.025,0.975))
3983   2.5%    97.5%
3984 0.1696684 0.1949381

```

3985 which yields a density estimate of about  $0.18 \text{ ind}/\text{km}^2$ , and a 95% Bayesian confidence  
 3986 interval of  $(0.170, 0.195)$ . Our estimate of density should be reliable if we have faith in  
 3987 our stated value of the “sampled area”. Clearly though this is largely subjective, and not  
 3988 something we can formally evaluate (or estimate) from the data based on model  $M_0$ .

### 4.3 TEMPORALLY VARYING AND BEHAVIORAL EFFECTS

3989 The purpose of this chapter is mainly to emphasize the central importance of the binomial  
 3990 model in capture-recapture and so we have considered models for individual encounter  
 3991 frequencies—the number of times individuals are captured out of  $K$  occasions. Sometimes  
 3992 we can’t aggregate the encounter data for each individual, such as when encounter proba-  
 3993 bility varies over time among samples. Time-varying responses that are relevant in many

3994 capture-recapture studies are “effort” such as amount of search time, number of observers,  
 3995 or trap nights, or encounter probability varying over time, as a function of date or season  
 3996 (Kéry et al., 2010) due to species behavior. A common situation in many animal studies  
 3997 is that in which there exists a “behavioral response” to trapping (even if the animal is not  
 3998 physically trapped).

3999 Behavioral response is an important concept in animal studies because individuals  
 4000 might learn to come to baited traps or avoid traps due to trauma related to being encoun-  
 4001 tered. There are a number of ways to parameterize a behavioral response to encounter.  
 4002 The distinction between persistent and ephemeral was made by Yang and Chao (2005)  
 4003 who considered a general behavioral response model of the form:

$$\text{logit}(p_{ik}) = \alpha_0 + \alpha_1 y_{i,k-1} + \alpha_2 x_{ik}$$

4004 where  $x_{ik}$  is a covariate indicator variable of previous capture (i.e.,  $x_{ik} = 1$  if captured  
 4005 in any previous period). Therefore, encounter probability changes depending on whether  
 4006 an individual was captured in the immediate previous period (a Markovian or ephemeral  
 4007 behavioral response; (Yang and Chao, 2005)), described by the term  $\alpha_1 y_{i,k-1}$  or in *any*  
 4008 previous period (persistent behavioral response), described by the term  $\alpha_2 x_{ik}$ . Because  
 4009 spatial capture-recapture models allow us to include trap-specific covariates, we can de-  
 4010 scribe a 3rd type of behavioral response—a local behavioral response that is trap-specific  
 4011 (Royle et al., 2011b). In this local behavioral response, the encounter probability is mod-  
 4012 ified for an individual trap depending on previous capture in that trap. Models with  
 4013 temporal effects are easy to describe and analyze in the **BUGS** language and we provide  
 4014 a number of examples in Chapt. 7 and elsewhere.

#### 4.4 MODELS WITH INDIVIDUAL HETEROGENEITY

4015 Models in which encounter probability varies by individual have a long history in capture-  
 4016 recapture and, indeed, this so-called “model  $M_h$ ” is one of the elemental capture-recapture  
 4017 models in (Otis et al., 1978). Conceptually, we imagine that the individual-specific em-  
 4018 counter probability parameters,  $p_i$ , are random variables distributed according to some  
 4019 probability distribution,  $[\theta]$ . We denote this basic model assumption as  $p_i \sim [\theta]$ . This  
 4020 type of model is similar in concept to extending a GLM to a GLMM but in the capture-  
 4021 recapture context  $N$  is unknown. The basic class of models is often referred to as “model  
 4022  $M_h$ ” (“h” for heterogeneity), but really this is a broad class of models, each being dis-  
 4023 tinguished by the specific distribution assumed for  $p_i$ . There are many different varieties  
 4024 of model  $M_h$  including parametric and various non-parametric approaches (Burnham and  
 4025 Overton, 1978; Norris and Pollock, 1996; Pledger, 2004). One important practical matter  
 4026 is that estimates of  $N$  can be extremely sensitive to the choice of heterogeneity model  
 4027 (Fienberg et al., 1999; Dorazio and Royle, 2003; Link, 2003). Indeed, Link (2003) showed  
 4028 that in some cases it’s possible to find models that yield precisely the same expected data,  
 4029 yet produce wildly different estimates of  $N$ . In that sense,  $N$  for most practical pur-  
 4030 poses is not identifiable across classes of different heterogeneity models, and this should  
 4031 be understood before fitting any such model. One solution to this problem is to seek  
 4032 to model explicit factors that contribute to heterogeneity, e.g., using individual covariate  
 4033 models (See 4.5 below). Indeed, spatial capture-recapture models do just that, by mod-  
 4034 eling heterogeneity due to the spatial organization of individuals in relation to traps or

4035 other encounter mechanism. For additional background and applications of model  $M_h$  see  
 4036 Royle and Dorazio (2008, Chapt. 6) and Kéry and Schaub (2012, Chapt. 6).

4037 We will work with a specific type of model  $M_h$  here which is a natural extension of  
 4038 the basic binomial observation model of model  $M_0$  so that

$$\text{logit}(p_i) = \mu + \eta_i$$

4039 where  $\mu$  is a fixed parameter (the mean) to be estimated, and  $\eta_i$  is an individual random  
 4040 effect assumed to be normally distributed:

$$\eta_i \sim \text{Normal}(0, \sigma_p^2)$$

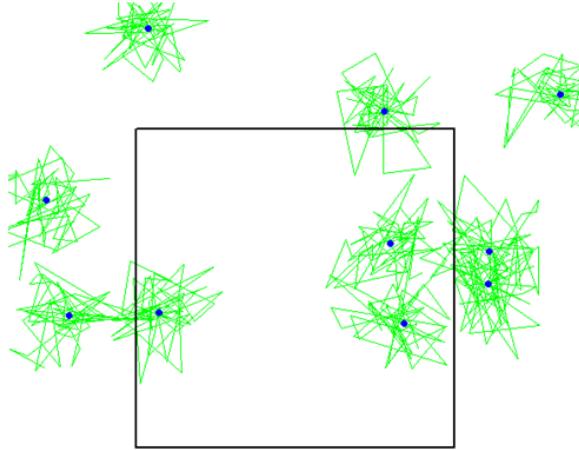
4041 We could as well combine these two steps and write  $\text{logit}(p_i) \sim \text{Normal}(\mu, \sigma_p^2)$ . This  
 4042 “logit-normal mixture” was analyzed by Coull and Agresti (1999) and elsewhere. It is  
 4043 a natural extension of the basic model with constant  $p$ , as a mixed GLMM, and similar  
 4044 models occur throughout statistics. It is also natural to consider a beta prior distribution  
 4045 for  $p_i$  (Dorazio and Royle, 2003) and so-called “finite-mixture” models are also popular  
 4046 (Norris and Pollock, 1996; Pledger, 2004). In the latter, individuals are assumed to belong  
 4047 to a finite number of latent classes, each of which has its own capture probability.

4048 Model  $M_h$  has important historical relevance to spatial capture-recapture situations  
 4049 (Karanth, 1995) because investigators recognized that the juxtaposition of individuals with  
 4050 the array of trap locations should yield heterogeneity in encounter probability, and thus it  
 4051 became common to use some version of model  $M_h$  in spatial trapping arrays to estimate  
 4052  $N$ . While this doesn’t resolve the problem of not knowing the effective sample area, it  
 4053 does yield an estimator that accommodates the heterogeneity in  $p$  induced by the spatial  
 4054 aspect of capture-recapture studies. To see how this juxtaposition induces heterogeneity,  
 4055 we have to understand the relevance of movement in capture-recapture models. Imagine a  
 4056 quadrat that can be uniformly searched by a crew of biologists for some species of reptile  
 4057 (see Royle and Young (2008)). Figure 4.3 shows a sample quadrat searched repeatedly  
 4058 over a period of time. Further, suppose that the species exhibits some sense of spatial  
 4059 fidelity in the form of a home range or territory, and individuals move about their home  
 4060 range (home range centroids are given by the solid dots) in some kind of random fashion.  
 4061 Heuristically, we imagine that each individual in the vicinity of the study area is liable  
 4062 to experience variable exposure to encounter due to the overlap of its home range with  
 4063 the sampled area - essentially the long-run proportion of times the individual is within  
 4064 the sample plot boundaries, say  $\phi$ . We might model the exposure or *availability* of an  
 4065 individual to capture by supposing that  $a_i = 1$  if individual  $i$  is available to be captured  
 4066 (i.e., within the survey plot) during any sample, and 0 otherwise. Then,  $\Pr(a_i = 1) = \phi$ .  
 4067 In the context of spatial studies, it is natural that  $\phi$  should depend on *where* an individual  
 4068 lives, i.e., it should be individual-specific  $\phi_i$  (Chandler et al., 2011). This system describes,  
 4069 precisely, that of “random temporary emigration” (Kendall et al., 1997) where  $\phi_i$  is the  
 4070 individual-specific probability of being “available” for capture.

4071 Conceptually, SCR models aim to deal with this problem of variable exposure to sam-  
 4072 pling due to movement in the proximity of the trapping array explicitly and formally with  
 4073 auxiliary spatial information. If individuals are detected with probability  $p_0$ , *conditional*  
 4074 on  $a_i = 1$ , then the marginal probability of detecting individual  $i$  is

$$p_i = p_0 \phi_i$$

so we see clearly that individual heterogeneity in encounter probability is induced as a result of the juxtaposition of individuals (i.e., their home ranges) with the sample apparatus and the movement of individuals about their home range.



**Figure 4.3.** A quadrat searched for lizards over some period of time (simulated data). The locations of encounter for each of 10 lizards are connected by lines—the dots are activity centers.

#### 4.4.1 Analysis of model $M_h$

If  $N$  is known, it is worth taking note of the essential simplicity of model  $M_h$  as a binomial GLMM. This is a type of model that is widely applied throughout statistics using standard methods of inference based either on integrated likelihood (Laird and Ware, 1982; Berger et al., 1999), which we discuss in Chapt. 6, or standard Bayesian methods. However, because  $N$  is not known, inference is somewhat more challenging. We address that here using Bayesian analysis based on data augmentation. Although we use data augmentation in the context of Bayesian methods here, we note that heterogeneity models formulated under DA are easily analyzed by conventional likelihood methods as zero-inflated binomial mixtures (Royle, 2006) and more traditional analysis of model  $M_h$  based on integrated likelihood, without using data augmentation, has been considered by Coull and Agresti (1999), Dorazio and Royle (2003), and others.

As with model  $M_0$ , we have the Bernoulli model for the zero-inflation variables:  $z_i \sim \text{Bernoulli}(\psi)$  and the model of the observations expressed conditional on these latent

4092 variables  $z_i$ . For  $z_i = 1$ , we have a binomial model with individual-specific  $p_i$ :

$$y_i | z_i = 1 \sim \text{Binomial}(K, p_i)$$

4093 and otherwise  $y_i | z_i = 0 \sim I(y = 0)$ , i.e., a point mass at  $y = 0$ . Further, we prescribe a  
4094 distribution for  $p_i$ . Here we assume

$$\text{logit}(p_i) \sim \text{Normal}(\mu, \sigma^2)$$

4095 For prior distributions we assume  $p_0 = \text{logit}^{-1}(\mu) \sim \text{Uniform}(0, 1)$  and, for the standard  
4096 deviation  $\sigma \sim \text{Uniform}(0, B)$  for some large  $B$ . Another common default prior is to assume  
4097  $\tau = 1/\sigma^2 \sim \text{Gamma}(1, 1)$ , although we usually choose  $\sigma \sim \text{Uniform}(0, B)$ .

#### 4098 4.4.2 Analysis of the Fort Drum data with model $M_h$

4099 Here we provide an analysis of the Fort Drum bear survey data using the logit-normal  
4100 heterogeneity model, and we used data augmentation to produce a data set of  $M = 700$   
4101 individuals. We have so far mostly used **WinBUGS** but we are now transitioning to  
4102 the use of **JAGS** run from within **R** using the useful packages **R2jags** or **rjags**. The  
4103 function **jags** from the **R2jags** package runs essentially like the **bugs** function which we  
4104 demonstrate here for setting up and running model  $M_h$  for the Fort Drum bear data:

```
4105 [...] get data as before ....]
4106
4107 > set.seed(2013)
4108
4109 > cat("
4110 model{
4111   p0 ~ dunif(0,1)           # prior distributions
4112   mup <- log(p0/(1-p0))
4113   sigmap ~ dunif(0,10)
4114   taup <- 1/(sigmap*sigmap)
4115   psi ~ dunif(0,1)
4116
4117   for(i in 1:(nind+nz)){
4118     z[i] ~ dbern(psi)        # zero inflation variables
4119     lp[i] ~ dnorm(mup,taup) # individual effect
4120     logit(p[i]) <- lp[i]
4121     mu[i] <- z[i]*p[i]
4122     y[i] ~ dbin(mu[i],K)    # observation model
4123   }
4124
4125   N<-sum(z[1:(nind+nz)])
4126 }
4127 ",file="modelMh.txt")
4128 > data1 <- list(y=y, nz=nz, nind=nind, K=K)
4129 > params1 <- c('p0','sigmap','psi','N')
```

---

```

4130 > inits <- function(){ list(z=as.numeric(y>=1), psi=.6, p0=runif(1),
4131   sigmap=runif(1,.7,1.2),lp=rnorm(M,-2)) }
4132 > library(R2jags)
4133 > wbout <- jags(data1, inits, params1, model.file = "modelMh.txt", n.chains = 3,
4134   n.iter = 1010000, n.burnin = 10000, working.directory = getwd())

```

4135 We provide an **R** function `modelMhBUGS` in the package `scrbook` which will fit the  
4136 model using either **JAGS** or **WinBUGS** as specified by the user. In addition, for fun,  
4137 we construct our own MCMC algorithm using a Metropolis-within-Gibbs algorithm for  
4138 model  $M_h$  in Chapt. 17, where we also develop MCMC algorithms for spatial capture-  
4139 recapture models. Using `modelMhBUGS`, we ran 3 chains of 1 *million* iterations (mixing is  
4140 poor for this model and this data set), which produced the posterior distribution for  $N$   
4141 shown in Fig. 4.4. Posterior summaries of parameters are given in Table 4.5.

**Table 4.5.** Posterior summaries from model  $M_h$  fitted to the Fort Drum black bear data. Results were obtained using **WinBUGS** running 3 chains, each with 1010000 iterations, discarding the first 10000 for a total of three *million* posterior samples.

Parameter	Mean	SD	2.5%	50 %	97.5%	Rhat	n.eff
$p_0$	0.072	0.056	0.002	0.060	0.203	1.008	540
$\sigma_p$	2.096	0.557	1.215	2.025	3.373	1.003	820
$\psi$	0.176	0.101	0.084	0.147	0.458	1.006	650
$N$	122.695	69.897	62.000	102.000	319.000	1.006	630

4142 We used  $M = 700$  for this analysis and we note that while the posterior mass of  $N$  is  
4143 concentrated away from this upper bound (Fig. 4.4), the posterior has an extremely long  
4144 right tail, with some MCMC draws at the upper boundary  $N = 700$ , suggesting that an  
4145 even higher value of  $M$  may be called for. To characterize the posterior distribution of  
4146 density we produce the relevant summaries of the posterior distribution of  $D = N/277.11$   
4147 (recall the buffered area of the convex hull is 277.11  $km^2$ ):

```

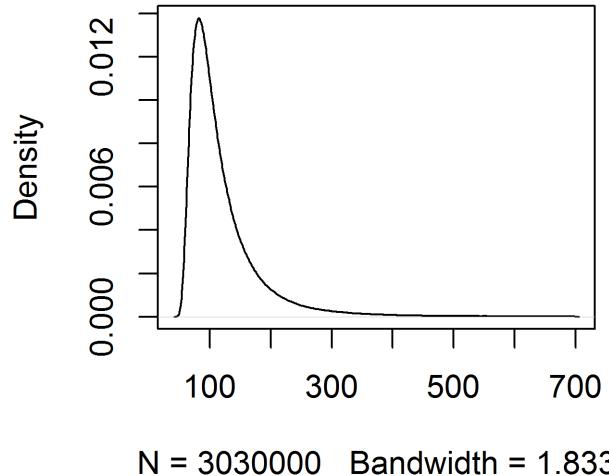
4148 > summary(wbout$sims.list$N/277.11)
4149   Min. 1st Qu. Median Mean 3rd Qu. Max.
4150 0.1696 0.2959 0.3681 0.4428 0.4944 2.5260
4151
4152 > quantile(wbout$sims.list$N/277.11,c(0.025,0.50,0.975))
4153   2.5%      50%    97.5%
4154 0.2237379 0.3680849 1.1511674

```

4155 Therefore, the point estimate, characterized by the posterior median, is around 0.37 bears  
4156 per square km and a 95% Bayesian credible interval is (0.224, 1.151).

#### 4.4.3 Comparison with MLE

4158 The posterior of  $N$  is highly skewed; therefore, we see that the posterior mean ( $N = 122.7$ )  
4159 is considerably higher than the posterior median ( $N = 102$ ). Further, it may be surprising  
4160 that these posterior summaries do not compare well with the MLE. We used the **R** code



**Figure 4.4.** Posterior of  $N$  for Fort Drum bear study data under the logit-normal version of model  $M_h$ .

4161 contained in Panel 6.1 from Royle and Dorazio (2008) to obtain the MLE of  $\log(n_0)$ ,  
 4162 the logarithm of the number of uncaptured individuals, is  $\widehat{\log(n_0)} = 3.86$  and therefore  
 4163  $\hat{N} = \exp(3.86) + 47 = 94.47$ , which is larger than the mode shown in Fig. 4.4. To see  
 4164 this, we compute the posterior mode, by finding the posterior value of  $N$  with the highest  
 4165 mass. Because  $N$  is discrete, we can use the `table()` function in **R** and find the most  
 4166 frequent value<sup>4</sup>. If we want to smooth out some of the Monte Carlo error a bit, we can  
 4167 use a smoother of some sort applied to the tabled posterior frequencies of  $N$ . Here we use  
 4168 a smoothing spline (**R** function `smooth.spline`) with the degree of smoothing chosen by  
 4169 cross-validation (the `cv=TRUE` argument):

```
4170 > N <- table(jout$BUGSoutput$sims.list$N)
4171 > xg <- as.numeric(names(N))
4172
4173 > sp <- smooth.spline(xg,N,cv=TRUE)
4174
4175 > sp
```

<sup>4</sup>For a continuous random variable we can use the function `density()` to smooth the posterior samples and obtain the mode.

```

4176
4177 Call:
4178 smooth.spline(x = xg, y = N, cv = TRUE)
4179
4180 Smoothing Parameter spar= 0.09339815 lambda= 8.201724e-09 (17 iterations)
4181 Equivalent Degrees of Freedom (Df): 121.1825
4182 Penalized Criterion: 2544481
4183 PRESS: 5903.4

```

4184 We obtain the mode of the smoothed frequencies as follows:

```

4185 sp$x[sp$y==max(sp$y)]
4186 [1] 82

```

4187 We don't dwell too much on the difference between the MLE and features of the posterior, but we do note here that the posterior distribution for the parameters of this model, for the Fort Drum data set, are very sensitive to the prior distributions. In the present case, the use of a Uniform(0, 1) prior for  $p_0 = \text{logit}^{-1}(\mu)$  is somewhat informative—in particular, it is not at all “flat” on the scale of  $\mu$ , and this affects the posterior. We generally always recommend use of a Uniform(0, 1) prior for  $\text{logit}^{-1}(\mu)$  in such models. That said, we were surprised at this result, and we experimented with other prior configurations including putting a flat prior on  $\mu$  directly. This kind of small sample instability has been widely noted in model  $M_h$  (Fienberg et al., 1999; Dorazio and Royle, 2003), as has extreme sensitivity to the specific form of model  $M_h$  (Link, 2003). In summary, while the mode is well-defined, the data set is relatively sparse and hence inferences are poor and sensitive to model choice.

## 4.5 INDIVIDUAL COVARIATE MODELS: TOWARD SPATIAL CAPTURE-RECAPTURE

4199 A standard situation in capture-recapture models is when a covariate which is thought  
 4200 to influence encounter probability is measured for each individual. These are often called  
 4201 “individual covariate models” but, in keeping with the classical nomenclature on closed  
 4202 population models, Kéry and Schaub (2012) referred to this class of models as “model  
 4203  $M_x$ ” (the  $x$  here being an explicit covariate). As with other closed population models, we  
 4204 begin with the basic binomial observation model:

$$y_i \sim \text{Binomial}(K, p_i).$$

4205 To model the covariate, we use a logit model for encounter probability of the form:

$$\text{logit}(p_i) = \alpha_0 + \alpha_1 x_i \tag{4.5.1}$$

4206 where  $x_i$  is the covariate value for individual  $i$  and the parameters  $\boldsymbol{\alpha} = (\alpha_0, \alpha_1)$  are the  
 4207 regression coefficients. Classical examples of covariates influencing detection probability  
 4208 are type of animal (juvenile/adult or male/female), a continuous covariate such as body  
 4209 mass, or a discrete covariate such as group or cluster size. For example, in models of aerial  
 4210 survey data, it is natural to model the detection probability of a group as a function of the  
 4211 observation-level individual covariate, “group size” (Royle, 2008; Langtimm et al., 2011).

Model  $M_x$  is similar in structure to model  $M_h$ , except that the individual effects are observed for the  $n$  individuals that appear in the sample. These models are important here because spatial capture-recapture models can be described precisely as a form of model  $M_x$ , where the covariate describes where the individual is located in relation to the trapping array. Specifically, SCR models are individual covariate models, but where the individual covariate is only observed imperfectly (or partially observed) for each captured individual. Unlike model  $M_h$ , in SCR models (and model  $M_x$ ) we do have some direct information about the latent variable, which comes from the spatial locations/distribution of individual recaptures.

Traditionally, estimation of  $N$  in model  $M_x$  is achieved using methods based on ideas of unequal probability sampling (i.e., Horvitz-Thompson estimation<sup>5</sup>; Huggins (1989), Alho (1990) and Borchers et al. (2002)). An estimator of  $N$  is

$$\hat{N} = \sum_{i=1}^n \frac{1}{\tilde{p}_i}$$

where  $\tilde{p}_i$  is the probability that individual  $i$  appeared in the sample. This quantity is  $\tilde{p}_i = \Pr(y_i > 0)$  and, in closed population capture-recapture models, it can be computed as:

$$\Pr(y_i > 0) = 1 - (1 - p_i)^K$$

where  $p_i$  is a function of parameters  $\alpha_0$  and  $\alpha_1$  according to Eq. 4.5.1. In practice, parameters are estimated from the conditional-likelihood of the observed encounter histories which is, for observation  $y_i$ ,

$$\mathcal{L}_c(\boldsymbol{\alpha}|y_i) = \frac{\text{Binomial}(y_i|\boldsymbol{\alpha})}{\tilde{p}_i}. \quad (4.5.2)$$

This derives from a straightforward application of the law of total probability. Conceptually, we partition  $\Pr(y)$  according to  $\Pr(y) = \Pr(y|y > 0)\Pr(y > 0) + \Pr(y|y = 0)\Pr(y = 0)$ . For any positive value of  $y$  the 2nd term is necessarily 0, and so we rearrange to obtain  $\Pr(y|y > 0) = \Pr(y)/\Pr(y > 0)$  which, in the specific case where  $\Pr(y)$  is the binomial probability mass function (pmf) produces Eq. 4.5.2.

Here we take a formal model-based approach to Bayesian analysis of such models based on the joint likelihood using data augmentation (Royle, 2009b). Classical likelihood analysis of the so-called “full likelihood” is covered by Borchers et al. (2002). For Bayesian analysis of model  $M_x$ , because the individual covariate is unobserved for the  $n_0 = N - n$  uncaptured individuals, we require a model to describe variation in  $x$  among individuals, essentially allowing the sample to be extrapolated to the population. For example, if we have a continuous trait measured on each individual, then we might assume that  $x$  has a normal distribution:

$$x_i \sim \text{Normal}(\mu, \sigma^2)$$

Data augmentation can be applied directly to this class of models. In particular, reformulation of the model under DA yields a basic zero-inflated binomial model of the following

<sup>5</sup>For a quick summary of the idea see:  
[http://en.wikipedia.org/wiki/Horvitz-Thompson\\_estimator](http://en.wikipedia.org/wiki/Horvitz-Thompson_estimator)

4245 form, for each  $i = 1, 2, \dots, M$ :

$$\begin{aligned} z_i &\sim \text{Bernoulli}(\psi) \\ y_i | z_i = 1 &\sim \text{Binomial}(K, p_i(x_i)) \\ y_i | z_i = 0 &\sim I(y = 0) \\ x_i &\sim \text{Normal}(\mu, \sigma^2) \end{aligned}$$

4246 Fully spatial capture-recapture models use this formulation with a latent covariate that  
 4247 is directly related to the individual detection probability (see next section). As with  
 4248 the previous models, implementation is trivial in the **BUGS** language. The **BUGS**  
 4249 specification is very similar to that for model  $M_h$ , but we require the distribution of the  
 4250 covariate to be specified, along with priors for the parameters of that distribution.

4251 **4.5.1 Example: Location of capture as a covariate**

4252 Here we consider a special type of model  $M_x$  that is especially relevant to spatial capture-  
 4253 recapture. Intuitively, some measure of distance from home range center to traps for an  
 4254 individual should be a reasonable covariate to explain heterogeneity in encounter probabil-  
 4255 ity, i.e., individuals with more exposure to traps should have higher encounter probabilities  
 4256 and vice versa. So we can imagine *estimating* such a quantity, say average distance from  
 4257 home range center to “the trap array”, and then using it as an individual covariate in  
 4258 capture-recapture models. A version of this idea was put forth by Boulanger and McLel-  
 4259 lan (2001) (see also Ivan (2012)), but using the Huggins-Alho estimator and with covariate  
 4260 “distance from home range center to edge” of the trapping array, where the home range  
 4261 center is estimated by the average capture location. This is intuitively appealing because  
 4262 we can imagine, in some kind of an ideal situation where we have a dense grid of traps  
 4263 over some geographic region, that the average location of capture would be a decent esti-  
 4264 mate (heuristically) of an individual’s home range center. We provide an example of this  
 4265 type of approach using a fully model-based analysis of the version of model  $M_x$  described  
 4266 above, analyzed by data augmentation. We take a slightly different approach than that  
 4267 adopted by Boulanger and McLellan (2001). By analyzing the full likelihood and placing  
 4268 a prior distribution on the individual covariate, we will resolve the problem of having an  
 4269 ill-defined sample area. After you read later chapters of this book, it will be apparent that  
 4270 SCR models represent a formalization of this heuristic procedure.

4271 For our purposes here, we define the scalar individual covariate  $x_i$  to be the distance  
 4272 from the average encounter location of individual  $i$ , say  $\mathbf{s}_i$ , to the centroid of the trap  
 4273 array,  $\mathbf{x}_0$ :  $x_i = \|\mathbf{s}_i - \mathbf{x}_0\|$ . Note that  $\|\mathbf{u}\|$  is standard notation for Euclidean norm or  
 4274 magnitude of the vector  $\mathbf{u}$ , and we use it throughout the book. In practice, people have  
 4275 used distance from edge of the trap array but that is less easy to quantify, as “edge” itself  
 4276 is not precisely defined. Conceptually, individuals in the middle of the array should have  
 4277 a higher probability of encounter and, as  $x_i$  increases,  $p_i$  should therefore decrease. We  
 4278 note that we have defined  $\mathbf{s}_i$  in terms of a sample quantity—the observed mean encounter  
 4279 location—which, while ad hoc, is consistent with the use of individual covariate models in  
 4280 the literature. For an expansive, dense trapping grid we might expect the sample mean  
 4281 encounter location to be a good estimate of home range center but, clearly this is biased  
 4282 for individuals that live around the edge (or off) the trapping array.

4283 A key point is that  $s_i$  is missing for each individual that is not encountered and so  
 4284  $x_i$  is also missing. Therefore, it is a latent variable, and we need to specify a probability  
 4285 distribution for it. As a measurement of distance we know it must be positive-valued, and  
 4286 it seems sensible that an individual located extremely far from the array of traps would  
 4287 not be captured. Therefore, let's assume that  $x_i$  is uniformly distributed from 0 to some  
 4288 large number, say  $B$ , beyond which it would be difficult to imagine an individual being  
 4289 captured by the trap array:

$$x_i \sim \text{Uniform}(0, B)$$

4290 where  $B$  is a specified constant, which we may choose to be arbitrarily large. For example,  
 4291  $B$  should be at least a home range diameter past the furthest trap from the centroid of  
 4292 the array.

#### 4293 4.5.2 Fort Drum bear study

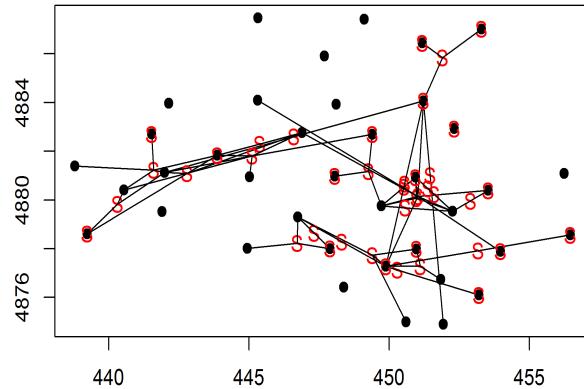
4294 We have to do a little bit of data processing to fit this individual covariate model to the  
 4295 Fort Drum data. We need to compute the individual covariate  $\mathbf{x}_i$  (distance from the  
 4296 centroid of the trapping array) using the **R** function `spiderplot` provided in `scrbook`.  
 4297 This function also produces the keen plot shown in Fig. 4.5 which we call a “spider plot”.  
 4298 The **R** commands for obtaining the individual covariate “distance from trap centroid”  
 4299 (the variable `xcent` returned by `spiderplot`) and making the spider plot are as follows:

```
4300 > library(scrbook)
4301 > data(beardata)
4302 > toad <- spiderplot(beardata$bearArray,beardata$trapmat)
4303 > xcent <- toad$xcent
```

4304 For the analysis of these data using the individual covariate “distance from centroid”  
 4305 we used  $x_i \sim \text{Uniform}(0, B)$  with  $B = 11.5 \text{ km}^2$ , which is about the distance from the  
 4306 array center to the furthest trap. Once we choose a value for  $B$ , the direct implication is  
 4307 that the population size parameter,  $N$ , applies to the area within 11.5 units of the trap  
 4308 centroid. Therefore, the model associates a precise area within which the population of  $N$   
 4309 individuals resides. We will see shortly that  $N$  does, in fact, scale with our choice of  $B$  to  
 4310 reflect the changing area over which the  $N$  individuals of the model reside. The **BUGS**  
 4311 model specification and **R** commands to package the data and fit the model are as follows:

```
4312 cat("
4313 model{
4314   p0 ~ dunif(0,1)                                # Prior distributions
4315   alpha0 <- log(p0/(1-p0))
4316   psi ~ dunif(0,1)
4317   beta ~ dnorm(0,.01)

4318 for(i in 1:(nind+nz)){
4319   xcent[i] ~ dunif(0,B)
4320   z[i] ~ dbern(psi)                               # DA variables
4321   lp[i] <- alpha0 + beta*xcent[i] # Individual effect
4322   logit(p[i]) <- lp[i]
```



**Figure 4.5.** Spider plot of the Fort Drum study data. The black dots represent the 47 trap locations with the "S" symbols being the average capture location of each bear. i.e., its estimated home range center. All traps in which a bear was captured are connected to its estimated home range center with a line.

```

4324     mu[i] <- z[i]*p[i]
4325     y[i] ~ dbin(mu[i],K)           # Observation model
4326 }
4327
4328 N <- sum(z[1:(nind+nz)])
4329 }
4330 ",file="modelMcov.txt")

4331 data2 <- list(y=y,nz=nz, nind=nind, K=K, xcent=xcent,B=11.5)
4332 params2 <- c('p0','psi','N','beta')
4333 inits <- function() {list(z=zst, psi=psi, p0=runif(1), beta=rnorm(1) ) }
4334 fit2 <- bugs(data2, inits, params2, model.file="modelMcov.txt",
4335                 n.chains=3, n.iter=11000, n.burnin=1000, n.thin=1)

```

4336 This produces the posterior summary statistics in Table 4.6.  
 4337 We note that the estimated  $N$  is much lower than obtained by model  $M_h$  but there  
 4338 is a good explanation for this which we discuss in the next section. That issue notwithstanding,  
 4339 it is worth pondering how this model could be an improvement (conceptually  
 4340 or technically) over some other model/estimator including  $M_0$  and  $M_h$  considered previously.  
 4341 Well, for one, we have accounted formally for heterogeneity due to spatial location

**Table 4.6.** Posterior summaries from the individual covariate model (model  $M_x$ ) with covariate “distance from the centroid of the trap array”, fitted to the Fort Drum black bear data. Results were obtained using WinBUGS running 3 chains, each with 11000 iterations, discarding the first 1000 for a total of 30000 posterior samples.

Parameter	Mean	SD	2.5%	50 %	97.5%	Rhat	n.eff
$p_0$	0.54	0.07	0.40	0.54	0.67	1	1100
$\psi$	0.34	0.05	0.25	0.34	0.44	1	3500
$N$	58.92	5.49	50.00	58.00	71.00	1	1900
$\beta$	-0.25	0.06	-0.36	-0.25	-0.12	1	780

of individuals relative to exposure to the trap array, characterized by the centroid of the array. Moreover, we have done so using a model that is based on an explicit mechanism, as opposed to a phenomenological one such as model  $M_h$ . In addition, and importantly, using our new model, *the estimated  $N$  applies to an explicit area which is defined by our prescribed value of  $B$* . That is, this area is a fixed component of the model and the parameter  $N$  therefore has explicit spatial context, as the number of individuals with home range centers less than  $B$  from the centroid of the trap array. As such, the implied “effective area” of the trap array for a given  $B$  is a precisely defined quantity—it is that of a circle with radius  $B$ .

### 4.5.3 Extension of the model

The model developed in the previous section is not a very good model for one important reason: Imposing a uniform prior distribution on  $x$  implies that density is *not constant* over space. In particular, this model implies that density *decreases* as we move away from the centroid of the trap array. That is,  $x_i \sim \text{Uniform}(0, B)$  implies constant  $N$  in each distance band from the centroid but obviously the *area* of each distance band is increasing. This is one reason we have a lower estimate of density than that obtained previously from model  $M_h$  (Sec. 4.4.2) and also why, if we were to increase  $B$ , we would see density continue to decrease.

Fortunately, we are not restricted to use of this specific distribution for the individual covariate. Clearly, it is a bad choice and, therefore, we should think about whether we can choose a better distribution for  $B$ —one that doesn’t imply a decreasing density as distance from the centroid increases. Conceptually, what we want to do is impose a prior on distance from the centroid,  $x$ , such that abundance should be proportional to the amount of area in each successive distance band as you move farther away from the centroid, so that density is *constant*. In fact, theory exists which tells us we should choose  $[x] = 2x/B^2$ . This can be derived by noting that  $F(x) = \Pr(X < x) = (\pi x^2)/(\pi * B^2)$ . Then,  $f(x) = dF/dx = 2 * x/(B^2)$ . This is a sort of triangular distribution in density induced because the incremental area in each additional distance band increases linearly with radius (i.e., distance from centroid). This can be verified empirically as follows:

```
> u <- runif(10000,-1,1)
4372 > v <- runif(10000,-1,1)
4373 > d <- sqrt(u*u+v*v)
```

```

4374 > hist(d[d<1])
4375 > hist(d[d<1],100)
4376 > hist(d[d<1],100,probability=TRUE)
4377 > abline(0,2)

```

4378 It would be useful if we could describe this distribution directly in **BUGS** but there  
 4379 is not a built-in way to do so. However, we can implement a discrete version of the pdf<sup>6</sup>.  
 4380 To do this, we break  $B$  into  $L$  distance classes of width  $\delta$ , with probabilities proportional  
 4381 to  $2 * x$ . In particular, if we denote the cut-points by  $g_1 = 0, g_2, \dots, g_{L+1} = B$  and the  
 4382 interval midpoints are  $m_i = g_{i+1} - \delta$ . Then the interval probabilities are, approximately<sup>7</sup>,  
 4383  $p_i = \delta(2m_i/B^2)$ , which we can compute once and then pass them to **BUGS** as data. The  
 4384 **R** commands for doing all of this (noting that we have already loaded and processed the  
 4385 Fort Drum bear data) are given in the following **R/BUGS** script:

```

4386 > delta <- .2
4387 > xbin <- xcent%/%delta + 1                      # Put x in bins
4388 > midpts <- seq(delta,Dmax,delta)
4389 > xprobs <- delta*(2*midpts/(B*B))
4390 > xprobs <- xprobs/sum(xprobs)
4391
4392 > cat("
4393 model{
4394 p0 ~ dunif(0,1)                                # Prior distributions
4395 alpha0 <- log(p0/(1-p0))
4396 psi ~ dunif(0,1)
4397 beta ~ dnorm(0,.01)
4398
4399 for(i in 1:(nind+nz)){
500   xbin[i] ~ dcat(xprobs[])
501   z[i] ~ dbern(psi)                               # DA variables
502   lp[i] <- alpha0 + beta*xbin[i]*delta          # Individual covariate model
503   logit(p[i]) <- lp[i]
504   mu[i] <- z[i]*p[i]
505   y[i] ~ dbin(mu[i],K)                          # Observation model
506 }
507
508 N <- sum(z[1:(nind+nz)])                      # N is derived
509 }
510 ",file="modelMcov.txt")

```

4411 In the model description, the variable  $x$  (observed distance from centroid of the trap  
 4412 array) has been rounded or binned (placed into a distance bin) so that the discrete version  
 4413 of the pdf of  $x$  can be used, as described previously. The new variable labeled **xbin** is  
 4414 then the *integer category* in units of  $\delta$  from 0. Thus, to convert back to distance in the

<sup>6</sup>We might also be able to use what is referred to in **WinBUGS** jargon as the “zeros trick” (see *Advanced BUGS tricks* in the manual) although we haven’t pursued this approach.

<sup>7</sup>This is just length  $\times$  width, the area of small rectangles approximating the integral.

4415 expression for `lp[i]`, `xbin[i]` has to be multiplied by  $\delta$ . To fit the model, keeping in  
 4416 mind that the data objects required below have been defined in previous analyses of this  
 4417 chapter, we do this:

```
4418 > data2 <- list(y=y, nz=nz, nind=nind, K=K, xbin=xbin, xprobs=xprobs,  

4419   delta=delta)  

4420 > params2 <- c('p0','psi','N','beta')  

4421 > inits <- function() {list(z=z, psi=psi, p0=runif(1),beta=rnorm(1) ) }  

4422 > fit <- bugs(data2, inits, params2, model.file="modelMcov.txt",  

4423   working.directory=getwd(), debug=FALSE, n.chains=3,  

4424   n.iter=11000, n.burnin=1000, n.thin=2)
```

4425 By specification of  $B$ , this model induces a clear definition of area in which the popu-  
 4426 lation of  $N$  individuals reside. The parameter  $N$  of the model is the population size that  
 4427 applies to the particular value of  $B$  and, as such, we will see that  $N$  scales with our choice  
 4428 of  $B$ . This might be disconcerting to some—we can get whatever value of  $N$  we want  
 4429 by changing  $B$ ! However, it is intuitively reasonable that, as we increase the area under  
 4430 consideration, there should be more individuals in it. Fortunately, we find empirically,  
 4431 that while  $N$  is highly sensitive to the prescribed value of  $B$ , density appears invariant to  
 4432  $B$  as long as  $B$  is sufficiently large. We fit the model for a set of values of  $B$  from  $B = 12$   
 4433 (restricting values of  $x$  to be in close proximity to the trap array) on up to 20. The results  
 4434 are given in Table 4.7.

**Table 4.7.** Analysis of Fort Drum bear hair snare data using the individual covariate model, for different values of  $B$ , the upper limit of the uniform distribution of ‘distance from centroid of the trap array’. “Density” is the posterior mean of density.

$B$	Density (post. mean)	Posterior SD
12	0.230	0.038
15	0.244	0.041
17	0.249	0.044
18	0.249	0.043
19	0.250	0.043
20	0.250	0.044

4435 We see that the posterior mean and SD of density (individuals per square km) appear  
 4436 insensitive to choice of  $B$  once we reach about  $B = 17$  or so. The estimated density of  
 4437 0.25 per km<sup>2</sup> is actually quite a bit lower than we reported using model  $M_h$  for which no  
 4438 relevant “area” quantity is explicit in the model (and so we had to make it up). Using  
 4439 MLEs of  $N$  in conjunction with buffer strips (see Tab. 1.1) our estimates were in the  
 4440 range of 0.32 – 0.43 and see Sec. 4.4 above. On the other hand our estimate of  $\hat{D} = 0.25$   
 4441 here (based on the posterior mean) is higher than that reported from model  $M_0$  using  
 4442 the buffered area ( $\hat{D} = 0.18$ ). There is no basis really for comparing or contrasting  
 4443 these various estimates. In particular, application of models  $M_0$  and  $M_h$  are distinctly  
 4444 *not* spatially explicit models—the area within which the population resides is not defined  
 4445 under either model. There is therefore no reason at all to think that the estimates produced  
 4446 under either closed population model, based on a buffered “trap area”, are justifiable by  
 4447 any theory. In fact, we would get exactly the same estimate of  $N$  no matter what we declare

4448 the area to be. On the other hand, the individual covariate model uses an explicit model  
 4449 for “distance from centroid” that is a reasonable and standard null model—it posits, in the  
 4450 absence of direct information, that individual home range centers are randomly distributed  
 4451 in space and that probability of detection depends on the distance between home range  
 4452 center and the centroid of the trap array. Under this definition of the system, we see that  
 4453 density is invariant to the choice of area, which seems like a desirable feature.

4454 **4.5.4 Invariance of density to  $B$**

4455 Under model  $M_x$ , and also under models that we consider in later chapters, a general  
 4456 property of the estimators is that while  $N$  increases with the prescribed area of the model  
 4457 (defined by  $B$  in this model), we expect that density estimators should be invariant to this  
 4458 area. In the model used above, we note that  $\text{Area}(B) = \pi B^2$  and  $\mathbb{E}(N(B)) = \lambda \text{Area}(B)$   
 4459 and thus  $\mathbb{E}(\text{Density}(B)) = \lambda$ , i.e., constant. This should be interpreted as the *prior*  
 4460 density. Absent data, then realizations under the model will have density  $\lambda$  regardless  
 4461 of what  $B$  is prescribed to be. As we verified empirically above, posterior summaries of  
 4462 density are also invariant to  $B$  as long as the prescribed area is sufficiently large.

4463 **4.5.5 Toward fully spatial capture-recapture models**

4464 While the use of an individual covariate model resolves two important problems inherent  
 4465 in almost all capture-recapture studies (induced heterogeneity and absence of a precise  
 4466 relationship between  $N$  and area), is not ideal for all purposes because it does not make  
 4467 full use of the spatial information in the data set, i.e., the trap locations and the locations  
 4468 of each individual encounter, so that we cannot use this model to model trap-specific  
 4469 effects (e.g., trap effort or type). Moreover, we applied this model for “data” being the  
 4470 average observed encounter location, and equated that summary to the home range center  
 4471  $s_i$ . Intuitively, taking the average encounter location as an estimate of home range center  
 4472 makes sense but more so when the trapping grid is dense and expansive relative to typical  
 4473 home range sizes which might not be reasonable in practice. Moreover, this approach  
 4474 also ignored the variable precision with which each  $s_i$  is estimated. Finally, it ignores  
 4475 that estimates of  $s_i$  around the “edge” (however we define that) are biased because the  
 4476 observations are truncated—we can only observe locations interior to the array.

4477 However, there is hope to extend this model in order to resolve these remaining defi-  
 4478 ciencies. In the next chapter we provide a further extension of this individual covariate  
 4479 model that definitively resolves the *ad hoc* nature of the approach we took here. In that  
 4480 chapter we build a model in which  $s_i$  are regarded as latent variables and the observation  
 4481 locations (i.e., trap specific encounters) are linked to those latent variables with an explicit  
 4482 model. We note that the model fitted previously could be adapted easily to deal with  $s_i$   
 4483 as a latent variable, simply by adding a prior distribution for  $s_i$ . This is actually easier,  
 4484 and less ad hoc in a number of respects, and you should try it out.

## 4.6 DISTANCE SAMPLING: A PRIMITIVE SCR MODEL

4485 Distance sampling is a class of methods for estimating animal density from measurements  
 4486 of distance from an observer to individual animals (or groups). The basic assumption

is that detection probability is a function of distance. Distance sampling is one of the most popular methods for estimating animal abundance (Burnham et al., 1980; Buckland et al., 2001; Buckland, 2004) because, unlike ordinary closed population models, distance sampling provides explicit estimates of *density*. In terms of methodological context, the distance sampling model is a special case of a closed population model with an individual covariate. The covariate in this case,  $x$ , is the distance between an individual's location say  $\mathbf{u}$  and the observation location or transect. In fact, distance sampling is precisely an individual-covariate model, except that observations are made at only  $K = 1$  sampling occasion. Distance sampling eliminates the need to explicitly identify individuals (except they need to be *distinguished* from other individuals) repeatedly and so distance sampling can be applied to unmarked populations. This first and most basic spatial capture-recapture model has been used routinely for decades and, formally, it is a spatially-explicit model in the sense that it describes, explicitly, the spatial organization of individual locations (although this is not always stated explicitly) and, as a result, somewhat general models of how individuals are distributed in space can be specified (Hedley et al., 1999; Royle et al., 2004; Johnson, 2010; Niemi and Fernández, 2010; Sillett et al., 2012).

As with other models we've encountered in this chapter, the distance sampling model, under data augmentation, includes a set of  $M$  zero-inflation variables  $z_i$  and a binomial observation model expressed conditional on  $z$  (binomial for  $z = 1$ , and fixed zeros for  $z = 0$ ). In distance sampling we pay for having only a single sample occasion (i.e.,  $K = 1$ ) by requiring constraints on the model of detection probability, normally imposed as the assumption that detection probability is 1.0 when distance equals 0. A standard model for detection probability is the "half-normal" model:

$$p_i = \exp(-\alpha_1 x_i^2)$$

for  $\alpha_1 > 0$ , where  $x_i$  denotes the distance at which the  $i$ th individual is detected relative to some reference location where perfect detectability ( $p = 1$ ) is assumed. This encounter probability model is more often written with  $\alpha_1 = 1/2\sigma^2$ . If  $K > 1$  then an intercept in this model, say  $\alpha_0$ , is identifiable and such models are usually called "capture-recapture distance sampling" (Alpizar-Jara and Pollock, 1996; Borchers et al., 1998).

As with previous examples, we require a distribution for the individual covariate  $x_i$ . The customary choice is

$$x_i \sim \text{Uniform}(0, B)$$

wherein  $B > 0$  is a known constant, being the upper limit of data recording by the observer (i.e., the point count radius, or transect half-width). Specification of this distance sampling model in the **BUGS** language is shown in Panel 4.2, taken from Royle and Dorazio (2008).

As with the individual covariate model in the previous section, the distance sampling model can be equivalently specified by putting a prior distribution on individual *location* instead of distance between individual and observation point (or transect). Thus we can write the general distance sampling model as

$$p_i = h(||\mathbf{u}_i - \mathbf{x}_0||, \alpha_1)$$

along with

$$\mathbf{u}_i \sim \text{Uniform}(\mathcal{S})$$

where  $\mathbf{x}_0$  is a fixed point (or line) and  $\mathbf{u}_i$  is the individual's location, which is observed for the sample of  $n$  individuals. In practice it is easier to record distance instead of location.

---

```

alpha1 ~ dunif(0,10)           # Prior distributions
psi ~ dunif(0,1)

for(i in 1:(nind+nz)){
  z[i] ~ dbern(psi)           # DA variables
  x[i] ~ dunif(0,B)           # B=strip width
  p[i] <- exp(logp[i])        # Detection function
  logp[i] <- - alpha1*(x[i]*x[i])
  mu[i] <- z[i]*p[i]
  y[i] ~ dbern(mu[i])         # Observation model
}

N <- sum(z[1:(nind+nz)])      # N is a derived parameter
D <- N/striparea               # D = N/total area of transects

```

---

Panel 4.2: Distance sampling model in **BUGS** for a line transect situation, using a half-normal detection function.

4527 Basic math can be used to argue that if individuals have a uniform distribution in space,  
 4528 then the distribution of Euclidean distance is also uniform. In particular, if a transect of  
 4529 length  $L$  is used and  $x$  is distance to the transect then  $F(x) = \Pr(X \leq x) = L*x/L*B =$   
 4530  $x/B$  and  $f(x) = dF/dx = (1/B)$ . For measurements of radial distance, we provided the  
 4531 analogous argument in the previous section.

4532 The preceding paragraph makes it clear that distance sampling is a special case of  
 4533 spatial capture-recapture models, such as those derived from model  $M_x$  of the previous  
 4534 section, where the encounter probability is related directly to *distance*, which is a reduced  
 4535 information summary of *location*,  $\mathbf{u}$ . Some intermediate forms of SCR/DS models can  
 4536 be described (Royle et al., 2011a). In the context of our general characterization of SCR  
 4537 models (Chapt. 2.6), we suggested that every SCR model can be described, conceptually,  
 4538 by a hierarchical model of the form:

$$[y|\mathbf{u}][\mathbf{u}|\mathbf{s}][\mathbf{s}].$$

4539 Distance sampling ignores the part of the model pertaining to  $\mathbf{s}$ , and deals only with the  
 4540 model components for the observed data  $\mathbf{u}$ <sup>8</sup>. Thus, we are left with a hierarchical model  
 4541 of the form

$$[y|\mathbf{u}][\mathbf{u}].$$

4542 In contrast, as we will see in the next chapters, many SCR models (Chapt. 5) ignore  $\mathbf{u}$   
 4543 and condition on  $\mathbf{s}$ , which is not observed:

$$[y|\mathbf{s}][\mathbf{s}]$$

4544 Since  $[\mathbf{u}]$  and  $[\mathbf{s}]$  are both assumed to be uniformly distributed, these are equivalent models!  
 4545 The main differences have to do with interpretation of model components and whether or  
 4546 not the latent variables are observable (in distance sampling they are).

<sup>8</sup>Equivalently, we could also say that  $[\mathbf{u}]$  in the distance sampling model is  $[\mathbf{u}] = \int [\mathbf{u}|\mathbf{s}][\mathbf{s}]ds$

4547 So why bother with SCR models when distance sampling yields density estimates and  
 4548 accounts for spatial heterogeneity in detection? For one, imagine trying to collect distance  
 4549 sampling data on species such as jaguars or tigers! Clearly, distance sampling requires  
 4550 that one can collect large quantities of distance data, which is not always possible. For  
 4551 tigers, it is much easier, efficient, and safer to employ camera traps or track plates and  
 4552 then apply SCR models. Furthermore, as we will see in Chapt. 15, SCR models can make  
 4553 use of distance data, allowing us to study distribution, movement, and density. Thus,  
 4554 SCR models are more general and versatile than distance sampling models (which clearly  
 4555 are a special case), and can accommodate data from virtually all animal survey designs.

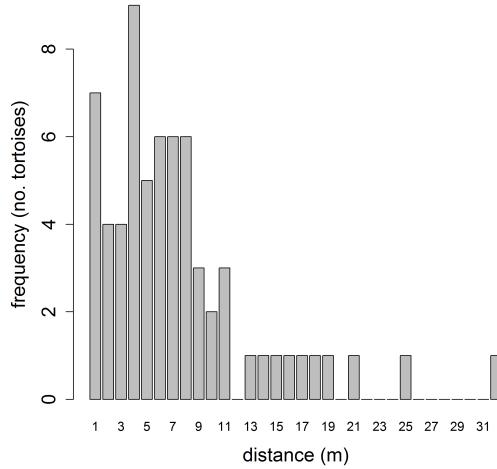
4556 **4.6.1 Example: Sonoran desert tortoise study**

4557 We illustrate the application of distance sampling models using data on the Sonoran desert  
 4558 tortoise (*Gopherus agassizii*), shown in Fig. 4.6, collected along transects in southern  
 4559 Arizona (see Zylstra et al. (2010) for details). The data are from 120 square transects  
 4560 having four 250-m sides, although we ignore this detail in our analysis here and regard  
 4561 them as 1 km transects, and we pooled the detection data from all 120 transects. The  
 4562 histogram of encounter distances from the 65 encountered individuals is shown in Fig. 4.7



**Figure 4.6.** Desert tortoise in its native habitat (*Photo credit: Erin Zylstra, Univ. of Arizona*).

4563  
 4564 Commands for reading in and organizing the data for analysis using **WinBUGS** are  
 4565 given in the help file `?tortoise` provided with the `scrbook` package. To compute density,



**Figure 4.7.** Distance histogram of  $n = 65$  Sonoran desert tortoise detections from a total of 120 km of survey transect.

4566 the total sampled area of the transects `striparea` is input as data, and computed as:  
 4567 120 (transects) multiplied by the length (1000 m) and half-width ( $B = 40$  m), then  
 4568 multiplied by 2, and divided by 10000 to convert to units of individuals per ha. We also  
 4569 provide commands for analyzing the data with `unmarked` (Fiske and Chandler, 2011) using  
 4570 hierarchical distance sampling models (Royle et al., 2004).

4571 Posterior summaries for the tortoise data are given in Tab. 4.8. Estimated density  
 4572 (posterior mean) is 0.54 individuals per ha and the estimated scale parameter of the  
 4573 distance function (posterior mean) is  $\sigma = 9.12$  meters. The R-hat statistics of around 1.02  
 4574 suggest that slightly longer MCMC simulations might be called for. The posterior mass  
 4575 of the data augmentation parameter  $\psi$  is located away from the upper bound  $\psi = 1$  and  
 4576 so the degree of data augmentation appears sufficient.

**Table 4.8.** Posterior summaries from the tortoise distance sampling data. Results were obtained using **WinBUGS** running 3 chains, each with 3000 iterations and the first 1000 discarded, thinning by 2.

Parameter	Mean	SD	2.5%	50 %	97.5%	Rhat	n.eff
$N$	516.67	54.71	415.00	516.00	632.00	1.02	100
$D$	0.54	0.06	0.43	0.54	0.66	1.02	100
$\alpha_1$	0.01	0.00	0.00	0.01	0.01	1.02	130
$\sigma$	9.12	0.77	7.77	9.07	10.77	1.02	130
$\psi$	0.61	0.07	0.49	0.61	0.75	1.02	96

## 4.7 SUMMARY AND OUTLOOK

4577 Traditional closed population capture-recapture models are closely related to binomial  
4578 generalized linear models. Indeed, the only real distinction is that in capture-recapture  
4579 models, the population size parameter  $N$  (corresponding also to the size of a hypothetical  
4580 “complete” data set) is unknown. This requires special consideration in the analysis of  
4581 capture-recapture models. The classical approach to inference recognizes that the observa-  
4582 tions don’t have a standard binomial distribution but, rather, a truncated binomial (from  
4583 which which the so-called *conditional likelihood* derives) since we only have encounter fre-  
4584 quency data on observed individuals. If instead we analyze the models using data augmen-  
4585 tation, which arises under a  $\text{Uniform}(0, M)$  prior for  $N$ , the observations can be modeled  
4586 using a zero-inflated binomial distribution. When we deal with the unknown- $N$  problem  
4587 using data augmentation then we are left with zero-inflated GLMs and GLMMs instead  
4588 of ordinary GLMs or GLMMs. The analysis of such zero-inflated models is practically  
4589 convenient, especially using the **BUGS** variants.

4590 Spatial capture-recapture models that we will consider in the rest of the chapters  
4591 of this book are closely related to individual covariate models (model  $M_x$ ). Naturally,  
4592 spatial capture-recapture models arise by defining individual covariates based on observed  
4593 locations of individuals—we can think of using some function of mean encounter location as  
4594 an individual covariate. We did this in a novel way, by using distance to the centroid of the  
4595 trapping array as a covariate. We analyzed the *full likelihood* using data augmentation,  
4596 and placed a prior distribution on the individual covariate which was derived from an  
4597 assumption that individual locations are, *a priori*, uniformly distributed in space. This  
4598 assumption provides for invariance of the density estimator to the choice of population  
4599 size area (induced by maximum distance from the centroid of the trap array). The model  
4600 addressed some important problems in the use of closed population models: it allows for  
4601 heterogeneity in encounter probability due to the spatial juxtaposition of individuals with  
4602 the array of traps, and it also provides a direct estimate of density because area is a  
4603 feature of the model (via the prior on the individual covariate). The model is still not  
4604 completely general, however, because it does not make full use of the spatial encounter  
4605 histories, which provide direct information about the locations and density of individuals.

4606 A specific individual covariate model that is in widespread use is classical distance  
4607 sampling. The model underlying distance sampling is precisely a special kind of SCR  
4608 model—but one without replicate samples. Understanding distance sampling and individ-  
4609 ual covariate models more broadly provides a solid basis for understanding and analyzing  
4610 spatial capture-recapture models. In fact if, instead of placing an explicit model on *dis-*  
4611 *tance* in the classical distance sampling model, we were to place the prior distribution on  
4612 *location*,  $s$ , of each individual, then the form of the distance sampling model more closely  
4613 resembles the SCR model we introduce in the next chapter.



4614

## Part II

4615

---

4616

# Basic SCR Models



---

# 5

---

## FULLY SPATIAL CAPTURE-RECAPTURE MODELS

4621 In the previous chapter, we discussed models that could be viewed as primitive spatial  
4622 capture-recapture models. We looked at a basic distance sampling model, and we also  
4623 considered a classical individual covariate modeling approach in which we defined a co-  
4624 variate to be the distance from the (estimated) home range center to the center of the  
4625 trap array. The individual covariate model that we conjured up was “spatial” in the sense  
4626 that it included some characterization of where individuals live but, on the other hand,  
4627 only a primitive or no characterization of trap location. That said, there is only a small  
4628 step from this model to spatial capture-recapture models that we consider in this chapter,  
4629 which fully recognize the spatial attribution of both individual animals *and* the locations  
4630 of encounter devices.

4631 Capture-recapture models must accommodate the spatial organization of individuals  
4632 and the encounter devices because the encounter process occurs at the level of individual  
4633 traps. Failure to consider the trap-specific data is one of the key deficiencies with classical  
4634 ad-hoc approaches which aggregate encounter information to the resolution of the entire  
4635 trap array. We have previously addressed some problems that this causes including induced  
4636 heterogeneity in encounter probability, imprecise notation of “sample area” and not being  
4637 able to accommodate trap-specific effects or trap-specific missing values. In this chapter  
4638 we resolve these issues by developing our first fully spatial capture-recapture model. This  
4639 model is not too different from that considered in Sec. 4.5 but, instead of defining the  
4640 individual covariate to be distance to the centroid of the array we define  $J$  individual  
4641 covariates - the distance to *each* trap. And, instead of using estimates of individual  
4642 locations  $\mathbf{s}$ , we consider a fully hierarchical model in which we regard  $\mathbf{s}$  as a latent variable  
4643 and impose a prior distribution on it.

4644 In this chapter we investigate the basic spatial capture-recapture model, which we re-  
4645 fer to as “model SCR0”, and address some important considerations related to its analysis  
4646 in **BUGS**. We demonstrate how to summarize posterior output for the purposes of pro-  
4647 ducing density maps or spatial predictions of density. The key aspect of the SCR models

4648 considered in this chapter is the formulation of a model for encounter probability that is  
 4649 a function of distance between individual home range center and trap locations. We also  
 4650 discuss how encounter probability models are related to explicit models of space usage  
 4651 or “home range area.” Understanding this allows us to compute, for example, the area  
 4652 used by an individual during some prescribed time. While it is intuitive that SCR models  
 4653 should be related to some model of space usage, this has not been discussed much in the  
 4654 literature (but see Royle et al. (2012a) which we address further in Chapt. 13).

## 5.1 SAMPLING DESIGN AND DATA STRUCTURE

4655 In our development here, we will assume a standard sampling design in which an array  
 4656 of  $J$  traps is operated for  $K$  sample occasions (say, nights) producing encounters of  $n$   
 4657 individuals. Because sampling occurs by traps and also over time, the most general data  
 4658 structure yields temporally *and* spatially indexed encounter histories for *each individual*.  
 4659 Thus a typical data set will include an encounter history *matrix* for each individual indicating  
 4660 which trap the individual was captured, during each sample occasion. For example,  
 4661 suppose we sample at 4 traps over 3 nights. A plausible data set for a single individual  
 4662 captured one time in trap 1 on the first night and one time in trap 3 on the 3rd night is:

```
4663     night1 night2 night3
4664 trap1    1    0    0
4665 trap2    0    0    0
4666 trap3    0    0    1
4667 trap4    0    0    0
```

4668 This data structure would be obtained for *each* of the  $i = 1, 2, \dots, n$  captured individuals.

4669 We develop models in this chapter for passive detection devices such as “hair snares” or  
 4670 other DNA sampling methods (Kéry et al., 2010; Gardner et al., 2010b) and related types of  
 4671 sampling devices in which (i) devices (“traps”) may capture any number of individuals (i.e.,  
 4672 they don’t fill up); (ii) an individual may be captured in more than one trap during each  
 4673 occasion but (iii) individuals can be encountered at most 1 time by each trap during any  
 4674 occasion. Hair snares for sampling DNA from bears and other species function according  
 4675 to these rules. An individual bear wandering about its territory might come into contact  
 4676 with  $> 1$  devices; a device may encounter multiple bears; however, in practice, it will  
 4677 often not be possible to attribute multiple visits of the same individual during a single  
 4678 occasion (e.g., night) to distinct encounter events. Thus, an individual may be captured  
 4679 at most 1 time in each trap during any occasion. While this model, which we refer to  
 4680 as SCR0, is most directly relevant to hair snares and other DNA sampling methods for  
 4681 which multiple detections of an individual are not distinguishable, we will also make use  
 4682 of the model for data that arise from camera-trapping studies. In practice, with camera  
 4683 trapping, individuals might be photographed several times in a night but it is common to  
 4684 distill such data into a single binary encounter event for reasons discussed later in Chapt.  
 4685 9.

4686 The statistical assumptions we make to build a model for these data are that individual  
 4687 encounters within and among traps are independent, and this allows us to regard individual- and trap-specific encounters as *independent* Bernoulli trials (see next section).  
 4688 These basic (but admittedly at this point somewhat imprecise) assumptions define the  
 4689

**Table 5.1.** Hypothetical spatial capture-recapture data set showing 6 individuals captured in 4 traps. Each entry is the number of captures out of  $K = 3$  nights of sampling.

Individual	Trap 1	Trap 2	Trap 3	Trap 4
1	1	0	0	0
2	0	2	0	0
3	0	0	0	1
4	0	1	0	0
5	0	0	1	1
6	1	0	1	0

4690 basic spatial capture-recapture model, SCR0. We will make things more precise as we  
 4691 develop a formal statistical definition of the model shortly.

## 5.2 THE BINOMIAL OBSERVATION MODEL

4692 We begin by considering the simple model in which there are no time-varying covariates  
 4693 that influence encounter, there are no explicit individual-specific covariates, and there are  
 4694 no covariates that influence density. In this case, we can aggregate the binary encounters  
 4695 over the  $K$  sample occasions and record the total number of encounters out of  $K$ . We will  
 4696 denote these individual- and trap-specific encounter frequencies by  $y_{ij}$  for  $i = 1, 2, \dots, n$   
 4697 captured individuals and  $j = 1, 2, \dots, J$  traps. For example, suppose we observe 6 individuals  
 4698 in sampling at 4 traps over 3 nights of sampling then a plausible data set is the  $6 \times 4$   
 4699 matrix of encounters (out of 3 sampling occasions) shown in Table 5.1. We assume that  
 4700  $y_{ij}$  are mutually independent outcomes of a binomial random variable which we express  
 4701 as:

$$y_{ij} \sim \text{Binomial}(K, p_{ij}) \quad (5.2.1)$$

4702 This is the basic model underlying standard closed population models (Chapt. 4) except  
 4703 that, in the present case, the encounter frequencies are individual- *and* trap-specific, and  
 4704 encounter probability  $p_{ij}$  depends on both individual *and* trap.

4705 As we did in Sec. 4.5, we will make explicit the notion that  $p_{ij}$  is defined conditional  
 4706 on *where* individual  $i$  lives. Naturally, we think about defining an individual home range  
 4707 and then relating  $p_{ij}$  explicitly to a summary of its location relative to each trap. For  
 4708 example, the centroid of the individuals home range, or its center of activity (Efford, 2004;  
 4709 Borchers and Efford, 2008; Royle and Young, 2008). In what follows, we define  $\mathbf{s}_i$ , a two-  
 4710 dimensional spatial coordinate, to be the home range or activity center of individual  $i$ .  
 4711 Then, the SCR model postulates that encounter probability,  $p_{ij}$ , is a decreasing function  
 4712 of distance between  $\mathbf{s}_i$  and the location of trap  $j$ ,  $\mathbf{x}_j$  (also a two-dimensional spatial  
 4713 coordinate). A standard model for modeling binomial counts is the logistic regression,  
 4714 where we model the dependence of  $p_{ij}$  on distance according to:

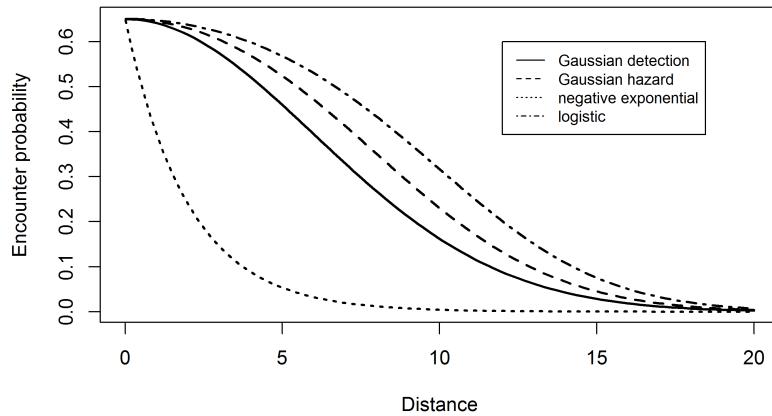
$$\text{logit}(p_{ij}) = \alpha_0 + \alpha_1 \|\mathbf{x}_j - \mathbf{s}_i\| \quad (5.2.2)$$

4715 where, here,  $\|\mathbf{x}_j - \mathbf{s}_i\|$  is the distance between  $\mathbf{s}_i$  and  $\mathbf{x}_j$ . We sometimes write  $\|\mathbf{x}_j - \mathbf{s}_i\| =$   
 4716  $\text{dist}(\mathbf{x}_j, \mathbf{s}_i) = d_{ij}$ . Alternatively, a popular model is

$$p_{ij} = p_0 \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{x}_j - \mathbf{s}_i\|^2\right) \quad (5.2.3)$$

4717 which is similar to the “half-normal” model in distance sampling, except with an intercept  
 4718  $p_0 \leq 1$  which can be estimated in SCR studies. Because it is the kernel of a bivariate  
 4719 normal, or Gaussian, probability density function for the random variable “individual  
 4720 location” we will refer to it as the “(bivariate) normal” or “Gaussian” model although  
 4721 the distance sampling term “half-normal” is widely used. In the context of 2-dimensional  
 4722 space, the model is clearly interpretable as a primitive model of movement outcomes or  
 4723 space usage (we discuss this in Sec. 5.4).

4724 There are a large number of standard detection models commonly used (see Chapt. 7).  
 4725 All other standard models that relate encounter probability to  $\mathbf{s}$  will also have a parameter  
 4726 that multiplies distance in some non-linear function. To be consistent with parameter  
 4727 naming across models, we will sometimes parameterize any encounter probability model  
 4728 so that the coefficient on distance (or distance squared) is  $\alpha_1$ . So, for the Gaussian model,  
 4729  $\alpha_1 = 1/(2\sigma^2)$ . A characteristic of the common parametric forms is they are monotone de-  
 4730 creasing with distance, but vary in their characteristic behavior as they approach distance  
 4731 = 0. We show the standard Gaussian, Gaussian hazard, negative exponential and logistic  
 4732 models in Fig. 5.1. The negative exponential model has  $p_{ij} = p_0 \exp(-\alpha_1 \|\mathbf{x}_j - \mathbf{s}_i\|)$  and  
 4733 the Gaussian hazard model has  $p_{ij} = 1 - \exp(-\lambda_0 k(\mathbf{x}_j, \mathbf{s}_i))$  where  $k(\mathbf{x}_j, \mathbf{s}_i)$  is the Gaussian  
 kernel. Whatever model we choose for encounter probability, we should always keep in



**Figure 5.1.** Some common encounter probability models showing the characteristic monotone decrease of encounter probability with distance between activity center and trap location.

4734 mind that the activity center for individual  $i$ ,  $\mathbf{s}_i$ , is an unobserved random variable. To  
 4735 be precise about this in the model, we should express the observation model as  
 4736

$$y_{ij} | \mathbf{s}_i \sim \text{Binomial}(K, p(\mathbf{s}_i; \alpha_1))$$

4737 but sometimes, for notational simplicity, we abbreviate this by omitting some of the  
4738 arguments to  $p$ .

### 4739 5.2.1 Definition of home range center

4740 We define an individual's home range as *the area used by an organism during some time*  
4741 *period* which has a clear meaning for most species regardless of their biology. We therefore  
4742 define the home range center (or activity center) to be the center of the space that individ-  
4743 ual was occupying (or using) during the period in which traps were active. Thinking about  
4744 it in that way, it could even be observable (almost) as the centroid of a very large number  
4745 of radio fixes over the course of a survey period or a season. Thus, this practical version  
4746 of a home range center in terms of space usage is a well-defined construct regardless of  
4747 whether one thinks the home range itself is a meaningful concept. We use the terms home  
4748 range center and activity center interchangeably, and we recognize that this is a transient  
4749 thing which applies only to a well-defined period of study.

### 4750 5.2.2 Distance as a latent variable

4751 If we knew precisely every  $\mathbf{s}_i$  in the population (and population size  $N$ ), then the model  
4752 specified by Eqs. 5.2.1 and 5.2.2 would be just an ordinary logistic regression-type of  
4753 a model (with covariate  $d_{ij}$ ) which we learned how to fit using **WinBUGS** previously  
4754 (Chapt. 3). However, the activity centers are unobservable even in the best possible  
4755 circumstances. In that case,  $d_{ij}$  is an unobserved variable, analogous to the situation in  
4756 classical random effects models. We need to therefore extend the model to accommodate  
4757 these random variables with an additional model component – the random effects dis-  
4758 tribution. The customary assumption is the so-called “uniformity assumption,” which is  
4759 to assume that the  $\mathbf{s}_i$  are uniformly distributed over space (the obvious next question:  
4760 “which space?” is addressed below). This uniformity assumption amounts to a uniform  
4761 prior distribution on  $\mathbf{s}_i$ , i.e., the pdf of  $\mathbf{s}_i$  is constant, which we may express

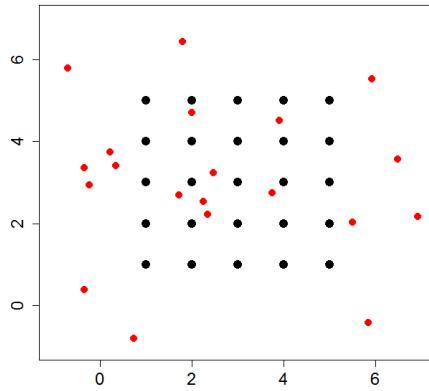
$$\Pr(\mathbf{s}_i) \propto \text{constant} \quad (5.2.4)$$

4762 As it turns out, this assumption is usually not precise enough to fit SCR models in practice  
4763 for reasons we discuss shortly. We will give another way to represent this prior distribution  
4764 that is more concrete, but depends on specifying the “state-space” of the random variable  
4765  $\mathbf{s}_i$ . The term state-space is a technical way of saying “the space of all possible outcomes”  
4766 of the random variable.

## 5.3 THE BINOMIAL POINT PROCESS MODEL

4767 In the SCR model, the individual activity centers are unobserved and thus we treat them  
4768 as random effects. Specifically, the collection of individual activity centers  $\mathbf{s}_1, \dots, \mathbf{s}_N$   
4769 represents a realization of a *binomial point process* (Illian et al., 2008, p. 61). The  
4770 binomial point process (BPP) is analogous to a Poisson point process in the sense that it  
4771 represents a “random scatter” of points in space – except that the total number of points  
4772 is *fixed*, whereas, in a Poisson point process, it is random (having a Poisson distribution).

4773 As an example, we show in Fig. 5.2 locations of 20 individual activity centers (black  
 4774 dots) in relation to a grid of 25 traps. For a Poisson point process the number of such  
 4775 points in the prescribed state-space would be random whereas often we will simulate fixed  
 4776 numbers of points, e.g., for evaluating the performance of procedures, e.g., how well does  
 our estimator perform when  $N = 50$ ?



**Figure 5.2.** Realization (small dots) of a binomial point process with  $N = 20$ . The large dots represent trap locations.

4777  
 4778 It is natural to consider a binomial point process in the context of capture-recapture  
 4779 models because it preserves  $N$  in the model and thus preserves the linkage directly with  
 4780 closed population models. In fact, under the binomial point process model, model  $M_0$   
 4781 and other closed models are simple limiting cases of SCR models, i.e., they arise as the  
 4782 coefficient on distance ( $\alpha_1$  above) tends to 0.

4783 While we often will express SCR models “conditional-on- $N$ ”, it will sometimes be  
 4784 convenient to impose specific prior distributions on  $N$ . By assuming  $N$  has a binomial  
 4785 distribution, we can make use of data augmentation, our preferred tool, for Bayesian  
 4786 analysis of the models as in Chapt. 4, thus yielding a methodologically coherent approach  
 4787 to analyzing the different classes of models. We might also assume that  $N$  has a Poisson  
 4788 distribution in some cases (see Chapt. 14). Of course, the two assumptions are closely  
 4789 related in the usual limiting sense.

4790 One consequence of having fixed  $N$  in the BPP model is that the model is not  
 4791 strictly a model of “complete spatial randomness”. This is because, if one forms counts  
 4792  $n(A_1), \dots, n(A_k)$  in any set of disjoint regions of the state-space, say  $A_1, \dots, A_k$ , then  
 4793 these counts are *not* independent. In fact, they have a multinomial distribution (see Illian  
 4794 et al., 2008, p. 61). Thus, the BPP model introduces a slight bit of dependence in the  
 4795 distribution of points. However, in most situations this will have no practical effect on any  
 4796 inference or analysis and, as a practical matter, we will usually regard the BPP model as

one of spatial independence among individual activity centers because each activity center is distributed independently of each other activity center. Despite this independence we see in Fig. 5.2 that *realizations* of randomly distributed points will typically exhibit distinct non-uniformity. Thus, independent, uniformly distributed points will almost never appear regularly, uniformly or systematically distributed. For this reason, the basic binomial (or Poisson) point process models are enormously useful in practical settings since they allow for a range of distribution patterns without violating the assumption of spatial randomness. More relevant for SCR models is that we actually have a little bit of data for some individuals and thus the resulting posterior point pattern can deviate strongly from uniformity, a point we come back to repeatedly in this book. The uniformity hypothesis is only a *prior* distribution which is directly affected by the quantity and quality of the observed data, to produce a posterior distribution which may appear distinctly non-uniform. In addition, we can build more flexible models for the point process, which we take up in Chapt. 11.

### 5.3.1 The state-space of the point process

Shortly we will focus on Bayesian analysis of model SCR0 with  $N$  known so that we can gain some basic experience with important elements of the model, and its analysis. To do this, we note that the individual activity centers  $\mathbf{s}_i, \dots, \mathbf{s}_N$  are unknown quantities and we will need to be able to simulate each  $\mathbf{s}_i$  in the population from the posterior distribution. In order to simulate the  $\mathbf{s}_i$ , it is necessary to describe precisely the region over which they are distributed. This is the quantity referred to above as the state-space, which is sometimes called the *observation window* in the point process literature. We denote the state-space henceforth (throughout this book) by  $\mathcal{S}$ , which is a region or a set of points comprising the potential values (the support) of the random variable  $\mathbf{s}$ . Thus, an equivalent explicit statement of the “uniformity assumption” is

$$\mathbf{s}_i \sim \text{Uniform}(\mathcal{S})$$

where  $\mathcal{S}$  is a precisely defined region. e.g., in Fig. 5.2,  $\mathcal{S}$  is the square defined by  $[-1, 7] \times [-1, 7]$ . Thus each of the  $N = 20$  points were generated by randomly selecting each coordinate on the line  $[-1, 7]$ . When points are distributed uniformly over some region, the point process is usually called a *homogeneous point process*.

#### Prescribing the state-space

Evidently, to define the model, we need to define the state-space,  $\mathcal{S}$ . How can we possibly do this objectively? Prescribing any particular  $\mathcal{S}$  seems like the equivalent of specifying a “buffer” which we have criticized as being ad hoc. How is it, then, that the choice of a state-space is *not* ad hoc? As we observed in Chapt. 4, it is true that  $N$  increases with  $\mathcal{S}$ , but only at the same rate as the area of  $\mathcal{S}$  increases under the prior assumption of constant density. As a result, we say that density is invariant to  $\mathcal{S}$  as long as  $\mathcal{S}$  is sufficiently large. Thus, while choice of  $\mathcal{S}$  is (or can be) essentially arbitrary, once  $\mathcal{S}$  is chosen, it defines the population being exposed to sampling, which scales appropriately with the size of the state-space.

For our simulated system developed previously in this chapter, we defined the state-space to be a square within which our trap array was centered. For many practical

4838 situations this might be an acceptable approach to defining the state-space, i.e., just a  
 4839 rectangle around the trap array. Although defining the state-space to be a regular polygon  
 4840 has computational advantages (e.g., we can implement this more efficiently in **BUGS** and  
 4841 cannot for irregular polygons), a regular polygon induces an apparent problem of admitting  
 4842 into the state-space regions that are distinctly non-habitat (e.g., oceans, large lakes, ice  
 4843 fields, etc.). It is difficult to describe complex regions in mathematical terms that can  
 4844 be used in **BUGS**. As an alternative, we can provide a representation of the state-space  
 4845 as a discrete set of points which the **R** package **secr** (Efford, 2011a) permits (**secr** uses  
 4846 the term “mask” for what we call the state-space). Defining the state-space by a discrete  
 4847 set of points is handy because it allows specific points to be deleted or not, depending on  
 4848 whether they represent available or suitable habitat (see Sec. 5.10). We can also define  
 4849 the state-space as an arbitrary collection of polygons stored as a GIS shapefile which can  
 4850 be analyzed easily by MCMC in **R** (see Sec. 17.7), but not so easily in the **BUGS** engines.  
 4851 In Sec. 5.10, we provide an analysis of the wolverine camera trapping data, in which we  
 4852 define the state-space to be a regular continuous polygon (a rectangle).

4853 **Invariance to the state-space**

4854 We will assert for all models we consider in this book that density is invariant to the size  
 4855 and extent of  $\mathcal{S}$ , if  $\mathcal{S}$  is sufficiently large, and as long as our model relating  $p_{ij}$  to  $\mathbf{s}_i$  is a  
 4856 decreasing function of distance. We can prove this easily by drawing an analogy with a 1-d  
 4857 case involving distance sampling. Let  $y_j$  be the number of individuals captured in some  
 4858 interval  $[d_{j-1}, d_j)$ , and define  $d_J = B$  for some large value of  $B$ . The observations from a  
 4859 survey are  $y_1, \dots, y_J$  and the likelihood is a multinomial likelihood, so the log-likelihood  
 4860 is of the form

$$\text{logL}(y_1, \dots, y_J) = \sum_{j=1}^J y_j \log(\pi_j)$$

4861 where  $\pi_j$  is the probability of detecting an individual in distance class  $j$ , which depends on  
 4862 parameters of the detection function (the manner of which is not relevant for the present  
 4863 discussion). Choosing  $B$  sufficiently large guarantees that  $\mathbb{E}(y_J) = 0$  and therefore the  
 4864 observed frequency in the “last cell” contributes nothing to the likelihood, in regular  
 4865 situations in which the detection function decays monotonically with distance and prior  
 4866 density is constant. We can think of  $B$  as being related to the state-space in an SCR  
 4867 model, as the width of a rectangular state-space with area  $B \times L$ ,  $L$  being the length  
 4868 of the transect. Thus, if we choose  $B$  large enough, then we ensure that the expected  
 4869 trap-frequencies beyond  $B$  will be 0, and thus contribute nothing to the likelihood.

4870 Sometimes our estimate of density can be affected by choosing  $\mathcal{S}$  too small. However,  
 4871 this might be sensible if  $\mathcal{S}$  is naturally well-defined. As we discussed in Chapt. 1,  $\mathcal{S}$  is  
 4872 part of the model, and thus it is sensible that estimates of density might be sensitive to  
 4873 its definition in problems where it is natural to restrict  $\mathcal{S}$ . One could imagine, however,  
 4874 in specific cases, e.g., a small population with well-defined habitat preferences, that a  
 4875 problem could arise because changing the state-space based on differing opinions, and  
 4876 GIS layers, might have substantial affects on the density estimate. But this is a real  
 4877 biological problem, and a natural consequence of the spatial formalization of capture-  
 4878 recapture models – a feature, not a bug or some statistical artifact – and it should be  
 4879 resolved with better information, research, and thinking. For situations where there is not  
 4880 a natural choice of  $\mathcal{S}$ , we should default to choosing  $\mathcal{S}$  to be very large in order to achieve

invariance or, otherwise, evaluate sensitivity of density estimates by trying a couple of different choices of  $\mathcal{S}$ . This is a standard “sensitivity to prior” argument that Bayesians always have to be conscious of. We demonstrate this in our analysis of Sec. 5.9 below. As an additional practical consideration, we note that the area of the state-space  $\mathcal{S}$  affects data augmentation. If you increase the size of  $\mathcal{S}$ , then there are more individuals to account for and therefore the size of the augmented data set  $M$  must increase. This has computational implications.

### 5.3.2 Connection to model $M_h$ and distance sampling

SCR models are closely related to “model  $M_h$ ” and also distance sampling. In SCR models, heterogeneity in encounter probability is induced by both the effect of distance in the model for detection probability and also from specification of the state-space. Hence, the state-space is an explicit element of the model. To understand this, suppose activity centers have the uniform distribution:

$$\mathbf{s} \sim \text{Uniform}(\mathcal{S})$$

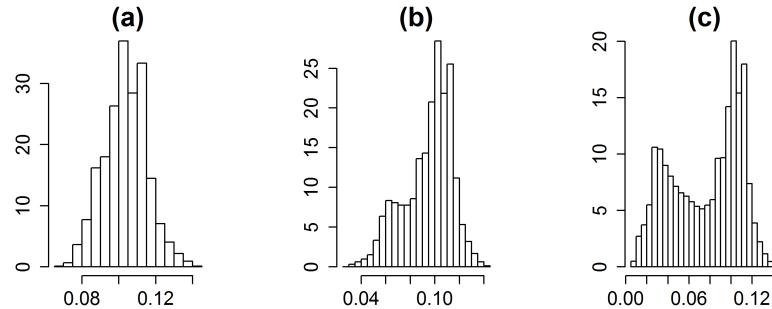
and encounter probability is a function of  $\mathbf{s}$ , denoted by  $p(\mathbf{s}) = p(y = 1|\mathbf{s})$ . For example, under Eq. 5.2.2 we have that

$$p(\mathbf{s}) = \text{logit}^{-1}(\alpha_0 - \alpha_1 \|\mathbf{x}_j - \mathbf{s}_i\|)$$

and we can work out, either analytically or empirically, what is the implied distribution of  $p$  for a population of individuals. Fig. 5.3 shows a histogram of  $p$  for a hypothetical population of 100000 individuals on a state-space enclosing our  $5 \times 5$  trap array above, under the logistic model for distance given by Eq. 5.2.2 with buffers of 0.2, 0.5 and 1.0. We see the mass shifts to the left as the buffer increases, implying more individuals with lower encounter probabilities, as their home range centers increase in distance from the trap array.

Another way to understand this is by representing  $\mathcal{S}$  as a set of discrete points on a grid. In the coarsest possible case where  $\mathcal{S}$  is a single arbitrary point, then every individual has exactly the same  $p$ . As we increase the number of points in  $\mathcal{S}$ , more distinct values of  $p$  are possible. Indeed, when  $\mathcal{S}$  is characterized by discrete points, then SCR models are precisely a type of finite-mixture model (Norris and Pollock, 1996; Pledger, 2004), except, in the case of SCR models, we have some information about which group an individual belongs to (i.e., where their activity center is), as a result of which traps it is captured in.

It is also worth re-emphasizing that the basic SCR encounter model is a binomial encounter model in which distance is a covariate. As such, it is strikingly similar to classical distance sampling models (Buckland et al., 2001). Both have distance as a covariate but, in classical distance sampling problems, the focus is on the distance between the observer and the animal at an instant in time, not the distance between a trap and an animal’s home range center. As a practical matter, in distance sampling, “distance” is *observed* for those individuals that appear in the sample. Conversely, in SCR problems, it is only imperfectly observed (we have partial information in the form of trap observations). Clearly, it is preferable to observe distance if possible, but distance sampling requires field methods that are not practical in many situations, e.g. when studying carnivores such as



**Figure 5.3.** Implied distribution of  $p_i$  for a population of individuals as a function of the size of the state-space buffer around the trap array. The state-space buffer is 0.2, 0.5 and 1.0 for panels (a), (b), (c), respectively. In each case, the trap array is fixed and centered within a square state-space.

bears or large cats. Furthermore, SCR models allow us to relax many of the assumptions made in classical distance sampling, such as perfect detection at distance zero, and SCR models allow for estimates of quantities other than density, such as home range size, and space usage (see Chaps. 12 and 13).

#### 5.4 THE IMPLIED MODEL OF SPACE USAGE

We developed the basic SCR model in terms of a latent variable,  $\mathbf{s}$ , the home range center or activity center. Surely the encounter probability model, which relates encounter of individuals in specific traps to  $\mathbf{s}$  must somehow imply a certain model for home range geometry and size. Here we explore the nature of that relationship and we argue that any given detection model implies a model of space usage – i.e., the amount and extent of area used some prescribed percentage of the time. So we might say, for example, 95% of animal movements are within some distance from an individual's activity center. While we have used the term “home range” or similar, what we really mean to imply is something that would be more clearly identified as resource selection or space usage (the latter term meaning resource selection, when the resource is only homogeneous space).

Intuitively, the detection function of SCR models is related to space usage by individuals. Indeed, it is natural to interpret the detection model as the composite of two processes: movement of an individual about its home range i.e., how it uses space within its home range (“space usage”), and detection *conditional on use* in the vicinity of a trapping device. It is natural to decompose encounter probability according to:

$$\Pr(\text{encounter at } \mathbf{x}|\mathbf{s}) = \Pr(\text{encounter}|\text{usage of } \mathbf{x}, \mathbf{s}) \Pr(\text{usage of } \mathbf{x}|\mathbf{s}).$$

In practice it might make sense to think about the first component, i.e.,  $\Pr(\text{encounter}|\text{usage of } \mathbf{x}, \mathbf{s})$  as being a constant (e.g., if traps are located within arbitrarily small grid cells) and then, in that case, the encounter probability model is directly

4942 proportional to this model for individual movements about their home range center deter-  
 4943 mining the use frequency of each  $\mathbf{x}$ . This is a sensible heuristic model for what ecologists  
 4944 would call a central place forager although, as we have stated previously, it may be mean-  
 4945 ingful as a description of transient space usage as well (that is, the space usage during the  
 4946 period of sampling).

4947 To motivate a specific model for space usage, imagine the area we are interested in  
 4948 consists of some large number of small pixels (i.e. we're looking at a discrete representation  
 4949 of space), and that we have some kind of perfect observation device (e.g., continuous  
 4950 telemetry) so that we observe every time an individual moves into a pixel. After a long  
 4951 period of time, we observe an enormous sample size of  $\mathbf{x}$  values. We tally those up into  
 4952 each pixel, producing the frequency  $m(\mathbf{x}, \mathbf{s})$ , which is something like the "true" usage of  
 4953 pixel  $\mathbf{x}$  by individual with activity center  $\mathbf{s}$ . So, then, the usage model should be regarded  
 4954 as a probability mass function for these counts and, naturally, we regard the counts  $m(\mathbf{x}, \mathbf{s})$   
 4955 as a multinomial observation with probabilities  $\pi(\mathbf{x}|\mathbf{s})$ , and prescribe a suitable model for  
 4956  $\pi(\mathbf{x}|\mathbf{s})$  that describes how use events should accumulate in space. A natural null model  
 4957 for  $\pi(\mathbf{x}|\mathbf{s})$  has a decreasing probability of use as  $\mathbf{x}$  gets far away from  $\mathbf{s}$ ; i.e., animals spend  
 4958 more time close to their activity centers than far away. We can regard points used by  
 4959 the individual with activity center  $\mathbf{s}$  as the realization of a point process with conditional  
 4960 intensity:

$$\pi(\mathbf{x}|\mathbf{s}) = \frac{k(\mathbf{x}, \mathbf{s})}{\sum_x k(\mathbf{x}, \mathbf{s})} \quad (5.4.1)$$

4961 where  $k(\mathbf{x}, \mathbf{s})$  is any positive function. In continuous space, the equivalent representation  
 4962 would be:

$$\pi(\mathbf{x}|\mathbf{s}) = \frac{k(\mathbf{x}, \mathbf{s})}{\int k(\mathbf{x}, \mathbf{s}) dx}.$$

4963 Clearly the space used by an individual will be proportional to whatever kernel,  $k(\mathbf{x}, \mathbf{s})$ ,  
 4964 we plug-in here. If we use a negative exponential function, then this produces a standard  
 4965 resource selection function (RSF) model (e.g., Manly et al., 2002, Chapt. 8). But, here  
 4966 we use a Gaussian kernel, i.e.,

$$k(\mathbf{x}, \mathbf{s}) = \exp(-d(\mathbf{x}, \mathbf{s})^2/(2\sigma^2))$$

4967 so that contours of the probability of space usage resemble a bivariate normal or Gaussian  
 4968 probability distribution function.

4969 To apply this model of space-usage to SCR problems we allow for imperfect detection  
 4970 by introducing a non-uniform "thinning rate" of the true counts  $m(\mathbf{x}, \mathbf{s})$ . This yields,  
 4971 precisely, our Gaussian encounter probability model where the thinning rate is our baseline  
 4972 encounter probability  $p_0$  for each pixel where we place a trap, and  $p = 0$  in each pixel  
 4973 where we don't place a trap.

4974 The main take-away point here is that underlying most SCR models is some kind of  
 4975 model of space-usage, implied by the specific choice of  $k(\mathbf{x}, \mathbf{s})$ . Whether or not we have  
 4976 perfect sampling devices, the function we use in the encounter probability model equates  
 4977 to some conditional distribution of points, a utilization distribution, as in Eq. 5.4.1, from  
 4978 which we can compute effective home range area, i.e., the area that contains some percent  
 4979 of the mass of a probability distribution proportional to  $k(\mathbf{x}, \mathbf{s})$ ; e.g., 95% of all space used  
 4980 by an individual with activity center  $\mathbf{s}$ .

4981 **5.4.1 Bivariate normal case**

4982 One encounter model that allows direct analytic computation of home range area is the  
 4983 Gaussian encounter probability model

$$p(\mathbf{x}, \mathbf{s}) = p_0 \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{x} - \mathbf{s}\|^2\right).$$

4984 For this model, encounter probability is proportional to the kernel of a bivariate normal  
 4985 (Gaussian) pdf and so the natural interpretation is that in which movement outcomes (or  
 4986 successive locations of an individual) are draws from a bivariate normal distribution with  
 4987 standard deviation  $\sigma$ . We say that use of this model implies a bivariate normal model of  
 4988 space usage. Under this model we can compute precisely the effective home range area. In  
 4989 particular, if use outcomes are bivariate normal, then  $\|\mathbf{x} - \mathbf{s}\|^2$  has a chi-square distribution  
 4990 with 2 d.f. and the quantity  $B(\alpha)$  that encloses  $(1 - \alpha)\%$  of all realized distances i.e.,  
 4991  $\Pr(d \leq B(\alpha)) = 1 - \alpha$ , is  $B(\alpha) = \sigma * \sqrt{q(\alpha, 2)}$  where  $q(\alpha, 2)$  is the 0.05 chi-square  
 4992 critical value on 2 df. For example, to compute  $q(.05, 2)$  in R we execute the command  
 4993 `qchisq(.95, 2)` which is  $q(2, \alpha) = 5.99$ . Then, for  $\sigma = 1$ ,  $B(\alpha) = 1 * \sqrt{5.99} = 2.447$ .  
 4994 Therefore 95% of the points used will be within 2.447 (standard deviation) units of the  
 4995 home range center. So, in practice, we can estimate  $\sigma$  by fitting the bivariate normal  
 4996 encounter probability model to some SCR data, and then use the estimated  $\sigma$  to compute  
 4997 the “95% radius”, say  $r_{.95} = \sigma\sqrt{5.99}$ , and convert this to the 95% use area – the area  
 4998 around  $\mathbf{s}$  which contains 95% of the movement outcomes – according to  $A_{.95} = \pi r_{.95}^2$ .

4999 An alternative bivariate normal model is the bivariate normal hazard rate model:

$$p(\mathbf{x}, \mathbf{s}) = 1 - \exp\left(-\lambda_0 * \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{x} - \mathbf{s}\|^2\right)\right) \quad (5.4.2)$$

5000 We use  $\lambda_0$  here because this parameter, the baseline encounter rate, can be  $> 1$ . This arises  
 5001 by assuming the latent “use frequency”  $m(\mathbf{x}, \mathbf{s})$  is a Poisson random variable with intensity  
 5002  $\lambda_0 k(\mathbf{x}, \mathbf{s})$ . The model is distinct from our Gaussian encounter model  $p(\mathbf{x}, \mathbf{s}) = p_0 k(\mathbf{x}, \mathbf{s})$   
 5003 used previously, although we find that they produce similar results in terms of estimates  
 5004 of density or 95% use area, as long as baseline encounter probability is low. We discuss  
 5005 these two formulations of the bivariate normal model further in Chapt. 9.

5006 **5.4.2 Empirical analysis**

5007 For any encounter model we can compute space usage quantiles empirically by taking a fine  
 5008 grid of points and either simulating movement outcomes with probabilities proportional to  
 5009  $p(\mathbf{x}, \mathbf{s})$  and accumulating area around  $\mathbf{s}$ , or else we can do this precisely by varying  $B(\alpha)$   
 5010 to find that value within which 95% of all movements are concentrated, i.e., the set of all  
 5011  $\mathbf{x}$  such that  $\|\mathbf{x} - \mathbf{s}\| \leq B(q)$ . Under any detection model, movement outcomes will occur  
 5012 in proportion to  $p(\mathbf{x}, \mathbf{s})$ , as long as the probability of encounter is constant, *conditional on*  
 5013 use, and so we can define our space usage distribution according to:

$$\pi(\mathbf{x} | \mathbf{s}) = \frac{p(\mathbf{x}, \mathbf{s})}{\sum_x p(\mathbf{x}, \mathbf{s})}$$

5014 Given the probabilities  $\pi(\mathbf{x}, \mathbf{s})$  for all  $\mathbf{x}$  we can find the value of  $B(q)$ , for any  $q$ , such that

$$\left( \sum_{\mathbf{x}: \|\mathbf{x} - \mathbf{s}\| \leq B(q)} \pi(\mathbf{x}, \mathbf{s}) \right) \leq 1 - q$$

5015 (here, we use  $\ni$  to mean “such that”). We have a function called `hra` in the `scrbook`  
 5016 package that computes the home range area for any encounter model and prescribed  
 5017 parameter values. The help file for `hra` has an example of simulating some data. The  
 5018 following commands illustrate this calculation for two different bivariate normal models  
 5019 of space usage:

```

5020 ##
5021 ## Define encounter probability model as R function
5022 ##
5023 > pGauss2 <- function(parms,Dmat){
5024   a0 <- parms[1]
5025   sigma <- parms[2]
5026   lp <- parms[1] -(1/(2*parms[2]*parms[2]))*Dmat*Dmat
5027   p <- 1-exp(-exp(lp))
5028   p
5029 }
5030
5031 > pGauss1 <- function(parms,Dmat){
5032   a0 <- parms[1]
5033   sigma <- parms[2]
5034   p <- plogis(parms[1])*exp( -(1/(2*parms[2]*parms[2]))*Dmat*Dmat )
5035   p
5036 }
5037
5038 ##
5039 ## Execute hra with sigma = .3993
5040 ##
5041 > hra(pGauss1,parms=c(-2,.3993),plot=FALSE,xlim=c(0,6),ylim=c(0,6),
5042   ng=500,tol=.0005)
5043
5044 [1] 0.9784019
5045 radius to achieve 95% of area: 0.9784019
5046 home range area: 3.007353
5047 [1] 3.007353
5048
5049
5050 ## Analytic solution:
5051 ##      true sigma that produces area of 3
5052 > sqrt(3/pi)/sqrt(5.99)
5053 [1] 0.3992751

```

5054 What this means is that  $B(q) = 0.978$  is the radius that encloses about 95% of all  
 5055 movements under the standard bivariate normal encounter model. Therefore, the area is  
 5056 about  $\pi * .978^2 = 3.007$  spatial units. You can change the intercept of the model and find  
 5057 that it has no effect. The true (analytic) value of  $\sigma$  that produces a home range area of 3.0  
 5058 is 0.3993 which is the value we initially plugged in to the `hra` function. We can improve  
 5059 on the numerical approximation to home range area (get it closer to 3.0) by increasing the  
 5060 resolution of our spatial grid (increase the `ng` argument) along with the `tol` argument.

5061 We can also reverse this process, and find, for any detection model, the parameter  
 5062 values that produce a certain  $(1 - q)\%$  home range area, which we imagine would be  
 5063 useful for doing simulation studies. The function `hra` will compute the value of the scale  
 5064 parameter that achieves a certain target  $(1 - q)\%$  home range area, by simply providing a  
 5065 non-null value of the variable `target.area`. Here we use `target.area = 3.00735` (from  
 5066 above) to obtain a close approximation to the value  $\sigma$  we started with (the parameter  
 5067 argument is meaningless here):

```
5068 > hra(pGauss1,parms=c(-2,.3993),plot=FALSE,xlim,ylim,ng=500,  

5069     target.area=3.00735,tol=.0005)  

5070  

5071 Value of parm[2] to achieve 95% home range area of 3.00735: 0.3993674
```

#### 5072 **5.4.3 Relevance of understanding space usage**

5073 One important reason that we need to be able to deduce “home range area” from a  
 5074 detection model is so that we can compare different models with respect to a common  
 5075 biological currency. Many encounter probability models have some “scale parameter”,  
 5076 which we might call  $\sigma$  no matter the model, but this relates to 95% area in a different  
 5077 manner under each model. Therefore, we want to be able to convert different models  
 5078 to the same currency. Another reason to understand the relationship between models of  
 5079 encounter probability and space usage is that it opens the door to combining traditional  
 5080 resource selection data from telemetry with spatial capture-recapture data. In Chapt. 13  
 5081 we consider this problem, for the case in which a sample of individuals produces encounter  
 5082 history data suitable for SCR models and, in addition, we have telemetry relocations on a  
 5083 sample of individuals. This is achieved by regarding the two sources of data as resulting  
 5084 from the same underlying process of space usage but telemetry data produce “perfect”  
 5085 observations, like always-on camera traps blanketing a landscape. We use this idea to  
 5086 model the effect of a measured covariate at each pixel, say  $C(\mathbf{x})$ , on home range size and  
 5087 geometry and, hence, the probability of encounter in traps.

#### 5088 **5.4.4 Contamination due to behavioral response**

5089 Interpretation of encounter probability models as models of animal home range and space  
 5090 usage can be complicated by a number of factors, including whether traps are baited or  
 5091 not. In the case of baited traps, this might lead to a behavioral response (Sec. 7.2.3)  
 5092 which could affect animal space usage. For example, if traps attract animals from a long  
 5093 distance, it could make typical home ranges appear larger than normal. More likely, in our  
 5094 view, it wouldn’t change the typical size of a range but would change how individuals use  
 5095 their range e.g., by moving from baited trap to baited trap, so that observed movement  
 5096 distances of individuals are typically larger than normal.

5097 In other cases, the reliance on Euclidean distance in models for encounter probability  
 5098 might be unrealistic, and can lead to biased estimates of density (Royle et al., 2013).  
 5099 For example, animals might concentrate their movements along trails, roads, or other  
 5100 landscape features. In this case, models that accommodate other distance metrics can be  
 5101 considered. We present models based on least-cost path in Chapt. 12.

## 5.5 SIMULATING SCR DATA

5102 It is always useful to simulate data because it allows you to understand the system that  
 5103 you're modeling and also calibrate your understanding with specific values of the model  
 5104 parameters. That is, you can simulate data using different parameter values until you  
 5105 obtain data that "look right" based on your knowledge of the specific situation that  
 5106 you're interested in. Here we provide a simple script to illustrate how to simulate spatial  
 5107 encounter history data. In this exercise we simulate data for 100 individuals and a 25 trap  
 5108 array laid out in a  $5 \times 5$  grid of unit spacing. The specific encounter model is the Gaussian  
 5109 model given above and we used this code to simulate data used in subsequent analyses.  
 5110 The 100 activity centers were simulated on a state-space defined by a  $8 \times 8$  square within  
 5111 which the trap array was centered (thus the trap array is buffered by 2 units). Therefore,  
 5112 the density of individuals in this system is fixed at  $100/64$ .

```

5113 > set.seed(2013)
5114 # Create 5 x 5 grid of trap locations with unit spacing
5115 > traplocs <- cbind(sort(rep(1:5,5)),rep(1:5,5))
5116 > ntraps <- nrow(traplocs)
5117 # Compute distance matrix:
5118 > Dmat <- e2dist(traplocs,traplocs)
5119
5120
5121 # Define state-space of point process. (i.e., where animals live).
5122 # "buffer" just adds a fixed buffer to the outer extent of the traps.
5123 #
5124 > buffer <- 2
5125 > xlim <- c(min(traplocs[,1] - buffer),max(traplocs[,1] + buffer))
5126 > ylim <- c(min(traplocs[,2] - buffer),max(traplocs[,2] + buffer))
5127
5128 > N <- 100    # population size
5129 > K <- 20     # number nights of effort
5130
5131 > sx <- runif(N,xlim[1],xlim[2])    # simulate activity centers
5132 > sy <- runif(N,ylim[1],ylim[2])
5133 > S <- cbind(sx,sy)
5134 # Compute distance matrix:
5135 > D <- e2dist(S,traplocs) # distance of each individual from each trap
5136
5137 > alpha0 <- -2.5      # define parameters of encounter probability
5138 > sigma <- 0.5        # scale parameter of half-normal
5139 > alpha1 <- 1/(2*sigma*sigma) # convert to coefficient on distance
5140
5141 # Compute Probability of encounter:
5142 #
5143 > probcap <- plogis(-2.5)*exp( - alpha1*D*D)
5144
5145 # Generate the encounters of every individual in every trap

```

```

5146 > Y <- matrix(NA,nrow=N,ncol=ntraps)
5147 > for(i in 1:nrow(Y)){
5148   Y[i,] <- rbinom(ntraps,K,probcap[i,])
5149 }

```

5150     We remind the reader that, in presenting **R** or other code snippets throughout the  
 5151     book, we will deviate from our standard variable expressions for some quantities. In  
 5152     particular, we sometimes substitute words for integer variable designations: **nind** (for  $n$ ),  
 5153     **ntraps** (for  $J$ ), and **nocc** (for  $K$ ). In our opinion this leaves less to be inferred by the  
 5154     reader in trying to understand code snippets.

5155     Subsequently we will generate data using this code packaged in an **R** function called  
 5156     **simSCRO** in the package **scrbook** which takes a number of arguments including **discard0**  
 5157     which, if TRUE, will return only the encounter histories for captured individuals. A second  
 5158     argument is **array3d** which, if TRUE, returns the 3-dimensional encounter history array  
 5159     instead of the aggregated **nind**  $\times$  **ntraps** encounter frequencies (see below). Finally we  
 5160     provide a random number seed, **rnd** = 2013 to ensure repeatability of the analysis here.  
 5161     We obtain a data set as above using the following command:

```

5162 > data <- simSCRO(discard0=TRUE, array3d=FALSE, rnd=2013)

```

5163     The **R** object **data** is a list, so let's take a look at what's in the list and then harvest some  
 5164     of its elements for further analysis below.

```

5165 > names(data)
5166 [1] "Y"      "traplocs" "xlim"      "ylim"      "N"       "alpha0"    "beta"
5167 [8] "sigma"   "K"
5168
5169 ## Grab encounter histories from simulated data list
5170 > Y <- data$Y
5171 ## Grab the trap locations
5172 > traplocs <- data$traplocs

```

### 5173 5.5.1 Formatting and manipulating real data sets

5174     Conventional capture-recapture data are easily stored and manipulated as a 2-dimensional  
 5175     array, an **nind**  $\times$  **K** (individuals by sample occasions) matrix, which is maximally informative  
 5176     for any conventional capture-recapture model, but not for spatial capture-recapture  
 5177     models. For SCR models we must preserve the spatial information in the encounter history  
 5178     information. We will routinely analyze data from 3 standard formats:

- 5179     (1) The basic 2-dimensional data format, which is an **nind**  $\times$  **ntraps** encounter frequency  
 5180         matrix such as that simulated previously. These are the total number of encounters in  
 5181         each trap, summed over the  $K$  sample occasions.
- 5182     (2) The maximally informative 3-dimensional array, for which we establish here the con-  
 5183         vention that it has dimensions **nind**  $\times$  **ntraps**  $\times$  **K**.
- 5184     (3) We use a compact format – the “encounter data file” – which we describe below in  
 5185         Sec. 5.9.

5186 To simulate data in the most informative format - the “3-d array” - we can use the **R**  
 5187 commands given previously but replace the last 4 lines with the following:

```
5188 > Y <- array(NA,dim=c(N,ntraps,K))
5189
5190 > for(i in 1:nrow(Y)){
5191   for(j in 1:ntraps){
5192     Y[i,j,1:K] <- rbinom(K,1,probcap[i,j])
5193   }
5194 }
```

5195 We see that a collection of  $K$  binary encounter events are generated for *each* individual  
 5196 and for *each* trap. The probabilities of those Bernoulli trials are computed based on the  
 5197 distance from each individual’s home range center and the trap (see calculation above),  
 5198 and those are housed in the matrix `probcap`. Our data simulator function `simSRC0` will  
 5199 return the full 3-d array if `array3d=TRUE` is specified in the function call. To recover the  
 5200 2-d matrix from the 3-d array, and subset the 3-d array to individuals that were captured,  
 5201 we do this:

```
5202 # Sum over the ‘‘sample occasions’’ dimension (3rd margin of the array)
5203 > Y2d <- apply(Y,c(1,2),sum)
5204
5205 # Compute how many times each individual was captured
5206 > ncaps <- apply(Y2d,1,sum)
5207
5208 # Keep those individuals that were captured
5209 > Y <- Y[ncaps>0,,]
```

## 5.6 FITTING MODEL SCR0 IN BUGS

5210 Clearly if we somehow knew the value of  $N$  then we could fit this model directly because,  
 5211 in that case, it is a special kind of logistic regression model, one with a random effect (`s`)  
 5212 that enters into the model in a peculiar fashion, and also with a distribution (uniform)  
 5213 which we don’t usually think of as standard for random effects models. So our aim here is  
 5214 to analyze the known- $N$  problem, using our simulated data, as an incremental step in our  
 5215 progress toward fitting more generally useful models. To begin, we use our simulator to  
 5216 grab a data set and then harvest the elements of the resulting object for further analysis.

```
5217 > data <- simSRC0(discard0=FALSE,rnd=2013)
5218 > y <- data$Y
5219 > traplocs <- data$traplocs
5220
5221 # In this case nind=N because we’re doing the known-N problem
5222 #
5223 > nind <- nrow(y)
5224 > X <- data$traplocs
5225 > J <- nrow(X)    # number of traps
5226 > K <- data$K
```

```
5227 > xlim <- data$xlim
5228 > ylim <- data$ylim
```

5229 Note that we specify `discard0 = FALSE` so that we have a “complete” data set, i.e.,  
 5230 one with the all-zero encounter histories corresponding to uncaptured individuals. Now,  
 5231 within an **R** session, we can create the **BUGS** model file and fit the model using the  
 5232 following commands.

```
5233 cat("
5234   model{
5235     alpha0 ~ dnorm(0,.1)
5236     logit(p0) <- alpha0
5237     alpha1 ~ dnorm(0,.1)
5238     sigma <- sqrt(1/(2*alpha1))
5239     for(i in 1:N){ # note N here -- N is KNOWN in this example
5240       s[i,1] ~ dunif(xlim[1],xlim[2])
5241       s[i,2] ~ dunif(ylim[1],ylim[2])
5242       for(j in 1:J){
5243         d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
5244         y[i,j] ~ dbin(p[i,j],K)
5245         p[i,j] <- p0*exp(- alpha1*d[i,j]*d[i,j])
5246       }
5247     }
5248   }
5249 ",file = "SCR0a.txt")
```

5250 This model describes the Gaussian encounter probability model, but it would be trivial  
 5251 to modify that to various others including the logistic described above. One consequence  
 5252 of using the half-normal is that we have to constrain the encounter probability to be in  
 5253  $[0, 1]$  which we do here by defining `alpha0` to be the logit of the intercept parameter `p0`.  
 5254 Note that the distance covariate is computed within the **BUGS** model specification given  
 5255 the matrix of trap locations, `X`, which is provided to **WinBUGS** as data.

5256 Next we do a number of organizational activities including bundling the data for **Win-**  
 5257 **BUGS**, defining some initial values, the parameters to monitor and some basic MCMC  
 5258 settings. We choose initial values for the activity centers `s` by generating uniform random  
 5259 numbers in the state-space but, for the observed individuals, we replace those values by  
 5260 each individual’s mean trap coordinate for all encounters

```
5261 ### Starting values for activity centers, s
5262 > sst <- cbind(runif(nind,xlim[1],xlim[2]),runif(nind,ylim[1],ylim[2]))
5263 > for(i in 1:nind){
5264   if(sum(y[i,])==0) next
5265   sst[i,1] <- mean( X[y[i,>0,1] )
5266   sst[i,2] <- mean( X[y[i,>0,2] )
5267 }
5268
5269 > data <- list (y=y, X=X, K=K, N=nind, J=J, xlim=xlim, ylim=ylim)
5270 > inits <- function(){
```

---

```

5271     list (alpha0=rnorm(1,-4,.4), alpha1=runif(1,1,2), s=sst)
5272   }
5273
5274 > library(R2WinBUGS)
5275 > parameters <- c("alpha0","alpha1","sigma")
5276 > out <- bugs (data, inits, parameters, "SCR0a.txt", n.thin=1, n.chains=3,
5277                   n.burnin=1000,n.iter=2000,debug=TRUE,working.dir=getwd())

```

5278 There is little to say about the preceding operations other than to suggest that you might  
 5279 explore the output and investigate additional analyses by running the `simSCR0` script  
 5280 provided in the **R** package `scrbook`.

5281 For purposes here, we ran 1000 burn-in and 1000 post-burn-in iterations, and 3 chains,  
 5282 to obtain 3000 posterior samples. Because we know  $N$  for this particular data set we only  
 5283 have 2 parameters of the detection model to summarize (`alpha0` and `alpha1`), along with  
 5284 the derived parameter  $\sigma$ , the scale parameter of the Gaussian kernel, i.e.,  $\sigma = \sqrt{1/(2\alpha_1)}$ .  
 5285 When the object `out` is produced we print a summary of the results as follows:

```

5286 > print(out,digits=2)
5287 Inference for Bugs model at "SCR0a.txt", fit using WinBUGS,
5288   3 chains, each with 2000 iterations (first 1000 discarded)
5289   n.sims = 3000 iterations saved
5290       mean    sd   2.5%   25%   50%   75% 97.5% Rhat n.eff
5291 alpha0    -2.50  0.22  -2.95  -2.65  -2.48  -2.34  -2.09  1.01   190
5292 alpha1     2.44  0.42   1.64   2.15   2.44   2.72   3.30  1.00   530
5293 sigma      0.46  0.04   0.39   0.43   0.45   0.48   0.55  1.00   530
5294 deviance  292.80 21.16 255.60 277.50 291.90 306.00 339.30 1.01   380
5295
5296
5297 [...some output deleted...]
5298

```

5299 We know the data were generated with `alpha0 = -2.5` and `alpha1 = 2`. The estimates  
 5300 look reasonably close to those data-generating values and we probably feel pretty good  
 5301 about the performance of the Bayesian analysis and MCMC algorithm that **WinBUGS**  
 5302 cooked-up based on our sample size of 1 data set. It is worth noting that the `Rhat`  
 5303 statistics indicate reasonable convergence but, as a practical matter, we might choose to  
 5304 run the MCMC algorithm for additional time to bring these closer to 1.0 and to increase  
 5305 the effective posterior sample size (`n.eff`). Other summary output includes “deviance”  
 5306 and related things including the deviance information criterion (DIC). We discuss general  
 5307 issues of convergence and other MCMC considerations in Chapt. 17, and DIC and model  
 5308 selection in Chapt. 8.

## 5.7 UNKNOWN N

5309 In all real applications  $N$  is unknown. We handled this important issue in Chapt. 4  
 5310 using the method of data augmentation (DA) which we apply here to achieve a realistic  
 5311 analysis of model SCR0. As with the basic closed population models considered previously,

5312 we formulate the problem by augmenting our observed data set with a number of “all-  
 5313 zero” encounter histories - what we referred to in Chapt. 4 as potential individuals. If  
 5314  $n$  is the number of observed individuals, then let  $M - n$  be the number of potential  
 5315 individuals in the data set. For the 2-dimensional  $y_{ij}$  data structure ( $n$  individual  $\times J$   
 5316 traps encounter frequencies) we simply add additional rows of all-zero observations to  
 5317 that data set. Because such “individuals” are unobserved, they therefore necessarily have  
 5318  $y_{ij} = 0$  for all  $j$ . A data set, say with 4 traps and 6 individuals, augmented with 4  
 5319 pseudo-individuals therefore might look like this:

```
5320      trap1 trap2 trap3 trap4
5321 [1,]    1    0    0    0
5322 [2,]    0    2    0    0
5323 [3,]    0    0    0    1
5324 [4,]    0    1    0    0
5325 [5,]    0    0    1    1
5326 [6,]    1    0    1    0
5327 [7,]    0    0    0    0
5328 [8,]    0    0    0    0
5329 [9,]    0    0    0    0
5330 [10,]   0    0    0    0
```

5331 We typically have more than 4 traps and, if we’re fortunate, many more individuals in  
 5332 our data set.

5333 For the augmented data set, we introduce a set of binary latent variables (the data  
 5334 augmentation variables),  $z_i$ , and the model is extended to describe  $\Pr(z_i = 1)$  which is, in  
 5335 the context of this problem, the probability that an individual in the augmented data set  
 5336 is a member of the population of size  $N$  that was exposed to sampling. In other words,  
 5337 if  $z_i = 1$  for one of the all-zero encounter histories, this is implied to be a sampling zero  
 5338 whereas observations for which  $z_i = 0$  are “structural zeros” under the model. Under DA,  
 5339 we also express the binomial observation model *conditional on  $z_i$*  as follows:

$$y_{ij}|z_i \sim \text{Binomial}(K, z_i p_{ij})$$

5340 where we see that the binomial probability evaluates to 0 if  $z_i = 0$  (so  $y_{ij}$  is a fixed 0 in  
 5341 that case) and evaluates to  $p_{ij}$  if  $z_i = 1$ .

5342 How big does the augmented data set have to be? We discussed this issue in Chapt. 4  
 5343 where we noted that the size of the data set is equivalent to the upper limit of a uniform  
 5344 prior distribution on  $N$ . Practically speaking, it should be sufficiently large so that the  
 5345 posterior distribution for  $N$  is not truncated. On the other hand, if it is too large then  
 5346 unnecessary calculations are being done. An approach to choosing  $M$  by trial-and-error  
 5347 is indicated. Do a short MCMC run and then consider whether you need to increase  $M$ .  
 5348 See Chapt. 17 for an example of this. Kéry and Schaub (2012, Chapt. 6) provide an  
 5349 assessment of choosing  $M$  in closed population models. The useful thing about DA is that  
 5350 it removes  $N$  as an explicit parameter of the model. Instead,  $N$  is a derived parameter,  
 5351 computed by  $N = \sum_{i=1}^M z_i$ . Similarly, *density*,  $D$ , is also a derived parameter computed  
 5352 as  $D = N/\text{area}(\mathcal{S})$ .

---

### 5.7.1 Analysis using data augmentation in WinBUGS

5353 We provide a complete **R** script for simulating and organizing a data set, and analyzing  
 5354 the data in **WinBUGS**. As before we begin by obtaining a data set using our `simSCR0`  
 5355 function and then harvesting the required data objects from the resulting data list. Note  
 5356 that we use the `discard0=TRUE` option this time so that we get a “real looking” data set  
 5357 with no all-zero encounter histories:

```
5359 ##  
5360 ## Simulate the data and extract the required objects  
5361 ##  
5362 > data <- simSCR0(discard0=TRUE,rnd=2013)  
5363 > y <- data$Y  
5364 > nind <- nrow(y)  
5365 > X <- data$traplocs  
5366 > K <- data$K  
5367 > J <- nrow(X)  
5368 > xlim <- data$xlim  
5369 > ylim <- data$ylim
```

5370 After harvesting the data we augment the data matrix  $y$  with  $M - n$  all-zero encounter  
 5371 histories, and create starting values for the variables  $z_i$  and also the activity centers  $s_i$   
 5372 of which, for each, we require  $M$  values. One thing to take care of in using the **BUGS**  
 5373 engines is the starting values for the activity centers. It is usually helpful to start the  $s_i$   
 5374 for each observed individual at or near the trap(s) it was captured. All of this happens as  
 5375 follows:

```
5376 ## Data augmentation  
5377 > M <- 200  
5378 > y <- rbind(y,matrix(0,nrow=M-nind,ncol=ncol(y)))  
5379 > z <- c(rep(1,nind),rep(0,M-nind))  
5380  
5381 ## Starting values for s  
5382 > sst <- cbind(runif(M,xlim[1],xlim[2]),runif(M,ylim[1],ylim[2]))  
5383 > for(i in 1:nind){  
5384   sst[i,1] <- mean( X[y[i,]>0,1] )  
5385   sst[i,2] <- mean( X[y[i,]>0,2] )  
5386 }
```

5387 Next, we write out the **BUGS** model specification and save it to an external file  
 5388 called `SCR0b.txt`. The model specification now includes  $M$  encounter histories including  
 5389 the augmented potential individuals, the data augmentation parameters  $z_i$ , and the data  
 5390 augmentation parameter  $\psi$ :

```
5391 > cat("model{  
5392   alpha0 ~ dnorm(0,.1)  
5393   logit(p0) <- alpha0
```

```

5395 alpha1 ~ dnorm(0,.1)
5396 sigma <- sqrt(1/(2*alpha1))
5397 psi ~ dunif(0,1)

5398
5399 for(i in 1:M){
  5400   z[i] ~ dbern(psi)
  5401   s[i,1] ~ dunif(xlim[1],xlim[2])
  5402   s[i,2] ~ dunif(ylim[1],ylim[2])
  5403   for(j in 1:J){
    5404     d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
    5405     y[i,j] ~ dbin(p[i,j],K)
    5406     p[i,j] <- z[i]*p0*exp(- alpha1*d[i,j]*d[i,j])
  5407   }
  5408 }
5409 N <- sum(z[])
5410 D <- N/64
5411 }
5412 ",file = "SCR0b.txt")

```

5413     The remainder of the code for bundling the data, creating initial values and executing **WinBUGS** looks much the same as before except with more or differently named arguments:

```

5416 > data <- list (y=y, X=X, K=K, M=M, J=J, xlim=xlim, ylim=ylim)
5417 > inits <- function(){
5418   list (alpha0=rnorm(1,-4,.4), alpha1=runif(1,1,2), s=sst, z=z)
5419 }
5420
5421 > library(R2WinBUGS)
5422 > parameters <- c("alpha0","alpha1","sigma","N","D")
5423 > out <- bugs (data, inits, parameters, "SCR0b.txt", n.thin=1,n.chains=3,
5424   n.burnin=1000,n.iter=2000,debug=TRUE,working.dir=getwd())

```

5425     Note the differences in this new **WinBUGS** model with that appearing in the known-  
5426  $N$  version – there are not many! The loop over individuals goes up to  $M$  now, and there is a  
5427 model component for the DA variables  $z$ . We are also computing some derived parameters:  
5428 population size  $N(\mathcal{S})$  is computed by summing up all of the data augmentation variables  
5429  $z_i$  (as we've done previously in Chapt. 4) and density,  $D$ , is also a derived parameter,  
5430 being a function of  $N$ . The input data has changed slightly too, as the augmented data  
5431 set has more rows to include excess all-zero encounter histories. Previously we knew that  
5432  $N = 100$  but in this analysis we pretend not to know  $N$ , but think that  $N = 200$  is a  
5433 good upper bound. This analysis can be run directly using the **SCR0bayes** function once  
5434 the **scrbook** package is loaded, by issuing the following commands:

```

5435 > library(scrbook)
5436 > data <- simSCR0(discard0=TRUE,rnd=2013)
5437 > out1 <- SCR0bayes(data,M=200,engine="winbugs",ni=2000,nb=1000)

```

5438 Summarizing the output from **WinBUGS** produces:

```

5439 > print(out1,digits=2)
5440 Inference for Bugs model at "SCR0b.txt", fit using WinBUGS,
5441   3 chains, each with 2000 iterations (first 1000 discarded)
5442   n.sims = 3000 iterations saved
5443     mean    sd   2.5%   25%   50%   75% 97.5% Rhat n.eff
5444 alpha0   -2.57  0.23  -3.04  -2.72  -2.56  -2.41  -2.15 1.01   320
5445 alpha1    2.46  0.42   1.63   2.16   2.46   2.73   3.33 1.02   120
5446 sigma     0.46  0.04   0.39   0.43   0.45   0.48   0.55 1.02   120
5447 N        113.62 15.73  86.00 102.00 113.00 124.00 147.00 1.01   260
5448 D        1.78  0.25   1.34   1.59   1.77   1.94   2.30 1.01   260
5449 deviance 302.60 23.67 261.19 285.47 301.50 317.90 354.91 1.00 1400
5450
5451 [...some output deleted...]
5452

```

5453     The **Rhat** statistic (discussed in Secs. 3.5.2 and 17.6.4) for this analysis indicates  
5454     satisfactory convergence. We see that the estimated parameters ( $\alpha_0$  and  $\alpha_1$ ) are comparable  
5455     to the previous results obtained for the known- $N$  case, and also not too different  
5456     from the data-generating values. The posterior of  $N$  overlaps the data-generating value  
5457     substantially.

#### 5458     **Use of other BUGS engines: JAGS**

5459     There are two other popular **BUGS** engines in widespread use: **OpenBUGS** (Thomas  
5460     et al., 2006) and **JAGS** (Plummer, 2003). Both of these are easily called from **R**. **Open-**  
5461     **BUGS** can be used instead of **WinBUGS** by changing the package option in the **bugs**  
5462     call to `package='OpenBUGS'`. **JAGS** can be called using the function **jags()** in package  
5463     **R2jags** which has nearly the same arguments as **bugs()**. Or, it can be executed from the  
5464     **R** package **rjags** (Plummer, 2011) which has a slightly different implementation that we  
5465     demonstrate here as we reanalyze the simulated data set in the previous section (note:  
5466     the same **R** commands are used to generate the data and package the data, inits and  
5467     parameters to monitor). The function **jags.model** is used to initialize the model and run  
5468     the MCMC algorithm for an adaptive period during which tuning of the MCMC algorithm  
5469     might take place. These samples cannot be used for inference. Then the Markov chains  
5470     are updated using **coda.samples()** to obtain posterior samples for analysis, as follows:

```

5471 > jinit <- jags.model("SCR0b.txt", data=data, inits=inits,
5472           n.chains=3, n.adapt=1000)
5473 > jout <- coda.samples(jinit, parameters, n.iter=1000, thin=1)

```

5474     These commands can be executed using the function **SCR0bayes** provided with the **R**  
5475     package **scrbook**. Hobbs (2011) provides a good introduction to ecological modeling with  
5476     **JAGS** which we recommend.

#### 5477     **5.7.2 Implied home range area**

5478     Here we apply the method described in Sec. 5.4 to compute the effective home range  
5479     area under different encounter probability models fit to simulated data. We simulated a  
5480     data set from the Gaussian kernel model as in Sec. 5.7 and then we fitted 4 models to it:

**Table 5.2.** Posterior mean of model parameters for 4 different models fitted to a single simulated data set, and the effective home range area under each detection model.

	Gaussian	Cloglog	Exponential	Logit
N	113.62	114.16	119.69	118.29
D	1.78	1.78	1.87	1.85
$\alpha_0$	-2.57	-2.60	-1.51	-0.47
$\alpha_1$	2.46	2.56	3.59	3.86
hra	3.85	3.78	5.51	2.64

(1) the true data-generating Gaussian encounter probability model; (2) the “hazard” or complementary log-log link model (Eq. 5.4.2); (3) the negative exponential model and (4) the logit model (Eq. 5.2.2). We modified the function `SCR0bayes` for this purpose which you should be able to do with little difficulty. We fit each model to the same simulated data set using **WinBUGS**, based only on 1000 post-burn-in samples and 3 chains, which produced the posterior summaries given in Table 5.2. The main thing we see is that, while the implied home range area can vary substantially, there are smaller differences in the estimated  $N$  and hence  $D$ .

### 5.7.3 Realized and expected density

In Bayesian analysis of the SCR model, we estimate a parameter  $N$  which is the size of the population for the prescribed state-space (presumably the state-space is defined so as to be relevant to where our traps were located, so  $N$  can be thought of as the size of the sampled population). In the context of Efford and Fewster (2012) this is the *realized* population size. Conversely, sometimes we see estimates of *expected* population size reported, which are estimates of  $\mathbb{E}(N)$ , the expected size of some hypothetical, unspecified population. Usually the distinction between realized and expected population size is not made in SCR models, because almost everyone only cares about actual populations – and their realized population size.

If you do likelihood analysis of SCR models, then the distinction between realized and expected is often discussed by whether the estimator is “conditional on  $N$ ” (realized) or not (expected). The naming arises because in obtaining the MLE of  $N$ , its properties are evaluated *conditional* on  $N$  – in particular, if the estimator is unbiased then  $\mathbb{E}(\hat{N}|N) = N$  and  $\text{Var}(\hat{N}|N) = \tilde{\sigma}_{\hat{N}}^2$  is the sampling variance. This does not conform to any concept or quantity that is relevant to Bayesian inference. If we care about  $N$  for the population that we sampled it is understood to be a realization of a random variable, but the relevance of “conditional on  $N$ ” is hard to see. Bayesian analysis will provide a prediction of  $N$  that is based on the posterior  $[N|y, \theta]$  – which is certainly *not* conditional on  $N$ .

There is a third type of inference objective that is relevant in practice and that is prediction of  $N$  for a population that was not sampled – i.e., a “new” population. To elaborate on this, consider a situation in which we are concerned about the tiger population in 2 distinct reserves in India. We do a camera trapping study on one of the reserves to estimate  $N_1$  and we think the reserves are similar and homogeneous so we’re willing to apply a density estimate based on  $N_1$  to the 2nd reserve. For the 2nd reserve, do we want a prediction of the realized population size,  $N_2$ , or do we want an estimate of its expected

5515 value? We believe the former is the proper quantity for inference about the population  
 5516 size in the 2nd reserve. An estimate of  $N_2$  should include the uncertainty with which  
 5517 the mean is estimated (from reserve 1) and it should also include “process variation” for  
 5518 making the prediction of the latent variable  $N_2$ .

5519 As a practical matter, to do a Bayesian analysis of this you could just define the state-  
 5520 space to be the union of the two state-spaces, increase  $M$  so that the posterior of the  
 5521 total population size is not truncated, and then have MCMC generate a posterior sample  
 5522 of individuals on the joint state-space. You can tally-up the ones that are on  $\mathcal{S}_2$  as an  
 5523 estimate of  $N_2$ . Alternatively, we can define  $\mu = \psi M/A_1$  and then simulate posterior  
 5524 samples of  $N_s \sim \text{Binomial}(M, \mu A_2/M)$  for the new state-space area,  $A_2$ .

5525 To carry out a classical likelihood analysis of this 2nd type of problem, what should we  
 5526 do? The argument for making a prediction of a new value of  $N$  would go something like  
 5527 this: If you obtain an MLE of  $N$ , say  $\hat{N}$ , then the inference procedure tells us the variance  
 5528 of this *conditional* on  $N$ . i.e.,  $\text{Var}(\hat{N}|N)$ . This is fine, if we care about the specific value  
 5529 of  $N$  that generated our data set. However, if we don’t care about the specific one in  
 5530 question then we want to “uncondition” on  $N$  to introduce a new variance component.  
 5531 Law of total variance says:

$$\text{Var}(\hat{N}) = \mathbb{E}[\text{Var}(\hat{N}|N)] + \text{Var}[\mathbb{E}(\hat{N}|N)]$$

5532 If  $\hat{N}$  is unbiased then we say the unconditional variance is

$$\text{Var}(\hat{N}) = \sigma_{\hat{N}}^2 + \text{Var}(N)$$

5533 The first part is estimation error and the 2nd component is the “process variance.” If  
 5534 you do Bayesian analysis, then you don’t have to worry too much about how to compute  
 5535 variances properly. You decide if you care about  $N$ , or its expected value, or predictions  
 5536 of some “new”  $N$ , and you tabulate the correct posterior distribution from your MCMC  
 5537 output.

5538 The considerations for estimating density are the same. Density can be  $N/A$  where  
 5539  $N$  is the realized population, which we understand it to be unless we put an expectation  
 5540 operator around the  $N$  like  $\mathbb{E}(N)/A$ . Classically, density is thought of as being defined as  
 5541 the expected value of  $N$  but this might not always be meaningful because the context of  
 5542 whether we mean realized density, of an actual population, or expected density for some  
 5543 hypothetical unspecified population, should matter. The formula for obtaining “expected  
 5544 density” is slightly different depending on whether we assume  $N$  has a Poisson distribution  
 5545 or whether we assume a binomial distribution (under data augmentation). In the latter  
 5546 case  $\psi$  is related to the point process intensity (see Chapt. 11) in the sense that, under  
 5547 the binomial prior:

$$\mathbb{E}(N) = M \times \psi$$

5548 so, what we think of as “density”,  $D$ , is  $D = M\psi/A$ . Under the Poisson point process  
 5549 model we have:

$$\mathbb{E}(N) = D \times A.$$

5550 In summary, there are 3 basic inference problems that relate to estimating population  
 5551 size (or density):

5552 (1) What is the value of  $N$  for some population that was sampled. This is what Efford  
 5553 and Fewster call “realized N” In general, we want the uncertainty to reflect having to  
 5554 estimate  $n_0$ , the part of the population not seen.

- 5555 (2) We need to estimate  $N$  for some population that we didn't sample but it is "similar"  
5556 to the population that we have information on. In this case, we have to account for  
5557 both variation in having to estimate parameters of the distribution of  $N$  and we have  
5558 to account for process variation in  $N$  (i.e., due to the stochastic model of  $N$ ).  
5559 (3) In some extremely limited cases we might care about estimating the expected value of  
5560  $N$ ,  $\mathbb{E}(N)$ . This is only useful as a hypothetical statement that we might use, e.g., if we  
5561 were to establish a new million ha refuge somewhere, then we might say its expected  
5562 population size is 200 tigers.

## 5.8 THE CORE SCR ASSUMPTIONS

5563 It's always a good idea to sit down and reflect on the meaning of any particular model,  
5564 its various assumptions, and what they mean in a specific context. From the statistician's  
5565 point of view, the basic assumption, the omnibus assumption, as in all of statistics, and  
5566 for every statistical model, is that "the model is correctly specified". So, naturally, that  
5567 precludes everything that isn't explicitly addressed by the model. To point this out to  
5568 someone seems to cause a lot of anxiety, so we enumerate here what we think are the most  
5569 important statistical assumptions of the basic SCR0 model:

- 5570 • **Demographic closure.** The model does not allow for demographic processes. There  
5571 is no recruitment or entry into the sampled population. There is no mortality or exit  
5572 from the sampled population.
- 5573 • **Geographic closure.** We assume no permanent emigration or immigration from the  
5574 state-space. However, we allow for "temporary" movements around the state-space  
5575 and variable exposure to encounter as a result. The whole point of SCR models is to  
5576 accommodate this dynamic. In ordinary capture-recapture models we have to assume  
5577 geographic closure to interpret  $N$  in a meaningful way.
- 5578 • **Activity centers are randomly distributed.** That is, uniformity and independence  
5579 of the underlying point process  $s_1, \dots, s_N$  (see next section).
- 5580 • **Detection is a function of distance.** A detection model that describes how encounter  
5581 probability declines as a function of distance from an individual's home range center.
- 5582 • **Independence of encounters** among individuals. Encounter of any individual is  
5583 independent of encounter of each other individual.
- 5584 • **Independence of encounters** of the same individual. Encounter of an individual  
5585 in any trap is independent of its encounter in any other trap, and subsequent sample  
5586 occasion.

5587 It's easy to get worried and question the whole SCR enterprise just on the grounds that  
5588 these assumptions combine to form such a simplistic model, one that surely can't describe  
5589 the complexity of real populations. On this sentiment, a few points are worth making.  
5590 First, you don't have inherently fewer assumptions by using an ordinary capture-recapture  
5591 model but, rather, the SCR model relaxes a number of important assumptions compared  
5592 to the non-spatial counterpart. For one, here, we're not assuming that  $p$  is constant for all  
5593 individuals but rather that  $p$  varies substantially as a matter of the spatial juxtaposition of  
5594 individuals with traps. So maybe the manner in which  $p$  varies isn't quite right, but that's  
5595 not an argument that supports doing less modeling. Fundamentally a distance-based  
5596 model for  $p$  has some basic biological justification in virtually every capture-recapture

study. Secondly, for some of these core assumptions such as uniformity, and independence of individuals and of encounters, we expect a fair amount of robustness to departures. They function primarily to allow us to build a model and an estimation scheme and we don't usually think they represent real populations (of course, no model does!). Third, we can extend these assumptions in many different ways and we do that to varying extents in this book, and more work remains to be done in this regard. Forth, we can also evaluate the reasonableness of the assumptions formally in some cases using standard methods of assessing model fit (Chapt. 8).

Finally, we return back to our sentiment about the omnibus assumptions which is that the model is properly specified. This precludes *everything* that isn't in the model. Sometimes you see in capture-recapture literature statements like "we assume no marks are lost", "marks are correctly identified" and similar things. We might as well also assume that, a shopping mall is not built, or a meteor does not crash down into our study area, the sun does not go super-nova, and so forth. Our point is that we should separate statistical assumptions about model parameters or aspects of the probability model from what are essentially logistical or operational assumptions about how we interpret our data, or based on our ability to conduct the study. It is pointless to enumerate all of the possible explanations for apparent *departures*, because there are an infinity of such cases.

## 5.9 WOLVERINE CAMERA TRAPPING STUDY

We provide an illustration of some of the concepts we've introduced previously in this chapter by analyzing data from a camera trapping data from a study of wolverines *Gulo gulo* (Magoun et al., 2011; Royle et al., 2011b). The study took place in SE Alaska (Fig. 5.4) where 37 cameras were operational for variable periods of time (min = 5 days, max = 108 days, median = 45 days). A consequence of this is that the number of sampling occasions,  $K$ , is variable for each camera. Thus, we must provide a vector of sample sizes as data to **BUGS** and modify the model specification in Sec. 5.7 accordingly.

### 5.9.1 Practical data organization

To carry out an analysis of these data, we require the matrix of trap coordinates and the encounter history data. We usually store data in 2 distinct data files which contain all the information needed for an analysis. These files are

- The encounter data file (EDF) containing a record of which traps and when each individual encounter occurred.
- The trap deployment file (TDF) which contains the coordinates of each trap, along with information indicating which sample occasions each trap was operating.

**Encounter Data File (EDF)** – We store the encounter data in the an efficient file format which is easily manipulated in **R** and easy to create in Excel and other spreadsheets which are widely used for data management. The file structure is a simple matrix with 4 columns, those being: (1) **session ID**: the trap *session* which usually corresponds to a year or a primary period in the context of a Robust Design situation, but it could also correspond to a distinct spatial unit (see Sec. 6.5.4 and Chapt. 14). For a single-year study (as considered here) this should be an integer that is the same for all records;



**Figure 5.4.** Wolverine camera trap locations (black dots) from a study that took place in SE Alaska. See Magoun et al. (2011) for details.

(2) **individual ID:** the individual identity, being an integer from 1 to  $n$  (repeated for multiple captures of the same individual) indicating which individual the record (row) of the matrix belongs to; (3) **occasion ID:** The integer sample occasion which generated the record, and (4) **trap ID:** the trap identity, an integer from 1 to  $J$ , the number of traps. The structure of the EDF is the same as used in the **secr** package (Efford, 2011a) and similar to that used in the **SPACECAP** (Gopalaswamy et al., 2012a), and **SCRbayes** (Russell et al., 2012) packages, both of which have a 3-column format (**trapID**, **indID**, **sampID**). We note that the naming of the columns is irrelevant as far as anything we do in this book, although **secr** and other software may have requirements on variable naming.

To illustrate this format, the wolverine data are available in the package **scrbook** by typing:

```
5648 > data(wolverine)
```

5649 which contains a list having elements `wcaps` (the EDF) and `wtraps` (the TDF). We see  
 5650 that `wcaps` has 115 rows, each representing a unique encounter event including the trap  
 5651 identity, the individual identity and the sample occasion index (`sample`). The first 5 rows  
 5652 of `wcaps` are:

```
5653 > wolverine$wcaps[1:5,]
5654   year individual day trap
5655   [1,]    1        2 127    1
5656   [2,]    1        2 128    1
5657   [3,]    1        2 129    1
5658   [4,]    1       18 130    1
5659   [5,]    1        3 106    2
```

5660 The 1st column here, labeled `year`, is an integer indicating the year or session of the  
 5661 encounter. All these data come from a single year (2008) and so `year` is set to 1. Variable  
 5662 `individual` is an integer identity of each individual captured, `day` is the sample occasion of  
 5663 capture (in this case, the sample occasions correspond to days), and `trap` is the integer trap  
 5664 identity. The variable `trapid` will have to correspond to the row of a matrix containing  
 5665 the trap coordinates - in this case the TDF file `wtraps` which we describe further below.

5666 Note that the information provided in this encounter data file `wcaps` does not represent  
 5667 a completely informative summary of the data. For example, if no individuals were  
 5668 captured in a certain trap or during a certain period, then this compact data format will  
 5669 have no record. Thus we will need to know  $J$ , the number of traps, and  $K$ , the number of  
 5670 sample occasions when reformatting this SCR data format into a 2-d encounter frequency  
 5671 matrix or 3-d array. In addition, the encounter data file does not provide information  
 5672 about which periods each trap was operated. This additional information is also necessary  
 5673 as the trap-specific sample sizes must be passed to **BUGS** as data. We provide this  
 5674 information along with trap coordinates, in the “trap deployment file” (TDF) which is  
 5675 described below.

5676 For our purposes, we need to convert the `wcaps` file into the  $n \times J$  array of binomial  
 5677 encounter frequencies, although more general models might require an encounter-history  
 5678 formulation of the model which requires a full 3-d array. To obtain our encounter frequency  
 5679 matrix, we do this the hard way by first converting the encounter data file into a 3-d array  
 5680 and then summarize to trap totals. We have a handy function `SCR23darray` which takes  
 5681 the compact encounter data file, and converts it to a 3-d array, and then we use the **R**  
 5682 function `apply` to summarize over the sample occasion dimension (by convention here,  
 5683 this is the 2nd dimension). To apply this to the wolverine data in order to compute the  
 5684 3-d array we do this:

```
5685 > y3d <- SCR23darray(wolverine$wcaps,wolverine$wtraps)
5686 > y <- apply(y3d,c(1,2),sum)
```

5687 See the help file for more information on `SCR23darray`. The 3-d array is necessary to  
 5688 fit certain types of models (e.g., behavioral response) and this is why we sometimes will  
 5689 require this maximally informative 3-d data format but, here, we analyze the summarized  
 5690 data.

5691 **Trap Deployment File (TDF)** – The other important information needed to fit SCR  
 5692 models is the “trap deployment file” (TDF) which provides additional information not

5693 contained in the encounter data file. The traps file has  $K + 3$  columns. The first column is  
 5694 assumed to be a trap identifier, columns 2 and 3 are the easting and northing coordinates  
 5695 (assumed to be in a Euclidean coordinate system), and columns 4 to  $K + 3$  are binary  
 5696 indicators of whether each trap was operational during each sample occasion. The first 10  
 5697 rows (out of 37) and 10 columns (out of 167) of the trap deployment file for the wolverine  
 5698 data are shown as follows:

```
5699 > wolverine$wtraps[1:10,1:10]
5700
5701      Easting Northing 1 2 3 4 5 6 7 8
5702 1    632538  6316012 0 0 0 0 0 0 0 0
5703 2    634822  6316568 1 1 1 1 1 1 1 1
5704 3    638455  6309781 0 0 0 0 0 0 0 0
5705 4    634649  6320016 0 0 0 0 0 0 0 0
5706 5    637738  6313994 0 0 0 0 0 0 0 0
5707 6    625278  6318386 0 0 0 0 0 0 0 0
5708 7    631690  6325157 0 0 0 0 0 0 0 0
5709 8    632631  6316609 0 0 0 0 0 0 0 0
5710 9    631374  6331273 0 0 0 0 0 0 0 0
5711 10   634068  6328575 0 0 0 0 0 0 0 0
```

5712 This tells us that trap 2 was operated during occasions (days) 1-7 but the other traps  
 5713 were not operational during those periods. It is extremely important to recognize that  
 5714 each trap was operated for a variable period of time and thus the binomial “sample size”  
 5715 is different for each, and this needs to be accounted for in the **BUGS** model specification.  
 5716 To compute the vector of sample sizes  $K$ , and extract the trap locations, we do this:

```
5717 > traps <- wolverine$wtraps
5718 > traplocs <- traps[,1:2]
5719 > K <- apply(traps[,3:ncol(traps)],1,sum)
```

5720 This results in a matrix `traplocs` which contains the coordinates of each trap and a vector  
 5721 `K` containing the number of days that each trap was operational. We now have all the  
 5722 information required to fit a basic SCR model in **BUGS**.

5723 Summarizing the data for the wolverine study, we see that 21 unique individuals were  
 5724 captured a total of 115 times. Most individuals were captured 1-6 times, with 4, 1, 4, 3, 1,  
 5725 and 2 individuals captured 1-6 times, respectively. In addition, 1 individual was captured  
 5726 each 8 and 14 times and 2 individuals each were captured 10 and 13 times. The number  
 5727 of unique traps that captured a particular individual ranged from 1-6, with 5, 10, 3, 1, 1,  
 5728 and 1 individual captured in each of 1 to 6 different traps, respectively, for a total of 50  
 5729 unique wolverine-trap encounters. These numbers might be hard to get your mind around  
 5730 whereas some tabular summary is often more convenient. For that it seems natural to  
 5731 tabulate individuals by trap and total encounter frequencies. The spatial information in  
 5732 SCR data is based on multi-trap captures, and so, it is informative to understand how  
 5733 many unique traps each individual is captured in, and the total number of encounters.  
 5734 For the wolverine data, we reproduce Table 1 from Royle et al. (2011b) as Table 5.3.

**Table 5.3.** Individual frequencies of capture for wolverines captured in camera traps in Southeast Alaska in 2008. Rows index unique traps of capture for each individual and columns represent total number of captures (e.g., we captured 4 individuals 1 time, necessarily in only 1 trap; we captured 3 individuals 3 times but in 2 different traps).

No. of traps	No. of captures									
	1	2	3	4	5	6	8	10	13	14
1	4	1	0	0	0	0	0	0	0	0
2	0	0	3	2	0	2	1	2	0	0
3	0	0	1	1	0	0	0	0	0	1
4	0	0	0	0	0	0	0	0	1	0
5	0	0	0	0	1	0	0	0	0	0
6	0	0	0	0	0	0	0	0	1	0

### 5.9.2 Fitting the model in WinBUGS

Here we fit the simplest SCR model with the Gaussian encounter probability model, although we revisit these data and fit additional models in later chapters. Model SCR0 is summarized by the following 4 elements:

- (1)  $y_{ij} | \mathbf{s}_i \sim \text{Binomial}(K, z_i p_{ij})$
- (2)  $p_{ij} = p_0 \exp(-\alpha_1 ||\mathbf{x}_j - \mathbf{s}_i||^2)$
- (3)  $\mathbf{s}_i \sim \text{Uniform}(\mathcal{S})$
- (4)  $z_i \sim \text{Bernoulli}(\psi)$

We assume customary flat priors on the structural (hyper-) parameters of the model,  $\alpha_0 = \text{logit}(p_0)$ ,  $\alpha_1$  and  $\psi$ .

It remains to define the state-space  $\mathcal{S}$ . For this, we nested the trap array (Fig. 5.4) in a rectangular state-space extending 20 km beyond the traps in each cardinal direction. We scaled the coordinate system so that a unit distance was equal to 10 km, producing a rectangular state-space of dimension  $9.88 \times 10.5$  units ( $\text{area} = 10374 \text{ km}^2$ ) within which the trap array was nested. As a general rule, we recommend scaling the state-space so that it is defined near the origin  $(x, y) = (0, 0)$ . While the scaling of the coordinate system is theoretically irrelevant, a poorly scaled coordinate system can produce Markov chains that mix poorly. The buffer of the state space should be large enough so that individuals beyond the state-space boundary are not likely to be encountered (Sec. 5.3.1). To evaluate this, we fit models for various choices of a rectangular state-space based on buffers from 1.0 to 5.0 units (10 km to 50 km). In the **R** package **scrbook** we provide a function **wolvSCR0** which will fit model SCR0. For example, to fit the model in **WinBUGS** using data augmentation with  $M = 300$  potential individuals, using 3 Markov chains each of 12000 total iterations, discarding the first 2000 as burn-in, we execute the following **R** commands:

```

5760 > library(scrbook)
5761 > data(wolverine)
5762 > traps <- wolverine$wtraps
5763 > y3d <- SCR23darray(wolverine$wcaps,wolverine$wtraps)
5764 > wolv <- wolvSCR0(y3d,traps,nb=2000,ni=12000,buffer=1,M=300)

```

**Table 5.4.** Posterior summaries of SCR model parameters for the wolverine camera trapping data from SE Alaska, using state-space buffers from 10 up to 50 km. Each analysis was based on 3 chains, 12000 iterations, 2000 burn-in, for a total of 30000 posterior samples.

Buffer	$\sigma$			N			D		
	Mean	SD	n.eff	Mean	SD	n.eff	Mean	SD	n.eff
10	0.65	0.06	1800	39.63	6.70	7100	5.97	1.00	7100
15	0.64	0.06	510	48.77	9.19	3300	5.78	1.09	3300
20	0.64	0.06	1200	59.84	11.89	20000	5.77	1.15	20000
25	0.64	0.05	3600	72.40	14.72	2700	5.79	1.18	2700
30	0.63	0.05	5600	86.42	17.98	3900	5.82	1.21	3900
35	0.63	0.05	4500	101.79	21.54	30000	5.85	1.24	30000
40	0.64	0.05	410	118.05	26.17	410	5.87	1.30	450
45	0.64	0.05	10000	134.43	28.68	3300	5.83	1.24	3300
50	0.63	0.05	4700	151.61	31.65	3400	5.79	1.21	3400

**Table 5.5.** Posterior summaries of SCR model parameters for the wolverine camera trapping data from SE Alaska. The model was run with the trap array centered in a state-space with a 20 km rectangular buffer.

Parameter	Mean	SD	2.5%	25%	50%	75%	97.5%	Rhat
$N$	59.84	11.89	40.00	51.00	59.00	67.00	86.00	1
$D$	5.77	1.15	3.86	4.92	5.69	6.46	8.29	1
$\alpha_1$	1.26	0.21	0.87	1.11	1.25	1.40	1.71	1
$p_0$	0.06	0.01	0.04	0.05	0.06	0.06	0.08	1
$\sigma$	0.64	0.06	0.54	0.60	0.63	0.67	0.76	1
$\psi$	0.20	0.05	0.12	0.17	0.20	0.23	0.30	1

5765 The argument `buffer` determines the buffer size of the state-space in the scaled units  
 5766 (i.e., 10 km). Note that this analysis takes between 1-2 hours on many machines (in 2013)  
 5767 so we recommend testing it with lower values of  $M$  and fewer iterations. The posterior  
 5768 summaries are shown in Table 5.9.2.

### 5769 5.9.3 Summary of the wolverine analysis

5770 We see that the estimated density is roughly consistent as we increase the state-space  
 5771 buffer from 15 to 55 km. We do note that the data augmentation parameter  $\psi$  (and,  
 5772 correspondingly,  $N$ ) increase with the size of the state space in accordance with the deter-  
 5773 ministic relationship  $N = D * A$ . However, density is more or less constant as we increase  
 5774 the size of the state-space beyond a certain point. For the 10 km state-space buffer, we see  
 5775 a slight effect on the posterior distribution of  $D$  because the state-space is not sufficiently  
 5776 large. The full results from the analysis based on 20 km state-space buffer are given in  
 5777 Table 5.5.

5778 Our point estimate of wolverine density from this study, using the posterior mean from  
 5779 the state-space based on the 20 km buffer, is approximately 5.77 individuals/1000 km<sup>2</sup>

5780 with a 95% posterior interval of [3.86, 8.29]. Density is estimated imprecisely which might  
 5781 not be surprising given the low sample size ( $n = 21$  individuals!). This seems to be a  
 5782 basic feature of carnivore studies although it should not (in our view) preclude the study  
 5783 of their populations by capture-recapture nor attempts to estimate density or vital rates.

5784 It is worth thinking about this model, and these estimates, computed under a rect-  
 5785 angular state space roughly centered over the trapping array (Fig. 5.4). Does it make  
 5786 sense to define the state-space to include, for example, ocean? What are the possible  
 5787 consequences of this? What can we do about it? There's no reason at all that the state  
 5788 space has to be a regular polygon – we defined it as such here strictly for convenience and  
 5789 for ease of implementation in **WinBUGS** where it enables us to specify the prior for the  
 5790 activity centers as uniform priors for each coordinate. While it would be possible to define  
 5791 a more realistic state-space using some general polygon GIS coverage, it might take some  
 5792 effort to implement that in the **BUGS** language but it is not difficult to devise custom  
 5793 MCMC algorithms to do that (see Chapt. 17). Alternatively, we recommend using a  
 5794 discrete representation of the state-space – i.e., approximate  $\mathcal{S}$  by a grid of  $G$  points. We  
 5795 discuss this in Sec. 5.10.

#### 5796 5.9.4 Wolverine space usage

5797 The parameter  $\alpha_1$  is related to the home range radius (Sec. 5.4). For the Gaussian model  
 5798 we interpret the scale parameter  $\sigma$ , related to  $\alpha_1$  by  $\alpha_1 = 1/(2\sigma^2)$ , as the radius of a  
 5799 bivariate normal model of space usage. In this case  $\sigma = 0.64$  standardized units (10 km),  
 5800 which corresponds to  $0.64 \times 10 = 6.4$  km. It can be argued then that 95% of space used  
 5801 by an individual is within  $6.4 \times \sqrt{5.99} = 15.66$  km of the home range center. The effective  
 5802 “home range area” is then the area of this circle, which is  $\pi \times 15.66^2 = 770.4$  km<sup>2</sup>. Using  
 5803 our handy function **hra** we do this:

```
5804 hra(pGauss1,parms=c(-2,1/(2*.64*.64)),xlim=c(-1,7),ylim=c(-1,7))
5805
5806 [1] 7.731408
```

5807 which is in units of 100 km<sup>2</sup>, so 773.1. The difference in this case is due to numerical  
 5808 approximation of our all-purpose tool **hra**. This home range size is relatively huge for  
 5809 measured home ranges, which range between 100 and 535 km<sup>2</sup> (Whitman et al., 1986).

5810 Royle et al. (2011b) reported estimates for  $\sigma$  in the range 6.3 – 9.8 km depending on  
 5811 the model, which isn't too different than here<sup>1</sup>. However, these estimates are larger than  
 5812 the typical home range sizes suggested in the literature. One possible explanation is that  
 5813 if a wolverine is using traps as a way to get yummy chicken, so it's moving from trap to  
 5814 trap instead of adhering to “normal” space usage patterns, then the implied home range  
 5815 size might not be worth much biologically. Thus, interpretation of detection models in  
 5816 terms of home range area depends on some additional context or assumptions, such as  
 5817 that traps don't effect individual space usage patterns. As such, we caution against direct

---

<sup>1</sup> Royle et al. (2011b) expressed the model as  $\text{cloglog}(p_{ij}) = \alpha_0 - (1/\sigma^2) * d_{ij}^2$ , but the estimates of  $\sigma$  reported in their Table 2 are actually based on the model according to  $\text{cloglog}(p_{ij}) = \alpha_0 - \frac{1}{2\sigma^2} * d_{ij}^2$ , and so the estimates of  $\sigma$  they report in units of km are consistent to what we report here except based on the complementary log-log (Gaussian hazard) model, instead of the Gaussian encounter probability model.

5818 biological interpretations of home range area based on  $\sigma$ , although SCR models can be  
5819 extended to handle more general, non-Euclidean, patterns of space usage. See Chaps. 12  
5820 and 13.

5821 We can calibrate the desired size of the state-space by looking at the estimated home  
5822 range radius of the species. We should target a buffer of width 2 to  $3 \times \sigma$  in order that  
5823 the probability of encountering an individual is very close to 0 beyond the prescribed  
5824 state-space. Essentially, by specifying a state-space, we're setting  $p = 0$  for individuals  
5825 beyond the prescribed state-space. For the wolverine data, with  $\sigma$  in the range of 6-9 km,  
5826 a state-space buffer of 20 km is sufficiently large.

## 5.10 USING A DISCRETE HABITAT MASK

5827 The SCR model developed previously in this chapter assumes that individual activity  
5828 centers are distributed uniformly over the prescribed state-space. Clearly this will not  
5829 always be a reasonable assumption. In Chapt. 11, we develop models that allow explicitly  
5830 for non-uniformity of the activity centers by modeling covariate effects on density. A  
5831 simplistic method of affecting the distribution of activity centers, which we address here,  
5832 is to modify the shape and organization of the state-space explicitly. For example, we  
5833 might be able to classify the state-space into distinct blocks of habitat and non-habitat.  
5834 In that case we can remove the non-habitat from the state-space and assume uniformity of  
5835 the activity centers over the remaining portions judged to be suitable habitat. There are  
5836 several ways to approach this: We can use a grid of points to represent the state-space, i.e.,  
5837 by the set of coordinates  $s_1, \dots, s_G$ , and assign equal probabilities to each possible value.  
5838 Alternatively, we can retain the continuous formulation of the state-space but attempt  
5839 to describe constraints analytically, or we can use polygon clipping methods to enforce  
5840 constraints on the state-space in the MCMC analysis. We focus here on the formulation of  
5841 the basic SCR model in terms of a discrete state-space but in Chapt. 17 we demonstrate  
5842 the latter approach based on using polygon operations to define an irregular state-space.  
5843 Use of a discrete state-space can be computationally expensive in **WinBUGS**. That said,  
5844 it isn't too difficult to perform the MCMC calculations in **R** (discussed in Chapt. 17).  
5845 The **R** package **SPACECAP** (Gopalaswamy et al., 2012a) arose from the **R** implementation  
5846 of the SCR model in Royle et al. (2009a).

5847 While clipping out non-habitat seems like a good idea, we think investigators should  
5848 go about this very cautiously. We might prefer to do it when non-habitat represents a  
5849 clear-cut restriction on the state-space such as a reserve boundary or a lake, ocean or  
5850 river. But, having the capability to do this also causes people to start defining "habitat"  
5851 vs. "non-habitat" based on their understanding of the system whereas it can't be known  
5852 whether the animal being studied has the same understanding. Moreover, differentiating  
5853 the landscape by habitat or habitat quality must affect the geometry and morphology of  
5854 home ranges (see Chapt. 13) much more so than the plausible locations of activity centers.  
5855 That is, a home range centroid could, in actual fact, occur in a shopping mall parking lot  
5856 if there is pretty good habitat around the shopping mall, so there is probably no sense  
5857 preclude it as the location for an activity center. It would generally be better to include  
5858 some definition of habitat quality in the model for the detection probability (Royle et al.,  
5859 2013) which we address in Chaps. 12 and 13.

---

**5860 5.10.1 Evaluation of coarseness of habitat mask**

5861 The coarseness of the state-space should not really have much of an effect on estimates  
 5862 if the grain is sufficiently fine relative to typical animal home range sizes. Why is this?  
 5863 We have two analogies that can help us understand. First is the relationship to model  
 5864  $M_h$ . As noted in Sec. 5.3.2 above, we can think about SCR models as a type of finite  
 5865 mixture (Norris and Pollock, 1996; Pledger, 2004) where we are fortunate to be able to  
 5866 obtain direct information about which group individuals belong to (group being location  
 5867 of activity center). In the standard finite mixture models we typically find that a small  
 5868 number of groups (e.g., 2 or 3 at the most) can explain high levels of heterogeneity and  
 5869 are adequate for most data sets of small to moderate sample sizes. We therefore expect a  
 5870 similar effect in SCR models when we discretize the state-space. We can also think about  
 5871 discretizing the state-space as being related to numerical integration where we find (see  
 5872 Chapt. 6) that we don't need a very fine grid of support points to evaluate the integral to  
 5873 a reasonable level of accuracy. We demonstrate this here by reanalyzing simulated data  
 5874 using a state-space defined by a different number of support points. We provide an **R**  
 5875 script called **SCR0bayesDss** in the **R** package **scrbook**. We note that for this comparison  
 5876 we generated the actual activity centers as a continuous random variable and thus the  
 5877 discrete state-space is, strictly speaking, an approximation to truth. That said, we regard  
 5878 all state-space specifications as approximations to truth in the sense that they represent  
 5879 a component of the SCR model.

5880 As with our **R** function **SCR0bayes**, the modification **SCR0bayesDss** will use either  
 5881 **WinBUGS** or **JAGS**. In addition, it requires a grid resolution argument (**ng**) which  
 5882 is the dimension of 1 side of a square state-space. To execute this function we do, for  
 5883 example:

```
5884 > library(scrbook)
5885 > data <- simSCR0(discard0=TRUE,rnd=2013)    # Generate data set
5886
5887 # Run with JAGS
5888 > out1 <- SCR0bayesDss(data,ng=8,M=200,engine="jags",ni=2000,nb=1000)
5889
5890 # Run with WinBUGS
5891 > out2 <- SCR0bayesDss(data,ng=8,M=200,engine="winbugs",ni=2000,nb=1000)
```

5892 We fit this model to the same simulated data set for  $6 \times 6$ ,  $9 \times 9$ ,  $12 \times 12$ ,  $15 \times 15$   
 5893 state-space grids. For **WinBUGS**, we used 3 chains of 5000 total length with 1000 burn-in,  
 5894 which yields 12000 total posterior samples. Summary results are shown in Table 5.6.  
 5895 The results are broadly consistent except for the  $6 \times 6$  case. We see that the run time  
 5896 increases with the size of the state-space grid (not unexpected), such that we imagine it  
 5897 would be impractical to run models with more than a few hundred state-space grid points.  
 5898 We found (not shown here) that the runtime of **JAGS** is much faster and, furthermore,  
 5899 relatively *constant* as we increase the grid size. We suspect that **WinBUGS** is evaluating  
 5900 the full-conditional for each activity center at all  $G$  possible values whereas it may be  
 5901 that **JAGS** is evaluating the full-conditional only at a subset of values or perhaps using  
 5902 previous calculations more effectively. While this might suggest that one should always  
 5903 use **JAGS** for this analysis, we found in our analysis of the wolverine (next section) that  
 5904 **JAGS** could be extremely sensitive to starting values, producing MCMC algorithms that

**Table 5.6.** Comparison of the effect of state-space grid coarseness on estimates of  $N$  for a simulated data set. Posterior summaries and run time are given. Results obtained using **WinBUGS** run from **R2WinBUGS**.

Grid Size	Mean	SD	NaiveSE	Time-seriesSE	runtime (sec)
6 × 6	111.6699	16.61414	0.1516657	0.682008	2274
9 × 9	114.2294	17.99109	0.1642355	0.833291	4300
12 × 12	115.9806	17.3843	0.1586964	0.762756	7100
15 × 15	115.379	17.93721	0.1637436	0.832483	13010

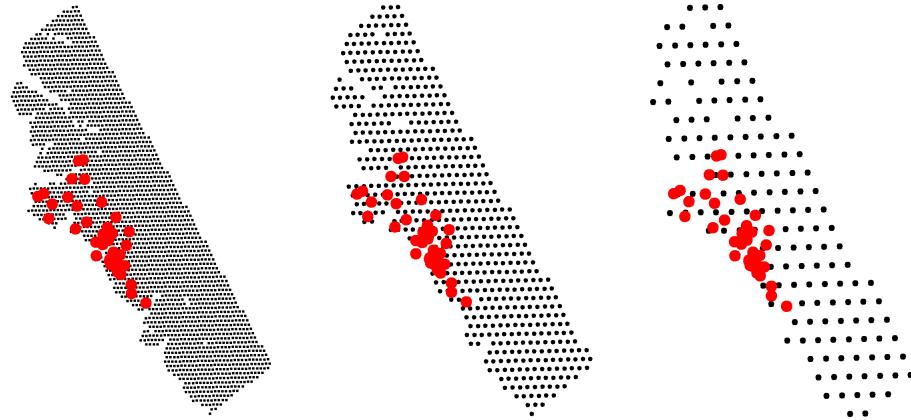
5905 often simply do not work for some problems, so be careful when using **JAGS**. To improve  
 5906 its performance, always start the latent activity centers at values near where individuals  
 5907 were captured. The performance of either should improve if we compute the full distance  
 5908 matrix outside of **BUGS** and pass it as data, although we haven't fully evaluated this  
 5909 approach.

### 5910 5.10.2 Analysis of the wolverine camera trapping data

5911 We reanalyzed the wolverine data using discrete state-space grids with points spaced by  
 5912 2, 4 and 8 km (see Fig. 5.5). These were constructed from a 40 km buffered state-space,  
 5913 and deleting the points over water (see Royle et al., 2011b). Our interest in doing this  
 5914 was to evaluate the relative influence of grid resolution on estimated density because the  
 5915 coarser grids will be more efficient from a computational stand-point and so we would  
 5916 prefer to use them, but only if there is no strong influence on estimated density. The  
 5917 posterior summaries for the 3 habitat grids are given in Table 5.7. We see that the  
 5918 density estimates are quite a bit larger than obtained in our analysis (Table 5.9.2) based  
 5919 on a rectangular, continuous state-space. We also see that there are slight differences  
 5920 depending on the resolution of the state-space grid. Interestingly, the effectiveness of the  
 5921 MCMC algorithms, as measured by effective sample size (**n.eff**) is pretty remarkably  
 5922 different. Furthermore, the finest grid resolution (2 km spacing) took about 6 days to run  
 5923 and thus it would not be practical for large problems or with many models.

## 5.11 SUMMARIZING DENSITY AND ACTIVITY CENTER LOCATIONS

5924 One of the most useful aspects of SCR models is that they are parameterized in terms of  
 5925 individual locations – i.e., *where* each individual lives – and, thus, we can compute many  
 5926 useful and interesting summaries of the activity centers using output from an MCMC sim-  
 5927 ulation, including maps of density (the number of activity centers per unit area), estimates  
 5928 of  $N$  for any well-defined polygon, or estimates of where the activity centers for specific  
 5929 individuals reside. In Bayesian analysis by MCMC, obtaining such summaries entails no  
 5930 added calculations, because we need only post-process the output for the individual ac-  
 5931 tivity centers to obtain the desired summaries. We demonstrate that in this section. Note  
 5932 that you have to be sure to retain the MCMC history for the **s** variables and also the data  
 5933 augmentation variables  $z$  in order to do the following analyses.



**Figure 5.5.** Three habitat mask grids used in the comparison of the effect of pixel size on the estimated density surface of wolverines. The 3 cases are 2 (left), 4 (center) and 8 (right) km spacing of state-space points, extending 40 km from the vicinity of the trap array.

### 5934 5.11.1 Constructing density maps

5935 Because SCR models are spatially-explicit, it is natural to want to summarize the results  
 5936 of fitting a model by producing a map of density. Using Bayesian analysis by MCMC, it is  
 5937 most easy to make a map of *realized* density. We can do this by tallying up the number of  
 5938 activity centers  $\mathbf{s}_i$  in pixels of arbitrary size and then producing a nice multi-color spatial  
 5939 plot of the result. Specifically, let  $B(\mathbf{x})$  indicate a pixel centered at  $\mathbf{x}$  then

$$N(\mathbf{x}) = \sum_{i=1}^M I(\mathbf{s}_i \in B(\mathbf{x}))$$

5940 (here,  $I(arg)$  is the indicator function which evaluates to 1 if  $arg$  is true, and 0 otherwise)  
 5941 is the population size of pixel  $B(\mathbf{x})$ , and  $D(\mathbf{x}) = N(\mathbf{x})/\|B(\mathbf{x})\|$  is the local density. Note  
 5942 that these  $N(\mathbf{x})$  parameter are just “derived parameters” as we normally obtain from  
 5943 posterior output using the appropriate Monte Carlo average (see Chapt. 3).

5944 One thing to be careful about, in the context of models in which  $N$  is unknown, is that,  
 5945 for each MCMC iteration  $m$ , we only tabulate those activity centers which correspond to  
 5946 individuals in the sampled population, i.e., for which the data augmentation variable  
 5947  $z_i = 1$ . In this case, we take all of the output for MCMC iterations  $m = 1, 2, \dots, \text{niter}$   
 5948 and compute this summary:

$$N(\mathbf{x}, m) = \sum_{i: z_{i,m}=1} I(\mathbf{s}_{i,m} \in B(\mathbf{x}))$$

**Table 5.7.** Posterior summaries for the wolverine camera trapping data, using model SCR0, with a Gaussian hazard encounter probability model, and a discrete habitat mask of 3 different resolutions: 2, 4 and 8 km. Parameters are  $\lambda_0$  = baseline encounter rate,  $p_0 = 1 - \exp(-\lambda_0)$ ,  $\sigma$  is the scale parameter of the Gaussian kernel,  $\psi$  is the data augmentation parameter,  $N$  and  $D$  are population size and density, respectively. Models fitted using **WinBUGS**, 3 chains, each with 11000 iterations (first 1000 discarded) producing 30000 posterior samples.

2 km spacing										
	Mean	SD	2.5%	25%	50%	75%	97.5%	Rhat	n.eff	
N	86.56	16.94	57.00	75.00	85.00	97.00	124.00	1.00	510	
D	8.78	1.72	5.78	7.60	8.62	9.83	12.57	1.00	510	
$\lambda_0$	0.05	0.01	0.04	0.04	0.05	0.06	0.07	1.01	320	
$p_0$	0.05	0.01	0.03	0.04	0.05	0.05	0.06	1.01	320	
$\sigma$	0.62	0.05	0.54	0.59	0.62	0.65	0.73	1.01	160	
$\psi$	0.43	0.09	0.27	0.37	0.43	0.49	0.63	1.00	560	
4 km spacing										
	Mean	SD	2.5%	25%	50%	75%	97.5%	Rhat	n.eff	
N	89.25	17.44	59.00	77.00	88.00	100.00	127.00	1	1100	
D	9.01	1.76	5.96	7.77	8.88	10.10	12.82	1	1100	
$\lambda_0$	0.05	0.01	0.04	0.05	0.05	0.06	0.07	1	2500	
$p_0$	0.05	0.01	0.03	0.04	0.05	0.05	0.07	1	2500	
$\sigma$	0.61	0.04	0.53	0.58	0.61	0.64	0.71	1	1600	
$\psi$	0.45	0.09	0.28	0.38	0.44	0.50	0.64	1	1300	
8 km spacing										
	Mean	SD	2.5%	25%	50%	75%	97.5%	Rhat	n.eff	
N	83.18	16.14	56.00	72.00	82.00	93.00	119.00	1.00	700	
D	8.28	1.61	5.57	7.17	8.16	9.26	11.84	1.00	700	
$\lambda_0$	0.05	0.01	0.03	0.04	0.05	0.05	0.06	1.00	560	
$p_0$	0.05	0.01	0.03	0.04	0.04	0.05	0.06	1.00	560	
$\sigma$	0.68	0.05	0.59	0.64	0.67	0.71	0.77	1.01	220	
$\psi$	0.42	0.09	0.26	0.36	0.41	0.47	0.61	1.00	940	

5949 Thus,  $N(\mathbf{x}, 1), N(\mathbf{x}, 2), \dots$ , is the Markov chain for parameter  $N(\mathbf{x})$ . In what follows we  
 5950 will provide a set of **R** commands for doing this calculation and making a basic image  
 5951 plot from the MCMC output.

5952 **Step 1:** Define the center points of each pixel  $B(\mathbf{x})$ , or point at which local density will  
 5953 be estimated:

```
5954 > xg <- seq(xlim[1], xlim[2], , 50)
  5955 > yg <- seq(ylim[1], ylim[2], , 50)
```

5956 **Step 2:** Extract the MCMC histories for the activity centers and the data augmentation  
 5957 variables. Note that these are each  $N \times \text{niter}$  matrices. Here we do this assuming that  
 5958 **WinBUGS** was run producing the **R** object named **out**:

```
5959 > Sxout <- out$sims.list$s[, , 1]
  5960 > Syout <- out$sims.list$s[, , 2]
  5961 > z <- out$sims.list$z
```

5962   **Step 3:** We associate each coordinate with the proper pixel using the **R** command `cut()`.  
 5963   Note that we keep only the activity centers for which  $z = 1$  (i.e., individuals that belong  
 5964   to the population of size  $N$ ):

```
5965 > Sxout <- cut(Sxout[z==1], breaks=xg, include.lowest=TRUE)
5966 > Syout <- cut(Syout[z==1], breaks=yg, include.lowest=TRUE)
```

5967   **Step 4:** Use the `table()` command to tally up how many activity centers are in each  
 5968    $B(\mathbf{x})$ :

```
5969 > Dn <- table(Sxout, Syout)
```

5970   **Step 5:** Use the `image()` command to display the resulting matrix.

```
5971 > image(xg, yg, Dn/nrow(z), col=topo.colors(10))
```

5972   It is worth emphasizing here that density maps will not usually appear uniform despite  
 5973   that we have assumed that activity centers are uniformly distributed. This is because  
 5974   the observed encounters of individuals provide direct information about the location of  
 5975   the  $i = 1, 2, \dots, n$  activity centers and thus their “estimated” locations will be affected  
 5976   by the observations. In a limiting sense, were we to sample space intensely enough,  
 5977   every individual would be captured a number of times and we would have considerable  
 5978   information about all  $N$  point locations. Consequently, the uniform prior would have  
 5979   almost no influence at all on the estimated density surface in this limiting situation.  
 5980   Thus, in practice, the influence of the uniformity assumption decreases as the fraction of  
 5981   the population encountered, and the total number of encounters per individual, increases.

5982   **On the non-intuitiveness of `image()`** – the **R** function `image()`, invoked for a  
 5983   matrix  $M$  by `image(M)`, might not be very intuitive to some – it plots  $M[1, 1]$  in the lower  
 5984   left corner. If you want  $M[]$  to be plotted “as you look at it” then  $M[1, 1]$  should be in the  
 5985   upper left corner. We have a function `rot()` which does that. If you do `image(rot(M))`  
 5986   then it puts it on the monitor as if it was a map you were looking at. You can always  
 5987   specify the  $x$ - and  $y$ -labels explicitly as we did above.

5988   **Spatial dot plots** – A cruder version of the density map can be made using our  
 5989   “spatial dot map” function `spatial.plot` (in `scrbook`). This function requires, as input,  
 5990   point locations and the value to be displayed. A simplified version of this function is as  
 5991   follows:

```
5992 > spatial.plot <- function(x,y){
  5993   nc <- as.numeric(cut(y,20))
  5994   plot(x,pch=" ")
  5995   points(x,pch=20,col=topo.colors(20)[nc],cex=2)
  5996   image.scale(y,col=topo.colors(20))
  5997 }
  5998 #
  5999 # To execute the function do this:
  6000 #
  6001 > spatial.plot(cbind(xg,yg), Dn/nrow(z))
```

**5.11.2 Example: Wolverine density map**

6003 We return to the wolverine study which took place in 2008 in SE Alaska (Fig. 5.4) and  
 6004 we produce a density map of wolverines from that analysis. We include the function  
 6005 **SCRdensity** which requires a specific data structure as shown below. In particular, we  
 6006 have to package up the MCMC history for the activity centers and the data augmentation  
 6007 variables  $z$  into a list. This also requires that we add those variables to the parameters-  
 6008 to-be-monitored list when we pass things to **BUGS**.

6009 We used the posterior output from the wolverine model fitted previously to compute  
 6010 a relatively coarse version of a density map, using 100 pixels in a  $10 \times 10$  grid (Fig. 5.6  
 6011 top panel) and using 900 pixels arranged in a  $30 \times 30$  grid (Fig. 5.6 lower panel) for a  
 6012 fine-scale map. The **R** commands for producing such a plot (for a short MCMC run) are  
 6013 as follows:

```
6014 > library(scrbook)
6015 > data(wolverine)
6016 > traps <- wolverine$wtraps
6017 > y3d <- SCR23darray(wolverine$wcaps,wolverine$wtraps)
6018
6019 # This takes 341 seconds on a standard CPU circa 2011
6020 > out <- wolvSCRO(y3d,traps,nb=1000,ni=2000,buffer=1,M=100,keepz=TRUE)
6021
6022 > Sx <- out$sims.list$s[,,1]
6023 > Sy <- out$sims.list$s[,,2]
6024 > z <- out$sims.list$z
6025 > obj <- list(Sx=Sx,Sy=Sy,z=z)
6026 > tmp <- SCRdensity(obj,nx=10,ny=10,scalein=100,scaleout=100)
```

6027 In these figures density is expressed in units of individuals per  $100 \text{ km}^2$ , while the area of  
 6028 the pixels is about  $103.7 \text{ km}^2$  and  $11.5 \text{ km}^2$ , respectively. That calculation is based on:

```
6029 > total.area <- (ylim[2]-ylim[1])*(xlim[2]-xlim[1])*100
6030 > total.area/(10*10)
6031 [1] 103.7427
6032 > total.area/(30*30)
6033 [1] 11.52697
```

6034 A couple of things are worth noting: First is that as we move away from “where the  
 6035 data live” – away from the trap array – we see that the density approaches the mean  
 6036 density. This is a property of the estimator as long as the detection function decreases  
 6037 sufficiently rapidly as a function of distance. Relatedly, it is also a property of statistical  
 6038 smoothers such as splines, kernel smoothers, and regression smoothers – predictions tend  
 6039 toward the global mean as the influence of data diminishes. Another way to think of it is  
 6040 that it is a consequence of the prior, which imposes uniformity, and as you get far away  
 6041 from the data, the predictions tend to the expected constant density under the prior.  
 6042 Another thing to note about this map is that density is not 0 over water (although the  
 6043 coastline is not shown). This might be perplexing to some who are fairly certain that  
 6044 wolverines do not like water. However, there is nothing about the model that recognizes

6045 water from non-water and so the model predicts over water *as if* it were habitat similar to  
 6046 that within which the array is nested. But, all of this is OK as far as estimating density  
 6047 goes and, furthermore, we can compute valid estimates of  $N$  over any well-defined region  
 6048 which presumably wouldn't include water if we so wished. Alternatively, areas covered by  
 6049 water could be masked out, which we discuss in the next section.

### 6050 5.11.3 Predicting where an individual lives

6051 The density maps in the previous section show the expected number of individuals per  
 6052 unit area. A closely related problem is that of producing a map of the probable location  
 6053 of a specific individual's activity center. For any observed encounter history, we can easily  
 6054 generate a posterior distribution of  $\mathbf{s}_i$  for individual  $i$ . In addition, for an individual that  
 6055 is *not* captured, we can use the MCMC output to produce a corresponding plot of where  
 6056 such an individual might live, say  $\mathbf{s}_{n+1}$ . Obviously, all such uncaptured individuals (for  
 6057  $i = n + 1, \dots, N$ ) should have the same posterior distribution. To illustrate, we show the  
 6058 posterior distribution of  $\mathbf{s}_1$ , the activity center for the individual labeled 1 in the data  
 6059 set, in Fig. 5.7. This individual was captured a single time at trap 30 which is circled  
 6060 in Fig. 5.7. We see that the posterior distribution is affected by traps of capture *and*  
 6061 traps of non-capture in fairly intuitive ways. In particular, because there are other traps  
 6062 in close proximity to trap 30, in which individual 1 was *not* captured, the model pushes  
 6063 its activity center away from the trap array. The help file for `SCRdensity` shows how to  
 6064 calculate Fig. 5.7.

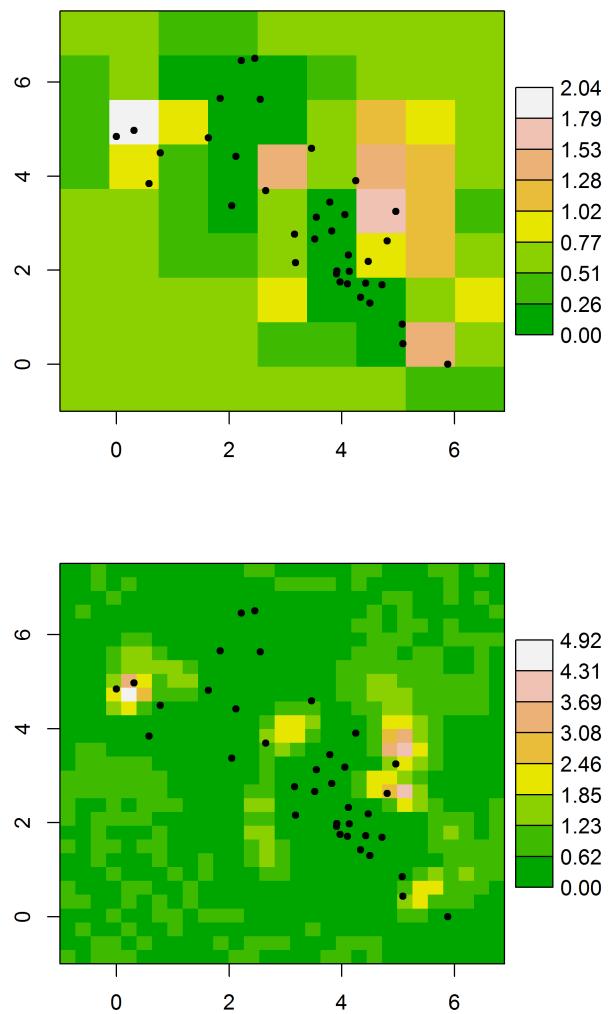
## 5.12 EFFECTIVE SAMPLE AREA

6065 One of the key issues in using ordinary capture recapture models which we've brought up  
 6066 over and over again is this issue that the area which is sampled by a trapping array is  
 6067 unknown – in other words, the  $N$  that is estimated by capture-recapture models does not  
 6068 have an explicit region of space associated with it. Classically this has been addressed in  
 6069 the ad hoc way of prescribing an area that contains the trap array, usually by adding a  
 6070 buffer of some width, which is not estimated as part of the capture-recapture model. In  
 6071 SCR models we avoid the problem of not having an explicit linkage between  $N$  and “area”,  
 6072 by prescribing explicitly the area within which the underlying point process is defined – the  
 6073 state-space of the point process. This state-space is *not* the effective sample (or sampled)  
 6074 area (ESA) – it is desirable that it be somewhat larger than the ESA, whatever that may  
 6075 be, in the sense that individuals at the edge of the state-space have no probability of being  
 6076 captured, but as part of the SCR model we don't need to try to estimate or otherwise  
 6077 characterize the ESA explicitly.

6078 However, it is possible to provide a characterization of effective sampled area under  
 6079 any SCR model. This is directly analogous to the calculation of “effective strip width” in  
 6080 distance sampling (Buckland et al., 2001; Borchers et al., 2002). The conceptual definition  
 6081 of ESA follows from equating density to “apparent density” – ESA is the magic number  
 6082 that satisfies that equivalence:

$$D = N/A = n/\text{ESA}$$

6083 In other words, the ratio of  $N$  to the area of the state-space should be equal to the ratio  
 6084 of the observed sample size  $n$  to this number ESA. Both of these should equal density.



**Figure 5.6.** Density of wolverines (individuals per 100 km<sup>2</sup>) in SE Alaska in 2007 based on model SCR0. Map grid cells are about 103.7 km<sup>2</sup> (top panel) and 11.5 km<sup>2</sup> (bottom panel) in area. Dots are the trap locations.

6085 So, to compute ESA for a model, we substitute  $\mathbb{E}(n)$  for  $n$  into the above equation, and  
 6086 solve for  $ESA$ , to get:

$$ESA = \mathbb{E}(n)/D.$$

6087 Our following development assumes that  $D$  is constant, but these calculations can be  
 6088 generalized to allow for  $D$  to vary spatially. Imagine our habitat mask for the wolverine  
 6089 data, or the bins we just used to produce a density map, then we can write  $\mathbb{E}(n)$  according  
 6090 to

$$\mathbb{E}(n) = \sum_s \Pr(\text{encounter}|\mathbf{s})\mathbb{E}(N(\mathbf{s}))$$

6091 where if we prefer to think of this more conceptually we could replace the summation with  
 6092 an integration (which, in practice, we would just replace with a summation, and so we  
 6093 just begin there). In this expression note that  $\mathbb{E}(N(\mathbf{s}))$  is the expected population size at  
 6094 pixel  $\mathbf{s}$  which is the density times the area of the pixel, i.e.,  $\mathbb{E}(N(\mathbf{s})) = D \times a$ . Therefore

$$\mathbb{E}(n) = D \times a \times \sum_s \Pr(\text{encounter}|\mathbf{s})$$

6095 and (plugging this into the expression above for ESA)

$$ESA = \frac{D \times a \times \sum_s \Pr(\text{encounter}|\mathbf{s})}{D}$$

6096 We see that  $D$  cancels and we have  $ESA = a \times \sum_s \Pr(\text{encounter}|\mathbf{s})$  So what you have to  
 6097 do here is substitute in  $\Pr(\text{encounter}|\mathbf{s})$  and just sum them up over all pixels. For the  
 6098 Bernoulli model of model SCR0

$$\Pr(\text{encounter}|\mathbf{s}) = 1 - (1 - p(\mathbf{s}))^K$$

6099 with slight modifications when encounter probability depends on covariates. Thus,

$$ESA = a \sum_s 1 - (1 - p(\mathbf{s}))^K \tag{5.12.1}$$

6100 Clearly the calculation of ESA is affected by the use of a habitat mask, because the  
 6101 summation in Eq. 5.12.1 only occurs over pixels that define the state-space.

6102 For the wolverine camera trapping data, we used the  $2 \times 2$  km habitat mask and the  
 6103 posterior means of  $p_0$  and  $\sigma$  (see Sec. 5.10.2) to compute the probability of encounter for  
 6104 each  $\mathbf{s}$  of the mask points. The result is shown graphically in Fig. 5.8. The ESA is the  
 6105 sum of the values plotted in that figure multiplied by 4, the area of each pixel. For the  
 6106 wolverine study, the result is  $2507.152 \text{ km}^2$ . We note that the probability of encounter  
 6107 declines rapidly to 0 as we move away from the periphery of the camera traps, indicating  
 6108 the state-space constructed from a 40 km buffered trap array was indeed sufficient for the  
 6109 analysis of these data. An **R** script for producing this figure is in the `wolvESA` function of  
 6110 the `scrbook` package.

## 5.13 SUMMARY AND OUTLOOK

6111 In this chapter, we introduced the simplest SCR model – “model SCR0” – which is an ordi-  
 6112 nary capture-recapture model like model  $M_0$ , but augmented with a set of latent individual

6113 effects,  $s_i$ , which relate encounter probability to some sense of individual location using a  
6114 covariate, “distance”, from  $s_i$  to each trap location. Thus, individuals in close proximity  
6115 to a trap will have a higher probability of encounter, and *vice versa*. The explicit modeling  
6116 of individual locations and distance in this fashion resolves classical problems related to  
6117 estimating density: unknown sample area, and heterogeneous encounter probability due  
6118 to variable exposure to traps.

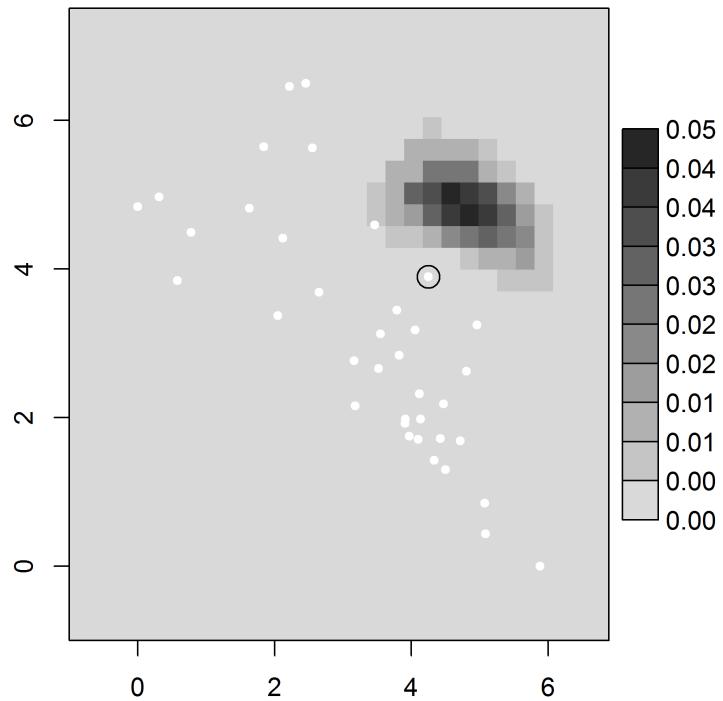
6119 SCR models are closely related to classical individual covariate models (“model  $M_x$ ”,  
6120 as introduced in Chapt. 4), but with imperfect information about the individual covari-  
6121 ate. Therefore, they are also not too dissimilar from standard GLMMs used throughout  
6122 statistics and, as a result, we find that they are easy to analyze using standard MCMC  
6123 methods encased in black boxes such as **WinBUGS** or **JAGS**. We will also see that they  
6124 are easy to analyze using likelihood methods, which we address in Chapt. 6.

6125 Formal consideration of the collection of individual locations ( $s_1, \dots, s_N$ ) is funda-  
6126 mental to all models considered in this book. In statistical terminology, we think of the  
6127 collection of points  $\{s_i\}$  as a realization of a point process. Because SCR models formally  
6128 link individual encounter history data to an underlying point process, we can obtain for-  
6129 mal inferences about the point process. For example, we showed how to produce a density  
6130 map (Fig. 5.6), or even a probability map for an individual’s home range center (Fig.  
6131 5.7). We can also use SCR models as the basis for doing more traditional point process  
6132 analyses, such as testing for “complete spatial randomness” (CSR) (see Chapt. 8), and  
6133 computing other point process summaries (Illian et al., 2008).

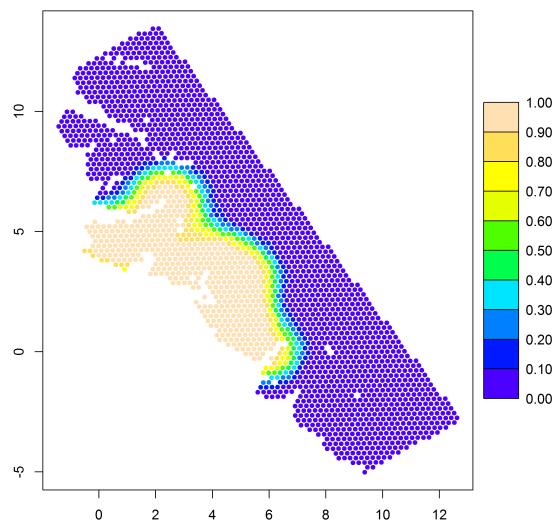
6134 Part of the promise, and ongoing challenge, of SCR models is to develop models that  
6135 reflect interesting biological processes, for example interactions among points or temporal  
6136 dynamics in point locations. In this chapter we considered the simplest possible point  
6137 process model in which points are independent and uniformly (“randomly”) distributed  
6138 over space. Despite the simplicity of this model, it should suffice in many applications of  
6139 SCR models, although we do address generalizations in later chapters. Moreover, even  
6140 though the *prior* distribution on the point locations is uniform, the realized pattern may  
6141 deviate markedly from uniformity as the observed encounter data provide information to  
6142 impart deviations from uniformity. Thus, estimated density maps will typically appear  
6143 distinctly non-uniform (as we saw in the wolverine example). In applications of the basic  
6144 SCR model, we find that this simple *a priori* model can effectively reflect or adapt to  
6145 complex realizations of the underlying point process. For example, if individuals are  
6146 highly territorial then the data should indicate this in the form of individuals not being  
6147 encountered in the same trap – the resulting posterior distribution of point locations should  
6148 therefore reflect non-independence. Obviously the complexity of posterior estimates of the  
6149 point pattern will depend on the quantity of data, both number of individuals and captures  
6150 per individual. Because the point process is such an integral component of SCR models,  
6151 the state-space of the point process plays an important role in developing SCR models.  
6152 As we emphasized in this chapter, the state-space is part of the model. It can have an  
6153 influence on parameter estimates and other inferences, such as model selection (see chapter  
6154 8).

6155 One concept we introduced in this chapter, which has not been discussed much in  
6156 the literature on SCR models, is the manner in which the encounter probability model  
6157 relates to a model of space usage by individuals. The standard SCR models of encounter  
6158 probability can all be motivated as simplistic models of space usage and movement, in

6159 which individuals make random use decisions from a probability distribution proportional  
6160 to the encounter probability model. This both clarifies the simplicity of the underlying  
6161 model of space usage and also suggests a direct extension to produce more realistic models,  
6162 which we discuss in Chapt. 13. We consider some other important extensions of the basic  
6163 SCR model in later chapters. For example, we consider models that include covariates that  
6164 vary by individual, trap, or over time (Chapt. 7), spatial covariates on density (Chapt.  
6165 11), open populations (Chapt. 16), and methods for model assessment and selection  
6166 (Chapt. 8) among other topics. We also consider technical details of maximum likelihood  
6167 (Chapt. 6) and Bayesian (Chapt. 17) estimation, so that the interested reader can develop  
6168 or extend methods to suit their own needs.



**Figure 5.7.** Posterior probability distribution of  $s_1$ , the activity center for individual 1 in the wolverine data set. This individual was captured a single time in one trap (trap 30) which is circled. White dots are trap locations.



**Figure 5.8.** Probability of encounter used in computing effective sampled area for the wolverine camera trapping array, using the parameter estimates (posterior means) for the  $2 \times 2$  km habitat mask.



6169

6170

6171

6172

# 6

---

## LIKELIHOOD ANALYSIS OF SPATIAL CAPTURE-RECAPTURE MODELS

6173 We have so far mainly focused on Bayesian analysis of spatial capture-recapture models.  
 6174 And, in the previous chapters we learned how to fit some basic spatial capture-recapture  
 6175 models using a Bayesian formulation of the models analyzed in **BUGS** engines including  
 6176 **WinBUGS** and **JAGS**. Despite our focus on Bayesian analysis, it is instructive to de-  
 6177 velop the basic concepts and ideas behind classical analysis based on likelihood methods  
 6178 and frequentist inference for SCR models. We recognized earlier (Chapt. 5) that SCR  
 6179 models are versions of binomial (or other) GLMs, but with random effects (i.e., GLMMs).  
 6180 Throughout statistics, such models are routinely analyzed by likelihood methods. In par-  
 6181 ticular, likelihood analysis is based on the integrated or marginal likelihood in which the  
 6182 random effects are removed, by integration, from the conditional-on-s likelihood (*s* being  
 6183 the individual activity center). This has been the approach taken by Borchers and Ef-  
 6184 ford (2008); Dawson and Efford (2009) and related papers. Therefore, in this chapter, we  
 6185 provide some conceptual and technical foundation for likelihood-based analysis of spatial  
 6186 capture-recapture models.

6187 We will show here that it is straightforward to compute the maximum likelihood esti-  
 6188 mates (MLE) for SCR models by integrated likelihood. We develop the MLE framework  
 6189 using **R**, and we also provide a basic introduction to the **R** package **secr** (Efford, 2011a)  
 6190 which does likelihood analysis of SCR models (see also the stand-alone program **DEN-**  
 6191 **SITY** (Efford et al., 2004)). To set the context for likelihood analysis of SCR models,  
 6192 we first analyze the SCR model when  $N$  is known because, in that case, analysis is no  
 6193 different at all than a standard GLMM. We generalize the model to allow for unknown  $N$   
 6194 using both conventional ideas based on the “full likelihood” (e.g., Borchers et al., 2002)  
 6195 and also using a formulation based on data augmentation. We obtain the MLEs for the  
 6196 SCR model from the wolverine camera trapping study (Magoun et al., 2011) analyzed in  
 6197 previous chapters to compare/contrast the results.

## 6.1 MLE WITH KNOWN N

We noted in Chapt. 5 that, with  $N$  known, the basic SCR model is a type of binomial model with a random effect. For such models we can obtain maximum likelihood estimators of model parameters based on integrated likelihood. The integrated likelihood is based on the marginal distribution of the data  $y$  in which the random effects are removed by integration from the conditional-on-s distribution of the observations. See Chapt. 2 for a review of marginal, conditional and joint distributions. Conceptually, any SCR model begins with a specification of the conditional-on-s model  $[y|\mathbf{s}, \boldsymbol{\alpha}]$  and we have a “prior distribution” for  $\mathbf{s}$ , say  $[\mathbf{s}]$ . Then, the marginal distribution of the data  $y$  is

$$[y|\boldsymbol{\alpha}] = \int_{\mathcal{S}} [y|\mathbf{s}, \boldsymbol{\alpha}][\mathbf{s}]d\mathbf{s}.$$

When viewed as a function of  $\boldsymbol{\alpha}$  for purposes of estimation, the marginal distribution  $[y|\boldsymbol{\alpha}]$  is often referred to as the *integrated likelihood*.

It is worth analyzing the simplest SCR model with known- $N$  in order to understand the underlying mechanics and basic concepts. These are directly relevant to the manner in which many capture-recapture models are classically analyzed, such as model  $M_h$ , and individual covariate models (see Chapt. 4).

To develop the integrated likelihood for SCR models, we first identify the conditional-on-s likelihood. The observation model for each encounter observation  $y_{ij}$ , for individual  $i$  and trap  $j$ , specified conditional on  $\mathbf{s}_i$ , is

$$y_{ij}|\mathbf{s}_i \sim \text{Binomial}(K, p_{\boldsymbol{\alpha}}(\mathbf{x}_j, \mathbf{s}_i)) \quad (6.1.1)$$

where we have indicated the dependence of encounter probability,  $p_{ij}$ , on  $\mathbf{s}$  and parameters  $\boldsymbol{\alpha}$  explicitly. For example,  $p_{ij}$  might be the Gaussian model given by

$$p_{ij} = \text{logit}^{-1}(\alpha_0) \exp(-\alpha_1 \|\mathbf{x}_j - \mathbf{s}_i\|^2)$$

where  $\alpha_1 = 1/(2\sigma^2)$ . The joint distribution of the data for individual  $i$  is the product of  $J$  such terms (i.e., contributions from each of  $J$  traps).

$$[\mathbf{y}_i|\mathbf{s}_i, \boldsymbol{\alpha}] = \prod_{j=1}^J \text{Binomial}(K, p_{\boldsymbol{\alpha}}(\mathbf{x}_j, \mathbf{s}_i))$$

We note this assumes that encounter of individual  $i$  in each trap is independent of encounter in every other trap, conditional on  $\mathbf{s}_i$ . This is the fundamental property of the basic model SCR0. The marginal likelihood is computed by removing  $\mathbf{s}_i$ , by integration from the conditional-on-s likelihood, so we compute:

$$[\mathbf{y}_i|\boldsymbol{\alpha}] = \int_{\mathcal{S}} [\mathbf{y}_i|\mathbf{s}_i, \boldsymbol{\alpha}][\mathbf{s}_i]d\mathbf{s}_i$$

In most SCR models,  $[\mathbf{s}] = 1/A(\mathcal{S})$  where  $A(\mathcal{S})$  is the area of the prescribed state-space  $\mathcal{S}$  (but see Chapt. 11 for alternative specifications of  $[\mathbf{s}]$ ).

The joint likelihood for all  $N$  individuals, assuming independence of encounters among individuals, is the product of  $N$  such terms:

$$\mathcal{L}(\boldsymbol{\alpha}|\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N) = \prod_{i=1}^N [\mathbf{y}_i|\boldsymbol{\alpha}]$$

6227 We emphasize that two independence assumptions are explicit in this development: independence of trap-specific encounters within individuals and also independence among  
 6228 individuals. In particular, this would only be valid when individuals are not physically  
 6229 restrained or removed upon capture, and when traps do not “fill up.”

6230 The key operation for computing the likelihood is solving a 2-dimensional integration  
 6231 problem. There are some general purpose **R** packages that implement a number of multi-  
 6232 dimensional integration routines including **adapt** (Genz et al., 2007) and **R2cuba** (Hahn  
 6233 et al., 2010). In practice, we won’t rely on these extraneous **R** packages (except see Chapt.  
 6234 11 for an application of **R2cuba**) but instead will use perhaps less efficient methods in which  
 6235 we replace the integral with a summation over an equal area mesh of points on the state-  
 6236 space  $\mathcal{S}$  and explicitly evaluate the integrand at each point. We invoke the rectangular  
 6237 rule for integration here<sup>1</sup> in which we evaluate the integrand on a regular grid of points  
 6238 of equal area and compute the average of the integrand over that grid of points. Let  
 6239  $u = 1, 2, \dots, nG$  index a grid of  $nG$  points,  $\mathbf{s}_u$ , where the area of grid cells is constant, say  
 6240  $A$ . In this case, the integrand, i.e., the marginal pmf of  $\mathbf{y}_i$ , is approximated by

$$[\mathbf{y}_i | \boldsymbol{\alpha}] = \frac{1}{nG} \sum_{u=1}^{nG} [\mathbf{y}_i | \mathbf{s}_u, \boldsymbol{\alpha}] \quad (6.1.2)$$

6242 This is a specific case of the general expression that could be used for approximating  
 6243 the integral for any arbitrary distribution  $[\mathbf{s}]$ . The general case is

$$[\mathbf{y} | \boldsymbol{\alpha}] = \frac{A(\mathcal{S})}{nG} \sum_{u=1}^{nG} [y | \mathbf{s}_u, \boldsymbol{\alpha}] [\mathbf{s}_u]$$

6244 Under the uniformity assumption,  $[\mathbf{s}] = 1/A(\mathcal{S})$  and thus the grid-cell area cancels in the  
 6245 above expression to yield Eq. 6.1.2. The rectangular rule for integration can be seen as  
 6246 an application of the Law of Total Probability for a discrete random variable  $\mathbf{s}$ , having  
 6247  $nG$  unique values with equal probabilities  $1/nG$ .

### 6248 6.1.1 Implementation (simulated data)

6249 Here we will illustrate how to carry out this integration and optimization based on the  
 6250 integrated likelihood using simulated data (i.e., see Sec. 5.5). Using **simSCR0** we simulate  
 6251 data for 100 individuals and an array of 25 traps laid out in a  $5 \times 5$  grid of traps having unit  
 6252 spacing. The specific encounter model is the Gaussian model. The 100 activity centers  
 6253 were simulated on a state-space defined by an  $8 \times 8$  square within which the trap array was  
 6254 centered (thus the trap array is buffered by 2 units). Therefore, the density of individuals  
 6255 in this system is fixed at 100/64. In the following set of **R** commands we generate the  
 6256 data and then harvest the required data objects:

```
6257 ## simulate a complete data set (perfect detection)
6258 > data <- simSCR0(discard0=FALSE, rnd=2013)
6259   ## extract the objects that we need for analysis
6260 > y <- data$Y
```

---

<sup>1</sup>e.g., [http://en.wikipedia.org/wiki/Rectangle\\_method](http://en.wikipedia.org/wiki/Rectangle_method)

```

6261 > traplocs <- data$traplocs
6262 > nind <- nrow(y) ## in this case nind=N
6263 > J <- nrow(traplocs)
6264 > K <- data$K
6265 > xlim <- data$xlim
6266 > ylim <- data$ylim

```

6267 Now, we need to define the integration grid, say **G**, which we do with the following set of  
 6268 **R** commands (here, **delta** is the grid spacing):

```

6269 > delta <- .2
6270 > xg <- seq(xlim[1]+delta/2,xlim[2]-delta/2,by=delta)
6271 > yg <- seq(ylim[1]+delta/2,ylim[2]-delta/2,by=delta)
6272 > npix <- length(xg)           # valid for square state-space only
6273 > G <- cbind(rep(xg,npix),sort(rep(yg,npix)))
6274 > nG <- nrow(G)

```

6275 In this case, the integration grid is set up as a grid with spacing  $\delta = 0.2$  which produces,  
 6276 for our example, a  $40 \times 40$  grid of points for evaluating the integrand if the state-space  
 6277 buffer is set at 2. We note that the integration grid is set-up here to correspond exactly  
 6278 to the state-space used in simulating the data. However, in practice, we wouldn't know  
 6279 this, and our estimate of  $N$  (for the unknown case, see below) would be sensitive to choice  
 6280 of the extent of the integration grid. As we've discussed previously, density, which is  $N$   
 6281 standardized by the area of the state-space, will not be so sensitive in most cases.

6282 We are now ready to compute the conditional-on-s likelihood and carry out the  
 6283 marginalization described by Eq. 6.1.2. We need to do this by defining an **R** function  
 6284 that computes the likelihood for the integration grid, as a function of the data objects  
 6285 **y** and **traplocs** which were created above. However, it is a bit untidy to store the grid  
 6286 information in your workspace, and define the likelihood function in a way that depends  
 6287 on these things that exist in your workspace. Therefore, we build the **R** function so that  
 6288 it computes the integration grid *within* the function, thereby avoiding potential problems  
 6289 if our trapping grid locations change, or if we want to modify the state-space buffer easily.  
 6290 We therefore define the function, called **intlik1**, to which we pass the data objects and  
 6291 other information necessary to compute the marginal likelihood. This function is available  
 6292 in the **scrbook** package (use **?intlik1** at the **R** prompt). The code is reproduced here:

```

6293 intlik1 <- function(parm,y=y,X=traplocs, delta=.2, ssbuffer=2){
6294
6295   Xl <- min(X[,1]) - ssbuffer ## These lines of code are setting up the
6296   Xu <- max(X[,1]) + ssbuffer ## support for the integration which is
6297   Yu <- max(X[,2]) + ssbuffer ## the same as the state-space of "s"
6298   Yl <- min(X[,2]) - ssbuffer
6299   xg <- seq(Xl+delta/2,Xu-delta/2,,length=npix)
6300   yg <- seq(Yl+delta/2,Yu-delta/2,,length=npix)
6301   npix<- length(xg)
6302
6303   G <- cbind(rep(xg,npix),sort(rep(yg,npix)))

```

```

6304   nG <- nrow(G)
6305   D <- e2dist(X,G)
6306
6307   alpha0 <- parm[1]
6308   alpha1 <- exp(parm[2]) # alpha1 restricted to be positive here
6309
6310   probcap <- plogis(alpha0)*exp(-alpha1*D*D)
6311   Pm <- matrix(NA,nrow=nrow(probcap),ncol=ncol(probcap))
6312           # Frequency of all-zero encounter histories
6313   n0 <- sum(apply(y,1,sum)==0)
6314           # Encounter histories with at least 1 detection
6315   ymat <- y[apply(y,1,sum)>0,]
6316   ymat <- rbind(ymat,rep(0,ncol(ymat)))
6317   lik.marg <- rep(NA,nrow(ymat))
6318
6319   for(i in 1:nrow(ymat)){
6320       ## Next line: log conditional likelihood for ALL possible values of s
6321       Pm[1:length(Pm)] <- dbinom(rep(ymat[i,],nG),K,probcap[1:length(Pm)],
6322                                     log=TRUE)
6323       ## Next line: sum the log conditional likelihoods, exp() result
6324       ## same as taking the product
6325       lik.cond <- exp(colSums(Pm))
6326       ## Take the average value == computing marginal
6327       lik.marg[i] <- sum(lik.cond*(1/nG))
6328   }
6329   ## n0 = number of all-0 encounter histories
6330   nv <- c(rep(1,length(lik.marg)-1),n0)
6331   return( -1*(sum(nv*log(lik.marg)) ) )
6332 }
```

6333 We emphasize that this function (and subsequent) are not meant to be general-purpose  
 6334 routines for solving all of your SCR problems but, rather, they are meant for illustrative  
 6335 purposes – so you can see how the integrated likelihood is constructed and how we connect  
 6336 it to data and other information that is needed.

6337 The function `intlik1` accepts as input the encounter history matrix, `y`, the trap locations,  
 6338 `X`, and the state-space buffer. This allows us to vary the state-space buffer and easily  
 6339 evaluate the sensitivity of the MLE to the size of the state-space. Note that we have a  
 6340 peculiar handling of the encounter history matrix `y`. In particular, we remove the all-zero  
 6341 encounter histories from the matrix and tack-on a single all-zero encounter history as the  
 6342 last row which then gets weighted by the number of such encounter histories (`n0`). This is  
 6343 a bit long-winded and strictly unnecessary when  $N$  is known, but we did it this way be-  
 6344 cause the extension to the unknown- $N$  case is now transparent (as we demonstrate in the  
 6345 following section). The matrix `Pm` holds the log-likelihood contributions of each encounter  
 6346 frequency for each possible state-space location of the individual. The log contribu-  
 6347 tions are summed up and the result exponentiated on the next line, producing `lik.cond`, the  
 6348 conditional-on-s likelihood (Eq. 6.1.1 above). The marginal likelihood (`lik.marg`) sums  
 6349 up the conditional elements weighted by the probabilities [`s`] (Eq. 6.1.2 above).

6350     This is a fairly primitive function which doesn't allow much flexibility in the data  
 6351     structure. For example, it assumes that  $K$ , the number of replicates, is constant for each  
 6352     trap. Further, it assumes that the state-space is a square. We generalize this to some  
 6353     extent later in this chapter.

6354     Here is the **R** command for maximizing the likelihood using **nlm** (the function **optim**  
 6355     could also be used) and saving the results into an object called **frog**. The output is a list  
 6356     of the following structure and these specific estimates are produced using the simulated  
 6357     data set:

```
6358 # should take 15-30 seconds
6359
6360 > starts <- c(-2,2)
6361 > frog <- nlm(intlik1,starts,y=y,X=traplocs,delta=.1,ssbuffer=2,hessian=TRUE)
6362 > frog
6363
6364 $minimum
6365 [1] 297.1896
6366
6367 $estimate
6368 [1] -2.504824 2.373343
6369
6370 $gradient
6371 [1] -2.069654e-05 1.968754e-05
6372
6373 $hessian
6374 [,1]      [,2]
6375 [1,] 48.67898 -19.25750
6376 [2,] -19.25750 13.34114
6377
6378 $code
6379 [1] 1
6380
6381 $iterations
6382 [1] 11
```

6383     Details about this output can be found on the help page for **nlm**. We note briefly that  
 6384     **frog\$minimum** is the negative log-likelihood value at the MLEs, which are stored in the  
 6385     **frog\$estimate** component of the list. The order of the parameters is as they are defined  
 6386     in the likelihood function so, in this case, the first element (value =  $-2.504824$ ) is the  
 6387     logit transform of  $p_0$  and the second element (value =  $2.373343$ ) is the value of  $\alpha_1$  the  
 6388     “coefficient” on distance-squared. The Hessian is the observed Fisher information matrix,  
 6389     which can be inverted to obtain the variance-covariance matrix using the command:

```
6390 > solve(frog$hessian)
```

6391     It is worth drawing attention to the fact that the estimates are slightly different than  
 6392     the Bayesian estimates reported previously in Sec. 5.6. There are several reasons for this.  
 6393     First Bayesian inference is based on the posterior distribution and it is not generally the

case that the MLE should correspond to any particular value of the posterior distribution. If the prior distributions in a Bayesian analysis are uniform, then the (multivariate) mode of the posterior is the MLE, but note Bayesians almost always report posterior *means* and so there will typically be a discrepancy there. Secondly, we have implemented an approximation to the integral here and there might be a slight bit of error induced by that. We will evaluate that shortly. Third, the Bayesian analysis by MCMC is itself subject to some amount of Monte Carlo error which the analyst should always be aware of in practical situations. All of these different explanations are likely responsible for some of the discrepancy. Accounting for these, we see general consistency between the two estimates.

In summary, for the basic SCR model, computing the integrated likelihood is a simple task when  $N$  is known. Even for  $N$  unknown it is not too difficult, and we will do that shortly. However, if you can solve the known- $N$  problem then you should be able to do a real analysis, for example by considering different values of  $N$  and computing the results for each value and then making a plot of the log-likelihood or AIC and choosing the value of  $N$  that produces the best log-likelihood or AIC. As a homework problem we suggest that you can take the code given above and try to estimate  $N$  without modifying the code by just repeatedly applying it for different values of  $N$  in attempt to deduce the best value. We will formalize the unknown- $N$  problem next.

## 6.2 MLE WHEN N IS UNKNOWN

Here we build on the previous introduction to integrated likelihood but we consider now the case in which  $N$  is unknown. We will see that adapting the analysis based on the known- $N$  model is straightforward for the more general problem. The main distinction is that we don't observe the all-zero encounter history so we have to make sure we compute the probability for that encounter history, which we do by tacking a row of zeros onto the encounter history matrix. In addition, we include the number of such all-zero encounter histories (that is, the number of individuals *not* encountered) as an unknown parameter of the model. Call that unknown quantity  $n_0$ , so that  $N = n_0 + n$  where  $n$  is the number of unique individuals encountered. We will usually parameterize the likelihood in terms of  $n_0$  because optimization over a parameter space in which  $\log(n_0)$  is unconstrained is preferred to a parameter space in which  $N$  must be constrained  $N \geq n$ . With  $n_0$  unknown, we have to be sure to include a combinatorial term to account for the fact that, of the  $n$  observed individuals, there are  $\binom{N}{n}$  ways to realize a sample of size  $n$ . The combinatorial term involves the unknown  $n_0$  and thus it must be included in the likelihood. In evaluating the log-likelihood, we have to compute terms such as the log-factorial,  $\log(N!) = \log((n_0+n)!)$ . We do this in **R** by making use of the log-gamma function (`lgamma`) and the identity

$$\log(N!) = \text{lgamma}(N + 1).$$

Therefore, to compute the likelihood, we require the following 3 components: (1) The marginal probability of each  $\mathbf{y}_i$  as before,

$$[\mathbf{y}_i | \boldsymbol{\alpha}] = \int_{\mathcal{S}} [\mathbf{y}_i | \mathbf{s}_i, \boldsymbol{\alpha}] [\mathbf{s}_i] d\mathbf{s}_i.$$

6431 (2) We compute the probability of an all-0 encounter history:

$$\pi_0 = [\mathbf{y} = \mathbf{0} | \boldsymbol{\alpha}] = \int_{\mathcal{S}} \text{Binomial}(\mathbf{0} | \mathbf{s}_i, \boldsymbol{\alpha}) [\mathbf{s}_i] d\mathbf{s}_i$$

6432 (3) The combinatorial term:  $\binom{N}{n}$ . Then, the marginal likelihood has this form:

$$\mathcal{L}(\boldsymbol{\alpha}, n_0 | \mathbf{y}) = \frac{N!}{n! n_0!} \left\{ \prod_{i=1}^n [\mathbf{y}_i | \boldsymbol{\alpha}] \right\} \pi_0^{n_0}. \quad (6.2.1)$$

6433 This is discussed in Borchers and Efford (2008, p. 379) as the conditional-on- $N$  form of the  
6434 likelihood – we also call it the “binomial form” of the likelihood because of its appearance.

6435 Operationally, things proceed much as before: We compute the marginal probability  
6436 of each observed  $\mathbf{y}_i$ , i.e., by removing the latent  $\mathbf{s}_i$  by integration. In addition, we com-  
6437 pute the marginal probability of the “all-zero” encounter history  $\mathbf{y}_{n+1}$ , and make sure to  
6438 weight it  $n_0$  times. We accomplish this by “padding” the data set with a single encounter  
6439 history having  $y_{n+1,j} = 0$  for all traps  $j = 1, 2, \dots, J$ . Then we be sure to include the  
6440 combinatorial term in the likelihood or log-likelihood computation. We demonstrate this  
6441 shortly. To analyze a specific case, we’ll simulate our fake data set (simulated using the  
6442 parameters given above). To set some things up in our workspace we do this:

```
6443 ## Obtain a simulated data set
6444 > data <- simSCRO(discard0=TRUE, rnd=2013)
6445
6446 ## Extract the items we need for analysis
6447 > y <- data$Y
6448 > nind <- nrow(y)
6449 > traplocs <- data$traplocs
6450 > J <- nrow(traplocs)
6451 > K <- data$K
```

6452 Recall that these data are simulated by default with  $N = 100$ , on an  $8 \times 8$  unit state-  
6453 space representing the trap locations buffered by 2 units, although you can modify the  
6454 simulation script easily.

6455 As before, the likelihood is defined in the **R** workspace as an **R** function, **intlik2**,  
6456 which takes an argument being the unknown parameters of the model and additional  
6457 arguments as prescribed. In particular, we provide the encounter history matrix **y**, the  
6458 trap locations **traplocs**, the spacing of the integration grid (argument **delta**) and the  
6459 state-space buffer. Here is the new likelihood function:

```
6460 intlik2 <- function(parm,y=y,X=traplocs,delta=.3,ssbuffer=2){
6461
6462   Xl <- min(X[,1]) - ssbuffer
6463   Xu <- max(X[,1]) + ssbuffer
6464   Yu <- max(X[,2]) + ssbuffer
6465   Yl <- min(X[,2]) - ssbuffer
6466
6467   xg <- seq(Xl+delta/2,Xu-delta/2,delta)
```

```

6468     yg <- seq(Yl+delta/2,Yu-delta/2,delta)
6469     npix.x <- length(xg)
6470     npix.y <- plength(yg)
6471     area <- (Xu-Xl)*(Yu-Yl)/((npix.x)*(npix.y))
6472     G <- cbind(rep(xg,npix.y),sort(rep(yg,npix.x)))
6473     nG <- nrow(G)
6474     D <- e2dist(X,G)
6475     # extract the parameters from the input vector
6476     alpha0 <- parm[1]
6477     alpha1 <- exp(parm[2])
6478     n0 <- exp(parm[3]) # note parm[3] lives on the real line
6479     probcap <- plogis(alpha0)*exp(-alpha1*D*D)
6480     Pm <- matrix(NA,nrow=nrow(probcap),ncol=ncol(probcap))
6481     ymat <- rbind(y,rep(0,ncol(y)))
6482
6483     lik.marg <- rep(NA,nrow(ymat))
6484     for(i in 1:nrow(ymat)){
6485       Pm[1:length(Pm)] <- (dbinom(rep(ymat[i,],nG),K,probcap[1:length(Pm)],
6486                                     log=TRUE))
6487       lik.cond <- exp(colSums(Pm))
6488       lik.marg[i] <- sum(lik.cond*(1/nG) )
6489     }
6490     nv <- c(rep(1,length(lik.marg)-1),n0)
6491     ## part1 here is the combinatorial term.
6492     ## math: log(factorial(N)) = lgamma(N+1)
6493     part1 <- lgamma(nrow(y)+n0+1) - lgamma(n0+1)
6494     part2 <- sum(nv*log(lik.marg))
6495     return( -1*(part1+ part2) )
6496   }

```

6497 To execute this function for the data that we created with `simSCR0`, we execute the  
 6498 following command (saving the result in our friend `frog`). This results in the usual output,  
 6499 including the parameter estimates, the gradient, and the numerical Hessian which is useful  
 6500 for obtaining asymptotic standard errors (see below):

```

6501 > starts <- c(-2.5,0,4)
6502 > frog <- nlm(intlik2,starts,hessian=TRUE,y=y,X=traplocs,delta=.2,ssbuffer=2)
6503
6504 Warning message:
6505 In nlm(intlik2, starts, hessian = TRUE, y = y, X = traplocs, delta = 0.2, :
6506 NA/Inf replaced by maximum positive value
6507
6508 > frog
6509 $minimum
6510 [1] 113.5004
6511
6512 $estimate

```

```
6513 [1] -2.538333 0.902807 4.232810
6514
6515 [... additional output deleted ...]
```

6516 Executing `nlm` here usually produces one or more **R** warnings due to numerical calculations  
 6517 happening on extremely small or large numbers (calculation of  $p$  near the edge of the  
 6518 state-space), and they also happen if a poor parameterization is used which produces  
 6519 evaluations of the objective function beyond the boundary of the parameter space (e.g.,  
 6520  $n_0 < 0$ ). Such numerical warnings can often be minimized or avoided altogether by picking  
 6521 judicious starting values of parameters or properly transforming or scaling the parameters  
 6522 but, in general, they can be ignored. You will see from the `nlm` output that the algorithm  
 6523 performed satisfactory in minimizing the objective function. The estimate of population  
 6524 size,  $\hat{N}$ , for the state-space (using the default state-space buffer) is

```
6525 > Nhat <- nrow(y) + exp(4.2328) #### This is n + MLE of n0
6526 > Nhat
6527 [1] 110.9099
```

6528 Which differs from the data-generating value ( $N = 100$ ), as we might expect for a single  
 6529 realization. We usually will present an estimate of uncertainty associated with this MLE  
 6530 which we can obtain by inverting the Hessian. Note that  $\text{Var}(\hat{N}) = n + \text{Var}(\hat{n}_0)$ . Since  
 6531 we have parameterized the model in terms of  $\log(n_0)$  we use the delta method described  
 6532 in Williams et al. (2002, Appendix F4) (see also Ver Hoef, 2012) to obtain the variance  
 6533 on the scale of  $n_0$  as follows:

```
6534 > (exp(4.2328)^2)*solve(frog$hessian)[3,3]
6535 [1] 260.2033
6536
6537 > sqrt(260)
6538 [1] 16.12452
```

6539 Therefore, the asymptotic “Wald-type” confidence interval for  $N$  is  $110.91 \pm 1.96 \times 16.125 =$   
 6540  $(79.305, 142.515)$ . To report this in terms of density, we scale appropriately by the area  
 6541 of the prescribed state-space which is 64 units of area (i.e., an  $8 \times 8$  square). Our MLE  
 6542 of  $D$  is  $\hat{D} = 110.91/64 = 1.733$  individuals per square unit. To get the standard error  
 6543 for  $\hat{D}$  we need to divide the SE for  $\hat{N}$  by the area of the state-space, and so  $\text{SE}(\hat{D}) =$   
 6544  $(1/64) * 16.12452 = 0.252$ .

### 6545 6.2.1 Integrated likelihood under data augmentation

6546 The likelihood analysis developed in the previous sections is based on the likelihood in  
 6547 which  $N$  (or  $n_0$ ) is an explicit parameter. This is usually called the “full likelihood” or  
 6548 sometimes “unconditional likelihood” (Borchers et al., 2002) because it is the likelihood  
 6549 for all individuals in the population, not just those which have been captured, i.e., not that  
 6550 which is *conditional on capture*. It is also possible to express an alternative unconditional  
 6551 likelihood using data augmentation, replacing the parameter  $N$  with  $\psi$  (e.g., see Sec. 7.1.6  
 6552 Royle and Dorazio, 2008, for an example). We don’t go into detail here, but we note that  
 6553 the likelihood under data augmentation is a zero-inflated binomial mixture – precisely an

occupancy type model (Royle, 2006). Thus, while it is possible to carry out likelihood analysis of models under data augmentation, we primarily advocate data augmentation for Bayesian analysis.

### 6.2.2 Extensions

We have only considered basic SCR models with no additional covariates. However, in practice, we are interested in covariate effects including “behavioral response”, sex-specificity of parameters, and potentially others. Some of these can be added directly to the likelihood if the covariate is fixed and known for all individuals captured or not. An example is a behavioral response, which amounts to having a covariate  $x_{ik} = 1$  if individual  $i$  was captured prior to occasion  $k$  and  $x_{ik} = 0$  otherwise. For uncaptured individuals,  $x_{ik} = 0$  for all  $k$ . Royle et al. (2011b) called this a global behavioral response because the covariate is defined for all traps, no matter the trap in which an individual was captured. We could also define a *local* behavioral response which occurs at the level of the trap, i.e.,  $x_{ijk} = 1$  if individual  $i$  was captured in trap  $j$  prior to occasion  $k$ , etc... Trap-specific covariates such as trap type or status, or time-specific covariates such as date, are easily accommodated as well. As an example, Kéry et al. (2010) develop a model for the European wildcat *Felis silvestris* in which traps are either baited or not (a trap-specific covariate with only 2 values), and also encounter probability varies over time in the form of a quadratic seasonal response. We consider models with behavioral response or fixed covariates in Chapt. 7. The integrated likelihood routines we provided above can be modified directly for such cases, which we leave to the interested reader to investigate.

Sex-specificity is more difficult to deal with since sex is not known for uncaptured individuals (and sometimes not even for all captured individuals). To analyze such models, we do Bayesian analysis of the joint likelihood using data augmentation (Gardner et al., 2010b; Russell et al., 2012), discussed further in Chapt. 7. For such covariates (i.e., that are not fixed and known for all individuals), it is somewhat more challenging to do MLE based on the joint likelihood as we have developed above. Instead it is more conventional to use what is colloquially referred to as the “Huggins-Alho” type model which is one of the approaches taken in the software package **secr** (Efford, 2011a). We introduce the **secr** package in Sec. 6.5 below.

## 6.3 CLASSICAL MODEL SELECTION AND ASSESSMENT

In most analyses, one is interested in choosing from among various potential models, or ranking models, or something else to do with assessing the relative merits of a set of models. A good thing about classical analysis based on likelihood is we can apply Akaike Information Criterion (AIC) methods (Burnham and Anderson, 2002) without difficulty. AIC is convenient for assessing the relative merits of these different models although if there are only a few models it is not objectionable to use hypothesis tests or confidence intervals to determine importance of effects. A second model selection context has to do with choosing among various detection models, although, as a general rule, we don't recommend this application of model selection. This is because there is hardly ever (if at all) a rational subject-matter based reason motivating specific distance functions. As a result, we believe that doing too much model selection will invariably lead to over-fitting

and thus over-statement of precision. This is the main reason that we haven't loaded you down with a basket of models for detection probability so far, although we discuss many possibilities in Chapt. 7.

**Goodness-of-fit or model-checking** – For many standard capture-recapture models, it is possible to identify goodness-of-fit statistics based on the multinomial likelihood, (Cooch and White, 2006, Chapt. 5), and evaluate model adequacy using formal statistical tests. Similar strategies can be applied to SCR models using expected cell-frequencies based on the marginal distribution of the observations. Also, because computing MLEs is somewhat more efficient in many cases compared to Bayesian analysis, it is sometimes feasible to use bootstrap methods. At the present time, we don't know of any applications of goodness-of-fit testing for SCR models based on likelihood inference, although we discuss the use of Bayesian p-values for assessing model fit in Chapt. 8. An important practical problem in trying to evaluate goodness-of-fit is that, in realistic sample sizes, fit tests often lack the power to detect departures from the model under consideration and so they may not be generally useful in practice.

## 6.4 LIKELIHOOD ANALYSIS OF THE WOLVERINE CAMERA TRAPPING DATA

Here we compute the MLEs for the wolverine data using an expanded version of the function we developed in the previous section. To accommodate that each trap might be operational a variable number of nights, we provided an additional argument to the likelihood function (allowing for a vector  $\mathbf{K} = (K_1, \dots, K_J)$ ), which requires also a modification to the construction of the likelihood. In addition, we accommodate the state-space is a general rectangle, and we included a line in the code to compute the state-space area which we apply below for computing density. The more general function (`intlik3`) is given in the **R** package `scrbook`. Incidentally, this function also returns the area of the state-space for a given set of parameter values, as an attribute to the function value, which will be used in converting  $\hat{N}$  to  $\hat{D}$ . To use this function to obtain the MLEs for the wolverine camera trap study, we execute the following commands (note: these are in the help file and will execute if you type `example(intlik3)`):

```

6622 > library(scrbook)
6623 > data(wolverine)
6624
6625 > traps <- wolverine$traps
6626 > traplocs <- traps[,2:3]/10000
6627 > K.wolv <- apply(traps[,4:ncol(traps)],1,sum)
6628
6629 > y3d <- SCR23darray(wolverine$wcaps,traps)
6630 > y2d <- apply(y3d,c(1,2),sum)
6631
6632 > starts <- c(-1.5,0,3)
6633
6634 > wolv <- nlm(intlik3,starts,hessian=TRUE,y=y2d,K=K.wolv,X=traplocs,
6635           delta=.2,ssbuffer=2)
6636

```

```

6637 > wolv
6638 $minimum
6639 [1] 220.4313
6640
6641 $estimate
6642 [1] -2.8176120 0.2269395 3.5836875
6643
6644 [.... output deleted ....]

```

6645 Of course we're interested in obtaining an estimate of population size for the prescribed  
 6646 state-space, or density, and associated measures of uncertainty which we do using the delta  
 6647 method (Williams et al., 2002, Appendix F4). To do all of that we need to manipulate the  
 6648 output of `nlm` since we have our estimate in terms of  $\log(n_0)$ . We execute the following  
 6649 commands:

```

6650 > wolv <- nlm(intlik3,starts,hessian=TRUE,y=y2d,K=K.wolv,X=traplocs,delta=.2,
6651           ssbuffer=2)
6652 > Nhat <- nrow(y2d)+exp(wolv$estimate[3])
6653 > area <- attr(intlik3(starts,y=y2d,K=K.wolv,X=traplocs,delta=.2,ssbuffer=2),
6654           "SSarea")
6655 > Dhat <- Nhat/area
6656
6657 > Dhat
6658 [1] 0.5494947
6659
6660 > SE <- (1/area)*exp(wolv$estimate[3])*sqrt(solve(wolv$hessian)[3,3])
6661
6662 > SE
6663 [1] 0.1087073

```

6664 Our estimate of density is 0.55 individuals per “standardized unit” which is  $100 \text{ km}^2$ ,  
 6665 because we divided UTM coordinates by 10000. So this is about 5.5 individuals per  $1000 \text{ km}^2$ ,  
 6666 with a SE of around 1.09 individuals. This compares closely with 5.77 reported in  
 6667 Sec. 5.9 based on Bayesian analysis of the model.

#### 6668 6.4.1 Sensitivity to integration grid and state-space buffer

6669 The effect of approximating the integral by a discrete mesh of points is that it induces  
 6670 some numerical error in evaluation of the integral and, further, that error increases as the  
 6671 coarseness of the mesh increases. To evaluate the effect (or sensitivity) of the integration  
 6672 grid spacing, we obtained the MLEs for a state-space buffer of 2 (standardized units) and  
 6673 for integration grid with spacing  $\delta = .3, .2, .1, .05$ . The MLEs for these 4 cases including  
 6674 the relative runtime are given in Table 6.1. We see the results change only slightly as the  
 6675 integration grid changes. Conversely, the runtime on the platform of the day for the 4 cases  
 6676 increases rapidly. These runtimes could be regarded in relative terms, across platforms,  
 6677 for gaging the decrease in speed as the fineness of the integration grid increases.

6678 We studied the effect of the state-space buffer on the MLEs, using a fixed  $\delta = .2$  for  
 6679 all analyses. We used state-space buffers of 1 to 4 units stepped by .5. As we can see

**Table 6.1.** Runtime and MLEs for different integration grid resolutions for the wolverine camera trapping data.

$\delta$	Estimates			
	runtime (s)	$\hat{\alpha}_0$	$\hat{\alpha}_1$	$\widehat{\log(n_0)}$
0.30	9.9	-2.819786	1.258468	3.569731
0.20	32.3	-2.817610	1.254757	3.583690
0.10	115.1	-2.817570	1.255112	3.599040
0.05	407.3	-2.817559	1.255281	3.607158

6680 (Table 6.2), the estimates of  $D$  stabilize rapidly and the incremental difference is within  
 6681 the numerical error associated with approximating the integral.

**Table 6.2.** Results of the effect of the state-space buffer on the MLE. Given here are the state-space buffer, area of the state-space (area), the MLE of  $N$  ( $\hat{N}$ ) for the prescribed state-space and the corresponding MLE of density ( $\hat{D}$ ).

Buffer	Area	$\hat{N}$	$\hat{D}$
1.0	66.98212	37.73338	0.5633352
1.5	84.36242	46.21008	0.5477567
2.0	103.74272	57.00617	0.5494956
2.5	125.12302	69.03616	0.5517463
3.0	148.50332	82.17550	0.5533580
3.5	173.88362	96.44018	0.5546249
4.0	201.26392	111.83524	0.5556646

#### 6682 6.4.2 Using a habitat mask (Restricted state-space)

6683 In Sec. 5.10 we used a discrete representation of the state-space in order to have control  
 6684 over its extent and shape. This makes it easy to do things like clip out non-habitat, or  
 6685 create a *habitat mask* which defines suitable habitat. Clearly that formulation of the model  
 6686 is relevant to the calculation of the marginal likelihood in the sense that the discrete state-  
 6687 space is equivalent to the integration grid. Thus, for example, we could easily compute  
 6688 the MLE of parameters under some model with a restricted state-space merely by creating  
 6689 the required state-space at whatever grid resolution is desired, and then inputting that  
 6690 state-space into the likelihood function above, instead of computing it within the function.  
 6691 We can easily create an explicit state-space grid for integration from arbitrary polygons or  
 6692 GIS shapefiles which we demonstrate here. Our approach is to create the integration grid  
 6693 (or state-space grid) outside of the likelihood evaluation, and then determine which points  
 6694 of the grid lie in the polygon defined by the shapefile using functions in the **R** packages **sp**  
 6695 and **maptools**. For each point in the state-space grid (object **G** in the code below which is  
 6696 assumed to exist), we determine whether it is inside the polygon<sup>2</sup>, identifying such points

<sup>2</sup>We perform this check using the `over` function. This function takes as its second argument (among others) an object of the class “`SpatialPolygons`” or “`SpatialPolygonsDataFrame`”, which

6697 with a value of `mask=1` and `mask=0` for points that are *not* in the polygon. We load the  
 6698 shapefile which originates by an application of the `readShapeSpatial` function. We have  
 6699 saved the result into an **R** data object called `SSp` which is in the `scrbook` package. Here  
 6700 are the **R** commands for doing this (see the helpfile `?intlik4`):

```
6701 > library(mapproj)
6702 > library(sp)
6703 > library(scrbook)
6704
6705 ##### If we have the .shp file in place, we would use this command:
6706 ##### SSp <- readShapeSpatial('Sim_Polygon.shp')
6707 ##### The object SSp is in data(fakeshapefile)
6708 > data(fakeshapefile)
6709 > Pcoord <- SpatialPoints(G)
6710 > PinPoly <- over(Pcoord,SSp) #### determine if each point is in polygon
6711 > mask <- as.numeric(!is.na(PinPoly[,1])) ## convert to binary 0/1
6712 > G <- G[mask==1,]
```

6713 We created the function `intlik4` which accepts the integration grid as an explicit argument,  
 6714 and this function is also available in the package `scrbook`.

6715 We apply this modification to the wolverine camera trapping study. Royle et al.  
 6716 (2011b) created 2, 4 and 8 km state-space grids so as to remove “non-habitat” (mostly  
 6717 ocean, bays, and large lakes). We previously analyzed the model using **JAGS** and **Win-**  
**6718 BUGS** in Chapt. 5. To set up the wolverine data and fit the model using maximum  
 6719 likelihood we execute the following commands:

```
6720 > library(scrbook)
6721 > data(wolverine)
6722
6723 > traps <- wolverine$wtraps
6724 > traplocs <- traps[,2:3]/10000
6725 > K.wolv <- apply(traps[,4:ncol(traps)],1,sum)
6726
6727 > y3d <- SCR23darray(wolverine$wcaps,traps)
6728 > y2d <- apply(y3d,c(1,2),sum)
6729 > G <- wolverine$grid2/10000
6730
6731 > starts <- c(-1.5,0,3)
6732 > wolv <- nlm(intlik4, starts, y=y2d, K=K.wolv, X=traplocs, G=G)
6733
6734 > wolv
```

---

can hold additional information for each polygon, and the output value of the function differs slightly for these two classes: if using a “`SpatialPolygons`” object, the function returns a vector of length equal to the number of points (e.g., in the example above), but if using a “`SpatialPolygonsDataFrame`” it returns a data frame (e.g., see Sec. 17.7 in Chapt. 17). If you use the `over` function, make sure you know the class of your second argument so that when processing the function output you index it correctly.

**Table 6.3.** Maximum likelihood estimates (MLEs) and asymptotic standard errors (SE) for the wolverine camera trapping data using 2, 4 and 8 km state-space grids.

Grid	$\alpha_0$	$\alpha_1$	$\log(n_0)$	$N$	SE	D(1000)	SE
2	-3.00	1.27	4.11	81.98	16.31	8.31	1.65
4	-2.99	1.34	4.16	84.88	16.76	8.57	1.69
8	-3.05	1.08	4.06	78.89	15.31	7.85	1.52

```

6735
6736 $minimum
6737 [1] 225.8355
6738
6739 $estimate
6740 [1] -2.9955424 0.2350885 4.1104757
6741
6742 [... some output deleted ...]

```

6743 Next we convert the parameter estimates to estimates of total population size for the  
6744 prescribed state-space, and then obtain an estimate of density (per 1000 km<sup>2</sup>) using the  
6745 area computed as the number of pixels in the state-space grid, G, multiplied by the area  
6746 per grid cell. In the present case (the calculation above) we used a state-space grid with 2  
6747 km × 2 km pixels. Finally, we compute a standard errors using the delta approximation:

```

6748 > area <- nrow(G)*4
6749 # Nhat = n (observed) + MLE of n0 (not observed)
6750 > Nhat <- 21 + exp(wolv$estimate[3])
6751 > SE <- exp(wolv$estimate[3])*sqrt(solve(wolv$hessian)[3,3])
6752 > D <- (Nhat/(nrow(G)*area))*1000
6753 > SE.D <- (SE/(nrow(G)*area))*1000

```

6754 We did this for each the 2 km, 4 km and 8 km state-space grids which produced the  
6755 estimates summarized in Table 6.3. These estimates compare with the 8.6 (2 km grid)  
6756 and 8.2 (8 km grid) reported in Royle et al. (2011b) based on a clipped state-space as  
6757 described in Sec. 5.10.

## 6.5 DENSITY AND THE R PACKAGE SECR

6758 **DENSITY** is a software program developed by Efford (2004) for fitting spatial capture-  
6759 recapture models based mostly on classical maximum likelihood estimation and related  
6760 inference methods. Efford (2011a) has also released an **R** package called **secr**, that con-  
6761 tains much of the functionality of **DENSITY** but also incorporates new models and  
6762 features. Here, we briefly introduce the **secr** package which we prefer to use over **DEN-**  
6763 **SITY**, because it allows us to remain in the **R** environment for data processing and  
6764 summarization. We provide a brief introduction to **secr** and some of its capabilities here,  
6765 and we also use it for doing some analysis in other parts of this book. We believe that **secr**  
6766 will be sufficient for many (if not most) of the SCR problems that one might encounter.  
6767 It provides a flexible analysis platform, with a large number of summary features, and

6768 “publication ready” output. Its user-interface is clean and intuitive to **R** users, and it has  
 6769 been stable, efficient and reliable in the (fairly extensive) evaluations that we have done.

6770 To install and run models in **secr**, you must download the package and load it in **R**.

```
6771 > install.packages("secr")
6772 > library(secr)
```

6773 **secr** allows the user to simulate data and fit a suite of models with various detection func-  
 6774 tions and covariate responses. It also contains a number of helpful constructor functions  
 6775 for creating objects of the proper class that are recognized by other **secr** functions. We  
 6776 provide a brief overview of the capabilities here, but the **secr** help manual can be accessed  
 6777 with the command:

```
6778 > RShowDoc("secr-manual", package = "secr")
```

6779 We note that **secr** has many capabilities that we will not cover or do so only sparingly.  
 6780 We encourage you to read through the manual, the extensive documentation, and the  
 6781 vignettes, in order to get a better understanding of what the package is capable of. We  
 6782 also cover certain capabilities of **secr** in other chapters.

6783 The main model-fitting function in **secr** is called **secr.fit**, which makes use of the  
 6784 standard **R** model specification framework with tildes. As an example, the equivalent of  
 6785 the basic model SCR0 is fitted as follows:

```
6786 > secr.fit(capturedata, model = list(D ~ 1, g0 ~ 1, sigma ~ 1),
6787   buffer = 20000)
```

6788 where **capturedata** is the object created by **secr** containing the encounter history data  
 6789 and the trap information, and the model expression  $g0^1$  indicates the intercept-only (i.e.,  
 6790 constant) model. Note that we use  $p_0$  for the baseline encounter probability parameter,  
 6791 which is  $g_0$  in **secr** notation. A number of possible models for encounter probability can  
 6792 be fitted including both pre-defined variables (e.g., **t** and **b** corresponding to “time” and  
 6793 “behavior”), and user-defined covariates of several kinds. For example, to include a global  
 6794 behavioral response, this would be written as  $g0^1b$ . The discussion of this (global versus  
 6795 local trap-specific behavioral response) and other covariates is developed more in Chapt.  
 6796 7. We can also model covariates on density in **secr**, which we discuss in Chapt. 11. It  
 6797 is important to note that **secr** requires the buffer distance to be defined in meters and  
 6798 density will be returned as number of animals per hectare. Thus to make comparisons  
 6799 between **secr** and output from other programs, we will often have to convert the density  
 6800 to the same units.

6801 Before we can fit the models, the data must first be packaged properly for **secr**.  
 6802 We require data files that contain two types of information: trap layout (location and  
 6803 identification information for each trap), which is equivalent to the trap deployment file  
 6804 (TDF) described in Sec. 5.9 and the capture data file containing sampling *session*, animal  
 6805 identification, trap occasion, and trap location, equivalent in information content to the  
 6806 encounter data file (EDF). Sample session can be thought of as primary period identifier  
 6807 in a robust design like framework – it could represent a yearly sample or multiple sample  
 6808 periods within a year, each of them producing data on a closed population. We discuss  
 6809 “multi-session” models in more detail below, in Sec. 6.5.4 and Chapt. 14.

6810 There are three important constructor functions that help package-up your data for  
 6811 use in **secr**: **read.traps**, **make.capthist** and **read.mask**. We provide a brief description  
 6812 of each here, but apply them to our wolverine camera trapping data in the next section:

6813 (1) **read.traps**: This function points to an external file or **R** data object containing the  
 6814 trap coordinates, and other information, and also requires specification of the type of  
 6815 encounter devices (described in the next section). A typical application of this function  
 6816 looks like the following, invoking the **data=** option when there is an existing **R** object  
 6817 containing the trap information:

6818 > trapfile <- **read.traps**(**data=traps**, **detector="proximity"**)

6819 (2) **make.capthist**: This function takes the EDF and combines it with trap information,  
 6820 and the number of sampling occasions. A typical application looks like this:

6821 > capturedata <- **make.capthist**(**enc.data**, **trapfile**, **fmt="trapID"**,  
 6822 **noccasions=165**)

6823 See **?make.capthist** for definition of distinct file formats. Specifying **fmt = trapID** is  
 6824 equivalent to our EDF format.

6825 (3) **read.mask**: If there is a habitat mask available (as described in sec. 6.4.2), then this  
 6826 function will organize it so that **secr.fit** knows what to do with it. The function  
 6827 accepts either an external file name (see **?read.mask** for details of the structure) or a  
 6828  $NG \times 2$  **R** object, say **mask.coords**, containing the coordinates of the mask. A typical  
 6829 application looks like the following:

6830 > grid <- **read.mask**(**data=mask.coords**)

6831 These constructor functions produce output that can then be used in the fitting of models  
 6832 using **secr.fit**.

### 6833 6.5.1 Encounter device types and detection models

6834 The **secr** package requires that you specify the type of encounter device. Instead of  
 6835 describing models by their statistical distribution (Bernoulli, Poisson, etc..), **secr** uses  
 6836 certain operational classifications of detector types including ‘proximity’, ‘multi’, ‘single’,  
 6837 ‘polygon’ and ‘signal’. For camera trapping/hair snares we might consider ‘proximity’  
 6838 detectors or ‘count’ detectors. The ‘proximity’ detector type allows, at most, one detection  
 6839 of each individual at a particular detector on any occasion (i.e., it is equivalent to what  
 6840 we call the Bernoulli or binomial encounter process model, or model SCR0). The ‘count’  
 6841 detector designation allows repeat encounters of each individual at a particular detector  
 6842 on any occasion. There are other detector types that one can select such as: ‘polygon’  
 6843 detector type which allows for a trap to be a sampled polygon (Royle and Young, 2008)  
 6844 which we discuss further in Chapt. 15, and ‘signal’ detector which allows for traps that  
 6845 have a strength indicator, e.g., acoustic arrays (Dawson and Efford, 2009). The detector  
 6846 types ‘single’ and ‘multi’ refer to traps that retain individuals, thus precluding the ability  
 6847 for animals to be captured in other traps during the sampling occasion. The ‘single’ type  
 6848 indicates trap that can only catch one animal at a time (single-catch traps), while ‘multi’  
 6849 indicates traps that may catch more than one animal at a time (multi-catch). These are  
 6850 both variations of the multinomial encounter models described in Chapt. 9.

6851 As with all SCR models, **secr** fits an encounter probability model (“detection function”  
 6852 in **secr** terminology relating the probability of encounter to the distance of a detector from  
 6853 an individual activity center. **secr** allows the user to specify one of a variety of detection  
 6854 functions including the commonly used half-normal (“Gaussian”), hazard rate (“Gaussian  
 6855 hazard”), and (negative) exponential models. There are 12 different functions as of version  
 6856 2.3.1 (see Table 7.1 in Chapt. 7), but some are only available for simulating data. The  
 6857 different detection functions are defined in the **secr** manual and can be found by calling  
 6858 the help function for the detection function:

6859 > ?detectfn

6860 Most of the detection functions available in **secr** contain some kind of a scale parameter  
 6861 which is usually labeled  $\sigma$ . The units of this parameter default to meters in the **secr**  
 6862 output. We caution that the meaning of this parameter depends on the specific detection  
 6863 model being used, and it should not be directly compared as a measure of home-range size  
 6864 across models. Instead, as we noted in Sec. 5.4 most encounter probability models imply  
 6865 a model of space-usage and fitted encounter models should be converted to a common  
 6866 currency such as “area used.”

### 6867 6.5.2 Analysis using the **secr** package

6868 To demonstrate the use of the **secr** package, we will show how to do the same analysis on  
 6869 the wolverine study as shown in Sec. 5.9. To use the **secr** package, the data need to be  
 6870 formatted in a similar but slightly different manner than we use in **WinBUGS**.

6871 For example, in Sec. 5.9 we introduced a standard data format for the encounter data  
 6872 file (EDF) and trap deployment file (TDF). The EDF shares the same format as that used  
 6873 by the **secr** package with 1 row for every encounter observation and 4 columns representing  
 6874 trap session (‘Session’), individual identity (‘ID’), sample occasion (‘Occasion’), and trap  
 6875 identity (‘trapID’). For a standard closed population study that takes place during a single  
 6876 season, the ‘Session’ column in our case is all 1’s, to indicate a single primary sampling  
 6877 occasion. In addition to providing the encounter data file (EDF), we must tell **secr** infor-  
 6878 mation about the traps, which is formed as a matrix with column labels ‘trapID’, ‘x’ and  
 6879 ‘y’, the last two being the coordinates of each trap, with additional columns representing  
 6880 the operational state of each trap during each occasion (1=operational, 0=not).

6881 We demonstrate these differences now by walking through an analysis of the wolverine  
 6882 camera trapping data using **secr**. To read in the trap locations and other related infor-  
 6883 mation, we make use of the constructor function **read.traps** which also requires that we  
 6884 specify the detector type. The detector type is important because it will determine the  
 6885 likelihood that **secr** will use to fit the model. Here, we have selected “proximity” which  
 6886 corresponds to the Bernoulli encounter model in which individuals are captured at most  
 6887 once in each trap during each sampling occasion:

```
6888 > library(secr)
6889 > library(scrbook)
6890 > data(wolverine)
6891 > traps <- as.matrix(wolverine$wtraps)
```

---

```

6893 > dimnames(traps) <- list(NULL,c("trapID","x","y",paste("day",1:165,sep="")))
6894 > traps1 <- as.data.frame(traps[,1:3])
6895 > trapfile1 <- read.traps(data=traps1,detector="proximity")

```

6896     Here we note that trap coordinates are extracted from the wolverine data but we do  
6897     not scale them. This is because **secr** defaults to coordinate scaling of meters which is  
6898     the extant scaling of the wolverine trap coordinates. Note that we add a 'trapID' column  
6899     to the trap coordinates and provide appropriate column labels to the 'traps' matrix. An  
6900     important aspect of the wolverine study is that while the camera traps were operated over  
6901     a 165 day period, each trap was operational during only a portion of that period. We need  
6902     to provide the trap operation information which is contained in the columns to the right  
6903     of the trap coordinates in our standard trap deployment file (TDF). Unfortunately, this is  
6904     less easy to do in **secr**<sup>3</sup>, which requires an external file with a single long string of 1's and  
6905     0's indicating the days in which each trap was operational (1) or not (0). The **read.traps**  
6906     function will not allow for this information on trap operation if the data exists as an **R**  
6907     object – instead, we can create this external file and then read it back in with **read.traps**  
6908     using these commands:

```

6909 > hold <- rep(NA,nrow(traps))
6910 > for(i in 1:nrow(traps)){
6911 >   hold[i] <- paste(traps[i,4:ncol(traps)],collapse="")
6912 > }
6913 > traps1 <- cbind(traps[,1:3],"usage"=hold)
6914
6915 > write.table(traps1, "traps.txt", row.names=FALSE, col.names=FALSE)
6916 > trapfile2 <- read.traps("traps.txt",detector="proximity")

```

6917 These operations can be accomplished using the function **scr2secr** which is provided in  
6918 the **R** package **scrbook**.

6919 After reading in the trap data, we now need to create the encounter matrix or array  
6920 using the **make.capthist** command, where we provide the capture histories in EDF format,  
6921 which is the existing format of the data input file **wcaps**. In creating the capture history,  
6922 we provide also the trapfile created previously, the format (e.g., here EDF format is  
6923 **fmt= "trapID"**), and finally, we provide the number of occasions.

```

6924 #
6925 # Grab the encounter data file and format it:
6926 #
6927 wolv.dat <- wolverine$wcaps
6928 dimnames(wolv.dat) <- list(NULL,c("Session","ID","Occasion","trapID"))
6929 wolv.dat <- as.data.frame(wolv.dat)
6930 wolvcapt2 <- make.capthist(wolv.dat,trapfile2,fmt="trapID",noccasions=165)

```

6931 We also set up a habitat mask using the  $2 \times 2$  km grid which we used previously in the  
6932 analysis of the wolverine data and then pass the relevant objects to **secr.fit** as follows:

---

<sup>3</sup>as of v. 2.3.1

```

6933 #
6934 # Grab the habitat mask (2 x 2 km) and format it:
6935 #
6936 gr2 <- (as.matrix(wolverine$grid2))
6937 dimnames(gr2) <- list(NULL,c("x","y"))
6938 gr2 <- read.mask(data=gr2)
6939 #
6940 # To fit the model we use secr.fit:
6941 #
6942 wolv.secr2 <- secr.fit(wolvcapt2,model=list(D ~ 1, g0 ~ 1, sigma ~ 1),
6943                         buffer=20000,mask=gr2)

```

6944 We are using the “proximity detector” model (SCR0), so we do not need to make any  
 6945 specifications in the command line because we have specified the detector type using the  
 6946 constructor function `read.traps`, except to provide the buffer size (in meters). To specify  
 6947 different models, you can change the default model `D~1`, `g0~1`, `sigma~1`. We provide all  
 6948 of these commands and additional analyses in the `scrbook` package with the function called  
 6949 `secr_wolverine`. Printing the output object produces the following (slightly edited):

```

6950 > wolv.secr2
6951
6952 secr 2.3.1, 15:52:45 29 Aug 2012
6953
6954 Detector type      proximity
6955 Detector number    37
6956 Average spacing    4415.693 m
6957 x-range            593498 652294 m
6958 y-range            6296796 6361803 m
6959 N animals          : 21
6960 N detections       : 115
6961 N occasions        : 165
6962 Mask area          : 987828.1 ha
6963
6964 Model              : D ~ 1 g0 ~ 1 sigma ~ 1
6965 Fixed (real)       : none
6966 Detection fn       : halfnormal
6967 Distribution        : poisson
6968 N parameters       : 3
6969 Log likelihood     : -602.9207
6970 AIC                : 1211.841
6971 AICc               : 1213.253
6972
6973 Beta parameters (coefficients)
6974           beta   SE.beta      lcl      ucl
6975 D      -9.390124 0.22636698 -9.833795 -8.946452
6976 g0     -2.995611 0.16891982 -3.326688 -2.664535
6977 sigma  8.745547 0.07664648  8.595323  8.895772

```

```

6978
6979 Variance-covariance matrix of beta parameters
6980          D      g0      sigma
6981 D      0.0512420110 -0.0004113326 -0.003945371
6982 g0     -0.0004113326  0.0285339045 -0.006269477
6983 sigma  -0.0039453711 -0.0062694767  0.005874683
6984
6985 Fitted (real) parameters evaluated at base levels of covariates
6986      link   estimate   SE.estimate      lcl      ucl
6987 D      log 8.354513e-05 1.915674e-05 5.360894e-05 1.301982e-04
6988 g0    logit 4.762453e-02 7.661601e-03 3.466689e-02 6.509881e-02
6989 sigma  log 6.282651e+03 4.822512e+02 5.406315e+03 7.301037e+03

```

6990 The object returned by `secr.fit` provides extensive default output when printed.  
6991 Much of this is basic descriptive information about the model, the traps, or the encounter  
6992 data. We focus here on the parameter estimates. Under the fitted (real) parameters, we  
6993 find  $D$ , the density, given in units of individuals/hectare (1 hectare = 10000  $m^2$ ). To  
6994 convert this into individuals/1000 km<sup>2</sup>, we multiply by 100000, thus our density estimate  
6995 is 8.35 individuals/1000 km<sup>2</sup>. The parameter  $\sigma$  is given in units of meters, and so this  
6996 corresponds to 6.283 km. Both of these estimates are very similar to those obtained in  
6997 our likelihood analysis summarized in Table 6.3 which, for the 2 × 2 km grid, we obtained  
6998  $\hat{D} = 8.31$  with a SE of  $100000 \times 1.915674e - 05 = 1.9156$  and, accounting for the scale  
6999 difference (1 unit = 10000 m in the previous analysis),  $\hat{\sigma} = \sqrt{1/(2\hat{\alpha}_1)} * 10000 = 6.289$   
7000 km. The difference in the MLE between Table 6.3 and those produced by `secr` could be  
7001 due to subtle differences in internal tuning of optimization algorithms, starting values or  
7002 other numerical settings. In addition, the likelihood is based on a Poisson prior for  $N$  (see  
7003 the next section). On the other hand, the SE is slightly larger based on `secr` which is due  
7004 to a subtle difference in the interpretation of  $D$  under the `secr` model (See below).

### 7005 6.5.3 Likelihood analysis in the `secr` package

7006 The `secr` package does likelihood analysis of SCR models for most classes of models  
7007 as developed by Borchers and Efford (2008). Their formulation deviates slightly from  
7008 the binomial form we presented in Sec. 6.2 above (though Borchers and Efford (2008)  
7009 also mention the binomial form). Specifically, the likelihood that `secr` implements is that  
7010 based on removing  $N$  from the likelihood by integrating the binomial likelihood (Eq. 6.2.1  
7011 above) over a Poisson prior for  $N$  – what we will call the *Poisson-integrated likelihood* as  
7012 opposed to the conditional-on- $N$  (*binomial-form*) considered previously.

7013 To develop the Poisson-integrated likelihood we compute the marginal probability of  
7014 each  $\mathbf{y}_i$  and the probability of an all-0 encounter history,  $\pi_0$ , as before, to arrive at the  
7015 marginal likelihood in the binomial-form:

$$\mathcal{L}(\boldsymbol{\alpha}, n_0 | \mathbf{y}) = \frac{N!}{n! n_0!} \left\{ \prod_i [\mathbf{y}_i | \boldsymbol{\alpha}] \right\} \pi_0^{n_0}$$

7016 Now, what Borchers and Efford (2008) do is assume that  $N \sim \text{Poisson}(\Lambda)$  and they do a

7017 further level of marginalization over this prior distribution:

$$\sum_{n_0=0}^{\infty} \frac{N!}{n_0! n_0!} \left\{ \prod_i [\mathbf{y}_i | \boldsymbol{\alpha}] \right\} \pi_0^{n_0} \frac{\exp(-\Lambda) \Lambda^N}{N!}$$

7018 In Chapt. 11 we write  $\Lambda = \mu ||\mathcal{S}||$  where  $||\mathcal{S}||$  is the area of the state-space, and  $\mu$  is the  
 7019 density (“intensity”) of the point process. Carrying out the summation above produces  
 7020 exactly this marginal likelihood:

$$\mathcal{L}_2(\boldsymbol{\alpha}, \Lambda | \mathbf{y}) = \left\{ \prod_i [\mathbf{y}_i | \boldsymbol{\alpha}] \right\} \Lambda^n \exp(-\Lambda(1 - \pi_0))$$

7021 which is Eq. 2 of Borchers and Efford (2008) except for notational differences. It also  
 7022 resembles the binomial-form of the likelihood in Eq. 6.2.1 with  $\Lambda^n \exp(-\Lambda\pi_0)$  replacing  
 7023 the combinatorial term and the  $\pi_0^{n_0}$  term. We emphasize there are two marginalizations  
 7024 going on here: (1) the integration to remove the latent variables  $\mathbf{s}$ ; and, (2) summation  
 7025 to remove the parameter  $N$ . We provide a function for computing this in the **scrbook**  
 7026 package called **intlik3Poisson**. The help file for that function shows how to conduct a  
 7027 small simulation study to compare the MLE under the Poisson-integrated likelihood with  
 7028 that from the binomial form.

7029 The essential distinction between our MLE and Borchers and Efford as implemented in  
 7030 **secr** is whether you keep  $N$  in the model or remove it by integration over a Poisson prior.  
 7031 If you have prescribed a state-space explicitly with a sufficiently large buffer, then we  
 7032 imagine there should be hardly any difference at all between the MLEs obtained by either  
 7033 the Poisson-integrated likelihood or the binomial-form of the likelihood which retains  $N$ .  
 7034 There is a subtle distinction in the sense that under the binomial form, we estimate the  
 7035 realized population size  $N$  for the state-space whereas, for the Poisson-integrated form we  
 7036 estimate the *prior* expected value which would apply to a hypothetical new study of a  
 7037 similar population (see Sec. 5.7.3).

7038 Both models (likelihoods) assume  $\mathbf{s}$  is uniformly distributed over space, but for the  
 7039 binomial model we make no additional assumption about  $N$  whereas we assume  $N$  is  
 7040 Poisson using the formulation in **secr** from (Borchers and Efford, 2008). Using data  
 7041 augmentation we could do a similar kind of integration but integrate  $N$  over a binomial  
 7042 ( $M, \psi$ ) prior – which we referred to as the binomial-integrated likelihood in Sec. 4.2.4.  
 7043 So obviously the two approaches (data augmentation and Poisson-integrated likelihood)  
 7044 are approximately the same as  $M$  gets large. However, doing a Bayesian analysis by  
 7045 MCMC, we obtain an estimate of both  $N$ , the *realized population size*, and the parameter  
 7046 controlling its expected value  $\psi$  which are, in fact, both identifiable from the data even  
 7047 using likelihood analysis (Royle et al., 2007). That said we can integrate  $N$  out completely  
 7048 and just estimate  $\psi$  as we noted in Sec. 6.2.1 above.

#### 7049 6.5.4 Multi-session models in **secr**

7050 In practice we will often deal with SCR data that have some meaningful stratification or  
 7051 group structure. For example, we might conduct mist-netting of birds on  $K$  consecutive  
 7052 days, repeated, say,  $T$  times during a year, or perhaps over  $T$  years. Or we might collect  
 7053 data from  $R$  distinct trapping grids. In these cases, we have  $T$  or  $R$  groups which we might

7054 reasonably regard as being samples of independent populations. While the groups might  
 7055 be distinct sites, year, or periods within years, they could also be other biological groups  
 7056 such as sex or age. Conveniently, **secr** fits a specific model for stratified populations –  
 7057 referred to as *multi-session* models. These models build on the Poisson assumption which  
 7058 underlies the integrated likelihood used in **secr** (as described in the previous section). To  
 7059 understand the technical framework, let  $N_g$  be the population size of group  $g$  and *assume*

$$N_g \sim \text{Poisson}(\Lambda_g).$$

7060 Naturally, we model group-specific covariates on  $\Lambda_g$ :

$$\log(\Lambda_g) = \beta_0 + \beta_1 z_g$$

7061 where  $z_g$  is some group-specific covariate such as a categorical index to the group, or a  
 7062 trend variable, or a spatial covariate, such as treatment effect or habitat structure, if the  
 7063 groups represent spatial units. Under this model, we can marginalize *all*  $N_g$  parameters  
 7064 out of the likelihood to concentrate the likelihood on the parameters  $\beta_0$  and  $\beta_1$  precisely  
 7065 as discussed in the previous section. This Poisson hierarchical model is the basis of the  
 7066 multi-session models in **secr**.

7067 To implement a multi-session model (or stratified population model) in **secr**, we pro-  
 7068 vide the relevant stratification information in the ‘Session’ variable of the input encounter  
 7069 data file (EDF). If ‘Session’ has multiple values then a “multi-session” object is created  
 7070 by default and session-specific variables can be described in the model. For example, if  
 7071 the session has 2 values for males and females then we have sex-specific densities , and  
 7072 baseline encounter probability  $p_0$  ( $g_0$  in **secr**) by just doing this (see Chapt. 8 for the **R**  
 7073 code to set this up):

```
7074 > out <- secr.fit(capdata, model=list(D ~ session, g0 ~ session, sigma^ 1),  

  7075           buffer=20000)
```

7076 More detailed analysis is given in Sec. 8.1 where we fit a number of different models and  
 7077 apply methods of model selection to obtain model-averaged estimates of density.

7078 We can also easily implement stratified population models in the various **BUGS** en-  
 7079 gines using data augmentation (Converse and Royle, 2012; Royle and Converse, in review)  
 7080 which we discuss, with examples, in Chapt. 14.

### 7081 6.5.5 Some additional capabilities of **secr**

7082 The **secr** package has capabilities to do a complete analysis of SCR data sets, including  
 7083 model fitting, selection, and many summary analyses. In the previous sections, we’ve  
 7084 given a basic overview, and we do more in later chapters of this book. Here we mention a  
 7085 few of these other capabilities that you should know about as you use **secr**. Of course, you  
 7086 should skim through the associated documentation (**?secr**) to see more of what’s available.

#### 7087 Alternative observation models

7088 **secr** fits a wide range of alternative observation models besides the Bernoulli encounter  
 7089 model, including multinomial encounter models for “multi-catch” and “single catch” traps,  
 7090 models for sound attenuation from acoustic detection devices, and many others. We  
 7091 discuss many of these other methods in Chapt. 9 and elsewhere in the book.

7092 **Summary statistics**

7093 `secr` provides a useful default summary of the data, but it also has summary statistics  
 7094 about animal movement including mean-maximum distance moved (the function `MMDM`).  
 7095 For example, see the help page `?MMDM` which lists a number of other summary functions  
 7096 which take a `capthist` object:

```
7097 > moves(capthist)
7098 > dbar(capthist)
7099 > RPSV(capthist)
7100 > MMDM(capthist, min.recapt = 1, full = FALSE)
7101 > ARL(capthist, min.recapt = 1, plt = FALSE, full = FALSE)
```

7102 The function `moves` returns the observed distances moved, `dbar` returns the average dis-  
 7103 tance moved, `RPSV` produces a measure of dispersion about the home-range center, and  
 7104 `ARL` gives the *Asymptotic Range Length* which is the asymptote of an exponential model  
 7105 fit to the observed range length vs. the number of detections of each individual (Jett and  
 7106 Nichols, 1987).

7107 **State-space buffer**

7108 `secr` will produce a warning if the state-space buffer is chosen too small. For example,  
 7109 in fitting the wolverine data as in Sec. 6.5.2 but with a 1000 m buffer, and we see the  
 7110 following warning message:

```
7111 Warning message:
7112 In secr.fit(wolvcapt2, model=list(D ~ 1, g0 ~ 1, sigma ~ 1), buffer=1000):
7113   predicted relative bias exceeds 0.01 with buffer = 1000
```

7114 This should cause you to contemplate modifying the state-space buffer if that is a reason-  
 7115 able thing to do in the specific application.

7116 **Model selection and averaging**

7117 `secr` does likelihood ratio tests to compare nested models using the function `LR.test`.  
 7118 You can create model selection tables based on AIC or AICc, using the function `AIC`,  
 7119 and obtain model-averaged parameter estimates using the function `model.average` (See  
 7120 Chapt. 8 for examples).

7121 **Population closure test**

7122 `secr` has a population closure test with the function `closure.test` which implements the  
 7123 tests of Stanley and Burnham (1999) or Otis et al. (1978). The function is used like this:  
 7124 `closure.test(object, SB = FALSE)`. Here `object` is a `capthist` object and `SB` is a logical  
 7125 variable that, if TRUE, produces the Stanley and Burnham (1999) test.

7126 **Density mapping and effective sample area**

7127 `secr` produces likelihood versions of the various summaries of posterior density and effec-  
 7128 tive sample area that we discussed in Chapt. 5. For example, while `secr` reports estimates  
 7129 of the expected value of  $N$  or density directly in the summary output from fitting a model,  
 7130 you can use the function `region.N` to produce estimates of  $N$  for any given region. In  
 7131 addition, `secr` has functions for creating maps of detection contours for individuals traps,  
 7132 or for the entire trap array. See the function `pdot.contour`, and also `fxi.contour` for

7133 computing the 2-dimensional pdf of the locations of one or more individual activity cen-  
 7134 ters (as in Sec. 5.11.3). In the context of likelihood analysis, estimation of a random effect  
 7135 **s** is based on a plug-in application of Bayes' Rule. When **s** has a uniform distribution, and  
 7136 we use a discrete evaluation of the integral, it can be computed simply by renormalizing  
 7137 the likelihood:

$$[s|y, \theta] = \frac{[y|s, \theta]}{\sum_s [y|s, \theta]}.$$

7138 Any of the **intlik** functions given previously in this chapter can be easily modified to  
 7139 return the posterior distribution of **s** for any, or all, individuals, or an individual that is  
 7140 not encountered.

7141 Effective sample area (see Sec. 5.12) can be calculated in **secr** using the functions **esa**  
 7142 and **esa.plot**).

#### 7143 Covariate models

7144 **secr** has many capabilities for modeling covariates. It has a number of built-in models  
 7145 that allow certain covariates on encounter probability, which we cover to a large extent  
 7146 in Chapt. 7, and also see Chapt. 8 for more examples. **secr** also allows covariates to be  
 7147 built into the density model (see Chapt. 11). It has some built in response surface models,  
 7148 allowing for the fitting of linear or quadratic response surfaces. This is done by modifying  
 7149 the density model in **secr.fit**. For example,  $D \sim 1$  is a constant density surface, and  
 7150  $D \sim x + y$  fits a linear response surface, etc.. See the manual **secr-densitysurfaces.pdf**  
 7151 for the details.

7152 There are a number of ways to model your own "custom" covariates (as opposed to  
 7153 pre-specified models). One way is to use the **addCovariates** function and supply it a  
 7154 **mask** or **traps** object along with some "spatialdata." Or, if you have covariates at each  
 7155 trap location then it will extrapolate to all points on the habitat mask. There's also a  
 7156 method by which the user can create a function of geographic coordinates, **userDfn**, which  
 7157 seems to provide additional flexibility, although we haven't used this method. There is a  
 7158 handy function **predictDsurface** for producing density maps under the specified model  
 7159 for density.

## 6.6 SUMMARY AND OUTLOOK

7160 In this chapter, we discussed basic concepts related to classical analysis of SCR models  
 7161 based on likelihood methods. Analysis is based on the so-called integrated or marginal  
 7162 likelihood in which the individual activity centers (random effects) are removed from the  
 7163 conditional-on-**s** likelihood by integration. We showed how to construct the integrated  
 7164 likelihood and fit some simple models in the **R** programming language. In addition,  
 7165 likelihood analysis for some broad classes of SCR models can be accomplished using the  
 7166 **R** library **secr** (Efford, 2011a) which we provided a brief introduction to. In later chapters  
 7167 we provide more detailed analyses of SCR data using likelihood methods and the **secr**  
 7168 package.

7169 Why or why not use likelihood inference exclusively? For certain specific models, it  
 7170 is may be more computationally efficient to produce MLEs (for an example see Chapt.  
 7171 12). And, likelihood analysis makes it easy to do model-selection by AIC and compute  
 7172 standard errors or confidence intervals. However, **BUGS** is extremely flexible in terms  
 7173 of describing models and we can devise models in the **BUGS** language easily that we

7174 cannot fit in **secr**. For example, in Chapt 16 we consider open population models which  
7175 are straightforward to develop in **BUGS** but, so far, there is no available platform for  
7176 doing MLE of such models. We can also fit models in **BUGS** that accommodate missing  
7177 covariates in complete generality (e.g., unobserved sex of individuals), and we can adopt  
7178 SCR models to include auxiliary data types. For example, we might have camera trapping  
7179 and genetic data and we can describe the models directly in **BUGS** and fit a joint model  
7180 (Gopalaswamy et al., 2012b). To do maximum likelihood estimation, we have to write a  
7181 custom new piece of code for each model<sup>4</sup> or hope someone has done it for us. You should  
7182 have some capability to develop your own MLE routines with the tools we provided in  
7183 this chapter.

---

<sup>4</sup>Although we may be able to handle multiple survey methods together in **secr** using the multi-session models.



7184  
7185

---

7186

# 7

7187

## MODELING VARIATION IN ENCOUNTER PROBABILITY

7188 In previous chapters we showed how to fit basic spatial capture-recapture models using  
7189 Bayesian analysis (in **WinBUGS** or **JAGS**; Chapt. 5) or by classical likelihood meth-  
7190 ods (Chapt. 6 or using **secr**). We mostly focused on a specific observation model, the  
7191 Bernoulli or binomial model for devices such as “proximity detectors” (although we extend  
7192 this model to Poisson and multinomial type observation models in Chapt. 9). We have  
7193 not, however, described a general framework for modeling covariates that might influence  
7194 encounter probability of individuals, traps or over time. In practice, investigators are  
7195 invariably concerned with explicit factors or covariates that might influence variation in  
7196 parameters. Such covariates include time (e.g., day of year, or season), behavior (e.g., is  
7197 there an effect of trapping on subsequent capture probabilities), sex of the individual, and  
7198 trap type (e.g., various camera types, or different constructions for hair snares). Tradition-  
7199 ally, in the non-spatial capture recapture literature, such models were called “model  $M_t$ ”,  
7200 “model  $M_h$ ”, or “model  $M_b$ ”, identifying models that account for variation in detection  
7201 probability as a function of time, “individual heterogeneity” or “behavior”, where behav-  
7202 ior describes whether or not an individual had been previously captured. In SCR models,  
7203 more complex covariate models are possible because we might also have trap-specific co-  
7204 variates, or covariates that vary spatially over the landscape, and because we generally  
7205 have more than one parameter describing the detection function: Most encounter proba-  
7206 bility functions include a baseline encounter rate ( $\lambda_0$ ) or probability ( $p_0$ ) parameter, and  
7207 a scale parameter ( $\sigma$ ), which takes on different interpretations depending on the specific  
7208 encounter probability function under consideration.

7209 In this chapter, we generalize the basic SCR model to accommodate both alternative  
7210 detection functions as well as many different kinds of covariates. We focus on the binomial  
7211 observation model used throughout Chaps. 5 and 6 and the Gaussian encounter model  
7212 (also called the “half-normal” model in the distance sampling literature), but the extension  
7213 to other observation models is straightforward (and other encounter probability models  
7214 with different functions of distance are considered in Sec. 7.1). Specifically, we consider

7215 three distinct types of covariates – those which are fixed, partially observed or completely  
 7216 unobserved (latent). Fixed covariates are those that are fully observed; for example, the  
 7217 date of all sampling occasions. Partially observed covariates are those which are not known  
 7218 for all observations; for example, the sex of an individual cannot always be determined  
 7219 from photos taken during camera trapping. Even if we are able to observe the sex of all  
 7220 individuals sampled, we cannot know it for those individuals never observed during the  
 7221 study. And finally, unobserved covariates are those which we cannot observe at all, for  
 7222 example, the home range size of individuals, or unstructured random “individual effects”.

7223 We will see that models containing these different types of covariates are relatively easy  
 7224 to describe in **WinBUGS** or **JAGS**, and therefore to analyze using Bayesian analysis  
 7225 of the joint likelihood based on data augmentation thus providing a coherent and flexible  
 7226 framework for inference for all classes of SCR models. Throughout the chapter, we will  
 7227 continue to develop the analysis of the black bear study introduced in Chapt. 4, using the  
 7228 software **JAGS**. We also consider the likelihood analysis of many of these models; to do so,  
 7229 we continue to use the **R** package **secr** , and we introduce some ideas of model comparison  
 7230 using AIC (Sec. 7.4 at the end of the chapter). There are other types of covariates that  
 7231 we do *not* cover in this chapter; for example, covariates that vary across the landscape  
 7232 might affect density, and we consider these covariates in Chapt. 11. Alternatively, these  
 7233 landscape covariates might affect the way individuals use space. There are probably very  
 7234 few circumstances under which animals use all space uniformly and we develop more  
 7235 realistic models of encounter probability in which covariates affect space usage in Chapt.  
 7236 12.

## 7.1 ENCOUNTER PROBABILITY MODELS

7237 In Chapt. 5, we developed a basic spatial capture recapture model using a standard  
 7238 encounter probability function based on the kernel of a normal (Gaussian) probability  
 7239 distribution:

$$p_{ij} = p_0 \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

7240 where  $||\mathbf{x}_j - \mathbf{s}_i||$  is the distance between  $\mathbf{x}_j$  and  $\mathbf{s}_i$  and  $\alpha_1 = 1/(2 * \sigma^2)$ . We argued (see  
 7241 Sec. 5.4) that one can view this model as corresponding to an explicit model of space  
 7242 usage – namely, that individual locations are draws from a bivariate normal distribution.  
 7243 We also mentioned that other detection models are possible, including a logit model of  
 7244 the form:

$$\text{logit}(p_{ij}) = \alpha_0 + \alpha_1 ||\mathbf{x}_j - \mathbf{s}_i||. \quad (7.1.1)$$

7245 However, there's nothing preventing us from constructing a myriad of other models for  
 7246 encounter probability as a function of distance. The most commonly used detection prob-  
 7247 ability models are also those used in the distance sampling literature: the half-normal  
 7248 (Gaussian), the hazard, and the negative exponential. The negative exponential model is:

$$p_{ij} = p_0 * \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||)$$

7249 where we define  $\alpha_1 = 1/\sigma$ . We could use the general power model (Russell et al., 2012):

$$p_{ij} = p_0 * \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^\theta)$$

7250 of which the Gaussian and exponential models are special cases. Another model that could  
 7251 be considered is the Gaussian hazard rate model (Hayes and Buckland, 1983):

$$p_{ij} = 1 - \exp(-\lambda_0 * \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2))$$

7252 which was previously discussed in Sec. .

7253 In each of the cases, the relationship of  $\alpha_1$  to  $\sigma$  varies and must be properly spec-  
 7254 ified. The **R** package **secr** allows the user to access 12 different encounter probability  
 7255 models (termed “distance functions” in **secr**), of which some are only used for simulating  
 7256 data (see Table 7.1). These encounter probability models can also be implemented in **R**,  
 7257 **WinBUGS**, **JAGS** etc..

**Table 7.1.** Basic encounter probability models (“distance functions”) available in **secr**. (Table taken from the **secr** help files). Notation deviates from that used in the text. In this table  $g_0$  is the baseline encounter rate or probability parameter used in **secr** which is equivalent to our  $p_0$  or  $\lambda_0$  depending on context.  $d$  is distance defined as we have done throughout, as the distance between the activity center and the trap. One can read more on this specific table by loading the **secr** package and using the **help** command in **R** (**?detectfn**).

	Name	Params	Function
0	half-normal	$g_0, \sigma$	$g(d) = g_0 e^{-d^2/(2\sigma^2)}$
1	hazard rate	$g_0, \sigma, z$	$g(d) = g_0(1 - e^{-(d/\sigma)^{-z}})$
2	exponential	$g_0, \sigma$	$g(d) = g_0 e^{-d/\sigma}$
3	compound half-normal	$g_0, \sigma, z$	$g(d) = g_0[1 - \{1 - e^{-d^2/(2\sigma^2)}\}^z]$
4	uniform	$g_0, \sigma$	$g(d) = g_0, d \leq \sigma;$ $g(d) = 0, \text{ otherwise}$
5	w exponential	$g_0, \sigma, w$	$g(d) = g_0, d < w;$ $g(d) = g_0 e^{(-(d-w)/\sigma)}, \text{ otherwise}$
6	annular normal	$g_0, \sigma, w$	$g(d) = g_0 e^{(-(d-w)^2/(2\sigma^2))}$
7	cumulative lognormal	$g_0, \sigma, z$	$g(d) = g_0[1 - F(d - \mu)/s)]$
8	cumulative gamma	$g_0, \sigma, z$	$g(d) = g_0\{1 - G(d; k, \theta)\}$
9	binary signal strength	$b_0, b_1$	$g(d) = 1 - F\{-(b_0 + b_1 d)\}$
10	signal strength	$\beta_0, \beta_1, S$	$g(d) = 1 - F[\{c - (\beta_0 + \beta_1 d)\}/S]$
11	signal strength spherical	$\beta_0, \beta_1, S$	$g(d) = 1 - F[\{c - (\beta_0 + \beta_1(d-1) - 10 * \log_{10}(d^2))\}/S]$

7258 Insofar as all these encounter probability models are symmetric and stationary, they  
 7259 are pretty crude descriptions of space usage by real animals. This is not to say they are  
 7260 inadequate descriptions of the data and, as we discuss in Chaps. 13 and 12, we can use  
 7261 them as the basis for producing more realistic models of space usage.

7262 By changing the encounter probability model and the specification of  $\alpha_1$ , we can  
 7263 basically create any function of distance for the data. It is important to note that  $\sigma$  is not  
 7264 comparable under these different encounter probability models and should not be regarded  
 7265 as “home range radius” in general. While there is generally a relationship between  $\sigma$  and  
 7266 home range size, that relationship varies depending on the model under consideration. We  
 7267 demonstrate how to fit different encounter probability models in the Bayesian framework  
 7268 here, and then provide information on the likelihood analysis (in **secr**) in a separate  
 7269 section below.

7270 **7.1.1 Bayesian analysis with bear.JAGS**

7271 To demonstrate how to incorporate various types of covariates into models for encounter  
 7272 probability using **JAGS**, we return to the data collected during the Fort Drum bear study.  
 7273 This data set was first introduced in Chapt. 4, but, to refresh your memory, there were  
 7274 38 baited hair snares that were operated between June and July 2006. The snares were  
 7275 checked each week for a total for  $K = 8$  sample occasions and  $n = 47$  individual bears  
 7276 were encountered at least once. The data are provided in the **R** package **scrbook** and an  
 7277 **R** function called **bear.JAGS** allows the user to easily pick which model to analyze. The  
 7278 function **bear.JAGS** will set up the data, write the model, define the MCMC specifications  
 7279 (e.g., initial values, etc.) and, finally, run the selected model in **JAGS**. In addition to  
 7280 choosing which model to run, the user can also specify the number of chains, iterations and  
 7281 length of the burn-in phase. Calling the function will provide all the code to implement  
 7282 the models independently as well. In the following sections we will present the model code  
 7283 and output for the most commonly employed models; for all analyses we ran 3 chains with  
 7284 a burn-in of 500 iterations and 20000 saved iterations.

7285 **7.1.2 Bayesian analysis of encounter probability models**

7286 In Panel 7.1, we present the basic SCR model and show how to specify the negative exponential  
 7287 encounter probability model. To call each of these from the function **bear.JAGS** set  
 7288 **model='SCRO'** or **model='SCRexp'** in the function call, respectively. To reduce repetition  
 7289 of the R coding, we include the basic code here and then only show modifications when  
 7290 necessary throughout the chapter. All of the R coding can be found within the **bear.JAGS**  
 7291 function as well. The function begins by loading the required **R** libraries as well as the  
 7292 Ft. Drum bear data set. This data set includes a 3-d data array (called **bearArray** in our  
 7293 code), with dimensions **nind**  $\times$  **ntraps**  $\times$  **nreps** representing the capture histories of **nind**  
 7294 captured individuals at **ntraps** trap locations. In the Bayesian analysis, data augmentation  
 7295 is used to estimate  $N$  and therefore the **bearArray** data must be augmented with  
 7296  $M - nind$  all zero encounter histories. In models without time dependence, the augmented  
 7297 **bearArray** (called **Yaug** in the code) will be reduced to a 2 dimensional array (denoted **y**  
 7298 in the code) that has dimensions **M**  $\times$  **ntraps**.

```
7299 > library(rjags) # Load the necessary libraries
7300 > library(scrbook)
7301
7302 > data(beardata) # Attach the bear data for Ft. Drum
7303 > ymat <- beardata$bearArray
7304 > trapmat <- beardata$trapmat
7305 > nind <- dim(beardata$bearArray)[1]
7306 > K <- dim(beardata$bearArray)[3]
7307 > ntraps <- dim(beardata$bearArray)[2]
7308 > M <- 650
7309 > nz <- M-nind
7310
7311 # Create augmented array
7312 > Yaug <- array(0, dim=c(M,ntraps,K))
```

---

```

7313 > Yaug[1:nind,,] <- ymat
7314 > y <- apply(Yaug,1:2, sum)

```

7315     The function `bear.JAGS` also establishes the upper and lower limits on the state space  
 7316     by centering the trap array coordinates (which are imported with the `beardata` and saved  
 7317     in the code above as `trapmat`) and then buffering by 20km.

---

```

model{
  alpha0 ~ dnorm(0,.1)                               # Prior distributions
  logit(p0) <- alpha0
  alpha1 <- 1/(2*sigma*sigma)
  sigma ~ dunif(0, 15)
  psi ~ dunif(0,1)

  for(i in 1:M){
    z[i] ~ dbern(psi)
    s[i,1] ~ dunif(xlim[1],xlim[2])
    s[i,2] ~ dunif(ylim[1],ylim[2])
    for(j in 1:J){
      d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
      y[i,j] ~ dbin(p[i,j],K)
      p[i,j] <- z[i]*p0*exp(-alpha1*d[i,j]*d[i,j]) # Gaussian model
      #p[i,j] <- z[i]*p0*exp(-alpha1*d[i,j])        # exponential model
    }
  }
  N <- sum(z[])
  D <- N/area
}

```

---

Panel 7.1: **JAGS** model specification for a basic SCR model with Gaussian encounter probability function and the alternative exponential encounter probability function.

7318     Applying the SCR model with Gaussian encounter probability model provides an  
 7319     estimate (posterior mean) of  $D = 0.167$  bears per  $km^2$  and with the negative exponential  
 7320     encounter probability model the posterior mean is virtually the same  $D = 0.167$ . In  
 7321     distance sampling, the use of different encounter probability models often results in very  
 7322     different estimates of density (especially when using the negative exponential model).  
 7323     There are two main reasons why the different models may have less of an impact on the  
 7324     density estimates under the SCR models. First, we can estimate the baseline encounter  
 7325     probability parameter ( $p_0$ ). In most distance sampling models, detection at distance 0  
 7326     is set to 1. In Table 7.2, the posterior mean of  $p_0$  is 0.11 under the Gaussian model  
 7327     and 0.34 under the negative exponential model. The larger baseline encounter probability

under the negative exponential model reduces the impact of the quick decline in detection as a function of distance. Secondly, the detection probability function here is governing 'movement' of individuals (which we have more information on than in distance sampling), not the whole detection process, so the shape of the detection probability function does not impact the density estimation as much.

In all analyses it is important to check that the size of the augmented data set ( $M$ ) is sufficiently large and does not impact the estimate of  $N$ . Here, the 97.5% percentile for  $N$  is 628 (Table 7.2), thus not reaching our  $M = 650$  value. We could also increase  $M$  and compare the posterior of  $N$  under the different scenarios as another check that the data augmentation is sufficient.

**Table 7.2.** Posterior summaries of SCR model parameters having different encounter probability models, for the Fort Drum black bear data.

Parameter	Mean	SD	2.5%	97.5%
<b>Gaussian</b>				
$N$	500.63	66.652	371.000	628.000
$D$	0.17	0.022	0.122	0.207
$p_0$	0.11	0.014	0.081	0.135
$\sigma$	1.99	0.131	1.762	2.275
$\psi$	0.77	0.104	0.566	0.966
<b>Exponential</b>				
$N$	512.06	65.771	382.000	634.000
$D$	0.17	0.022	0.130	0.210
$p_0$	0.34	0.056	0.246	0.465
$\sigma$	1.12	0.095	0.951	1.323
$\psi$	0.79	0.102	0.584	0.974

A very important consideration when using different detection probability functions is the interpretation of  $\sigma$ . The estimate (posterior mean) of  $\sigma$  under the negative exponential model is 1.12, which is distinct from our estimate of  $\sigma$  under the Gaussian model,  $\sigma = 1.996$ . The interpretation of  $\sigma$  in the two models is really quite distinct. In the normal model it can be interpreted as the standard deviation of a bivariate normal movement model whereas the manner in which  $\sigma$  relates to "area used" for the negative exponential model has nothing to do with a bivariate normal model of movement. This highlights that it is important for the user to know what detection probability function is used and what the interpretation of  $\sigma$  might be in relation to the home range size. This relationship was discussed in Sec. 5.4.

We now move onto incorporating covariates into the model using the **JAGS** language. For this part, we will stick with the Gaussian encounter probability model shown in Panel 7.1 above.

## 7.2 MODELING COVARIATE EFFECTS

The basic strategy for modeling covariate effects is to include them on the baseline encounter rate or probability parameter,  $p_0$  (or  $\lambda_0$ ), or the scale parameter of the encounter model,  $\sigma$ , or in some cases, both parameters.

Broadly speaking, we recognize (here) 3 types of covariates. Fixed covariates are fully observable and might vary by trap alone (e.g., type of trap, baited or not, disturbance regime, even habitat), sample occasion (e.g., day of season or weather conditions), or both (e.g., behavior, weather - if over a large region). Another class of covariates are those which vary at the level of the individual (and possibly also over time). As a technical matter, and as noted before, these are different from fixed covariates because we cannot see all of the individuals and the covariates are almost always incompletely observed (if at all). The lone exception is the effect of previous capture, used to model a behavioral response to capture, which is known for all individuals, captured or not (an animal never captured/observed has never been captured before). We noted many times before that space itself (i.e., the activity centers) is a type of individual covariate and this notion actually helped us derive the fully spatial capture-recapture model from the traditional, non-spatial model (Chapt. 4). We do not get to observe the activity center for any individuals, but for individuals that are encountered we get to observe some information about it in the form of which traps the individual was encountered in. And finally, we have completely unobserved covariates such as heterogeneity in home range size. We consider heterogeneity in a separate section below since there are a suite of models for describing latent heterogeneity.

**Table 7.3.** Examples of different types of covariates in SCR models.

Covariate type	Examples
individual	sex, age, home range
trap	baited/not, habitat (see also Chapter 13)
time	season, shedding, weather
individual x time	global behavioral response
trap x time	trap failures
individual x trap x time	local behavioral response

To develop covariate models, we assume a standard sampling design in which an array of  $J$  traps is operated for  $K$  sample occasions, which produces encounter histories for  $n$  individuals. For the null model, there are no time-varying covariates that influence encounter, there are no explicit individual-specific covariates, and there are no covariates that influence density. For fixed effects, those which we observe fully, we can easily incorporate these into the encounter probability model, just as we would do in any standard GLM or GLMM, on some suitable scale for the encounter probability,  $p_{ijk}$ . For example,

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * C_{ijk}$$

$$p_{ijk} = p_{0,ijk} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

where  $C_{ijk}$  is some covariate that varies (potentially) by individual ( $i$ ), trap ( $j$ ) and occasions ( $k$ ), and  $\alpha_2$  is the coefficient to be estimated. How we define specific covariates (e.g., trap specific versus individual specific) will influence exactly how we include them in the model. Table 7.3 shows examples of covariates by type – trap, individual, and time – and also gives examples of some combined types. These are the types of covariates we will specifically address in this chapter, demonstrating how to analyze the different types in the following sections.

7386 **7.2.1 Date and time**

7387 Often, researchers are interested in modeling the effect of date or chronological time on  
 7388 encounter probability. For example, in a long term hair snare study, we may expect that  
 7389 seasonal shedding (Wegan et al., 2012) will influence encounter probabilities directly. Or,  
 7390 we may expect behaviors such as denning, mating, etc., to influence the encounter of  
 7391 certain species at certain times of year (Kéry et al., 2011). There are two common ways  
 7392 to incorporate date or time information into a model for encounter probability. For cases  
 7393 with a small number of sampling occasions we can fit a time-specific intercept (analogous  
 7394 to “model  $M_t$ ” in classical capture-recapture (Otis et al., 1978)). In this model, there are  
 7395  $K$  sampling occasion-specific parameters to reflect potential variation in sampling effort  
 7396 or other factors that might vary across samples. Alternatively, we can model parametric  
 7397 functions of date or time such as polynomial or sinusoidal functions.

7398 In the first case, we allow each sampling occasion,  $k$ , to have its own baseline encounter  
 7399 probability, e.g.,

$$\text{logit}(p_{0,k}) = \alpha_{0,k}$$

7400 so that

$$p_{ijk} = p_{0,k} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2).$$

7401 This description of the model includes  $k$  occasion-specific baseline encounter probabilities.  
 7402 Thus, if there are 4 sampling occasions, then there are 4 different baseline encounter  
 7403 probabilities. We imagine that complete time-specificity of  $p_0$  (i.e., one distinct value  
 7404 for each sample occasion) would be most useful in situations where there are just a few  
 7405 sampling occasions (if there are many, this formulation will dramatically increase the  
 7406 number of parameters to be estimated) or we do not expect systematic patterns over time  
 7407 (e.g., explainable by a polynomial function or time-varying covariates).

7408 To implement this in **JAGS**,  $\alpha_0$  has to be estimated for each time period  $k$  either  
 7409 using an index vector or dummy variables (as described in Chapt. 2 and Sec. 4.3) and this  
 7410 can be done by only changing only a few lines in Panel 7.1:

```
7411 alpha0[k] ~ dnorm(0,.1)
7412 logit(p0[k]) <- alpha0[k]
7413 .....
7414 .....
7415 y[i,j,k] ~ dbin(p[i,j,k],K)
7416 p[i,j,k] <- z[i]*p0[k]*exp(- alpha1*d[i,j]*d[i,j])
```

7417 Since the model contains a parameter for each time period, the encounter histories  
 7418 must be time-dependent. Thus, a 3-d data array (called **bearArray** in our code), with  
 7419 dimensions **nind** × **ntraps** × **nreps** is required (recall that we use the 3-d augmented array  
 7420 called **Yaug** with dimensions **M** × **ntraps** × **nreps** for the Bayesian analysis). In addition  
 7421 to using the 3-d data array, the initial values must be updated so that there are  $K$  values  
 7422 generated for  $\alpha_0$ . And finally, this means that another nested *for loop* is needed in the  
 7423 code to account for the  $K$  sample occasions. A side note: the computation time will  
 7424 increase quite a bit (this model for the bear data may take up to 15 hours or more on  
 7425 your machine to obtain a sufficient posterior sample).

7426 Running this model with the function **bear.JAGS** by setting **model=SCRt**, returns esti-  
 7427 mates of density similar to those from the model without covariates (see Table 7.4), but

now we have a characterization of variation in encounter probability over time. Encounter probability seems to increase for the first few time periods before stabilizing around 0.14, dropping off again at the end of the study. The differences in encounter probability from the first time periods to the others might actually be due to something like a behavioral response (see below) or possibly seasonal differences in the efficiency of the sampling technique. Researchers have found that hair snares are more effective at different times of the year (even within season) due to shedding (Wegan et al., 2012). In this particular example, our density estimates (posterior means) are similar to the base model, likely because the differences in encounter probability between occasion were not that large. In a longer term study or in one with greater variation in the encounter probability, the implication of such differences might have a bigger impact on the estimates of density and  $\sigma$ .

**Table 7.4.** Posterior summaries of parameter estimates from a SCR model with time-dependent baseline encounter probability for the Ft. Drum black bear data set.

Parameter	Mean	SD	2.5%	97.5%
$N$	509.24	66.13	381	632
$D$	0.17	0.02	0.13	0.21
$p_0(t = 1)$	0.06	0.02	0.03	0.10
$p_0(t = 2)$	0.05	0.02	0.02	0.09
$p_0(t = 3)$	0.15	0.03	0.09	0.22
$p_0(t = 4)$	0.14	0.03	0.09	0.21
$p_0(t = 5)$	0.15	0.03	0.09	0.22
$p_0(t = 6)$	0.12	0.03	0.07	0.19
$p_0(t = 7)$	0.15	0.03	0.09	0.22
$p_0(t = 8)$	0.08	0.02	0.04	0.13
$\sigma$	1.96	0.12	1.73	2.22
$\psi$	0.78	0.10	0.58	0.97

The occasion specific intercepts (baseline encounter probability) model might not be the most appropriate for all scenarios and could require the estimation of many parameters if we had many sampling occasions, take the wolverine example from Chapt. 5.9 where there were 165 daily sampling occasions. Particularly in such a case as the wolverine study, variation in the encounter process over time is to be expected. For example, if a camera trap study is conducted for an entire year, it is expected that there would be behavioral patterns in individuals due to mating or denning. Instead of fitting a model with  $K$  baseline encounter probabilities, we can include date as a linear (or quadratic, ...) effect. An example can be found in Kéry et al. (2011) who incorporated a day-of-year covariate, both as a linear and a quadratic effect, into their SCR model of European wildcats; the data had been collected over a year-long period and cat behavior was expected to vary seasonally thus influencing the probability of encounter. In these cases, we would specifically incorporate day of year (variable “Date”) as a numeric covariate as:

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * \text{Date}_k$$

$$p_{ijk} = p_{0,ijk} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

7452 or a quadratic effect of day-of-year:

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * \text{Date}_k + \alpha_3 * \text{Date}_k^2 \\ p_{ijk} = p_{0,ijk} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

7453 where the variable **Date** is an integer coding of day-of-year, indexed to some arbitrary  
7454 start point in time.

### 7455 7.2.2 Trap-specific covariates

7456 In some studies it makes sense to model encounter probability as a function of local or trap-  
7457 specific covariates. These can be one of two types: genuine trap covariates that describe  
7458 the trap or encounter site, such as whether a trap is baited or not, or how many traps were  
7459 set at a sampling location, or what kind of bait was used, etc., or local covariates that  
7460 describe the likelihood that an animal would use the habitat in the vicinity of the trap  
7461 (see Chapt. 13 for more on this situation). We imagine that these covariates, of either  
7462 type, should affect baseline encounter probability. For example, Sollmann et al. (2011)  
7463 found a large difference in the encounter probability of jaguars due to traps being located  
7464 on roads, which the animals were using to travel along, as opposed to traps placed off  
7465 of roads. In this case, the trap type is a binary variable – on/off road, (another binary  
7466 variable could be baited/non-baited). We can write this as:

$$\text{logit}(p_{0,j}) = \alpha_{0,type_j} \\ p_{ijk} = p_{0,j} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2).$$

7467 Here, we use an index variable, “type”, an integer value for the trap-specific covariate.  
7468 Thus for our example of on/off road, we would have  $type_j = 1$  if trap  $j$  is on a road and  
7469  $type_j = 2$  otherwise, and we would estimate two separate  $\alpha_0$  parameters – one for on-road  
7470 and one for off-road cameras. An alternative way to express the 2-category model, using  
7471 dummy variables, requires that we specify our “type” vector as  $Type_j = 0$  if trap  $j$  is on  
7472 a road and  $Type_j = 1$  otherwise, and write the model as

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * Type_j.$$

7473 Now,  $\alpha_0$  is the baseline encounter probability (on the logit scale) for traps on a road  
7474 ( $Type_j = 0$ ) and  $\alpha_2$  is the effect on baseline encounter probability of a trap being of  
7475  $Type = 1$ . This general set up also allows for more than 2 categories, say if 4 different  
7476 camera models were used in a study, we would use a set of 3 binary dummy variables  
7477 to allow for estimation of the different encounter rates (i.e., the intercept). While these  
7478 models are equivalent, and should yield identical results, sometimes one parameterization  
7479 might work better than the other in **WinBUGS** or **JAGS** (Kéry, 2010).

### 7480 7.2.3 Behavior or trap response by individual

7481 One of the most basic of encounter models is that which accommodates a change in  
7482 encounter probability as a result of initial encounter. This is colloquially referred to as  
7483 “trap happiness” or “trap shyness”, or in other words, a behavioral response of individuals

7484 to being captured (Otis et al., 1978). If a trap is baited with a food source, an individual  
 7485 might come back for more. On the other hand, if being captured is traumatic then an  
 7486 individual might learn to avoid traps. Both of these types of responses can occur in  
 7487 most species depending on the type of encounter mechanisms being employed. Moreover,  
 7488 behavioral response can be either global (Gardner et al., 2010b) or local (Royle et al.,  
 7489 2011b). The local response is a trap-specific response while a global response suggests that  
 7490 initial capture provides a net increase or decrease in subsequent probabilities of capture  
 7491 (across all traps). A behavioral response does not need to be enduring (i.e., persist for  
 7492 the entire study after the individual has been captured/observed for the first time) but  
 7493 can also be ephemeral, if, for example, an animal only avoids a trap on the occasion  
 7494 immediately after it was captured (Yang and Chao, 2005; Royle, 2008). While we will  
 7495 focus the examples in this chapter on enduring behavioral effects, extending such a model  
 7496 to the case of an ephemeral response should not pose any difficulties.

7497 To describe these behavioral models we need to create a binary matrix that indicates  
 7498 if an individual has been captured previously. For the global behavioral response, define  
 7499 the  $n \times K$  matrix,  $\mathbf{C}$ , where  $C_{ik} = 1$  if individual  $i$  was captured at least once prior to  
 7500 session  $k$ , otherwise  $C_{ik} = 0$ .

$$\text{logit}(p_{0,ik}) = \alpha_0 + \alpha_2 * C_{ik}$$

$$p_{ijk} = p_{0,ik} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2).$$

7501 For the local behavioral response, which is trap specific, we create an array,  $C_{ijk}$ , that  
 7502 indicates if an individual  $i$  has been previously captured in trap  $j$  at time  $k$ . (For the  
 7503 augmented individuals, the entries are all 0 since the animals were never captured.) We  
 7504 then include this in the model in the exact same form as above (with the sole difference  
 7505 that both  $C$  and  $p$  are now also indexed by  $k$ ):

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * C_{i,j,k}$$

$$p_{ijk} = p_{0,ijk} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2).$$

7506 Since the behavioral response is occasion specific, to implement either the local or  
 7507 global response model in **JAGS**, we will have to use the 3-d array of the augmented  
 7508 capture histories ( $M \times ntraps \times nreps$ ) as we did for the time-varying encounter probability  
 7509 model above. The code must loop over each sampling occasion, but otherwise, the model  
 7510 varies only a little from the basic SCR model shown in Panel 7.1. Here is the specification  
 7511 of the the occasion specific ( $k$ ) loop:

```
7512 for(k in 1:K){  

  7513   logit(p0[i,j,k]) <- alpha0 + alpha2*C[i,j,k]  

  7514   y[i,j,k] ~ dbin(p[i,j,k],1)  

  7515   p[i,j,k] <- z[i]*p0[i,j,k]*exp(- alpha1*d[i,j]*d[i,j]).  

  7516 }
```

7517 Despite only minor changes to the **BUGS** code, this model can require quite a bit  
 7518 of time and computational effort. Implementing the behavioral models with the function  
 7519 **bear.JAGS** by setting **model=SCRb** or **model=SCRb** for the local or global model respec-  
 7520 tively, returns the results shown in Table 7.5. There is a strong global behavioral response  
 7521 suggested by the posterior mean of  $\alpha_2 = 0.90$ . The estimate of  $N$  and subsequently  $D$  are

7522 larger than under the model without a behavioral response; here we estimate the posterior  
 7523 mean of  $N = 577.56$ , whereas in the SCR0 model, we estimated the posterior mean as  
 7524  $N = 500$ . This makes sense given the large estimate of  $\alpha_2$ , which suggests that bears  
 7525 are trap happy. In situations where animals are trap happy, the null model tends to over  
 7526 estimate encounter probability (i.e., the bears that are never observed have a lower en-  
 7527 counter probability than those that have been captured in the study) and thereby reduce  
 7528 the estimate of  $N$ . We do not include the results here, but the estimates were similar  
 7529 under the local behavioral response model.

**Table 7.5.** Posterior summaries of parameter estimates from the SCR model with a global behavioral response in encounter for the Fort Drum black bear data set.

Parameter	Mean	SD	2.5%	97.5%
$N$	577.56	54.30	452	648
$D$	0.19	0.02	0.15	0.21
$\alpha_0$	-2.81	0.24	-2.91	-2.36
$\alpha_2$	0.90	0.23	0.45	1.35
$\sigma$	2.00	0.13	1.77	2.28
$\psi$	0.88	0.08	0.69	0.99

#### 7530 7.2.4 Individual covariates

7531 Individual covariates are those which are measured (or measurable) on individuals, so  
 7532 we get to observe them only for the captured individuals. Sex is a simple example of an  
 7533 individual covariate, but one of the most commonly used in capture-recapture studies. The  
 7534 sex of an individual can influence many aspects of its ecology and behavior, including for  
 7535 example, the frequency of movement, seasonal behavior, and its home range size. This is  
 7536 common in studies of carnivores where females often have smaller home ranges than males  
 7537 (Gardner et al., 2010b; Sollmann et al., 2011). Additionally, we may find differences in  
 7538 the baseline encounter probability between males and females because females may move  
 7539 around less frequently, or possibly because they are less likely to use landscape structures  
 7540 that researchers may target with sampling devices in order to increase sample size, such  
 7541 as roads (e.g. Salom-Pérez et al., 2007). Therefore, we can imagine that sex may impact  
 7542 both the baseline encounter probability  $\alpha_0$  and the typical home range size, so that  $\alpha_1$   
 7543 might also be sex-specific also. The fully sex-specific model is:

$$\text{logit}(p_{0,i}) = \alpha_{0,sex_i}$$

$$p_{ijk} = p_{0,i} \exp(-\alpha_{1,sex_i} * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

7544 where  $sex_i$  is a vector indicating the sex of each individual (1 = male, 2 = female). While  
 7545 we might know the sex of all individuals observed in the study, we will never know the sex  
 7546 of individuals that are not observed (Gardner et al., 2010b). It is also possible that we  
 7547 may not be able to determine the sex of individuals that are observed during the study.  
 7548 For example photographic captures do not necessarily result in pictures that allow the sex  
 7549 to be absolutely determined, thus sometimes resulting in missing values of this covariate  
 7550 for animals captured in the study. We deal with this slightly differently depending on

7551 the inference framework that we adopt (Bayesian or likelihood). Here we demonstrate  
 7552 the Bayesian implementation and we discuss the likelihood approach using **secr** in detail  
 7553 below in Sec. 7.4.2. Before proceeding with that, we note that it would be possible also to  
 7554 model covariates directly on the parameter  $\sigma$  (or its logarithm), e.g.,  $\log(\sigma_i) = \theta_1 + \theta_2 \text{sex}_i$   
 7555 (see Sec. 8.1). One or the other (or perhaps *some* other) parameterization may yield a  
 7556 better performing MCMC algorithm or provide a more natural or preferred interpretation.  
 7557 In the context of Bayesian analysis, given that priors are not invariant to transformation of  
 7558 the parameters, this may be a consideration in choosing the particular parameterization.

7559 Specifying a fully sex-specific model for **JAGS** is similar to the time-specific model  
 7560 shown above. We need to use an index or dummy variable to let  $\alpha_0$  and/or  $\alpha_1$  be defined  
 7561 separately for males and females. The main difference in this specification is that we do  
 7562 not observe sex for the augmented individuals. Therefore, we have missing observations  
 7563 of the covariate for those individuals. As a result, sex is regarded as a random variable  
 7564 and so the missing values can be estimated along with the other structural parameters of  
 7565 the model.

7566 Because we are regarding sex as a random variable, we have to specify a distribution for  
 7567 it. With only two possible outcomes, it is natural to suppose that  $\text{Sex}_i \sim \text{Bernoulli}(\psi_{\text{sex}})$   
 7568 where the parameter  $\psi_{\text{sex}}$  is the sex ratio of the population. We assume our default non-  
 7569 informative prior for this parameter:  $\psi_{\text{sex}} \sim \text{Uniform}(0, 1)$ . The model specification in  
 7570 Panel 7.2 demonstrates how to incorporate a partially observed covariate (i.e., “sex”). It  
 7571 is important to note that in the previous equation,  $\text{sex}_i$  is a vector with two categories  
 7572 indicating the sex of each individual (e.g., 1 = male, 2 = female). This corresponds  
 7573 directly to having a binary indicator of sex (e.g.,  $\text{Sex}_i = 1$  if individual  $i$  is female, and 0  
 7574 otherwise). In the Bayesian formulation of the model, we use both the binary indicator  
 7575 (**Sex**) and a categorical indicator ( $\text{Sex2} = \text{Sex} + 1$ ). The former (termed **Sex** in Panel  
 7576 7.2) allows us to specify the Bernoulli distribution for the random variable, and the latter  
 7577 (termed **Sex2**) allows us to use the dummy or indicator variable specification in the model.

7578 In both **JAGS** or **BUGS** missing data are indicated by **NA** in the data objects passed  
 7579 to the program through the **bugs** or **jags** functions in **R**. To set up the data, we need to  
 7580 create a vector of length  $M$  with the first  $n$  elements being 0 if individual  $i$  is a female, or  
 7581 1 if  $i$  is a male (for the Fort Drum black bear data the function **bear.JAGS** extracts this  
 7582 information automatically from the **beardata** object), and the subsequent  $M - n$  elements  
 7583 being **NA**. It is generally a good idea to provide starting values for the missing data, but we  
 7584 cannot provide starting values for observed data; in this case where one vector (or other  
 7585 object) contains both observed and missing data, initial values for the observed data have  
 7586 to be specified as **NA**. The code snippet below shows you how to set up the data including  
 7587 the **Sex** vector and the initial values function (the remainder of the code is identical to  
 7588 what we've shown before).

```
7589 > sex <- beardata$sex #the sex data for captured individual
7590 > Sex <- c(sex-1, rep(NA, nz)) #sex enters as 1/2, this recodes it to 0/1
7591                                         #so we can use Bernoulli distribution
7592
7593 > data <- list(y=y,Sex=Sex, M=M,K=K, J=ntraps, xlim=xlim, ylim=ylim,area=areaX)
7594 > params <- c('psi','p0','N', 'D', 'sigma', 'psi.sex')
7595 > inits <- function() { list(z=c(rep(1,nind), rbinom(nz,1,0.5)),psi=rnorm(1),
7596                               s=cbind(rnorm(M, xlim[1],xlim[2]), rnorm(M,ylim[1],ylim[2])),
```

```
7597     psi.sex=runif(1,Sex=c(rep(NA, nind), rbinom(nz,1,0.5)),
7598     sigma=runif(2,2,3),alpha0=runif(2)) }
```

7599 The **BUGS** model specification is shown in Panel 7.2.

---

```
model{

psi ~ dunif(0,1)                                # Prior distributions
psi.sex ~ dunif(0,1)
for(t in 1:2){
  alpha0[t] ~ dnorm(0,.1)
  logit(p0[t]) <- alpha0[t]
  alphai[t] <- 1/(2*sigma[t]*sigma[t])
  sigma[t] ~ dunif(0, 15)
}

for(i in 1:M){
  z[i] ~ dbern(psi)
  Sex[i] ~ dbern(psi.sex)                      # Sex is binary
  Sex2[i] <- Sex[i] + 1                         # Convert to categorical
  s[i,1] ~ dunif(xlim[1],xlim[2])
  s[i,2] ~ dunif(ylim[1],ylim[2])

  for(j in 1:J){
    d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
    y[i,j] ~ dbin(p[i,j],K)
    p[i,j] <- z[i]*p0[Sex2[i]]*exp(-alphai[Sex2[i]]*d[i,j]*d[i,j])
  }
}
N <- sum(z[])
D <- N/area
}
```

---

Panel 7.2: **JAGS** model specification for an SCR model with sex-specific encounter probability parameters.

7600 Our estimate of density under the fully sex-specific model is still very similar to the  
 7601 previous models (Table 7.6), and while the baseline detection was not very different be-  
 7602 tween males and females, we can see that they had very different  $\sigma$  estimates (note that  
 7603 the BCIs do not overlap). As usual, you can reproduce this analysis by calling the function  
 7604 `bear.JAGS` and set `model='SCRsex'`.

**Table 7.6.** Posterior summaries of parameter estimates from sex-specific SCR models for the Fort Drum black bear data set.

Parameter	Mean	SD	2.5%	97.5%
$N$	509.982	66.355	376	631
$D$	0.168	0.022	0.12	0.21
$p_{0,female}$	0.136	0.025	0.09	0.19
$p_{0,male}$	0.092	0.017	0.06	0.13
$\sigma_{female}$	1.542	0.132	1.31	1.83
$\sigma_{male}$	2.682	0.389	2.09	3.62
$\psi_{sex}$	0.310	0.068	0.19	0.45
$\psi$	0.784	0.103	0.58	0.97

### 7.3 INDIVIDUAL HETEROGENEITY

7605 Here we consider SCR models with individual heterogeneity. Capture-recapture models  
 7606 with individual heterogeneity in detection probability, so-called model  $M_h$ , have a long  
 7607 history in classical capture recapture models and they have special relevance to SCR (Sec.  
 7608 4.4). While the advent of SCR models may appear to have rendered the use of classical  
 7609 model  $M_h$  obsolete (because one major source of heterogeneity, namely exposure to the  
 7610 trap array is being accounted for explicitly) we may still wish to consider heterogeneity  
 7611 models for other biological reasons. It is reasonable to expect in real populations that there  
 7612 exists heterogeneity in home range size and so we think that  $\alpha_1$  could exhibit heterogeneity  
 7613 among individuals. As we noted previously, it may be advantageous or desirable in some  
 7614 cases to model heterogeneity directly in terms of the scale parameter of the encounter  
 7615 probability function,  $\sigma$ , or some other transformation of the “distance coefficient”, perhaps  
 7616 even 95% home range area.

7617 In this section, we describe a class of spatial capture-recapture models to allow for  
 7618 individual heterogeneity in encounter probability. In particular, one class of models we  
 7619 propose explicitly admits individual heterogeneity in home range *size*. In addition, we con-  
 7620 sider a standard representation for heterogeneity in which an additive individual-specific  
 7621 random effect is included in the linear predictor for baseline encounter probability.

#### 7622 7.3.1 Models of heterogeneity

7623 An obvious extension to the SCR model is to include an additive individual effect, analo-  
 7624 gous to classical “model  $M_h$ ”. We’ll call this model “SCR+Mh”:

$$\begin{aligned} \text{logit}(p_{0,i}) &= \alpha_0 + \eta_i \\ p_{ijk} &= p_{0,i} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2) \end{aligned}$$

7625 where  $\eta_i$  is an individual random effect having distribution  $[\eta|\sigma_p]$ . A popular class of  
 7626 models arises by assuming  $\eta_i \sim \text{Normal}(0, \sigma_p^2)$  (Coull and Agresti, 1999; Dorazio and  
 7627 Royle, 2003). We show how to implement this specific SCR + Mh model in Panel 7.3,  
 7628 and this model can be used to analyze the Ft. Drum bear data by calling the function  
 7629 `bear.JAGS` and setting `model='SCRh'`. While we show one possible implementation here,  
 7630 many other random effects distributions are possible. A popular one is the finite-mixture

7631 of point masses (Norris and Pollock, 1996; Pledger, 2004) which we demonstrate how to  
 7632 fit using `secr` in Sec. 7.4.3.

---

```
model{

  alpha0 ~ dnorm(0,.1)                                # Prior distributions
  alpha1 <- 1/(2*sigma*sigma)
  sigma ~ dunif(0, 15)
  psi ~ dunif(0,1)
  tau_p ~ dgamma(.001,.001)

  for(i in 1:M){
    eta[i] ~ dnorm(0, tau_p)                         # Individual level variables
    z[i] ~ dbern(psi)
    s[i,1] ~ dunif(xlim[1],xlim[2])
    s[i,2] ~ dunif(ylim[1],ylim[2])

    for(j in 1:J){                                     # The "likelihood" etc..
      d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
      y[i,j] ~ dbin(p[i,j],K)
      logit(p0[i,j]) <- alpha0 + eta[i]
      p[i,j] <- z[i]*p0[i,j]*exp(-alpha1*d[i,j]*d[i,j])
    }
  }
  N <- sum(z[])
  # N, D are derived
  D <- N/area
}
```

---

Panel 7.3: **JAGS** model specification for the SCR + Mh model with Gaussian encounter probability model and additive normal random effect.

### 7633 7.3.2 Heterogeneity induced by variation in home range size

7634 An alternative heterogeneity model, one that has more of a direct biological motivation and  
 7635 interpretation, describes heterogeneity in home range size among individuals. To model  
 7636 heterogeneity in home range area, we can assume a distribution for a transformation of  
 7637 the scale parameter of the encounter probability model such as  $\sigma^2$ , or  $\log(\sigma^2)$ , etc.. We  
 7638 call this “model SCR + Ah” (Ah here for area-induced heterogeneity).

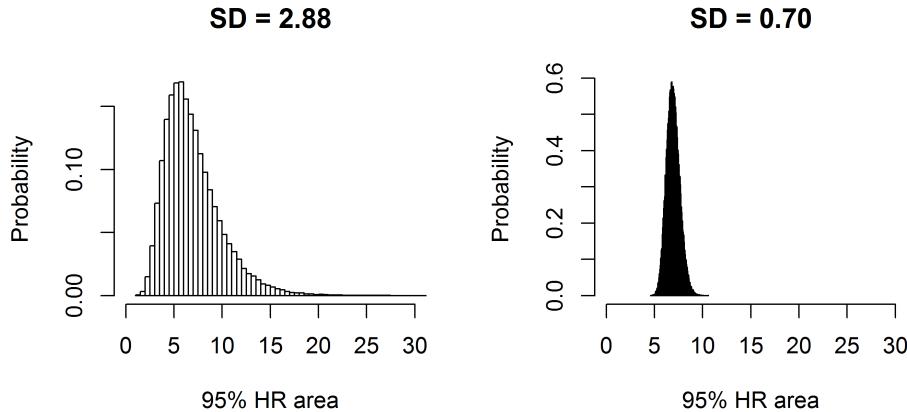
7639 Consider the following log-normal model for the individual scale parameter of the  
 7640 Gaussian encounter probability model,  $\sigma_i^2$ :

$$\log(\sigma_i^2) \sim \text{Normal}(\mu_{hra}, \tau_{hra}^2)$$

7641 then the 95% home range area has a scaled log-normal distribution with mean

$$6\pi \exp(\mu_{hra} + \tau_{hra}^2/2).$$

7642 The variance is slightly more complicated, but you can look up the variance of a log-normal  
 7643 distribution and combine it with the 95% home range area calculation in Sec. 5.4 to work  
 7644 out the implied variance of home range area under this model. We show two examples of  
 7645 the implied *population* distribution of home range area under this log-normal model that  
 7646 indicates a mean home range area of about 6.9 area units (Figure 7.1). The left panel  
 7647 shows a standard deviation in home range area of 2.88 units and the right panel shows  
 7648 a standard deviation in home range area of 0.70 units. The two cases were generated by  
 7649 tweaking the  $\mu_{hra}$  and  $\tau_{hra}^2$  parameters of the log-normal distribution to achieve a constant  
 7650 expected value of home range area, but modify the standard deviation.



**Figure 7.1.** Population distribution of home range area for a model in which  $\log(\sigma^2)$  has a normal distribution with mean  $\mu_{hra}$  and variance  $\tau_{hra}^2$ . The parameters were chosen to yield a constant expected value of about 6.9 units of area, but to produce two different levels of heterogeneity: A population standard deviation of 2.88 units (left panel) and 0.70 units (right panel).

## 7.4 LIKELIHOOD ANALYSIS IN SECR

7651 Previously, in Chapt. 6, we introduced the R package **secr** and described the likelihood  
 7652 based inference approach taken by that package (see Sec. 6.5.3). Here we discuss how  
 7653 to implement some standard covariate models in **secr** and provide an example of model  
 7654 selection using AIC. As we saw in Chapt. 6, **secr** uses the standard R model specifi-  
 7655 cation syntax, defining the dependent and independent variable relationship using tildes

(e.g., `y ~ x`). Thus, in `secr` we might have `g0 ~ behavior` or `sigma ~ time`; when left unspecified or set to 1 (e.g., `g0 ~ 1`), this will default to a model with no covariates (i.e., constant parameter values). A number of default model formulas for the baseline and scale parameter of the encounter probability model are available in `secr`. Additionally, `secr` allows us to specify covariates on density (we cover this in Chapt. 11), which are set for example as `D ~ habitat`.

To demonstrate models with various types of covariates using `secr`, we continue using the Fort Drum black bear data. We include in the `scrbook` package a function called `secr.bear` that will format the data (see Chapt. 6 for the `secr` data format) and then fit and compare 8 models (details shown in Panel 7.4). We have described all of these models in the previous sections, so we only briefly comment here on how to fit certain models in `secr` and compare them using AIC, and give a few helpful notes.

#### 7.4.1 Notes for fitting standard models

In the `secr` package, the encounter probability model is called the “detection function” and it is specified by changing the “`detectfn`” option (an integer code) within the `secr.fit` command. Table 7.1 shows the possible encounter probability models that `secr` allows; the default is that based on the kernel of a bivariate normal probability distribution function (hence we call this the Gaussian model, but it is referred to as “half-normal” in `secr`) and the (negative) exponential is `detectfn = 2`. See model 2 in Panel 7.4 for how to fit the exponential model to the Fort Drum bear data set.

The `secr` package easily fits a range of SCR equivalents of standard capture-recapture models. The package has pre-defined versions of the classic model  $M_t$  where each occasion has its own encounter probability, as well as a linear trend in baseline encounter probability over occasions (in a spatial modeling framework  $\sigma$  could also be an occasion specific parameter, but having encounter probability change with time seems like the more common case). For the classical time-effects type of model with  $K$  distinct parameters `secr` uses ‘t’ to denote this in the model specification formula (see model 3 in panel 7.4); whereas, for a linear trend over occasions `secr` uses ‘T’.

The global trap response model (what we called model  $M_B$ ), or a local trap-specific behavioral response (model  $M_b$ ) can be fitted in `secr` using formulae with “b” for the global response model and “bk” for the local trap response model (see models 4 and 5 in Panel 7.4; note that to fit the trap specific behavioral response model you need version 2.3.1 or newer of `secr`).

#### 7.4.2 Sex effects

Incorporating sex effects into models with `secr` can be done a few different ways, but there are not pre-defined models for this. A limitation of fitting models with sex effects in `secr` is that it does not accommodate missing values of the sex variable. Thus, in all cases, individuals that are of unknown sex must be removed from the data set (recall that in a Bayesian framework we can keep these individuals in the data set by specifying a distribution for the individual covariate “sex”). In `secr`, the easiest way to include sex effects is to code sex as a “session” variable using the multi-session models (see Sec. 6.5.4 for a description of the multi-session models), providing two sessions, one representing

---

```
1. null model with a bivariate normal encounter probability model  
bear_0=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ 1, sigma ~ 1))  
  
2. null model with an exponential encounter probability model  
bear_0exp=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ 1, sigma ~ 1),  
                    detectfn=2)  
  
3. model with fixed time effects  
bear_t=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ t, sigma ~ 1))  
  
4. global behavioral model  
bear_B=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ b, sigma ~ 1))  
  
5. trap specific behavioral response  
bear_b=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ bk, sigma ~ 1))  
  
6. global behavior model with fixed time effects  
bear_bt=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ b+t, sigma ~ 1))  
  
7. sex-specific model  
bear_sex=secr.fit(bear.cap, model=list(D ~ session, g0 ~ session,  
                                         sigma ~ session))  
  
8. heterogeneity model  
bear_h2=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ h2, sigma ~ h2))
```

---

Panel 7.4: Models called from **secr.bear** function. All models use **buffer = 20000**

7698 males and one for females (see model 7 in Panel 7.4). This method provides two separate  
7699 density estimates, which can then be combined into a total density.

7700 **7.4.3 Individual heterogeneity**

7701 To incorporate heterogeneity, **secr** fits a set of finite mixture models (Norris and Pollock,  
7702 1996; Pledger, 2004). These are expensive in terms of parameters but they have been  
7703 widely adopted because they are easy to analyze using likelihood methods, as the marginal  
7704 distribution of the data is just a sum of a small number of components. Using **secr**,  
7705 individual heterogeneity can be incorporated into the encounter probability model using  
7706 default models for either a 2- or 3-component finite mixture model using the “h2” or “h3”  
7707 model terms. The 2-part mixture is shown in model 8 of panel 7.4 and the 3-part mixture  
7708 can easily be fit by substituting **h3** for **h2**. We only showed the SCR + Mh logit-normal  
7709 mixture in the version above (see Sec. 7.3.1), but finite-mixture models can also be fit in  
7710 **JAGS** or **BUGS**.

7711 **7.4.4 Model selection in secr using AIC**

7712 One practical advantage to using the **secr** package, or likelihood inference in general, is  
7713 the convenience of automatic model selection using AIC (Burnham and Anderson, 2002).  
7714 The **secr** package has a number of convenient functions for computing AIC and producing  
7715 model selection tables, or doing model-averaging (as described in Chapt. 8). Running the  
7716 function **secr.bear**, which calls all of the models we have described, will return, in addition  
7717 to all model results, an AIC table with all of the summarized results including the AIC  
7718 values, delta AIC, and model weights (see Table 7.7 or reproduce results in R using **out<-**  
7719 **secr.bear()**; **out\$AIC.tab**).

7720 It is important to note that AIC is not comparable between a multi-session model and  
7721 a model that is not a multi-session model. Therefore, to compare the sex-specific model  
7722 (which uses “sessions”) with all the other models including the null, time, and behavioral  
7723 models, we coded the data set as a multi-session design when first loading it to **secr**. This  
7724 results in all the model outputs listing separate parameter estimates for each session, even  
7725 the null model with no covariates; however, the estimates are the same for both “sessions”  
7726 in all but the sex-specific model (in other words, we don’t specify any effect of session on  
7727 parameters, except in the sex specific model).

7728 The results from this AIC analysis are straightforward to interpret; the model with  
7729 a local trap response of encounter probability, “bk”, has a model weight of 1 and thus,  
7730 according to AIC, 100% support compared with the other models in this model set. The  
7731 2-part finite mixture model for  $g_0$  and  $\sigma$  has the second lowest AIC, but considering the  
7732 large dAICc compared to the local trap response model we would probably not consider  
7733 it any further.

## 7.5 SUMMARY AND OUTLOOK

7734 There are endless covariates and encounter probability models that can be defined and our  
7735 goal in this chapter was to introduce basic types of covariate models and demonstrate how  
7736 to implement them in **BUGS** and **secr**. Essentially, SCR’s are GLMMs and therefore

**Table 7.7.** Log-likelihood, AIC, deltaAIC and AIC weight for several models run in secr for the Fort Drum black bear data set.

model	logLik	AIC	AICc	dAICc	AICwt
bear.b	-641.7215	1291.443	1292.395	0.000	1
bear.h2	-653.8382	1319.676	1321.776	29.381	0
bear.0exp	-663.9152	1333.830	1334.389	41.994	0
bear.B	-677.6175	1363.235	1364.187	71.792	0
bear.bt	-668.3044	1358.609	1366.152	73.757	0
bear.sex	-677.7151	1367.430	1369.530	77.135	0
bear.t	-674.4134	1368.827	1374.938	82.543	0
bear.0	-686.2455	1378.491	1379.049	86.654	0

we develop covariate models in much the same way, using a suitable transformation (link function) of the parameter(s). In SCR models, we typically have 2 parameters of the encounter probability model for which we might specify covariate models – the baseline encounter probability (or rate) parameter, and a scale parameter that is related in many cases to the home range size of the species. A few examples of different covariate models are given in Table 7.3. We can also consider covariates by their classification as fixed, partially observed, or unobserved (see Table 7.8). This classification of covariate types can be important because the MLE and Bayesian approaches to dealing with partially and unobserved covariates is often different. This was seen above in how the covariate **Sex** was handled in the two frameworks.

**Table 7.8.** Examples of different covariate classifications.

Covariate class	Examples
Fixed	baited, weather, habitat
Partially observed	sex, age,
Unobserved	home range size, ind. effects

While the move to spatially explicit models in capture-recapture studies has largely rendered the basic CR models (Otis et al., 1978) obsolete, we continue to find this classification useful for categorizing the *spatial* extensions of these standard CR models. The extended models include the standard  $M_0$ ,  $M_t$ ,  $M_b$ , and  $M_h$ , but also new models that allow for trap-specific information such as "baited/not-baited" or "on/off road". In addition, in Chaps. 12, 13 and 11, we explore models for explaining variation in encounter probability and density based on spatial covariates that describe variation in landscape or habitat conditions.



7755  
7756

# 8

7757

## MODEL SELECTION AND ASSESSMENT

7758 Our purpose in life is to analyze models. By that, we mean one or more of the following  
7759 basic 4 tasks: (1) estimate parameters, (2) make predictions of unobserved random vari-  
7760 ables, (3) evaluate the relative merits of different models or choosing a best model (model  
7761 selection), and (4) checking whether a specific model appears to provide a reasonable de-  
7762 scription of the data or not (model checking, assessment, or “goodness-of-fit”). In previous  
7763 chapters we addressed the problems of estimation of model parameters, and also making  
7764 predictions of latent variables,  $s$  or  $z$ , or functions of these variables such as density or  
7765 population size. In this chapter, we focus on the last two of these basic inference tasks:  
7766 model selection (which model or models should be favored), and model assessment (do  
7767 the data appear to be consistent with a particular model).

7768 In this chapter we review basic strategies of model selection using both likelihood  
7769 methods (as implemented in the `secr` package) and Bayesian analysis. Specifically, we  
7770 review a number of standard methods of model selection that apply to “variable selection”  
7771 problems, when our set of models consists of distinct covariate effects and they represent  
7772 constraints of some larger model. For classical analysis based on likelihood, model selection  
7773 by Akaike Information Criterion (AIC) is the standard approach (Burnham and Anderson,  
7774 2002). For Bayesian analysis we rely on a number of different methods. We demonstrate  
7775 the use of the deviance information criterion (DIC) (Spiegelhalter et al., 2002) for variable  
7776 selection problems although it has deficiencies when applied to hierarchical models in some  
7777 cases (Millar, 2009). We use the Kuo and Mallick indicator variable selection approach  
7778 (Kuo and Mallick, 1998) which produces direct statements of posterior model probabilities  
7779 which we think are the most useful, and leads directly to model-averaged estimates of  
7780 density. There is a good review paper recently by O’Hara and Sillanpää (2009) that  
7781 discusses these and many other related ideas for variable selection. In addition to O’Hara  
7782 and Sillanpää (2009) we also recommend Link and Barker (2010, Chapt. 7) for general  
7783 information on model selection and assessment.

7784 To check model adequacy in a Bayesian framework, or whether a specific model pro-  
7785 vides a satisfactory description of our data set, we rely exclusively on the Bayesian p-value  
7786 framework (Gelman et al., 1996). For assessing fit of SCR models, part of the challenge

7787 is coming up with good measures of model fit, and there does not appear much definitive  
 7788 guidance in the literature on this point. Following Royle et al. (2011a), we break the prob-  
 7789 lem up into 2 components which we attack separately: (1) Conditional on the underlying  
 7790 point process, does the encounter model fit? (2) Do the uniformity and independence  
 7791 assumptions appear adequate for the point process model of activity centers? The latter  
 7792 component of model fit has a considerable precedence in the ecological literature as it  
 7793 is analogous to the classical problem of testing “complete spatial randomness” (Cressie,  
 7794 1991; Illian et al., 2008).

7795 We apply some of these methods to the wolverine camera trapping data first introduced  
 7796 in Chapt. 5 to investigate sex specificity of model parameters and whether there is a  
 7797 behavioral response to encounter. We note that individuals are drawn to the camera  
 7798 trap devices by bait and therefore it stands to reason that once an individual discovers a  
 7799 trap, it might be more likely to return subsequently, a response termed “trap happiness”.  
 7800 We evaluate whether certain models for encounter probability appear to be adequate  
 7801 descriptions of the data, and we evaluate the uniformity assumption for the underlying  
 7802 point process.

## 8.1 MODEL SELECTION BY AIC

7803 Using classical analysis based on likelihood, model selection is easily accomplished using  
 7804 AIC (Burnham and Anderson, 2002) which we demonstrate below. The AIC of a model is  
 7805 simply twice the negative log-likelihood evaluated at the MLE, penalized by the number  
 7806 of parameters ( $np$ ) in the model:

$$\text{AIC} = -2\log L(\hat{\theta}|\mathbf{y}) + 2np$$

7807 Models with small values of AIC are preferred. It is common to use a modified (“cor-  
 7808 rected”) AIC referred to as  $AIC_c$  for small sample sizes which is

$$\text{AIC}_c = -2\log L(\hat{\theta}|\mathbf{y}) + \frac{2np(np+1)}{n-np-1}$$

7809 where  $n$  is the sample size. Two important problems with the use of AIC and  $AIC_c$  are  
 7810 that they don’t apply directly to hierarchical models that contain random effects, unless  
 7811 they are computed directly from the marginal likelihood (for SCR models we can do this,  
 7812 see Chapt. 6). Moreover, it is not clear what should be the effective sample size  $n$  in  
 7813 calculation of  $AIC_c$ , as there can be covariates that affect individuals, that vary over  
 7814 time, or space. We do not offer strict guidelines as to when to use a small sample size  
 7815 adjustment.

7816 The R package **secr** computes and outputs AIC automatically for each model fitted  
 7817 and it provides some capabilities for producing a model selection table (function **AIC**) and  
 7818 also doing model-averaging (function **model.average**), which we recommend for obtaining  
 7819 estimates of density from multiple models.

### 8.1.1 AIC analysis of the wolverine data

7820 We provide an example of model selection for the wolverine camera trapping data using  
 7821 **secr**. We consider a model set with distinct models to accommodate various types of sex  
 7822 specificity of model parameters:

---

7824 Model 0: model SCR0 with constant density and constant encounter model parameters;  
 7825 Model 1: model SCR0 with constant parameter values for both male and female wolverines but with sex-specific density only;  
 7826 Model 2: Sex-specific density, sex-specific  $p_0$  but constant  $\sigma$ ;  
 7827 Model 3: Sex-specific density, sex-specific  $\sigma$  but constant  $p_0$ ;  
 7828 Model 4: Sex-specific density, sex-specific  $p_0$  and sex-specific  $\sigma$ .

7830 To model sex-specific abundance (density), we use the multi-session models provided  
 7831 by **secr** (introduced in Sec. 6.5.4), which allow one to model session-specific effects on  
 7832 density, baseline encounter probability,  $p_0$  (labeled  $g_0$  in **secr**), and also the scale parameter  
 7833  $\sigma$  of the encounter probability model. Using this formulation, we define the “Session”  
 7834 variable to be a *categorical* sex code having value 1 or 2 (demonstrated below) and thus  
 7835 *session*-specific parameters represent *sex*-specific parameters. For example, if we model  
 7836 session-specific density,  $D$ , then this corresponds to Model 1 in our list above. We note  
 7837 that “Model 0” in our list corresponds to a model where all of the encounter histories  
 7838 have the same session ID. This model is one of constant density, which implies that the  
 7839 population sex ratio is fixed at 0.5, i.e.,  $\psi_{\text{sex}} = 0.5$ .

7840 Although **secr** also uses the logit/log linear predictors as the default for modeling  
 7841 covariates on baseline encounter probability and the scale parameter, respectively, **secr**  
 7842 does something different with the multi-session models. It reports estimates in a *session*  
 7843 *mean* parameterization (equivalent to, in **BUGS**, using an index variable instead of a set  
 7844 of dummy variables), and not the *session effect* (i.e., deviation from the intercept) which  
 7845 arises from the use of dummy variables. We show this **BUGS** model description in Sec.  
 7846 8.2.2.

7847 To fit these models using **secr**, we load the wolverine data and do a slight bit of  
 7848 formatting to prepare the data objects for analysis by **secr**. The key difference from our  
 7849 analysis in Chapt. 6 is, here, we use the wolverine sex information (**wolverine\$wsex**)  
 7850 which is a binary 0/1 variable (1=male) and we add 1 so that we can define a categorical  
 7851 “Session” variable (having values 1 or 2). We also have a function **scr2secr** which converts  
 7852 a standard trap-deployment file (TDF) matrix into a **secr** object of class “traps.” The  
 7853 **R** commands are as follows (contained in the help file **?secr\_wolverine**):

```

7854
7855 > library(secr)
7856 > library(scrbook)
7857 > data(wolverine)
7858 > traps <- as.matrix(wolverine$wtraps)

7859 ## Name variables as required by secr
7860 > dimnames(traps) <- list(NULL,c("trapID","x","y",paste("day",1:165,sep="")))
7861 ## Convert trap information to a secr "traps" object
7862 > trapfile <- scr2secr(scrtraps=traps,type="proximity")

7863 ## Grab the wolverine state-space grid (2km here)
7864 > gr <- as.matrix(wolverine$grid2)
7865 > dimnames(gr) <- list(NULL,c("x","y"))
7866 > gr2 <- read.mask(data=gr)

```

```

7869
7870 ## Grab the encounter data, and re-name variables
7871 > wolv.dat <- wolverine$wcaps
7872 > dimnames(wolv.dat) <- list(NULL,c("Session","ID","Occasion","trapID"))
7873
7874 ## Convert binary 0/1 sex variable to categorical 1/2 for "session"
7875 > wolv.dat[,1] <- wolverine$wsex[wolv.dat[,2]]+1
7876 > wolv.dat <- as.data.frame(wolv.dat)
7877
7878 ## Convert to capthist object
7879 > wolvcapt <- make.capthist(wolv.dat,trapfile,fmt="trapID",noccasions=165)

```

Once the data have been prepared in this way, we use the `secr` model fitting function `secr.fit` to fit the different models, and then the function `AIC` to package the models together and summarize them in the form of an AIC table, with rows of the table ordered from best to worst. The function `model.average` performs AIC-based model-averaging of the parameters specified by the `realnames` variable (below this is demonstrated for the parameter density,  $D$ ). Because this function defaults to averaging by  $\text{AIC}_c$ , we slightly modified this function (called `model.average2`) to do model averaging by either  $\text{AIC}$  or  $\text{AIC}_c$  as specified by the user. The model fitting commands look like this (for Model 0 and Model 1):

```

7889 > model0 <- secr.fit(wolvcapt, model=list(D~1, g0~1, sigma~1),
7890                 buffer=20000)
7891 > model1 <- secr.fit(wolvcapt, model=list(D~session, g0~1, sigma~1),
7892                 buffer=20000)

```

Next we use the function `AIC`, passing the fit objects from all 5 models, and that produces the following output (abbreviated horizontally to fit on the page):

```

7893 > AIC (model0,model1,model2,model3,model4)
7894      model   ... npar logLik   AIC   AICc dAICc  AICwt
7895 model0 D~1 g0~1 sigma~1 ... 3 -627.2603 1260.521 1261.932 0.000 0.5831
7896 model1 .. ... 5 -624.9051 1259.810 1263.810 1.878 0.2280
7897 model2 .. ... 4 -627.2365 1262.473 1264.973 3.041 0.1275
7898 model3 .. ... 6 -624.6632 1261.326 1267.326 5.394 0.0393
7899 model4 .. ... 5 -627.2358 1264.472 1268.472 6.540 0.0222

```

Model averaging the results is done as follows:

```

7900 > model.average (model0,model1,model2,model3,model4,realnames="D")
7901      estimate  SE.estimate      lcl      ucl
7902 session=1 2.707190e-05 7.913577e-06 1.544474e-05 4.745224e-05
7903 session=2 2.927423e-05 8.270402e-06 1.700631e-05 5.039193e-05

```

As usual, estimates and standard errors of the individual model parameters can be obtained from the `secr.fit` summary output of any of the `modelX` objects shown above. The default output of estimated density is in individuals per ha, so we have to scale this up to something more reasonable. To get into units of per 1000 km<sup>2</sup>, we need to first

7911 multiply by 100 to get to units of  $\text{km}^2$  and then multiply by 1000. This produces an  
 7912 estimated density of about 2.71 for `session=1` (females) and 2.93 for `session=2` (males).  
 7913 We can use the generic **R** function `predict` applied to the `secr.fit` output to obtain  
 7914 specific information about the MLEs on the natural scale.

7915 We don't necessarily agree with the use of  $AIC_c$  here and think its better to use AIC,  
 7916 in general. This is because, as noted previously, it is not clear what the effective sample  
 7917 size is for most capture-recapture problems. While we have 21 individuals in the data  
 7918 set, most of the model structure has to do with encounter probability samples and for  
 7919 that there are hundreds of observations. We do note that the AIC and  $AIC_c$  results are  
 7920 not entirely consistent. By looking at the best model by AIC (Table 8.1), we find that  
 7921 the model with sex specific density and sex-specific baseline encounter probability,  $p_0$ , is  
 7922 preferred (Model 2). This is just slightly better than the null model (Model 0) with no  
 7923 sex effects at all and hence an implied fixed sex ratio of  $\psi_{\text{sex}} = 0.50$ .

**Table 8.1.** Model selection results for the wolverine models of sex specificity, with/without habitat mask. Fitting was done using `secr` with a half-normal (Gaussian) encounter probability model. Models are ordered by *AIC*. Density, *D*, is reported in units of individuals per 1000  $\text{km}^2$ . Model abbreviations indicate which parameters are sex-specific in order  $D/p_0/\sigma$ .

NO HABITAT MASK										
model	npar	Female			Male			D	$p_0$	$\sigma$
		AIC	AICc	D	$p_0$	$\sigma$				
2: sex/sex/1	5	1259.8	1263.8	2.45	0.08	6435.51	3.16	0.04	6435.51	
0: 1/1/1	3	1260.5	1261.9	2.83	0.06	6298.66	2.83	0.06	6298.66	
4: sex/sex/sex	6	1261.3	1267.3	2.59	0.08	6080.70	2.99	0.04	6833.16	
1: sex/1/1	4	1262.5	1265.0	2.69	0.06	6298.69	2.96	0.06	6298.69	
3: sex/1/sex	5	1264.5	1268.5	2.70	0.06	6280.49	2.95	0.06	6319.03	
WITH HABITAT MASK										
model	npar	Female			Male			D	$p_0$	$\sigma$
		AIC	AICc	D	$p_0$	$\sigma$				
2: sex/sex/1	5	1268.1	1272.1	3.64	0.07	6382.88	4.73	0.03	6382.88	
4: sex/sex/sex	6	1268.7	1274.7	3.87	0.07	5859.40	4.41	0.03	7039.09	
0: 1/1/1	3	1271.2	1272.6	4.18	0.05	6282.62	4.18	0.05	6282.62	
1: sex/1/1	4	1273.1	1275.6	3.98	0.05	6282.65	4.38	0.05	6282.65	
3: sex/1/sex	5	1275.1	1279.1	3.93	0.05	6357.26	4.41	0.05	6220.22	

7924 We fit the same models but now using a modified state-space which excludes the ocean  
 7925 (this is a habitat mask in `secr`). Results are shown in Table 8.1 along with the previous  
 7926 models without a mask. We see AIC values are smaller for the model without the mask.  
 7927 It is probably acceptable to compare these different fits (with and without habitat mask)  
 7928 by AIC because we recognize the mask as having the effect of modifying the random  
 7929 effects distribution (i.e., of the activity centers, *s*) and the results should be sensitive to  
 7930 choice of the distribution for *s*. That said, we tend to prefer the mask model because it  
 7931 makes sense to exclude the areas of open water from the state-space of *s*. For females the  
 7932 model-averaged density is 3.88 individuals per 1000  $\text{km}^2$  and for males the model-averaged  
 7933 density estimate is 4.46 individuals per 1000  $\text{km}^2$  as we see here:

7934 > `model.average (model0b,model1b,model2b,model3b,model4b,realnames="D")`

```

7935
7936      estimate   SE.estimate      lcl      ucl
7937 session=1 3.876615e-05 1.189102e-05 2.153795e-05 6.977518e-05
7938 session=2 4.459658e-05 1.323696e-05 2.523280e-05 7.882022e-05

```

7939 This is quite a bit higher than that based on the rectangular state-space (i.e., not  
 7940 specifying a habitat mask). This is not surprising given that **the state-space is part**  
 7941 **of the model** and the specific state-space modification we made here, which reduces the  
 7942 area from the rectangular state-space, should be extremely important from a biological  
 7943 standpoint (i.e., wolverines are not actively using open ocean).

## 8.2 BAYESIAN MODEL SELECTION

7944 Model selection is somewhat less straightforward as a Bayesian, and there is no canned  
 7945 all-purpose method like AIC. As such we recommend a pragmatic approach, in general,  
 7946 for all problems, based on a number of basic considerations:

- 7947 (1) For a small number of fixed effects we think it is reasonable to adopt a conventional  
 7948 “hypothesis testing” approach – i.e., if the posterior for a parameter overlaps zero  
 7949 substantially, then it is probably reasonable to discard that effect from the model.
- 7950 (2) Calculation of posterior model probabilities: In some cases we can implement methods  
 7951 which allow calculation of posterior model probabilities. One such idea is the indicator  
 7952 variable selection method from Kuo and Mallick (1998). For this, we introduce a latent  
 7953 variable  $w \sim \text{Bern}(.5)$  and expand the model to include the variable  $w$  as follows:

$$\text{logit}(p_{ijk}) = \alpha_0 + w * \alpha_1 * C_{ijk}.$$

7954 The importance of the covariate  $C$  is then measured by the posterior probability that  
 7955  $w = 1$ .

7956 (3) The Deviance Information Criterion (DIC): Bayesian model selection is now routinely  
 7957 carried out using DIC ((Spiegelhalter et al., 2002)), although its effectiveness in hier-  
 7958 archical models depends very much on the manner in which it is constructed (Millar,  
 7959 2009). We recommend using it if it leads to sensible results, but we think it should be  
 7960 calibrated to the extent possible for specific classes of models. This has not yet been  
 7961 done in the literature for SCR models, to our knowledge.

7962 (4) Logical argument: For something like sex specificity of certain parameters, it seems  
 7963 to make sense to leave an extra parameter in the model no matter what because, bio-  
 7964 logically, we might expect a difference (e.g., home range size). In some cases failure to  
 7965 apply logical argument leads to meaningless tests of gratuitous hypotheses (Johnson,  
 7966 1999).

7967 In all modeling activities, as in life itself, the use of logical argument should not be under-  
 7968 utilized.

### 8.2.1 Model selection by DIC

7970 The availability of AIC makes the use of likelihood methods convenient for problems where  
 7971 likelihood estimation is achievable. For Bayesian analysis, DIC seemed like a general-  
 7972 purpose equivalent, at least for a brief period of time after its invention. However, there

7973 seem to be many variations of DIC, and a consistent version is not always reported across  
 7974 computing platforms. Even statisticians don't have general agreement on practical issues  
 7975 related to the use of DIC (Millar, 2009). Despite this, it is still widely reported. We think  
 7976 DIC is probably reasonable for certain classes of models that contain only fixed effects,  
 7977 or for which the latent variable structure is the same across models so that only the fixed  
 7978 effects are varied (this covers many SCR model selection problems). However, it would be  
 7979 useful to see some calibration of DIC for some standardized model selection problems.

7980 Model deviance is defined as negative twice the log-likelihood; i.e., for a given model  
 7981 with parameters  $\theta$ :  $\text{Dev}(\theta) = -2 * \log L(\theta|\mathbf{y})$ . The DIC is defined as the posterior mean  
 7982 of the deviance,  $\overline{\text{Dev}}(\theta)$ , plus a measure of model complexity,  $p_D$ :

$$\text{DIC} = \overline{\text{Dev}}(\theta) + p_D$$

7983 The standard definition of  $p_D$  is

$$p_D = \overline{\text{Dev}}(\theta) - \text{Dev}(\bar{\theta})$$

7984 where the 2nd term is the deviance evaluated at the posterior mean of the model parameter(s),  $\bar{\theta}$ . The  $p_D$  here is interpreted as the effective number of parameters in the model.  
 7985 Gelman et al. (2004) suggest a different version of  $p_D$  based on one-half the posterior  
 7986 variance of the deviance:  
 7987

$$p_V = \text{Var}(\text{Dev}(\theta)|\mathbf{y})/2.$$

7988 This is what is produced from **WinBUGS** and **JAGS** if they are run from **R2WinBUGS** or  
 7989 **R2jags**, respectively. It is less easy to get DIC summaries from **rjags**, so we used **R2jags**  
 7990 in our analyses below.

### 7991 8.2.2 DIC analysis of the wolverine data

7992 We repeated the analysis of the wolverine models with sex specificity, but this time doing  
 7993 a Bayesian analysis paralleling the likelihood analysis we did above in **secr**, using the  
 7994 logit/log parameterization of the model parameters. To do so in **BUGS**, we used dummy  
 7995 variables. Thus, we can express models allowing for sex specificity using a dummy variable  
 7996 **Sex** and new parameters ( $\alpha_{sex}$ ,  $\beta_{sex}$ ) which represent the effect of **Sex** at level 1:

$$\text{logit}(p_{0,i}) = \alpha_0 + \alpha_{sex} \mathbf{Sex}_i$$

7997 and

$$\log(\sigma_i) = \log(\sigma_0) + \beta_{sex} \mathbf{Sex}_i.$$

7998 In these expressions, the sex variable  $\mathbf{Sex}_i$  is a binary variable where  $\mathbf{Sex}_i = 0$  corresponds to female, and  $\mathbf{Sex}_i = 1$  corresponds to male.

8000 Unlike the multi-session model in **secr**, we carry out the analysis of the sex-specific  
 8001 model here by putting all of the data into a single data set, and explicitly accounting for  
 8002 the covariate 'sex' in the model by assigning it a Bernoulli prior distribution with  $\psi_{sex}$   
 8003 being the proportion of males in the population. In this case, we produce "Model 0" above,  
 8004 the model with no sex effect on density, by setting the population proportion of males at  
 8005 one-half:  $\psi_{sex} = 0.5$  (see also Sec. 7.2.4). As usual, handling of missing values of the  
 8006 sex variable is done seamlessly which might be a practical advantage of Bayesian analysis

8007 in situations where sex is difficult to record in the field which may lead to individuals of  
 8008 unknown sex (i.e., missing values).

8009 The **BUGS** model specification for the most complex model, Model 4, is shown in  
 8010 Panel 8.1. This model has sex-specific intercept, scale parameter,  $\sigma$ , and density. We  
 8011 provide an **R** script named `wolvSCR0ms` in the `scrbook` package which will fit each model.  
 8012 The function uses **JAGS** by default for the fitting, using the `R2jags` package. The kernel  
 8013 of this function is the model specification in Panel 8.1, which gets modified depending on  
 8014 the model we wish to fit using a command line option `model`. For example, `model = 1`  
 8015 fits the model with constant parameter values for males and females, but sex-specific  
 8016 population sizes (`model = 0` constrains the male probability parameter,  $\psi_{sex}$ , to be 0.5).  
 8017 The **R** function fits each of the 5 models using a binary indicator variable to turn ‘on’ or  
 8018 ‘off’ each effect. Here is how we obtain the MCMC output for each of the 5 models:

```
8019 > wolv0 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=0)
8020 > wolv1 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=1)
8021 > wolv2 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=2)
8022 > wolv3 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=3)
8023 > wolv4 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=4)
```

8024 We fitted the 5 models to the wolverine data and summarize the DIC computation  
 8025 results in Table 8.2. The model rank has model 0, model 2, model 1, model 4, model 3.  
 8026 Interestingly, this is the same order as the models based on AIC<sub>c</sub> which we found above  
 8027 (see Table 8.1). The posterior mean and SD of model parameters under the 5 models are  
 8028 given in Table 8.3.

**Table 8.2.** DIC results for the 5 models of sex specificity fitted to the wolverine camera trapping data, using the function `wolvSCR0ms`. Results are based on 3 chains of length 61000 yielding 180000 posterior samples.

Model	Meandev	$p_D$	DIC	Rank
Model 0	441.01	77.09	518.10	1
Model 1	441.78	77.504	519.28	3
Model 2	440.12	78.440	518.56	2
Model 3	443.31	79.478	522.79	5
Model 4	441.24	80.078	521.32	4

### 8029 8.2.3 Bayesian model averaging with indicator variables

8030 A convenient way to deal with model selection and averaging problems in Bayesian analysis  
 8031 by MCMC is to use the method of model indicator variables (Kuo and Mallick,  
 8032 1998). Using this approach, we expand the model to include a set of prescribed models  
 8033 as specific reductions of a larger model. This has been demonstrated in some specific  
 8034 capture-recapture models in Royle and Dorazio (2008, Sec. 3.4.3), and Royle (2009b) and  
 8035 in the context of SCR by Tobler et al. (2012). A useful aspect of this method is that  
 8036 model-averaged parameters are produced by default. We emphasize the need to be care-  
 8037 ful of reporting model-averaged parameters that don’t have a common interpretation in

```

alpha.sex ~ dunif(-3,3)                      ## Prior distributions
beta.sex ~ dunif(-3,3)
sigma0 ~ dunif(0,50)
alpha0 ~ dnorm(0,.1)
psi ~ dunif(0,1)                             ## Data augmentation parameter
psi.sex ~ dunif(0,1)                          ## Probability of 'male'

for(i in 1:M){                                ## DA loop
  wsex[i] ~ dbern(psi.sex)                   ## Latent sex state (male = 1)
  z[i] ~ dbern(psi)                         ## DA variables, activity centers, etc..
  s[i,1] ~ dunif(Xl,Xu)
  s[i,2] ~ dunif(Yl,Yu)
  logit(p0[i]) <- alpha0 + alpha.sex*wsex[i]
  log(sigma.vec[i]) <- log(sigma0) + beta.sex*wsex[i]
  alpha1[i] <- 1/(2*sigma.vec[i]*sigma.vec[i])
  for(j in 1:ntraps){
    mu[i,j] <- z[i]*p[i,j]
    y[i,j] ~ dbin(mu[i,j],K[j])
    dd[i,j] <- pow(s[i,1] - traplocs[j,1],2) + pow(s[i,2] - traplocs[j,2],2)
    p[i,j] <- p0[i]*exp( - alpha1[i]*dd[i,j] )
  }
}

```

Panel 8.1: Part of the **BUGS** specification for a complete sex specificity of model parameters. This is a simplified version of the model contained in the **wolvSCR0ms** script, because it does not contain the on/off switches for creating the various sub-models.

**Table 8.3.** Posterior summaries of model parameters for models with varying sex specificity of model parameters. Model 0 = no sex specificity, model 4 = fully sex-specific (see text). Models are based on the Gaussian encounter probability model, each with 21000 iterations, 1000 burn-in, 3 chains for a total of 60000 posterior samples.

Parameter	model 0		model 1		model 2		model 3		model 4	
	Mean	SD								
$N$	60.02	11.91	60.24	11.93	59.37	11.97	59.67	11.97	58.77	11.75
$D$	5.79	1.15	5.81	1.15	5.72	1.15	5.75	1.15	5.66	1.13
$\alpha_0$	-2.81	0.18	-2.82	0.17	-2.44	0.25	-2.82	0.18	-2.43	0.25
$\alpha_{sex}$	0.00	1.73	0.00	1.73	-0.75	0.34	0.00	1.73	-0.79	0.36
$\sigma_0$	0.64	0.06	0.64	0.05	0.66	0.06	0.65	0.08	0.63	0.09
$\beta_{sex}$	0.00	1.73	-0.01	1.73	0.01	1.74	-0.01	0.17	0.10	0.18
$\psi_{sex}$	0.50	0.29	0.52	0.10	0.56	0.10	0.52	0.11	0.54	0.11
$\psi$	0.30	0.07	0.30	0.07	0.30	0.07	0.30	0.07	0.30	0.07
deviance	441.01	12.42	441.78	12.45	440.12	12.53	443.31	12.61	441.24	12.66
	$pD = 77.1$		$pD = 77.5$		$pD = 78.4$		$pD = 79.5$		$pD = 80.1$	
	$DIC = 518.1$		$DIC = 519.3$		$DIC = 518.6$		$DIC = 522.8$		$DIC = 521.3$	

the different models because they are meaningless (averaging apples and oranges....). For example, if a regression parameter is in a specific model then the posterior is informed by the data and a specific MCMC draw is from the appropriate posterior distribution. On the other hand, if the regression parameter is not in the model then the MCMC draw is obtained directly from the prior distribution, and so we need to think carefully about whether it makes sense to report an average of such a thing (in the vast majority of cases the answer is no). But some parameters like  $N$  or density,  $D$ , do have a consistent interpretation and we support producing model-averaged results of those parameters.

To implement the Kuo and Mallick approach, we expand the model to include the latent indicator variables, say  $w_m$ , for variable  $m$  in the model, such that

$$w_m = \begin{cases} 1 & \text{linear predictor includes covariate } m \\ 0 & \text{linear predictor does not include covariate } m \end{cases}$$

We assume that the indicator variables  $w_m$  are mutually independent with

$$w_m \sim \text{Bernoulli}(0.5)$$

for each variable  $m = 1, 2, \dots$ , in the model. For example, with 2 variables, the expanded model has the linear predictor:

$$\text{logit}(p_{ijk}) = \alpha_0 + \alpha_1 w_1 C_{1,i} + \alpha_2 w_2 C_{2,ijk}$$

where, let's suppose,  $C_{1,i}$  is an individual covariate such as sex, and  $C_{2,ijk}$  is a behavioral response covariate which is individual-, trap-, and occasion-specific. We can assume a parallel model specification on the parameter  $\sigma$  which is liable to vary by individual level covariates such as sex:

$$\log(\sigma_i) = \beta_0 + \beta_1 w_3 C_{1,i}.$$

Using this indicator variable formulation of the model selection problem we can characterize unique models by the sequence of  $w$  variables. In this case, each unique sequence  $(w_1, w_2, w_3)$  represents a model, and we can tabulate the posterior frequencies of each model by post-processing the MCMC histories of  $(w_1, w_2, w_3)$ , as we demonstrate shortly. This method then evaluates all possible combinations of covariates or  $2^m$  models.

Conceptually, analysis of this expanded model within the data augmentation framework does not pose any additional difficulty. One broader, technical consideration is that posterior model probabilities are well known to be sensitive to priors on parameters (Aitkin, 1991; Link and Barker, 2006). See also Royle and Dorazio (2008, Sec. 3.4.3) and Link and Barker (2010, Sec. 7.2.5). What might normally be viewed as vague or non-informative priors, are not usually innocuous or uninformative when evaluating posterior model probabilities. The use of AIC seems to avoid this problem largely by imposing a specific and perhaps undesirable prior that is a function of the sample size (Kadane and Lazar, 2004). One solution is to compute posterior model probabilities under a model in which the prior for parameters is fixed at the posterior distribution under the full model (Aitkin, 1991). At a minimum, one should evaluate the sensitivity of posterior model probabilities to different prior specifications.

### Analysis of the wolverine data

The **R** script `wolvSCR0ms` in the package `scrbook` provides the model indicator variable implementation for the fully sex-specific SCR model. It is run by setting `model=5` in the function call. We note again that it is not very useful to report most parameter estimates from this model because their marginal posterior is a mixture from the prior (when a value of the indicator variable of 0 is sampled) and draws informed by the data (i.e., from the posterior, when a 1 is drawn for the indicator variable  $w$ ). On the other hand, the parameters  $N$  and density  $D$  should be reported and they represent marginal posteriors over all models in the model set. In effect, model averaging is done as part of the MCMC sampling. The variable ‘mod’ contains the two binary indicator variables ( $w$  above) which pre-multiply the ‘sex’ term in each of the  $p_0$  and  $\sigma$  model components, like this:

$$\text{logit}(p_{0,i}) = \alpha_0 + \text{mod}[1]\alpha_{\text{sex}}\text{sex}_i$$

and

$$\log(\sigma_i) = \log(\sigma_0) + \text{mod}[2]\beta_{\text{sex}}\text{sex}_i$$

The third element of `mod` determines whether the  $\psi_{\text{sex}}$  parameter is estimated or fixed at  $\psi_{\text{sex}} = 0.5$  which is accomplished with the line of **BUGS** code as follows:  
`sex.ratio <- psi.sex*mod[3] + .5*(1-mod[3]).`  
 The MCMC output for ‘mod’ was post-processed to obtain the model-weights using the following **R** commands:

```

8089 > mod <- wolv5$BUGSoutput$sims.list$mod
8090 > mod <- paste(mod[,1],mod[,2],mod[,3],sep="")
8091 >
8092 > table(mod)
8093 mod
8094   000   001   010   011   100   101   110   111
8095 17181 4935 1057 296 25211 8337 2275 708
8096
8097 > round( table(mod)/length(mod) , 3)
8098 mod
8099   000   001   010   011   100   101   110   111
8100 0.286 0.082 0.018 0.005 0.420 0.139 0.038 0.012

```

8101 This results in a comparison of all 8 possible models (based on  $m = 3$  covariates) instead  
 8102 of just the 5 models we originally proposed. We see that the best model is that labeled  
 8103 100 which, according to our construction above, has `mod[1]=1, mod[2]=0` and `mod[3]=0`.  
 8104 This is the model having sex-specific baseline encounter probability  $p_0$ , and  $\psi_{sex} = 0.5$ .  
 8105 This model has posterior model probability 0.420. The model with no sex specificity at  
 8106 all (the model with label 000) has posterior probability 0.286 and the remaining posterior  
 8107 mass is distributed over the other six models. We could arrive at a qualitatively similar  
 8108 conclusion using a more ad hoc approach based on looking at the posterior mass for each  
 8109 parameter under the full model (model 4; see Table 8.3, in part). Considering the sex-  
 8110 specific intercept, it appears to be very important as its posterior mass is mostly away  
 8111 from 0. On the other hand, the coefficient on log-sigma is concentrated around 0, and  
 8112 the estimated  $\psi_{sex}$  (probability that an individual is a male) is 0.54 with a large posterior  
 8113 standard deviation. We might therefore be inclined to discard the sex effect on  $\log(\sigma)$   
 8114 based on classical thinking-like-a-hypothesis-testing-person and settle for the model with  
 8115 a sex-specific intercept in the encounter probability model. This is consistent with our  
 8116 indicator variable approach which found that model (1,0,0) has posterior probability of  
 8117 0.420. Looking at the posteriors for each parameter to thin the model down is consistent  
 8118 with these results. We can obtain model-averaged estimates from the indicator variable  
 8119 approach, which produces direct model-averaged estimates of  $N$  and  $D$ :

```
8120      mu.vect sd.vect   2.5%    25%    50%    75%  97.5% Rhat n.eff
8121 D      5.695   1.133  3.759  4.916  5.591  6.362  8.193 1.002 3600
8122 N     59.077  11.758 39.000 51.000 58.000 66.000 85.000 1.002 3600
```

8123 We obtain a model-averaged estimate (posterior mean) for density of  $D = 5.695$  which  
 8124 is hardly any different from our model specific estimates (Table 8.3) and, in particular,  
 8125 from model 2 which has only a sex-specific intercept.

#### 8126 8.2.4 Choosing among detection functions

8127 Another approach to implementing model indicator variables is to introduce a categorical  
 8128 “model identity” variable which is itself a parameter of the model. Using this approach,  
 8129 then each distinct model is associated with a unique set of covariates or other set of model  
 8130 features. This is convenient especially when we cannot specify the linear predictor as  
 8131 some general model that reduces to various alternative sub-models simply by switching  
 8132 binary variables on or off. In the context of SCR models, choosing among different en-  
 8133 counter probability models would be an example. For this case we do something like this  
 8134 `mod ~ dcat(probs[])` where `probs` is a vector with elements  $1/(\#models)$ , and the en-  
 8135 counter probability matrix is filled in depending on the value of `mod`. In particular, instead  
 8136 of a 2-dimensional array `p[i,j]`, we build `p[i,j,m]` for each of  $m = 1, 2, \dots, M$  models.  
 8137 An example with 3 distinct models is:

```
8138   mod ~ dcat(probs[])
8139   ##
8140   ## Using a double loop construction fill-in p[,] for each model:
8141   ##
8142   p[i,j,1] <- p0[1]*exp( - alpha1[1]*dist2[i,j] )
```

---

```

8143 p[i,j,2] <- 1-exp(-p0[2]*exp( - alpha1[2]*dist2[i,j] ) )
8144 logit(p[i,j,3]) <- p0[3] - alpha1[3]*dist2[i,j]
8145
8146 mu[i,j] <- z[i]*p[i,j,mod]
8147 y[i,j] ~ dbin(mu[i,j],K[j])

```

8148 As before the posterior probabilities can be highly sensitive to priors on the different  
8149 model parameters and sometimes mixing is really poor and, in general, we've experienced  
8150 mixed success trying to carry out model selection using this construction. We do provide  
8151 a template **R/JAGS** script (`wolvSCR0ms2`) in the `scrbook` package which has an example  
8152 of choosing among 3 different encounter probability models: The Gaussian encounter  
8153 probability, Gaussian hazard, and logistic model with the square of distance (defined  
8154 in Sec. 7.1). The key things to note are that there are 3 intercepts and 3 different  
8155 ‘`alpha1`’ parameters (the coefficient on distance). The parameters should not be regarded  
8156 as equivalent across the models, so it is important to have them separately defined (and  
8157 estimated) for each model. In our analysis we used a vague normal prior (precision = 0.1)  
8158 for the intercept parameter (either log or logit-scale of baseline encounter probability  $p_0$ )  
8159 and a `Uniform(0,5)` prior for one-half the inverse of the coefficient on distance-squared. In  
8160 the **BUGS** model specification the priors look like this:

```

8161 for(i in 1:3){
8162   alpha0[i] ~ dnorm(0,.1)
8163   sigma[i] ~ dunif(0,5)
8164   alpha1[i] <- 1/(2*sigma[i]*sigma[i])
8165 }

```

8166 Then, we create a probability of encounter for each individual, trap *and* model so that  
8167 the holder object “`p`” in the model description is a 3-dimensional array (sometimes this  
8168 would have to be a 4 or 5-d array in more complex models with time effects, etc..), so that  
8169 construction of the encounter probability models look like this:

```

8170 p[i,j,1] <- p0[1]*exp( - alpha1[1]*dist2[i,j] )
8171 p[i,j,2] <- 1-exp(-p0[2]*exp( - alpha1[2]*dist2[i,j] ) )
8172 logit(p[i,j,3]) <- p0[3] - alpha1[3]*dist2[i,j]

```

8173 where

```

8174 logit(p0[1]) <- alpha0[1]
8175 log(p0[2]) <- alpha0[2]
8176 p0[3] <- alpha0[3]

```

8177 You can experiment with the `wolvSCR0ms2` script to investigate the importance of different  
8178 models of encounter probability and whether they have an affect on the inferences.

### 8.3 EVALUATING GOODNESS-OF-FIT

8179 In practical settings, we estimate parameters of a desirable model, or maybe fit a bunch  
8180 of models and report estimates from all of them or a model-averaged summary of density.

8181 An important question is: Is our model worth anything? In other words, does the model  
8182 appear to be an adequate description of our data? Formal assessment of model adequacy or  
8183 goodness-of-fit is a challenging problem and there are no all-purpose algorithms for doing  
8184 this in either frequentist or Bayesian paradigms. Moreover, there are some philosophical  
8185 challenges to evaluating model fit, such as, if we do model averaging then should all of  
8186 the models have to fit? Or should the averaged model have to fit? What if none of the  
8187 models fit? We don't know the answers to these questions and we won't try to answer  
8188 them. Instead, we will provide what guidance we can on taking the first steps to evaluating  
8189 fit, of a single model, as if it were a cherished family heirloom of great importance. We  
8190 suggest that if you have a model that you really like, a single model, then it is a sensible  
8191 thing to check that the model is a good fit to your data. If it is not, we do not imagine  
8192 that the model is useless but just that some thought should be put into why the model  
8193 doesn't fit so that, perhaps, some remediation might happen as future data are collected.  
8194 After all, you may have spent 2, 3 or many more years of your life collecting that data set,  
8195 perhaps thousands of hours, and therefore it seems a reasonable proposition to expect to  
8196 do some estimation and analysis of the model regardless of model fit. You can still learn  
8197 something from a model that does not pass some technical litmus test of model fit.

8198 Conceptually, we can think of evaluation of model fit as follows: if we simulate data  
8199 under the model in question, do the simulated realizations resemble the data set that we  
8200 actually have? For either Bayesian or classical inference, the basic strategy to assessing  
8201 model fit is to come up with a fit statistic that depends on the parameters and the data  
8202 set, which we denote by  $T(\mathbf{y}, \theta)$ , and then we compute this for the observed data set, and  
8203 compare its value to that computed for perfect data sets simulated under the correct model.  
8204 In the case of classical inference, we will often rely on the standard practice of parametric  
8205 bootstrapping (Dixon, 2002), where we simulate data sets conditional on the MLE  $\hat{\theta}$  and  
8206 compare realizations with what we've observed. The R package **unmarked** (Fiske and  
8207 Chandler, 2011) contains generic bootstrapping methods for certain hierarchical models,  
8208 including distance sampling (e.g., see Sillett et al., 2012, for an application). In simple  
8209 cases, using classical inference methods, it is sometimes possible to identify a test statistic  
8210 of theoretical merit, perhaps with a known asymptotic distribution. For examples from  
8211 capture-recapture see Burnham et al. (1987), Lebreton et al. (1992), and Chapt. 5 of  
8212 Cooch and White (2006). For Bayesian analysis we use the Bayesian p-value method  
8213 (Gelman et al., 1996) (we introduced the Bayesian p-value in sec. 3.9.1). Using this  
8214 approach, data sets are simulated based on a posterior sample of the model parameters  
8215  $\theta$  and some fit statistic for the simulated data sets, usually based on the discrepancy of  
8216 the observed data from its expected values, is compared to that for the actual data. In  
8217 most cases, whether Bayesian or frequentist, the main idea for assessing model fit is the  
8218 same: We compare data sets from the model we're interested in with the data set we have  
8219 in hand. If they appear to be consistent with one another, then our faith in the model  
8220 increases, at least to some extent, and we say "the model fits."

8221 To date, we are unaware of any goodness-of-fit applications based on likelihood analysis  
8222 of SCR models. For Bayesian analysis of SCR models, there has not been a definitive or  
8223 general proposal for a fit statistic or even a class of fit statistics, although a few specialized  
8224 implementations of Bayesian p-values have been provided (Royle, 2009b; Gardner et al.,  
8225 2010a; Royle et al., 2011a; Gopalaswamy et al., 2012a,b; Russell et al., 2012). While  
8226 we universally adopt the Bayesian p-value approach, and suggest some fit statistics in

the following text, we caution that there is no general expectation to support how well they should do. As such, one might consider doing some kind of custom evaluation or calibration when using such methods, if the power of the test (ability to reject under specific departures from the model) is of paramount interest. We note that this uncertain power or performance of the Bayesian p-value is not a weakness of the Bayesian approach because the same issue applies in using bootstrap approaches applied to classical analysis of models, if we were to devise such methods.

## 8.4 THE TWO COMPONENTS OF MODEL FIT

For most SCR models, there are at least two distinct components of model fit, and we propose to evaluate these two distinct components individually. First, we can ask, are the data consistent with the *observation* model, conditional on the underlying point process? We can evaluate this based on the encounter frequencies of individuals *conditional* on (posterior samples of) the underlying point process  $\mathbf{s}_1, \dots, \mathbf{s}_N$ . We discuss some potential fit statistics for addressing this in the next section. Second, we can evaluate whether the data appear consistent with the *state* process model (i.e., the “uniformity” assumption of the point process). For the simple model of independence and uniformity, this is similar to the assumption of *complete spatial randomness* (CSR) which we consider in Sec. 8.4.1 below. Actually, this is not strictly the assumption of CSR because of the binomial assumption on  $N$  under data augmentation, so we instead use the term *spatial randomness*.

### 8.4.1 Testing uniformity or spatial randomness

Historically, especially in ecology, there has been an extraordinary amount of interest in whether a realization of a point process indicates “complete spatial randomness,” i.e., that the points are distributed uniformly and independently in space. Two good references for such things are Cressie (1991, Ch. 8) and Illian et al. (2008)<sup>1</sup>. In the context of animal capture-recapture studies, the spatial randomness hypothesis is manifestly false, purely on biological grounds. Typically individuals will be clustered, or more regular (for territorial species), than expected under spatial randomness and heterogeneous habitat will generate the appearance of clustering even if individuals are distributed independently of one another. While we recommend modeling spatial structure explicitly when possible (Chapts. 11, 12, 13), the uniformity assumption may be an adequate description of data sets in some situations. Further, we find that it is generally flexible enough to reflect non-uniform patterns in the data, because we do observe some direct information about some of the point locations.

The basic technical framework for evaluating the spatial randomness hypothesis is based on counts of activity centers in cells or bins. For that we use any standard goodness-of-fit test statistic, based on gridding (i.e., binning) the state-space of the point process into  $g = 1, 2, \dots, G$  cells or bins, and we tabulate  $N_g \equiv N(\mathbf{x}_g)$  the number of activity centers in bin  $g$ , centered at coordinate  $\mathbf{x}_g$ . Specifically, let  $B(\mathbf{x})$  indicate a bin centered at coordinate

<sup>1</sup>We also like Tony Smith’s lecture notes (Univ. of Penn. ESE 502), which can be found at [http://www.seas.upenn.edu/~ese502/NOTEBOOK/Part\\_I/3\\_Testing\\_Spatial\\_Randomness.pdf](http://www.seas.upenn.edu/~ese502/NOTEBOOK/Part_I/3_Testing_Spatial_Randomness.pdf), accessed January 24, 2013.

8264  $\mathbf{x}$ , then<sup>2</sup>  $N(\mathbf{x}) = \sum_{i=1}^N I(\mathbf{s}_i \in B(\mathbf{x}))$  is the population size of bin  $B(\mathbf{x})$ . In Sec. 5.11.1,  
 8265 we used the summaries  $N(\mathbf{x})$  for producing density maps from MCMC output. Here, we  
 8266 use them for constructing a fit statistic. We have used the Freeman-Tukey statistic of this  
 8267 form:

$$T(\mathbf{N}, \theta) = \sum_g (\sqrt{N_g} - \sqrt{\mathbb{E}(N_g)})^2$$

8268 where  $\mathbb{E}(N_g)$  is estimated by the mean bin count. An alternative conventional assessment  
 8269 of fit is based on the following statistic: Conditional on  $N$ , the total number of activity  
 8270 centers in the state-space  $\mathcal{S}$ , the bin counts  $N_g$  should have a binomial distribution. It will  
 8271 usually suffice to approximate the binomial cell counts by Poisson cell counts, in which  
 8272 case we can use the classical “index-of-dispersion” test (Illian et al., 2008, p. 87), based  
 8273 on the variance-to-mean ratio:

$$ID = (G - 1) * s^2 / \bar{N}$$

8274 where  $s^2$  is the sample variance of the bin counts and  $\bar{N}$  is the sample mean. When the  
 8275 point process realization is *observed*, as in classical point pattern modeling (but not in  
 8276 SCR), this statistic has approximately a Chi-square distribution on  $(G - 1)$  degrees-of-  
 8277 freedom under the spatial randomness hypothesis. If  $s^2 / \bar{N} > 1$ , clustering is suggested  
 8278 whereas,  $s^2 / \bar{N} < 1$  suggests the point process is too regular.

8279 Whatever statistic we choose as our basis for assessing spatial randomness, *the im-*  
 8280 *portant technical issue is that we don’t observe the point process and so the standard*  
 8281 *statistics for evaluating spatial randomness cannot be computed directly. However, using*  
 8282 *Bayesian analysis, we do have a posterior sample of the underlying point process and*  
 8283 *so we suggest computing the posterior distribution of any statistic in a Bayesian p-value*  
 8284 *framework. For a given posterior draw of all model parameters,  $N$  is known, based on the*  
 8285 *value of the data augmentation variables  $z_i$ , and so we can obtain a posterior sample of*  
 8286  *$N(\mathbf{x})$  by taking all of the output for MCMC iterations  $m = 1, 2, \dots$ , and doing this:*

$$N(\mathbf{x})^{(m)} = \sum_{z_i^{(m)}=1} I(\mathbf{s}_i^{(m)} \in B(\mathbf{x}))$$

8287 Thus,  $N(\mathbf{x})^{(1)}, N(\mathbf{x})^{(2)}, \dots$ , is the Markov chain for the derived parameter  $N(\mathbf{x})$ .

8288 In addition to computing the bin counts for each iteration of the MCMC algorithm,  
 8289 at the same time we generate a realization of the activity centers  $\mathbf{s}_i$  under the spatial  
 8290 randomness model, and we obtain bin counts for these “new” data,  $\tilde{N}(\mathbf{x})$ . For each of  
 8291 the posterior samples – that of the real data, and that of the posterior simulated data, we  
 8292 compute the fit-statistic. The fit statistic based on the actual data is:

$$T(\mathbf{N}, \theta) = \sum_x (\sqrt{N(\mathbf{x})} - \sqrt{\tilde{N}(\mathbf{x})})^2$$

8293 whereas the fit statistic based on a simulated realization of points under the spatial ran-  
 8294 domness hypothesis is:

$$T(\tilde{\mathbf{N}}, \theta) = \sum_x (\sqrt{\tilde{N}(\mathbf{x})} - \sqrt{\tilde{N}(\mathbf{x})})^2$$

---

<sup>2</sup> $I(arg)$  is the indicator function which evaluates to 1 if *arg* is true, otherwise 0

8295 And we compute the Bayesian p-value by tallying up the proportion of times that  $T(\tilde{\mathbf{N}}, \theta)$   
 8296 is larger than  $T(\mathbf{N}, \theta)$ , as an estimate of:  $p = \Pr(T(\tilde{\mathbf{N}}, \theta) > T(\mathbf{N}, \theta))$ . The **R** function  
 8297 **SCRgof** in our package **scrbook** will do this, given the output from **JAGS** (see below).

### 8298 Sensitivity to bin size

8299 Evaluating fit based on bin counts in point process models are sensitive to the number of  
 8300 bins (Illian et al., 2008, p. 87-88). This is related to the classical problem of fit testing  
 8301 for binary regression because in a point process model, as the number of grid cells gets  
 8302 small, the grid cell counts go to 0 or 1 and standard fit statistics (e.g., based on deviance  
 8303 or Pearson residuals) are known not to be very useful. There is some good discussion of  
 8304 this in McCullagh and Nelder (1989, Sec. 4.4.5). What it boils down to is, using the  
 8305 example of the Pearson residual statistic considered by McCullagh and Nelder (1989), the  
 8306 fit statistic is exactly a deterministic function of the sample size only, which clearly should  
 8307 not be regarded as useful for model fit. This is why, in order to do a check of model fit  
 8308 when you have a binary response, one must always aggregate the data in some fashion. In  
 8309 the context of testing spatial randomness, computing the test statistic we described above  
 8310 has us chop up the region  $\mathcal{S}$  into bins, and tally up  $N_g$ , the frequency of activity centers  
 8311 in each bin  $g$ . Suppose that we choose the bin size to be extremely small such that  $\mathbb{E}(N_g)$   
 8312 tends to  $N/G$  ( $N$  being the number of activity centers). Further,  $N_g$  tends to a binary  
 8313 outcome. Therefore the fit statistic has  $N$  components that have value  $N_g = 1$ , and it has  
 8314  $G - N$  components that have value  $N_g = 0$ . Therefore, the fit statistic resembles:

$$T(\mathbf{N}, \theta) = \sum_{g \ni N_g=1}^N (1 - \sqrt{N/G})^2 + \sum_{g \ni N_g=0}^{G-N} (N/G)^2 = N(1 + (G - N)/G)$$

8315 (here  $\ni$  means “such that”). If  $G$  is huge relative to  $N$ , then we see that this tends to  
 8316 about  $2 * N$ , which does not provide any meaningful assessment of model fit. So if you  
 8317 look at this in the limit in which the bin counts become binary, the fit statistic loses all  
 8318 its variability to the specific model used and is just a deterministic function of  $N$ . As a  
 8319 practical matter, it probably makes sense to restrict the number of bins to *fewer* than the  
 8320 number of observed individuals in the sample size. In typical SCR applications this will  
 8321 therefore result, usually, in very large (and few) bins, and presumably not much power.

8322 There are some extensions that help resolve the issue of sensitivity to bin size. We can  
 8323 construct fit statistics based not just on quadrat counts but also the neighboring quadrat  
 8324 counts – this is the Greig-Smith method (Greig-Smith, 1964). In addition, there are a  
 8325 myriad of “distance methods” for evaluating point process models, and we believe that  
 8326 many of these can (and will) be adapted to SCR models. Again the main feature is that  
 8327 the point process on which inference is focused is completely latent in SCR models – so  
 8328 this makes the fit assessment slightly different than in classical point processes. That said,  
 8329 the methods should be adaptable, e.g., in a Bayesian p-value kind of way.

### 8330 Sensitivity to state-space extent

8331 An issue that we have not investigated is that any model assessment that applies to a *latent*  
 8332 point process is probably sensitive to the size of the state-space. As the size of the state-  
 8333 space increases then the cell counts (far away from the data) *are* independent binomial  
 8334 counts with constant density, and so we can overwhelm the fit statistic with extraneous  
 8335 “data” simulated from the posterior, which is equal to the prior as we move away from the

8336 data, and therefore uninformed by the observed data that live in the vicinity of the trap  
 8337 array. Therefore we recommend computing these goodness-of-fit statistics in the vicinity  
 8338 of the trap array only. Perhaps, as an ad hoc rule-of-thumb, less than the average trap  
 8339 spacing from the rectangle enclosing the trap array. For example, if the average trap  
 8340 spacing is, say, 10 km, then the bins used to obtain the observed and predicted activity  
 8341 centers should not extend any further from the traps than 5 km. This should be a matter  
 8342 of future research.

8343 **8.4.2 Assessing fit of the observation model**

8344 In evaluating the spatial randomness hypothesis, we could draw on well-established ideas  
 8345 from point process modeling. On the other hand, it is less clear how to approach goodness-  
 8346 of-fit evaluation of the observation model. For most SCR problems, we have a 3-dimensional  
 8347 data array of *binary* observations,  $y_{ijk}$  for individual  $i$ , trap  $j$  and sample occasion  $k$ . As  
 8348 discussed in the previous section, we need to construct fit statistics based on observed and  
 8349 expected frequencies that are aggregated in some fashion. In practice, the data will be  
 8350 too sparse to have much power, unless the data are highly aggregated. We recommend  
 8351 focusing on summary statistics that represent aggregated versions of  $y_{ijk}$  over 1 or 2 of  
 8352 the dimensions. We describe 3 such fit statistics below. We recognize that, depending on  
 8353 the model, some information about model fit will be lost by summarizing the data in this  
 8354 way. For example if there is a behavioral response and we aggregate over time to focus  
 8355 on the individual and trap level summaries then some information about lack of fit due  
 8356 to temporal structure in the data is lost.

8357 **Fit statistic 1: individual x trap frequencies** We summarize the data by indi-  
 8358 vidual and trap-specific counts  $y_{ijk}$  aggregated over all sample occasions. Using standard  
 8359 “dot notation” to represent summed quantities, we express that as:  $y_{ij\cdot} = \sum_{k=1}^K y_{ijk}$ .  
 8360 Conditional on  $\mathbf{s}_i$ , the expected value under any encounter model is:

$$\mathbb{E}(y_{ij\cdot}) = p_{ij} K$$

8361 (or  $K_j$  if the traps are operational for variable periods). If there is time-varying structure  
 8362 to the model, then expected values would have to be computed according to  $\mathbb{E}(y_{ij\cdot}) =$   
 8363  $\sum_k p_{ijk}$ . Then we can define a fit statistic from the Freeman-Tukey residuals according  
 8364 to:

$$T_1(\mathbf{y}, \theta) = \sum_i \sum_j (\sqrt{y_{ij\cdot}} - \sqrt{\mathbb{E}(y_{ij\cdot})})^2$$

8365 where we use  $\theta$  here to represent the collection of all parameters in the model. This is  
 8366 conditional on  $\mathbf{s}$  as well as on the data augmentation variables  $\mathbf{z}$ . We compute this statistic  
 8367 for *each* iteration of the MCMC algorithm for the observed data set and also for a new  
 8368 data set simulated from the posterior distribution, say  $\tilde{\mathbf{y}}$ .

8369 We could also use a similar fit statistic derived from summarizing over traps to obtain  
 8370 an  $n_{ind} \times K$  matrix of count statistics. We imagine that either summary of the data will  
 8371 probably be too disaggregated (have mostly values of 0) in most practical settings to have  
 8372 much power.

8373 **Fit statistic 2: Individual encounter frequencies.** SCR models represent a  
 8374 type of model for heterogeneous encounter probability, like model  $M_h$ , but with an ex-  
 8375 plicit factor (space) that explains part of the heterogeneity. For model  $M_h$ , the individual

8376 encounter frequencies are the sufficient statistic for model parameters, and so it makes in-  
 8377 tuitive sense to provide some kind of omnibus fit assessment of the core heuristic that SCR  
 8378 model is adequately explaining the heterogeneity using a model  $M_h$ -like statistic based  
 8379 on individual encounter frequencies. So, we build a fit statistic based on the individual  
 8380 total encounters (Russell et al., 2012),  $y_{i..} = \sum_j \sum_k y_{ijk}$ . In addition, the expected value  
 8381 is a similar summary over traps and occasions:  $\mathbb{E}(y_{i..}) = \sum_j \sum_k p_{ijk}$ . Then, we define  
 8382 statistic  $T_2$  according to:

$$T_2(\mathbf{y}, \theta) = \sum_i (\sqrt{y_{i..}} - \sqrt{\mathbb{E}(y_{i..})})^2$$

8383 We imagine this test statistic should provide an omnibus test of extra-binomial variation  
 8384 and should therefore capture some effect of variable exposure to encounter of individuals,  
 8385 although we have not carried out any evaluations of power under specific alternatives.  
 8386 Obviously, in using this statistic, we lose information on departures from the model that  
 8387 might only be trap- or time-specific.

8388 **Fit Statistic 3: Trap frequencies.** We construct an analogous statistic based  
 8389 on aggregating over individuals and replicates to form trap encounter frequencies:  $y_{.j} =$   
 8390  $\sum_i \sum_k y_{ijk}$  (Gopalaswamy et al., 2012b) and the expected value is a similar summary  
 8391 over individuals and occasions:  $\mathbb{E}(y_{.j}) = \sum_i \sum_k p_{ijk}$ . Then statistic  $T_3$  is:

$$T_3(\mathbf{y}, \theta) = \sum_j (\sqrt{y_{.j}} - \sqrt{\mathbb{E}(y_{.j})})^2$$

8392 This seems like a sensible fit statistic because we can think of SCR models as spatial  
 8393 models for counts (Chandler and Royle, 2013). Therefore, we should seek models that  
 8394 provide good predictions of the observable spatial data, which are the trap totals. In this  
 8395 context, it might even make sense to pursue cross-validation based methods for model  
 8396 selection. Cross-validation is a standard method of evaluating models such as in kriging  
 8397 or spline smoothing, so we could as well develop such ideas based on the trap-specific  
 8398 frequencies.

#### 8399 8.4.3 Does the SCR model fit the wolverine data?

8400 We use the ideas described in the previous section to evaluate goodness-of-fit of the SCR  
 8401 model to the wolverine camera trapping data.

8402 We consider first whether the simple model of spatial randomness of the activity  
 8403 centers is adequate. We think that the encounter model shouldn't have a large effect  
 8404 on whether the spatial randomness assumption is adequate or not, so we fit "Model 0"  
 8405 (in which parameters are *not* sex-specific) using an **R** script provided in the function  
 8406 **wolvSCR0gof** which will default to fitting the model in **JAGS**. This is the same script as  
 8407 **wolvSCR0ms** except that it saves the MCMC output for the activity centers **s** and the data  
 8408 augmentation variables **z**, which are required in order to compute the Bayesian p-value  
 8409 test of spatial randomness.

8410 The MCMC output is processed with the **R** function **SCRgof** which computes the test  
 8411 of spatial randomness based on bin counts, using the Bayesian p-value calculation. The  
 8412 function **SCRgof** requires a few things as inputs: (1) the output from a **BUGS** run (in  
 8413 particular, the activity center coordinates and the data augmentation variables); (2) the

8414 number of bins to create for computing spatial frequencies of activity centers; (3) the trap  
 8415 locations and, (4) the buffer around the trap array to use in computing the bin counts.  
 8416 This buffer could be that used in defining the state-space for the model fitting, but we  
 8417 think it should be relatively tighter to the trap array than the state-space used in model-  
 8418 fitting. For the wolverine analysis, where we're using 10-km grid cells (1 unit = 10 km)  
 8419 and a 20 km buffer for model fitting, we'll use a state-space buffer of 0.4 units (4 km) for  
 8420 computing the fit statistic. The **R** code to fit the model and obtain the goodness-of-fit  
 8421 result is as follows:

```
8422 > wolv1 <- wolvSCR0gof(nb=1000,ni=6000,buffer=2,M=200,model=0)
8423
8424 > bugsout <- wolv1$BUGSoutput$sims.list
8425
8426 > traplocs <- wolverine$wtraps[,2:3]
8427 > traplocs[,1] <- traplocs[,1] - min(traplocs[,1])
8428 > traplocs[,2] <- traplocs[,2] - min(traplocs[,2])
8429 > traplocs <- traplocs/10000
8430
8431 > set.seed(2013) # set seed so Bayesian p-value is the same each time
8432
8433 > SCRgof(bugsout,5,5,traplocs=traplocs,buffer=.4)
8434
8435 Cluster index observed: 1.099822
8436 Cluster index simulated: 1.000453
8437 P-value index of dispersion: 0.408
8438 P-value2 freeman-tukey: 0.6842667
```

8439 The output produced by **SCRgof** is the index of dispersion based on the ratio of the variance to the mean (see above), which is computed as the posterior mean index of dispersion for the latent point process, and also the average value for simulated data. If this value is  $> 1$  then clustering is suggested, which we see a (very) minor amount of evidence for here. Two Bayesian p-values are produced: the first is based on the cluster index, and the 2nd is based on the Freeman-Tukey statistic calculated as described in Sec. 8.4.1. Because our p-values aren't close to 0 or 1, we judge that the model of spatial randomness provides an adequate fit to the data. You can verify that a similar result is obtained if we use the model with fully sex-specific parameters (Model 4).

8440 Next, we did a Bayesian p-value analysis of the observation component of the model,  
 8441 using the 3 fit statistics described in Sec. 8.4.2. These statistics can be calculated as  
 8442 part of the **BUGS** model specification or by post-processing the MCMC output returned  
 8443 from a **BUGS** run. The **R** script **wolvSCR0gof** contains the relevant calculations. For  
 8444 example, to compute fit statistic 1, we have to add some commands to the **BUGS** model  
 8445 specification such as this (note: this is only a fraction of the model specification):

```
8454 .....
8455 for(j in 1:ntraps){
8456   mu[i,j] <- w[i]*p[i,j]
8457
8458   y[i,j] ~ dbin(mu[i,j],K[j])
```

---

```

8459   ynew[i,j] ~ dbin(mu[i,j],K[j])
8460
8461   err[i,j] <- pow(pow(y[i,j],.5) - pow(K[j]*mu[i,j],.5),2)
8462   errnew[i,j] <- pow(pow(ynew[i,j],.5) - pow(K[j]*mu[i,j],.5),2)
8463 }
8464
8465 Tlobs <- sum(err[,])
8466 Tnew <- sum(errnew[,])
8467 .....
8468 Similar calculations are carried out to obtain the posterior samples of test statistics 2
8469 (individual totals) and 3 (trap totals). For the wolverine data, the Bayesian p-value
8470 calculations produce:
```

```

8471 > mean(wolv1$BUGSoutput$sims.list$T1new>wolv1$BUGSoutput$sims.list$T1obs)
8472 [1] 0
8473
8474 > mean(wolv1$BUGSoutput$sims.list$T2new>wolv1$BUGSoutput$sims.list$T2obs)
8475 [1] 0.17
8476
8477 > mean(wolv1$BUGSoutput$sims.list$T3new>wolv1$BUGSoutput$sims.list$T3obs)
8478 [1] 0.02066667
```

8479     Based on statistic  $T_2$ , we might conclude that the model is adequate for explaining  
8480 individual heterogeneity although the other two statistics suggest a general lack of fit of  
8481 the observation model. A similar result is obtained using the fully sex-specific model. We  
8482 note that one individual was captured 8 times in one trap, which is pretty extreme under  
8483 a model which assumes independent Bernoulli trials. We summarize that the trap-counts  
8484 simply are not well-explained by this model.

8485     In attempt to resolve this problem, we extended the model to include a local (trap-  
8486 specific) behavioral response (following Royle et al. (2011b)) which can be fitted using  
8487 the sample **R** script **wolvSCRMb**. To fit a model using **WinBUGS**, and then compute the  
8488 Bayesian p-values we do this:

```

8489 > wolv.Mb <- wolvSCRMb(nb=1000,ni=6000,buffer=2,M=200)
8490
8491 > mean(wolv.Mb$sims.list$T1new>wolv.Mb$sims.list$T1obs)
8492 [1] 0.9666667
8493
8494 > mean(wolv.Mb$sims.list$T2new>wolv.Mb$sims.list$T2obs)
8495 [1] 0.3644667
8496
8497 > mean(wolv.Mb$sims.list$T3new>wolv.Mb$sims.list$T3obs)
8498 [1] 0.4990667
```

8499     Given that this model seems to fit better, we might prefer reporting estimates under  
8500 this model, which we do in Table 8.4. (the behavioral response parameter is labeled  $\alpha_2$   
8501 in the table). Estimated density is about 1 individual higher per 1000 km<sup>2</sup> compared

with the various models that lack a behavioral response. It might be useful to try these fit assessment exercises using the habitat mask as described in Sec. 5.10. That takes an extremely long time to run in **BUGS** though, especially for the behavioral response model.

**Table 8.4.** Posterior summary statistics for local (trap-specific) behavioral response model  $M_b$  fitted to the wolverine camera trapping data using **WinBUGS**. The parameter  $\alpha_2$  is the local (trap-specific) behavioral response parameter.  $T_x()$  are the posterior summaries of fit statistics  $x = 1, 2, 3$  used in the Bayesian p-value analysis (See text for definitions). Results are based on 3 chains, each with 6000 iterations (first 1000 discarded) for a total of 15000 posterior samples.

Parameter	Mean	SD	2.5%	50%	97.5%	Rhat	n.eff
$N$	71.32	19.07	42.00	69.00	114.02	1.00	2100
$D$	6.87	1.84	4.05	6.65	10.99	1.00	2100
$\sigma$	0.88	0.13	0.68	0.86	1.17	1.00	730
$p_0$	0.01	0.00	0.01	0.01	0.02	1.01	530
$\alpha_1$	0.69	0.19	0.37	0.67	1.10	1.00	730
$\alpha_2$	2.50	0.27	1.99	2.50	3.04	1.00	700
$\psi$	0.36	0.10	0.20	0.35	0.58	1.00	2600
$T_1^{obs}$	54.71	6.12	43.69	54.39	67.47	1.00	3900
$T_1^{new}$	64.73	7.62	50.93	64.39	80.96	1.00	3900
$T_2^{obs}$	13.93	4.07	7.25	13.53	23.04	1.00	5700
$T_2^{new}$	12.65	3.35	6.93	12.36	20.07	1.00	2000
$T_3^{obs}$	12.80	1.74	9.80	12.64	16.61	1.00	2400
$T_3^{new}$	12.94	3.05	7.77	12.67	19.58	1.00	15000

## 8.5 QUANTIFYING LACK-OF-FIT AND REMEDIATION

Molinari-Jobin et al. (2013) used a strategy for assessing model fit in dynamic occupancy models (Royle and Kéry, 2007) similar to that which we suggested above. They constructed a fit statistic based on aggregating the data over replicate samples ( $k$ ), to obtain the total detections per site  $i$  and year  $j$ . They used a Bayesian p-value analysis based on a Chi-squared test statistic (also see Kéry and Schaub, 2012, Chapt. 12). Their analysis suggested a model that didn't fit, and, so they computed the "lack-of-fit ratio" (see Kéry and Schaub, 2012, Sec. 12.3) – the ratio of the fit statistic computed for the actual data to that of the replicate data sets. They interpret this analogous to the over-dispersion coefficient in generalized linear models (McCullagh and Nelder, 1989), usually called the c-hat statistic in capture-recapture literature (see Cooch and White, 2006, Chapt. 5). Molinari-Jobin et al. (2013) reported the lack-of-fit ratio for their model to be 1.14 which suggests a minor lack-of-fit, compared to perfect data having a value of 1, because the posterior standard deviations will be too small by a factor of  $\sqrt{1.14} = 1.07$ . In classical capture-recapture applications of goodness-of-fit assessment, inference for non-fitting models is dealt with by inflating the resulting SEs (of the non-fitting model), by the square-root of c-hat. We believe that these ideas related to quantifying lack-of-fit and understanding its effect could also be applied to SCR models, although we have not yet explored this.

## 8.6 SUMMARY AND OUTLOOK

8523 In this chapter, we offered some general strategies for model selection and model checking,  
8524 or assessment of model fit. We think the strategies we outlined for model selection are fairly  
8525 standard and can be effectively applied to many SCR modeling problems. Some technical  
8526 issues of Bayesian analysis need to be addressed (in general) before Bayesian methods  
8527 are more generally useful and accessible. For one thing, Bayesian model selection based  
8528 on the indicator variable approach of Kuo and Mallick (1998) can be tediously slow even  
8529 for small data sets, and so improved computation will improve our ability to do Bayesian  
8530 model selection in practical situations. Also, and most importantly, sensitivity to prior  
8531 distributions is an important issue. Further research and practice might identify preferred  
8532 prior configurations for SCR that provide a good calibration in relevant model selection  
8533 problems. Finally, we believe that cross-validation should prove to be a useful method  
8534 in model assessment and selection, as SCR models are a form of spatial model of counts,  
8535 and so it is natural to pick models that predict the observable spatial counts (i.e., at trap  
8536 locations) well.

8537 For Bayesian model assessment, or goodness-of-fit checking, we suggested a framework  
8538 based on independent testing of the spatial model of independence and uniformity, and  
8539 testing fit of the observation model conditional on the underlying point process. These  
8540 ideas are based on mostly *ad hoc* attempts in a number of published applications (Royle  
8541 et al., 2009a, 2011a; Gopalaswamy et al., 2012b; Russell et al., 2012, e.g.). While we think  
8542 this general strategy should be fruitful, we know of no studies on the power to detect  
8543 various model departures, and so the ideas should be viewed as experimental. We have  
8544 not discussed assessment of model fit for SCR models using likelihood methods, although  
8545 we imagine that standard bootstrapping ideas should be effective, perhaps based on the  
8546 fit statistics (or similar ones) we suggested here for computing Bayesian p-values.

8547 Clearly there is much research to be done on assessment of model fit in SCR models.  
8548 For testing the spatial randomness hypothesis, we used a classical approach based on  
8549 count frequencies, in which point locations are put into spatial bins. Other approaches  
8550 from spatial point process modeling should be pursued including nearest-neighbor methods  
8551 or distance-based methods. In addition, studies to evaluate the power to detect relevant  
8552 departures from the standard assumptions, and the robustness of inferences about  $N$  or  
8553 density, need to be conducted. If the spatial randomness model appears inadequate, it  
8554 is possible to fit models that allow for a non-uniform distribution of points (see Chapt.  
8555 11) and even point process models that allow for interactions among points (Reich et al.,  
8556 2012). On the other hand, we expect that most of these Bayesian p-value tests will have  
8557 low power in typical data sets consisting of a few to a few dozen individuals. As such,  
8558 failure to detect a lack of fit may not be that meaningful. But, on the other hand, it  
8559 may not make a difference in terms of density estimates either. We think inference about  
8560 density should be relatively insensitive to departures from spatial randomness, because  
8561 we get to observe direct information on some component of the population, component  
8562 of density is *observed*. For those activity centers, the assumed model of the point process  
8563 should exert little influence on the placement of the activity centers. Conversely, as is  
8564 the case with classical closed population models (Otis et al., 1978; Dorazio and Royle,  
8565 2003; Link, 2003), inferences may be somewhat more sensitive to bad-fitting models for  
8566 the observation process.



8567

8568

8569

# 9

---

## ALTERNATIVE OBSERVATION MODELS

8570 In previous chapters we considered various models of *encounter probability*, both in terms  
8571 of parametric functions of distance and also a myriad of covariate models (Chapt. 7 and  
8572 elsewhere). However, we have so far only considered a specific probability model for the  
8573 observations (we'll call this the "observation model") – the Bernoulli encounter process  
8574 model which, in **secr**, is the *proximity detector* model. This assumes that individual and  
8575 trap-specific encounters are independent Bernoulli trials.

8576 In this chapter, we focus on developing additional observation models. The observa-  
8577 tion model could be thought of as being determined by the type of device – or the type of  
8578 "detector" using the terminology of **secr** (Efford, 2011a). We consider models that apply  
8579 when observations are not binary and, in some cases, that do not require independence of  
8580 the observations. We present models when the data are encounter *frequencies*, based on the  
8581 Poisson distribution, and observation models based on the multinomial distribution. For  
8582 example, if sampling devices can detect an individual some arbitrary number of times dur-  
8583 ing an interval, then it is natural to consider observation models for encounter frequencies,  
8584 such as the Poisson model. Another type of encounter device is the "multi-catch" device  
8585 (Efford et al., 2009a) which is a physical device that can capture and hold an arbitrary  
8586 number of individuals. A typical example is a mist-net for birds (Borchers and Efford,  
8587 2008). It is natural to regard observations from these kinds of studies as independent  
8588 multinomial observations. A related type of device that produces *dependent* multinomial  
8589 observations are the so-called *single-catch* traps (Efford, 2004; Efford et al., 2009a). The  
8590 canonical example are small-mammal live traps which catch and hold a single individual.  
8591 Competition among individuals for traps induces a complex dependence structure among  
8592 individual encounters. To date, no formal inference framework has been devised for this  
8593 method although it stands to reason that the independent multinomial model should be  
8594 a good approximation in some situations (Efford et al., 2009a). We analyze a number of  
8595 examples of these different observation models using **JAGS** and also the **R** package **secr**  
8596 (Efford, 2011a).

## 9.1 POISSON OBSERVATION MODEL

The models we analyze in Chapt. 5 assumed binary observations – i.e., standard encounter history data – so that individuals are captured at most one time in a trap on any given sample occasion. This makes sense for many types of DNA sampling (e.g., based on hair snares) because distinct visits to sampled locations or devices cannot be differentiated. However, for some encounter devices, or methods, the potential number of encounters is *not* fixed, and so it is possible to encounter an individual some arbitrary number of times during any particular sampling episode. That is, we might observe encounter frequencies  $y_{ijk} > 1$  for individual  $i$ , trap  $j$  and sampling interval  $k$ . As an example, if a camera device is functioning properly it may be programmed to take photos every few seconds if triggered. For a second example, suppose we are searching a quadrat or length of trail for scat, we may find multiple samples from the same individual. Therefore, we seek observation models that accommodate such encounter frequency data. In general, any discrete probability mass function could be used for this purpose, including the standard models for count data used throughout ecology, the Poisson and negative binomial. Here we focus on using the Poisson model only although other count frequency models are possible for SCR models (Efford et al., 2009b).

Let  $y_{ijk}$  be the frequency of encounter for individual  $i$ , in trap  $j$ , during occasion  $k$ , then assume:

$$y_{ijk} \sim \text{Poisson}(\lambda_{ij})$$

where the expected encounter frequency  $\lambda_{ij}$  depends on both individual and trap. As we did in the binary model of Chapt. 5, we now seek to model the expected value of the observation (which was  $p_{ij}$  in Chapt 5) as a function of the individual activity center  $\mathbf{s}_i$ . We propose

$$\lambda_{ij} = \lambda_0 k(\mathbf{x}_j, \mathbf{s}_i)$$

Where  $k(\mathbf{x}, \mathbf{s})$  is any positive valued function, such as the negative exponential or the bivariate Gaussian kernel, and  $\lambda_0$  is the baseline encounter rate – the expected number of encounters if a trap is placed precisely at an individuals home range center (note: in `secr` the notation for this is  $g_0$ ). Then,  $\lambda_0 k(\mathbf{x}_j, \mathbf{s}_i)$  is the expected encounter rate in trap  $\mathbf{x}_j$  for an individual having activity center  $\mathbf{s}_i$ . Note that

$$\log(\lambda_{ij}) = \log(\lambda_0) + \log(k(\mathbf{x}_j, \mathbf{s}_i)).$$

Equating  $\alpha_0 \equiv \log(\lambda_0)$ , and, if  $k(\mathbf{x}, \mathbf{s}) \equiv \exp(-d(\mathbf{x}, \mathbf{s})^2/(2\sigma^2))$  (i.e., the Gaussian model), then:

$$\log(\lambda_{ij}) = \alpha_0 - \alpha_1 d(\mathbf{x}_j, \mathbf{s}_i)^2 \quad (9.1.1)$$

where  $\alpha_1 = 1/(2\sigma^2)$ , which is the same linear predictor as we have seen for the Bernoulli model in Chapt. 5. This Poisson SCR model is therefore a type of Poisson generalized linear mixed model (GLMM).

We can accommodate covariates at the level of individual-, trap- or sample occasion by including them on the baseline encounter rate parameter  $\lambda_0$ . For example, if  $C_j$  is some covariate that depends on trap only, then we express the relationship between  $\lambda_0$  and  $C_j$  as:

$$\log(\lambda_{0,ijk}) = \alpha_0 + \alpha_2 C_j$$

and therefore covariates on the logarithm of baseline encounter probability appear also as linear effects on  $\lambda_{ij}$ . In general, covariates might also affect the coefficient on the distance

term ( $\alpha_1$ ) (e.g., sex of individual). We don't get into too much discussion of general covariate models here, but we covered them in some detail in both Chaps. 7 and 8.

For models in which we do not have covariates that vary across the sample occasions  $k$ , we can aggregate the observed data by the property of compound additivity of the Poisson distribution (if  $x$  and  $y$  are *iid* Poisson with mean  $\lambda$  then  $x + y$  is Poisson with mean  $2\lambda$ ). Therefore,

$$y_{ij} = \left( \sum_{k=1}^K y_{ijk} \right) = \text{Poisson}(K\lambda_0 k(\mathbf{x}_j, \mathbf{s}_i))$$

We see that  $K$  and  $\lambda_0$  serve the same role as affecting the base encounter rate. Since the observation model is the same, probabilistically speaking, for all values of  $K$ , evidently we need only  $K = 1$  "survey" from which to estimate model parameters (Efford et al., 2009b). We know this intuitively, as sampling by multiple traps serves as replication in SCR models. This has great practical relevance to the conduct of capture-recapture studies and the use of SCR models. For example, if individuality is obtained by genetic information from scat sampling, one should only have to carry out a single spatial sampling of the study area. However, one must be certain that sufficient spatial recaptures will be obtained so that effective estimation is possible.

### 9.1.1 Poisson model of space usage

It is natural to interpret the Poisson encounter model as a model of space usage resulting from movement of individuals about their home range (Sec. 5.4). Imagine we have perfect samplers in every pixel of the landscape so that whenever an individual moves from one pixel to another, we can record it. Let  $m_{ij}$  be the number of times individual  $i$  was recorded in pixel  $j$  (i.e., it selected or used pixel  $j$ ). Then, we might think of the Poisson model for the observed *use* frequencies:

$$m_{ij} \sim \text{Poisson}(\lambda_0 k(\mathbf{x}_j, \mathbf{s}_i))$$

where  $\lambda_0$  is related to the baseline movement rate of the animal (how often it moves). This model of space usage gives rise to the standard resource selection function (RSF) models (see Chapt. 13). But now suppose our samplers are not perfect but, rather, record only a fraction of the resulting visits. A sensible model is

$$y_{ij}|m_{ij} \sim \text{Binomial}(m_{ij}, p).$$

The marginal distribution of  $y_{ij}$  is:

$$y_{ij} \sim \text{Poisson}(p_0 k(\mathbf{x}_j, \mathbf{s}_i)).$$

where  $p_0$  is a composite of the movement rate and conditional detection probability  $p$ . Therefore, we see that encounters accumulate in proportion to the frequency of outcomes of an individual using space (or "selecting resources").

We introduced an interpretation of SCR models in terms of movement and space usage in Sec. 5.4, and it is one of the main underlying concepts of SCR models that is not present in ordinary capture-recapture models. As we noted there, the underlying model of space usage is only as complex as the encounter probability model which has been, so far in this book, only symmetric and stationary (does not vary in space). We generalize this model of space usage substantially in Chapt. 13.

---

**9.1.2 Poisson relationship to the Bernoulli model**

8671 There is a sense in which the Poisson and Bernoulli models can be viewed as consistent with  
 8672 one another. Note that under the Poisson model, the relationship between the expected  
 8673 count and the probability of counting “at least 1”, is given by

$$\Pr(y > 0) = 1 - \exp(-\lambda) \quad (9.1.2)$$

8675 where  $\mathbb{E}(y) = \lambda$ . Therefore, if we equate the event “encountered” with the event that the  
 8676 individual was captured at least 1 time under the Poisson model, i.e.,  $y > 0$ , then it would  
 8677 be natural to set  $p_{ij} = \Pr(y > 0)$  according to Eq. 9.1.2. That is, we can use Eq. 9.1.2  
 8678 as the model for encounter probability for binary observations. This is the “hazard rate”  
 8679 model in distance sampling.

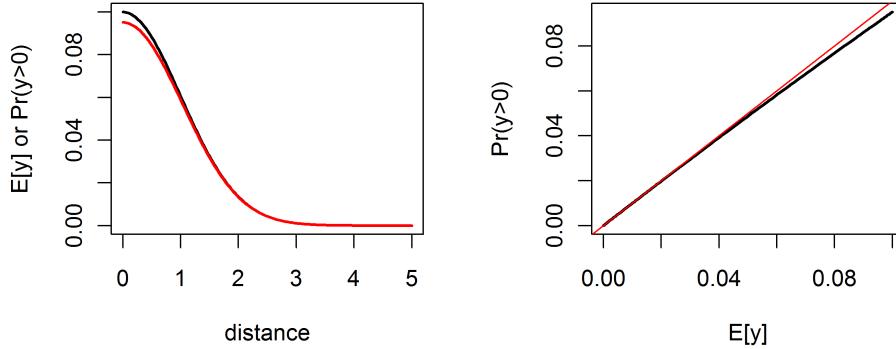
8680 In fact, as  $\lambda$  gets small, the Poisson model is a close approximation to the Bernoulli  
 8681 model in the sense that outcomes concentrate on  $\{0, 1\}$ , i.e.,  $\Pr(y \in \{0, 1\}) \rightarrow 1$  as  $\lambda \rightarrow 0$ .  
 8682 Indeed, under the Poisson model,  $\Pr(y > 0) \rightarrow \lambda$  for small values of  $\lambda$ . This phenomenon  
 8683 is shown in Fig. 9.1 where the left panel shows a plot of  $\lambda_{ij} = \lambda_0 k(\mathbf{x}_j, \mathbf{s}_i)$  vs. distance and  
 8684 superimposed on that is a plot of  $p_{ij} = 1 - \exp(-\lambda_{ij})$  vs. distance, for values  $\lambda_0 = 0.1$   
 8685 and  $\sigma = 1$ , and the right panel shows a plot of  $\Pr(y > 0)$  vs.  $\mathbb{E}(y)$ . We see that the two  
 8686 quantities are practically indistinguishable. This is convenient in some cases because the  
 8687 Poisson model might be more tractable to fit (or even vice versa). For an example, see the  
 8688 models described in Chapt. 18, and we also consider another case in Sec. 20.2.8 below.  
 8689 To evaluate the closeness of the approximation, you can use the following R commands  
 8690 which we used to produce Fig. 9.1:

```
8691 > x <- seq(0.001, 5, , 200)
8692 > lam0 <- .1
8693 > sigma <- 1
8694 > lam <- lam0*exp(-x**/(2*sigma*sigma))
8695
8696 > par(mfrow=c(1,2))
8697 > p1 <- 1-exp(-lam)
8698 > plot(x, lam, ylab="E[y] or Pr(y>0)", xlab="distance", type="l", lwd=2)
8699 > lines(x,p1,lwd=2,col="red")
8700 > plot(lam, p1, xlab="E[y]", ylab="Pr(y>0)", type="l", lwd=2)
8701 > abline(0,1,col="red")
```

8702 To summarize, if  $y$  is Poisson then, as  $\lambda$  gets small,

$$\begin{aligned} \Pr(y > 0) &\approx \mathbb{E}(y) \\ 1 - \exp(-\lambda_0 k(\mathbf{x}, \mathbf{s})) &\approx \lambda_0 k(\mathbf{x}, \mathbf{s}) \end{aligned} \quad (9.1.3)$$

8703 What all of this suggests it that if we have very few observations  $> 1$  in our SCR data  
 8704 set, then we won’t lose much information by using the Bernoulli model. On the other  
 8705 hand, the Poisson model may have some advantages in terms of analytic or numerical  
 8706 tractability in some cases. Further, this approximation explains the close correspondence  
 8707 we have found between these two versions of the Gaussian encounter probability model  
 8708 (Sec. 5.4). Namely, the Gaussian hazard model and the Gaussian encounter probability  
 8709 model are close approximations because  $1 - \exp(-\lambda) \approx \lambda$  if  $\lambda$  is small.



**Figure 9.1.** Poisson approximation to the binomial. As the Poisson mean approaches 0, then  $\Pr(y > 0)$  under the Poisson model approaches  $\lambda$  and therefore  $y \sim \text{Poisson}(\lambda)$  is well-approximated by a Bernoulli model with parameter  $\lambda$ .

Even in such cases where the Poisson and Bernoulli models are not quite equivalent, we might choose to truncate individual encounter frequencies to binary observations anyhow (transforming counts to 0/1 is called “quantizing”). We might do this intentionally in some cases, such as when the distinct encounter events are highly dependent as often happens in camera trap studies when the same individual moves back-and-forth in front of a camera during a short period of time. But sometimes, truncation is a feature of the sampling. For example, in the case of bear hair snares, the number of encounters might be well approximated by a Poisson distribution but we cannot determine unique visits and so only get to observe the binary event “ $y > 0$ ”. In this case, we might choose to model the encounter probability for the binary encounter using Eq. 9.1.4. This is equivalent to the complementary log-log link model, or the “Gaussian hazard” as we called it in Chapt. 5:

$$\text{cloglog}(p_{ij}) = \log(\lambda_0) + \log(k(\mathbf{x}, \mathbf{s}))$$

where  $\text{cloglog}(u) = \log(-\log(1 - u))$ .

### 9.1.3 A cautionary note on modeling encounter frequencies

Other models for counts might be appropriate. For example, ecologists are especially fond of negative binomial models for count data (Ver Hoef and Boveng, 2007; White and Bennetts, 1996; Kéry et al., 2005) but other models for excess-Poisson variation are possible. For example, we might add a normally distributed random effect to the linear predictor (Coul and Agresti, 1999).

As a general rule we favor the Bernoulli observation model even if our sampling scheme

8730 produces encounter frequencies. The main reason is that, with frequency data, we are  
 8731 forced to confront a model choice problem (i.e., Poisson, negative binomial, log-normal  
 8732 mixture) that is wholly unrelated to the fundamental space usage process that underlies  
 8733 the genesis of many types of SCR data. Repeated encounters over short time intervals are  
 8734 not likely to be the result of independent encounter events. E.g., an individual moving back  
 8735 and forth in front of a camera yields a cluster of observations that is not informative about  
 8736 the underlying spatial structure of the population. Similarly in scat surveys dogs are used  
 8737 to locate scats which are processed in the lab for individuality (Kohn et al., 1999; MacKay  
 8738 et al., 2008; Thompson et al., 2012). The process of local scat deposition is not strictly  
 8739 the outcome of movement or space usage but rather the outcome of complex behavioral  
 8740 considerations as well as dependence in detection of scat by dogs. For example, dogs find  
 8741 (or smell) one scat and then are more likely to find one or more nearby ones, if present, or  
 8742 they get into a den or latrine area and find many scats. The additional assumption required  
 8743 to model variation in observed frequencies (i.e., conditional on location) provides relatively  
 8744 no information about space usage and density, and we feel that the model selection issue  
 8745 should therefore be avoided.

8746 To elaborate on this, we suppose that an individual with activity center  $\mathbf{s}$  visits  
 8747 a particular pixel  $\mathbf{x}$  with some probability  $p(\mathbf{x}, \mathbf{s})$ , and then, once there, deposits a  
 8748 number of scat, or visits a camera some number of times with frequency  $y(\mathbf{x}, \mathbf{s}) \geq 0$ .  
 8749 We describe the outcome of this movement/usage process with a two-level hierarchical  
 8750 model of the form:  $[y|w][w|p(\mathbf{x}, \mathbf{s})]$  where  $w(\mathbf{x}, \mathbf{s})$  is a binary variable that indicates  
 8751 whether the individual with activity center  $\mathbf{s}$  used pixel  $\mathbf{x}$  during some interval, and let  
 8752  $w(\mathbf{x}, \mathbf{s}) \sim \text{Bernoulli}(p(\mathbf{x}, \mathbf{s}))$ . If we suppose encounter frequency  $y$  is independent of  $\mathbf{x}$  and  
 8753  $\mathbf{s}$  conditional on the use variable  $w$ , then we see that the model for  $y$  (amount of use) does  
 8754 not depend on  $\mathbf{s}$ .

#### 8755 9.1.4 Analysis of the Poisson SCR model in BUGS

8756 We consider the simplest possible model here in which we have no covariates that vary  
 8757 over sample occasions  $k = 1, 2, \dots, K$  so that we work with the aggregated individual-  
 8758 and trap-specific encounters:

$$y_{ij} = (\sum_{k=1}^K y_{ijk}) = \text{Poisson}(K\lambda_0 k(\mathbf{x}_j, \mathbf{s}_i))$$

8759 and we consider the bivariate normal form of  $k(\mathbf{x}, \mathbf{s})$ :

$$k(\mathbf{x}, \mathbf{s}) = \exp(-d(\mathbf{x}, \mathbf{s})^2 / (2\sigma^2))$$

8760 so that

$$\log(\lambda_{ij}) = \alpha_0 - \alpha_1 d(\mathbf{x}_j, \mathbf{s}_i)^2$$

8761 where  $\alpha_0 = \log(\lambda_0)$  and  $\alpha_1 = 1/(2\sigma^2)$ .

8762 As usual, we approach Bayesian analysis of these models using data augmentation  
 8763 (Sec. 4.2). Under data augmentation, we introduce a collection of all-zero encounter  
 8764 histories to bring the total size of the data set up to  $M$ , and a corresponding set of data  
 8765 augmentation variables  $z_i \sim \text{Bern}(\psi)$ . Then the observation model is specified conditional  
 8766 on  $z$  according to:

$$y_{ij} \sim \text{Poisson}(z_i K \lambda_{ij})$$

which evaluates to a point mass at  $y = 0$  if  $z = 0$ . In other words, the observation model under data augmentation is a zero-inflated Poisson model which is easily analyzed by Bayesian methods, e.g., in one of the **BUGS** dialects or, alternatively, using likelihood methods, which we neglect here although the same principles as in Chapt. 6 apply.

### 9.1.5 Simulating data and fitting the model

Simulating a sample SCR data set under the Poisson model requires only a couple minor modifications to the procedure we used in Chapt. 5 (see the function `simSCR0()`). In particular, we modify the block of code which defines the model to be that of  $E(y)$  and not  $\Pr(y = 1)$ , and we change the random variable generator from `rbinom` to `rpois`:

```
8776 ##  
8777 ## S =activity centers and traplocs defined as in simSCR0()  
8778 ##  
8779 ## Compute distance between activity centers and traps:  
8780 > D <- e2dist(S,traplocs)  
8781  
8782 ## Define parameter values:  
8783 > alpha0 <- -2.5  
8784 > sigma <- 0.5  
8785 > alpha1 <- 1/(2*sigma*sigma)  
8786  
8787 ## Encounter probability model:  
8788 > muy <- exp(alpha0)*exp(-alpha1*D*D)  
8789  
8790 ## Now generate the encounters of every individual in every trap  
8791 > Y <-matrix(NA,nrow=N,ncol=ntraps)  
8792 > for(i in 1:nrow(Y)){  
8793   Y[i,] <- rpois(ntraps,K*muy[i,])  
8794 }
```

We modified our simulation code from Chapt. 5 to simulate Poisson encounter frequencies for each trap and then we analyze an ideal data set using **BUGS**. This Poisson simulator function `simPoissonSCR` is available in the `scrbook` package (it can produce 3-d encounter history data too, although we don't do that here). Here is an example of simulating a data set and harvesting the required data objects, and doing the data augmentation:

```
8801 ## Simulate data and extract data elemements  
8802 ##  
8803 > data <- simPoissonSCR(discard0=TRUE,rnd=2013)  
8804 > y <- data$Y  
8805 > nind <- nrow(y)  
8806 > X <- data$traplocs  
8807 > K <- data$K  
8808 > J <- nrow(X)
```

```

8809 > xlim <- data$xlim
8810 > ylim <- data$ylim
8811
8812 ## Data augmentation
8813 > M <- 200
8814 > y <- rbind(y,matrix(0,nrow=M-nind,ncol=ncol(y)))
8815 > z <- c(rep(1,nind),rep(0,M-nind))

```

8816     The process for fitting the model in **WinBUGS** or **JAGS** is identical to what we've  
 8817     done previously in Chapt. 5. In particular, we set up some starting values, package  
 8818     the data and inits, identify the parameters to be monitored, and then send everything  
 8819     off to our MCMC engine. Here it all is for fitting the Poisson observation model (these  
 8820     commands are shown in the help file for `simPoissonSCR`):

```

8821 ## Starting values for activity centers
8822 ##
8823 > sst <- X[sample(1:J,M,replace=TRUE),]
8824 > for(i in 1:nind){
8825   if(sum(y[i,])==0) next
8826   sst[i,1] <- mean( X[y[i,>0,1] ) )
8827   sst[i,2] <- mean( X[y[i,>0,2] ) )
8828 }
8829 ## Dithered a little bit from trap locations
8830 > sst <- sst + runif(nrow(sst)*2,0,1)/8
8831 > data <- list (y=y,X=X,K=K,M=M,J=J,xlim=xlim,ylim=ylim)
8832 > inits <- function(){
8833   list (alpha0=rnorm(1,-2,.4),alpha1=runif(1,1,2),s=sst,z=z,psi=.5)
8834 }
8835 > parameters <- c("alpha0","alpha1","N","D")

```

8836     Next, we write the **BUGS** model to an external file:

```

8837 > cat("
8838 model{
8839   alpha0 ~ dnorm(0,.1)
8840   alpha1 ~ dnorm(0,.1)
8841   psi ~ dunif(0,1)
8842
8843   for(i in 1:M){
8844     z[i] ~ dbern(psi)
8845     s[i,1] ~ dunif(xlim[1],xlim[2])
8846     s[i,2] ~ dunif(ylim[1],ylim[2])
8847     for(j in 1:J){
8848       d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
8849       y[i,j] ~ dpois(lam[i,j])
8850       lam[i,j] <- z[i]*K*exp(alpha0)*exp(- alpha1*d[i,j]*d[i,j])
8851     }
8852   }

```

---

```

8853 N <- sum(z[])
8854 D <- N/64
8855 }
8856 ",file = "SCR-Poisson.txt")

```

8857 To fit the model we execute **bugs** in the usual way:

```

8858 > library(R2WinBUGS)
8859 > out1 <- bugs (data, inits, parameters, "SCR-Poisson.txt", n.thin=1,
8860           n.chains=3,n.burnin=1000,n.iter=2000,working.dir=getwd(),
8861           debug=TRUE)

```

8862 Or, using **JAGS** via **rjags** we would do something like this:

```

8863 > library(rjags)
8864 > jm <- jags.model("SCR-Poisson.txt", data=data, inits=inits,
8865           n.chains=3, n.adapt=1000)
8866 > out2 <- coda.samples(jm, parameters, n.iter=1000, thin=1)

```

8867 Summarizing the output from the **WinBUGS** run produces the following:

```

8868 > print(out1,digits=2)
8869 Inference for Bugs model at "SCR-Poisson.txt", fit using WinBUGS,
8870 3 chains, each with 2000 iterations (first 1000 discarded)
8871 n.sims = 3000 iterations saved
8872      mean   sd  2.5%   25%   50%   75% 97.5% Rhat n.eff
8873 alpha0  -2.57 0.19 -2.95 -2.69 -2.57 -2.44 -2.19 1.00 2600
8874 alpha1   2.34 0.36  1.69  2.08  2.32  2.57  3.12 1.00 3000
8875 N       114.13 15.25 87.97 103.00 113.00 124.00 147.00 1.01 370
8876 D       1.78 0.24  1.37  1.61  1.77  1.94  2.30 1.01 370
8877 deviance 329.95 21.92 290.00 314.20 329.50 344.40 375.80 1.00 1700
8878 ...
8879 [..some output deleted..]
8880 ...

```

### 8881 9.1.6 Analysis of the wolverine study data

8882 We reanalyzed the data from the wolverine camera trapping study that were first introduced in Sec. 5.9. We modified the **R** script from the function **wolvSCR0** to fit the Poisson model (see the help file for **wolvSCR0pois**). Executing this function produces the results shown in Table 9.1. The results are almost indistinguishable from the Bernoulli model fitted previously, where we had a posterior mean for  $N$  of 59.84 and  $\sigma$  was 0.64. You can edit the script **wolvSCR0pois** to obtain more posterior samples, or modify the model in some way.

**Table 9.1.** Results of fitting the SCR model with Poisson encounter frequencies to the wolverine camera trapping data. Posterior summaries were obtained using **WinBUGS** with 3 chains, each with 6000 iterations, discarding the first 1000 as burn-in, to yield a total of 15000 posterior samples.

Parameter	Mean	SD	2.5%	50%	97.5%	Rhat	n.eff
$N$	60.12	11.91	40.00	59.00	87.00	1	630
$D$	5.80	1.15	3.86	5.69	8.39	1	630
$\log(p_0)$	-2.89	0.17	-3.22	-2.89	-2.57	1	5000
$\lambda_0$	0.06	0.01	0.04	0.06	0.08	1	5000
$\sigma$	0.64	0.06	0.54	0.64	0.76	1	730
$\psi$	0.30	0.07	0.19	0.30	0.45	1	650

### 9.1.7 Count detector models in the secr package

The R package **secr** will fit Poisson or negative binomial encounter frequency models. The formatting of data and structure of the analysis proceeds in a similar fashion to the Bernoulli model described in Sec. 6.5, except that we specify the `detector='count'` option when the traps object is created. The set-up proceeds as follows:

```

8894 > library(secr)
8895 > library(scrbook)
8896 > data(wolverine)
8897
8898 > traps <- as.matrix(wolverine$wtraps)
8899 > dimnames(traps) <- list(NULL,c("trapID","x","y",paste("day",1:165,sep="")))
8900 > traps1 <- as.data.frame(traps[,1:3])
8901 > trapfile1 <- read.traps(data=traps1,detector="count")

```

You can proceed with analysis of these data and compare/contrast with the Bayesian analysis given above, or the results of the Bernoulli model fitted in Chapt. 6.

## 9.2 INDEPENDENT MULTINOMIAL OBSERVATIONS

Several types of encounter devices yield multinomial observations in which an individual can be caught in a single trap during a particular encounter occasion, but traps might catch any number of individuals. Mist netting is the canonical example of such a “multi-catch” device (Efford et al., 2009a). Also some kinds of bird or mammal cage-traps hold multiple animals, as do pit-fall traps which are commonly used for many species of herptiles. Another type of sample method that might be viewed (in some cases) as a multi-catch device are area-searches of, for example, reptiles where we think of a small polygon as the “trap” – we could get multiple individuals (turtles, lizards) in the same plot but not, in the same sample occasion, at different plots. The key features of this independent multinomial or multi-catch model are: (1) capture of an individual in a trap is *not* independent of its capture in other traps, because initial capture precludes capture in any other trap and (2) individuals behave independently of one another, so whether a trap captures some individual doesn’t have an affect on whether it captures another. A

8917 type of model in which the 2nd assumption is violated are the “single catch” trap systems  
 8918 which we address in Sec. 20.2.8 below.

8919 In this case we assume the observation  $\mathbf{y}_{ik}$  for individual  $i$  during sample occasion  $k$  is  
 8920 a multinomial observation which consists of a sequence of 0’s and a single 1 indicating the  
 8921 trap of capture, or “not captured”. For the “not captured” event we define an additional  
 8922 outcome, by convention element  $J + 1$  of the vector. As an example, if we capture an  
 8923 individual in trap 2 during some occasion of a study involving  $J = 6$  traps. Then, the  
 8924 multinomial observation has length  $J+1 = 7$ , and the observation is  $\mathbf{y}_i = (0, 1, 0, 0, 0, 0, 0)$ .  
 8925 An individual not captured at all would have the observation vector  $(0, 0, 0, 0, 0, 0, 1)$ . If  
 8926 we sample for 5 occasions in all and the individual is also caught in trap 4 during occasion  
 8927 3, but otherwise uncaptured, then the 5 encounter observations for that individual are as  
 8928 follows:

8929	occassion	trap						"not captured"
		1	2	3	4	5	6	
8930		-----	-----	-----	-----	-----	-----	-----
8931		-----	-----	-----	-----	-----	-----	-----
8932	1	0	1	0	0	0	0	0
8933	2	0	0	0	0	0	0	1
8934	3	0	0	0	1	0	0	0
8935	4	0	0	0	0	0	0	1
8936	5	0	0	0	0	0	0	1

8937 Statistically we regard the *rows* of this data matrix as *independent* multinomial trials.

8938 Analogous to our previous Bernoulli and Poisson models, we seek to construct the  
 8939 multinomial cell probabilities for each individual, as a function of *where* that individual  
 8940 lives, through its center of activity  $\mathbf{s}$ . Thus we suppose that

$$\mathbf{y}_{ik} | \mathbf{s}_i \sim \text{Multinomial}(1, \boldsymbol{\pi}(\mathbf{s}_i)) \quad (9.2.1)$$

8941 where  $\boldsymbol{\pi}(\mathbf{s}_i)$  is a vector of length  $J + 1$ , where  $\pi_{i,J+1}$ , the last cell, corresponds to the  
 8942 probability of the event “not captured”. Now we have to construct these cell probabili-  
 8943 ties in some meaningful way that depends on each individual’s  $\mathbf{s}$ . We use the standard  
 8944 multinomial logit with distance as a covariate:

$$\pi_{ij} = \frac{\exp(\alpha_0 - \alpha_1 d_{ij})}{1 + \sum_j \exp(\alpha_0 - \alpha_1 d_{ij})}$$

8945 for  $j = 1, 2, \dots, J$  and, for  $J + 1$ , i.e., “not captured”,

$$\pi_{i,(J+1)} = \frac{\exp(0)}{1 + \sum_j \exp(\alpha_0 - \alpha_1 d_{ij})}$$

8946 or, more commonly, we use  $d_{ij}^2$  to correspond to our Gaussian kernel model for encounter  
 8947 probability. Whatever function of distance we use in the construction of multinomial prob-  
 8948 abilities will have a direct correspondence to the standard encounter probability models  
 8949 we used in the Bernoulli or Poisson models as well (see Sec. 5.4).

8950 It is convenient to express these multinomial models short-hand as follows, e.g., for  
 8951 the Gaussian encounter probability model:

$$\text{mlogit}(\pi_{ij}) = \alpha_0 - \alpha_1 d_{ij}^2$$

8952 In this way we can refer to models with covariates in a more concise way. For example, a  
 8953 model with a trap-specific covariate, say  $C_j$ , is:

$$\text{mlogit}(\pi_{ij}) = \alpha_0 - \alpha_1 d_{ij}^2 + \alpha_2 C_j$$

8954 or we could include occasion-specific covariates too, such as behavioral response.

8955 A statistically equivalent distribution to the multinomial is the *categorical* distribution.

8956 If  $\mathbf{y}$  is a multinomial trial with probabilities  $\boldsymbol{\pi}$  than the *position* of the non-zero element of  
 8957  $\mathbf{y}$  is a categorical random variable with probabilities  $\boldsymbol{\pi}$ . We express this for SCR models  
 8958 as

$$\mathbf{y}|\mathbf{s} \sim \text{Categorical}(\boldsymbol{\pi}(\mathbf{s}))$$

8959 In the SCR context, the categorical version of the multinomial trial corresponds to the  
 8960 *trap of capture*. Using our example above with 6 traps then we could as well say  $y_{ik}$  is a  
 8961 categorical random variable with possible outcomes  $(1, 2, 3, 4, 5, 6, 7)$  where outcome  $y = 7$   
 8962 corresponds to “not captured.” Obviously, how this is organized or labeled is completely  
 8963 irrelevant, although it is convenient to use the integers 1 to  $(J + 1)$  where  $J + 1$  is the  
 8964 event not captured. Therefore, for our illustration in the previous table,  $y_{i1} = 2$ ,  $y_{i2} = 7$ ,  
 8965  $y_{i3} = 4$  and so on.

8966 For simulating and fitting data in the **BUGS** engines we will typically use the cat-  
 8967 egorical representation of the model because it is somewhat more convenient. We have  
 8968 found that fitting multinomial models in **WinBUGS** is less efficient than **JAGS** (Royle  
 8969 and Converse, in review), which we use in the subsequent examples involving multinomial  
 8970 observation models.

### 8971 9.2.1 Multinomial resource selection models

8972 The multinomial probabilities in Eq. 9.2.2 look similar to the multinomial resource selec-  
 8973 tion function (RSF) model for telemetry data (Manly et al., 2002; Lele and Keim, 2006).  
 8974 This suggests how we might model landscape or habitat covariates using such methods  
 8975 – i.e., by including them as explicit covariates in a larger multinomial model for “use” –  
 8976 which, if we take the product of use with encounter, produces a model for the observable  
 8977 encounter data. This leads naturally to the development of models that integrate RSF  
 8978 data from telemetry studies with SCR data (Royle et al., 2012a), which is the topic of  
 8979 Chapt. 13.

### 8980 9.2.2 Simulating data and analysis using JAGS

8981 We’re going to show the nugget of a simulation function which is used in the function  
 8982 **simMnSCR** found in the **R** package **scrbook**. The first lines of the following **R** code make  
 8983 use of some things that you need to define, but we omit them here (e.g., **xlim**, **ylim** are  
 8984 the boundaries of the state-space, **N** is the population size, etc.):

```
8985 ##
8986 ## Simulate random activity centers:
8987 ##      (first define N, xlim, ylim, etc...)
8988 ##
8989 > S <- cbind(runif(N,xlim[1],xlim[2]),runif(N,ylim[1],ylim[2]))
```

```

8990
8991 ## Distance from each individual to each trap
8992 > D <- e2dist(S,traplocs)
8993
8994 ## Set parameter values
8995 > sigma <- 0.5
8996 > alpha0 <- -1
8997 > alpha1 <- -1/(2*sigma*sigma)
8998
8999 ## make an empty data matrix and fill it up with data
9000 > Ycat <- matrix(NA,nrow=N,ncol=K)
9001 > for(i in 1:N){
9002   for(k in 1:K){
9003     lp <- alpha0 + alpha1*D[i,]*D[i,]
9004     cp <- exp(c(lp,0))
9005     cp <- cp/sum(cp)
9006     Ycat[i,k] <- sample(1:(ntraps+1),1,prob=cp)
9007   }
9008 }
```

9009 We save the data in the matrix `Ycat` to clarify that it is the categorical observation  
 9010 representing “trap of capture”. The matrix `Ycat` here has the maximal dimension  $N$   
 9011 and so, to do an analysis that mimics a real situation, we would have to discard the  
 9012 uncaptured individuals. The function `simMnSCR` in the package `scrbook` will also simulate  
 9013 data that includes a behavioral response which will be the typical situation in small-  
 9014 mammal trapping problems (see Converse and Royle, 2012, for details).

9015 Here we use our function `simMnSCR` to simulate a data set with  $K = 7$  occasions. We’ll  
 9016 run the model using `JAGS` which we have found is much more effective for this class of  
 9017 models. We get the data set-up for analysis by augmenting the size of the data set to  
 9018  $M = 200$ . In addition we choose starting values for  $s$  and the data augmentation variables  
 9019  $z$ . For starting values of  $s$  we cheat a little bit here and use the true values for the observed  
 9020 individuals and then augment the  $M \times 2$  matrix  $\mathbf{S}$  with  $M - n$  randomly selected activity  
 9021 centers. Our function `spiderplot` returns the mean observed location of individuals for  
 9022 use as starting values for the `nind` encountered individuals. The parameters input to  
 9023 `simMnSCR` are the intercept  $\alpha_0$ ,  $\sigma = \sqrt{1/(2\alpha_1)}$  for the Gaussian encounter probability  
 9024 model, and  $\alpha_2$  is the behavioral response parameter. The data simulation and set-up  
 9025 proceeds as follows:

```

9026 > set.seed(2013)
9027 > parms <- list(N=100,alpha0= -.40, sigma=0.5, alpha2= 0)
9028 > data <- simMnSCR(parms, K=7, ssbuff=2)
9029 > nind <- nrow(data$Ycat)
9030
9031 > M <- 200
9032 > Ycat <- rbind(data$Ycat,matrix(nrow(data$X)+1,nrow=(M-nind),ncol=data$K))
9033 > Sst <- rbind(data$S,cbind(runif(M-nind,data$xlim[1],data$xlim[2]),
9034                           runif(M-nind,data$ylim[1],data$ylim[2])))
```

```
9035 > zst <- c(rep(1,160),rep(0,40))
```

9036     The model specification is not much more complicated than the binomial or Poisson  
 9037     models given previously. The main consideration is that we define the cell probabilities for  
 9038     each trap  $j = 1, 2, \dots, J$  and then define the last cell probability,  $J+1$ , for “not captured”,  
 9039     to be the complement of the sum of the others. The code is shown in Panel 9.1. In the  
 9040     last lines of code here we specify  $N$  and density,  $D$ , as derived parameters.

9041     To fit the model, we need to package everything up (inits, parameters, data) and send  
 9042     it off to **JAGS** to build an MCMC simulator for us (these commands are executed in  
 9043     the help file for `simMnSCR`). In addition to the usual data objects, we also pass the limits  
 9044     of the assumed rectangular state-space (`ylim`, `xlim`, both  $1 \times 2$  vectors) and the scale of  
 9045     the standardized units, called `trap.space` here because we typically will define the trap  
 9046     coordinates to be an integer grid. If the trap spacing is 10 m and we want units of density  
 9047     computed in terms of individuals per meter-squared, then we input `trap.space=10`. The  
 9048     analysis is carried out as follows:

```
9049 > inits <- function(){ list (z=zst,sigma=rnorm(1,.5,1) ,S=Sst) }
9050
9051 # Parameters to monitor
9052 > parameters <- c("psi","alpha0","alpha1","sigma","N","D")
9053
9054 # Bundle the data. Note this reuses "data"
9055 > data <- list (X=data$X,K=data$K, trap.space=1,Ycat=Ycat,M=M,
9056   ntraps=nrow(data$X),ylim=data$ylim,xlim=data$xlim)
9057
9058 > library(R2jags)
9059 > out <- jags (data, inits, parameters, "model.txt", n.thin=1,
9060   n.chains=3, n.burnin=1000, n.iter=2000)
```

9061     The posterior summaries are provided in the following **R** output (recall that  $N = 100$ ,  
 9062      $\alpha_0 = -.40$ , and  $\sigma = 0.5$ ):

```
9063 > out
9064 Inference for Bugs model at "model.txt", fit using jags,
9065   3 chains, each with 2000 iterations (first 1000 discarded)
9066   n.sims = 3000 iterations saved
9067
9068   mu.vect sd.vect 2.5%    25%    50%    75% 97.5% Rhat n.eff
9069   D        1.873  0.189  1.531   1.750   1.859   2.000  2.250 1.006 1300
9070   N       119.867 12.107 98.000 112.000 119.000 128.000 144.000 1.006 1300
9071   alpha0   -0.435  0.151  -0.738  -0.535  -0.439  -0.331  -0.146 1.004  580
9072   alpha1   2.195  0.286  1.658   2.004   2.180   2.372   2.785 1.003 2400
9073   psi      0.599  0.069  0.465   0.552   0.599   0.645   0.739 1.006 1400
9074   sigma    0.480  0.032  0.424   0.459   0.479   0.500   0.549 1.003 2400
9075   deviance 892.164 21.988 850.922 877.417 891.561 906.246 937.728 1.003  950
9076 [... output deleted ....]
```

---

```

model{
psi ~ dunif(0,1)
alpha0 ~ dnorm(0,10)
sigma ~ dunif(0,10)
alpha1 <- 1/(2*sigma*sigma)

for(i in 1:M){
  z[i] ~ dbern(psi)
  S[i,1] ~ dunif(xlim[1],xlim[2])
  S[i,2] ~ dunif(ylim[1],ylim[2])
  for(j in 1:ntraps){
    #distance from capture to the center of the home range
    d[i,j] <- pow(pow(S[i,1]-X[j,1],2) + pow(S[i,2]-X[j,2],2),1)
  }
  for(k in 1:K){
    for(j in 1:ntraps){
      lp[i,k,j] <- exp(alpha0 - alpha1*d[i,j])*z[i]
      cp[i,k,j] <- lp[i,k,j]/(1+sum(lp[i,k,]))
    }
    cp[i,k,ntraps+1] <- 1-sum(cp[i,k,1:ntraps]) # last cell = not captured
    Ycat[i,k] ~ dcat(cp[i,k,])
  }
}
N <- sum(z[1:M])
A <- ((xlim[2]-xlim[1])*trap.space)*((ylim[2]-ylim[1])*trap.space)
D <- N/A
}

```

---

Panel 9.1: **BUGS** model specification for the independent multinomial observation model. For data simulation and model fitting see the help file `?simMnSCR` in the **R** package `scrbook`.

---

### 9077 9.2.3 Multinomial relationship to the Poisson

9078 The multinomial is related to the Poisson encounter rate model by a conditioning argument.  
 9079 Let  $y_{ij}$  be the number of encounters for individual  $i$  in trap  $j$ . If  $y_{ij} \sim \text{Poisson}(\lambda_{ij})$ ,  
 9080 then, conditional on the *total* number of captures (i.e., across all traps),  $y_i = \sum_j y_{ij}$ , the  
 9081 trap encounter frequencies are multinomial with probabilities

$$\pi_{ij} = \frac{\lambda_{ij}}{\sum_j \lambda_{ij}}$$

9082 for  $j = 1, 2, \dots, J$ . Or equivalently the *trap of capture* is categorical with probabilities  $\pi_{ij}$   
 9083 as given above. Under the Gaussian kernel model, these probabilities are:

$$\pi_{ij} = \frac{\exp(-\alpha_1 d(\mathbf{x}_i, \mathbf{s}_i)^2)}{\sum_j \exp(-\alpha_1 d(\mathbf{x}_i, \mathbf{s}_j)^2)} \quad (9.2.2)$$

9084 where, we note, the intercept  $\alpha_0$  has canceled from both the numerator and denominator.  
 9085 This makes sense because, here, these probabilities describe the trap-specific capture prob-  
 9086 abilities *conditional on capture*. Therefore, the model is not completely specified, absent  
 9087 a model for the “overall” probability of encounter or the expected frequency of captures,  
 9088 say  $\phi_i$ . Depending on how we specify a model for this quantity  $\phi_i$ , we can reconcile it  
 9089 directly with the Poisson model. Let  $y_i$  be the total number of encounters for individual  
 9090  $i$  and suppose  $y_i$  has a Poisson distribution with mean  $\phi_i$ . Then, marginalizing Eq. 9.2.1  
 9091 over the Poisson distribution for  $y_i$  produces the original set of *iid* Poisson frequencies  
 9092 with probabilities:

$$\lambda_{ij} = \phi_i \pi_{ij}$$

9093 for  $j = 1, 2, \dots, J$ . In particular, if we suppose that  $\phi_i = \sum_j \exp(\alpha_0 - \alpha_1 d(\mathbf{x}, \mathbf{s})^2)$  then  
 9094 the marginal distribution of  $y_{ij}$  is Poisson with mean  $\exp(\alpha_0 - \alpha_1 d(\mathbf{x}, \mathbf{s})^2)$ , equivalent to  
 9095 Eq. 9.1.1.

9096 In summary, the Poisson and multinomial models are equivalent in how they model  
 9097 the distribution of captures among traps. It stands to reason that, if the encounter  
 9098 rate of individuals is low, we could use the Poisson and multinomial models interchange-  
 9099 ably. In fact, based on our discussion in Sec. 9.1.2 above we could use any of the bino-  
 9100 mial/Poisson/multinomial models with little ill-effect when encounter rate is low.

### 9101 9.2.4 Avian mist-netting example

9102 We analyze data from a mist-netting study of ovenbirds, conducted at the Patuxent  
 9103 Wildlife Research Center, Laurel MD, by D.K. Dawson and M.G. Efford. The data from  
 9104 this study are available in the **secr** package, and have been analyzed previously by Efford  
 9105 et al. (2004), see also Borchers and Efford (2008). Forty-four mist nets spaced 30 m apart  
 9106 on the perimeter of a 600-m x 100-m rectangle were operated on 9 or 10 non-consecutive  
 9107 days in late May and June for 5 years from 2005-2009. The ovenbird data can be loaded  
 9108 as follows:

```
9109 > library(secr)
9110 > data(ovenbird)
```

9111 The data set consists of adult ovenbirds caught during sampling in each of 5 years, 2005-  
9112 2009. (one ovenbird was killed in 2009, indicated by a negative net number in the encounter  
9113 data file). As with most mist-netting studies, nets are checked multiple times during a  
9114 day (e.g., every hour during a morning session). However, for this data set, the within-day  
9115 recaptures are not included so each bird has at most a single capture per day. Therefore  
9116 the multinomial model (detector type ‘multi’ in **secr**) is appropriate. Although several  
9117 individuals were captured in more than one year, this information is not used in the models  
9118 presently offered in **secr**, but we do make use of it in the development of open models in  
9119 Chapt. 16.

9120 **Multiple sample sessions**

9121 Up to this point we have only dealt with a basic closed population sampling situation  
9122 consisting of repeated sample occasions on a single population of individuals using a single  
9123 array of traps. In practice, many studies produce repeated samples over longer periods  
9124 of time over which demographic closure isn’t valid, or at different locations where the  
9125 populations are completely distinct. We adopt the **secr** terminology of *session* for such  
9126 replication by groups of time or space, and the models are *multi-session* models, although  
9127 we think of such models as being relevant to any stratified population (see Chapt. 14).  
9128 We introduced **secr**’s multi-session models in Sec. 6.5.4. In the case of the ovenbird data,  
9129 sampling was carried out in multiple years, with a number of sample occasions within  
9130 each year (9 or 10), a type of data structure commonly referred to as “the robust design”  
9131 (Pollock, 1982). In this context, it stands to reason that there is recruitment and mortality  
9132 happening across years. In Chapt. 16 we model these processes explicitly but, here, we  
9133 provide an analysis of the data that does not require explicit models for recruitment and  
9134 survival, regarding the yearly populations as independent strata, and fitting a multi-session  
9135 model.

9136 When the sessions represent explicit time periods, the multi-session model of **secr** can  
9137 be thought of as a type of open population model. In particular, a special case of open  
9138 models arises when we assume  $N_t$  (time-specific population sizes) are independent from  
9139 one time period or session to the next – this can be thought of as a “random temporary  
9140 emigration” model of the Kendall et al. (1997) variety, and this is the multi-session model  
9141 implemented in **secr**. In particular, by assuming that  $N_t$  is Poisson with mean  $\Lambda_t$ , one can  
9142 model variation in abundance among sessions based on the Poisson-integrated likelihood  
9143 in which parameters of  $\Lambda_t$  appear directly in the likelihood as we noted in Sec. 6.5.4.  
9144 We provide an analysis (below) of the ovenbird data here using the multi-session models  
9145 in **secr**. We formalize the multi-session model approach from a Bayesian perspective  
9146 using data augmentation in Chapt. 14 (Converse and Royle, 2012; Royle and Converse,  
9147 in review).

9148 A 3rd way to develop models for stratified or grouped populations, not based on  
9149 multi-session models, but that is convenient in **BUGS**, is to regard the data from each  
9150 session as an independent data set with its own  $N_t$  parameter, and do  $T$  distinct data  
9151 augmentations. Because each  $N_t$  is regarded as a free parameter, independent of the  
9152 other parameters, we’ll call this the nonparametric multi-session model to distinguish it  
9153 from the multi-session model which assumes the  $N_t$  are related to one another by having  
9154 been generated from a common Poisson distribution. We can analyze this model in the  
9155 normal context of data augmentation by augmenting each year separately in the same  
9156 **BUGS** model specification. This approach avoids making explicit model assumptions

about the  $N_t$  parameters. This is distinct from the model implemented in **secr** in that **secr** is removing the  $N_t$  parameters by integrating the conditional-on- $N_t$  likelihood over the Poisson prior for  $N_t$ <sup>1</sup>

We demonstrate these 3 approaches to analyzing grouped/stratified data using the ovenbird data: (1) In the following section, we provide the nonparametric multi-session model with unconstrained  $N_t$ ; (2) we demonstrate the Poisson model-based multi-session models from **secr** both here (following section) and in Chapt. 14 from a Bayesian standpoint; (3) later, in Chapt. 16, we provide a fully dynamic “spatial Jolly-Seber” model and apply it to the ovenbird data.

### Analysis in JAGS

The ovenbird data are provided as a multi-session **capthist** object **ovenCH** which, by regarding years as independent strata, or sessions, allows for the fitting of the multi-session model. For doing a Bayesian analysis in one of the **BUGS** engines (we use **JAGS** here) there are a number of ways to structure the data and describe the model. We can analyze either a 2-d data set with all years (data augmented) “stacked” into a data set of dimension  $(5 * M) \times 10$  ( $5$  years,  $M$  = size of the augmented data set,  $K$  = 10 replicate sample occasions). Or, we could produce a 3-d array  $(M \times J \times K)$ . We adopted the former approach, analyzing the data as a 2-d array and creating an additional categorical variable for “year” to indicate which stratum (year) each record goes with.

Data on individual sex is included with **secr**, but we provide an analysis of a single model for all adults, constant  $\sigma$  across years, constant  $p_0$ , and year-specific values of  $N_t$  (and hence  $D_t$ ). There is a habitat mask provided with the data but the mask appears to just be a modified rectangle around the net locations, clipped to have rounded corners, and so we don’t use it here. Instead, we used a rectangular state-space buffer of 200 meters for our analysis. There was a single loss-on-capture which we accounted for by fixing  $p = 0$  for all subsequent encounters of that individual (indicated by the binary variable **dead**, as shown in Panel 9.2). We have an **R** script in **scrbook** package called **SCRovenbird**, so you can see how to set-up the data and run the model. Executing the script **SCRovenbird** produces the posterior summaries given in Table 9.2. Here, density is in units of birds per ha. The posterior mean of  $\sigma$  is about 76 meters, and there is considerable variability in density over the 5 year period with density peaking at 1.2 birds/ha in year 3, although there is considerable posterior uncertainty. The R-hat’s look a little bit peaked and so we might consider running the MCMC analysis longer.

### Analysis in secr

Included with the ovenbird data are a number of models fitted as examples. Those include:

```
9192 ovenbird.model.1    fitted secr model -- null
9193 ovenbird.model.1b   fitted secr model -- g0 net shyness
9194 ovenbird.model.1T   fitted secr model -- g0 time trend within years
9195 ovenbird.model.h2   fitted secr model -- g0 finite mixture
9196 ovenbird.model.D    fitted secr model -- trend in density across years
```

---

<sup>1</sup>We do not know of **secr** documentation that states this (or contradicts it). We think this is what is being done, based partially on conversations or emails with M.G. Efford, D.L. Borchers, the various publications on **secr**, and our own thinking about it.

```

model{
  alpha0 ~ dnorm(0,.1)
  sigma ~ dunif(0,200)
  alpha1 <- 1/(2*sigma*sigma)

  A <- ((xlim[2]-xlim[1]))*((ylim[2]-ylim[1]))
  for(t in 1:5){
    N[t] <- inprod(z[1:bigM],yrdummy[,t])
    D[t] <- (N[t]/A)*10000 # Put in units of per ha
    psi[t] ~ dunif(0,1)
  }

  for(i in 1:bigM){ # bigM = total size of jointly augmented data set
    z[i] ~ dbern(psi[year[i]])
    S[i,1] ~ dunif(xlim[1],xlim[2])
    S[i,2] ~ dunif(ylim[1],ylim[2])

    for(j in 1:ntraps){ # X = trap locations, S = activity centers
      d2[i,j] <- pow(pow(S[i,1]-X[j,1],2) + pow(S[i,2]-X[j,2],2),1)
    }
    for(k in 1:K){
      Ycat[i,k] ~ dcat(cp[i,k,])
      for(j in 1:ntraps){
        lp[i,k,j] <- exp(alpha0 - alpha1*d2[i,j])*z[i]*(1-dead[i,k])
        cp[i,k,j] <- lp[i,k,j]/(1+sum(lp[i,k,1:ntraps]))
      }
      cp[i,k,ntraps+1] <- 1-sum(cp[i,k,1:ntraps]) # Last cell = not captured
    }
  }
}

```

Panel 9.2: **BUGS** model specification for the non-parametric multi-session model in which each  $N_t$  is independent of the other. The implied prior (by data augmentation) is that  $N_t \sim \text{Uniform}(0, 100)$ . To fit this model to the ovenbird data, see `?SCRovenbird` in the **R** package `scrbook`.

**Table 9.2.** Posterior summary statistics for the ovenbird mist-netting data based on the independent multinomial (“multi-catch”) encounter process model. Parameters  $\psi$ ,  $N$  and  $D$  are indexed by year. MCMC was done using jags with 3 chains, each with 11000 iterations, discarding the first 1000, for a total of 30000 posterior samples.

Parameter	Mean	SD	2.5%	50%	97.5%	Rhat	n.eff
$D[1]$	0.983	0.211	0.636	0.966	1.455	1.002	1900
$D[2]$	1.023	0.209	0.673	1.003	1.492	1.001	7100
$D[3]$	1.208	0.238	0.807	1.186	1.749	1.004	740
$D[4]$	0.896	0.195	0.575	0.880	1.333	1.002	3000
$D[5]$	0.753	0.177	0.465	0.734	1.149	1.001	4000
$\alpha_0$	-3.479	0.160	-3.797	-3.477	-3.171	1.005	490
$\alpha_1$	0.000	0.000	0.000	0.000	0.000	1.003	1100
$\sigma$	76.214	6.125	65.569	75.758	89.360	1.003	1100
$N[1]$	80.423	17.283	52.000	79.000	119.000	1.002	1900
$N[2]$	83.685	17.077	55.000	82.000	122.000	1.001	7100
$N[3]$	98.822	19.483	66.000	97.000	143.000	1.004	740
$N[4]$	73.288	15.962	47.000	72.000	109.000	1.002	3000
$N[5]$	61.589	14.468	38.000	60.000	94.000	1.001	4000
$\psi[1]$	0.403	0.092	0.246	0.395	0.606	1.002	1600
$\psi[2]$	0.419	0.091	0.260	0.412	0.620	1.001	6400
$\psi[3]$	0.494	0.102	0.315	0.486	0.723	1.004	760
$\psi[4]$	0.368	0.086	0.221	0.361	0.555	1.002	3200
$\psi[5]$	0.310	0.079	0.178	0.302	0.485	1.002	3500

9197     The model fit objects provided in `secr` are based on the use of the habitat mask.  
 9198     To make the analyses consistent with our previous analysis in **JAGS**, we refit all of the  
 9199     models here without the habitat mask. The re-analysis proceeds as follows, changing the  
 9200     “trend in density across years” model to allow for year-specific density:

```
9201 ## Fit constant-density model
9202 > ovenbird.model.1 <- secr.fit(ovenCH)
9203 ## Fit net avoidance model
9204 > ovenbird.model.1b <- secr.fit(ovenCH, model = list(g0 ~ b))
9205 ## Fit model with time trend in detection
9206 > ovenbird.model.1T <- secr.fit(ovenCH, model = list(g0 ~ T))
9207 ## Fit model with 2-class mixture for g0
9208 > ovenbird.model.h2 <- secr.fit(ovenCH, model = list(g0 ~ h2))
9209 ## Fit a model with session (year)-specific Density
9210 > ovenbird.model.DT <- secr.fit(ovenCH, model = list(D ~ session))
```

9211     All of these can be fitted easily in **JAGS** but the model we fitted previously is roughly  
 9212     equivalent to the last model, `ovenbird.model.DT`, because we allowed for year-specific  
 9213     population sizes (and hence density). So, we’ll compare our results from **JAGS** to that  
 9214     model. The `secr` output is extensive and so we do not reproduce it completely here. By

9215 default, it summarizes the trap information for each year, encounter information, and then  
 9216 output for each year. Here is an abbreviated version for `ovenbird.model.DT`:

```

9217 > print(ovenbird.model.DT,digits=2)
9218
9219 secr.fit( capthist = ovenCH, model = list(D ~ session), buffer = 300 )
9220 secr 2.3.1, 14:46:52 23 Jan 2013
9221
9222 $`2005`
9223 Object class      traps
9224 Detector type    multi
9225 Detector number   44
9226 Average spacing   30.27273 m
9227 x-range           -50 49 m
9228 y-range           -285 285 m
9229
9230 [... deleted ...]
9231
9232          2005 2006 2007 2008 2009
9233 Occasions     9   10   10   10   10
9234 Detections    35   42   52   30   33
9235 Animals       20   22   26   19   16
9236 Detectors     44   44   44   44   44
9237
9238 Model          : D~session g0~1 sigma~1
9239 Fixed (real)   : none
9240 Detection fn   : halfnormal
9241 Distribution   : poisson
9242 N parameters   : 7
9243 Log likelihood : -1119.845
9244 AIC            : 2253.689
9245 AICc           : 2254.868
9246
9247 [... deleted ...]
```

9248 To do model selection we use the handy helper-function `AIC` as follows (output edited  
 9249 to fit on the page):

```

9250 AIC (ovenbird.model.1, ovenbird.model.1b, ovenbird.model.1T,
9251          ovenbird.model.h2, ovenbird.model.DT)
9252
9253          model detectfn npar logLik     AIC     AICc     dAICc
9254 ovenbird.model.1T [edited output]  4 -1111.850 2231.700 2232.109 0.000
9255 ovenbird.model.1b        ....      4 -1117.615 2243.229 2243.637 11.528
9256 ovenbird.model.h2        ....      3 -1121.164 2248.327 2248.570 16.461
9257 ovenbird.model.1         ....      5 -1119.762 2249.524 2250.143 18.034
9258 ovenbird.model.DT        ....      7 -1119.845 2253.689 2254.868 22.759
```

9259    We see that our DT model is way down at the bottom of the list. Instead, the model with  
 9260    a time-trend (within-season) in detection probability is preferred, followed by a behavioral  
 9261    response. We encourage you to adapt the **JAGS** model specification for such models which  
 9262    is easily done (see Chapt. 7 for many examples). We provide the summary results for the  
 9263    model having  $D \sim \text{session}$  as follows:

```

9264 > print(ovenbird.model.DT,digits=2)
9265
9266 secr.fit( capthist = ovenCH, model = list(D ~ session), buffer = 300 )
9267 secr 2.3.1, 14:46:52 23 Jan 2013
9268
9269 [...deleted....]
9270
9271 Fitted (real) parameters evaluated at base levels of covariates
9272
9273 session = 2005
9274      link estimate SE.estimate    lcl    ucl
9275 D      log     0.920       0.228  0.571  1.484
9276 g0     logit    0.028       0.004  0.021  0.037
9277 sigma   log    78.566      6.379 67.025 92.095
9278
9279 session = 2006
9280      link estimate SE.estimate    lcl    ucl
9281 D      log     0.963       0.238  0.598  1.553
9282 g0     logit    0.028       0.004  0.021  0.037
9283 sigma   log    78.566      6.379 67.025 92.095
9284
9285 session = 2007
9286      link estimate SE.estimate    lcl    ucl
9287 D      log     1.139       0.282  0.706  1.836
9288 g0     logit    0.028       0.004  0.021  0.037
9289 sigma   log    78.566      6.379 67.025 92.095
9290
9291 session = 2008
9292      link estimate SE.estimate    lcl    ucl
9293 D      log     0.832       0.206  0.516  1.341
9294 g0     logit    0.028       0.004  0.021  0.037
9295 sigma   log    78.566      6.379 67.025 92.095
9296
9297 session = 2009
9298      link estimate SE.estimate    lcl    ucl
9299 D      log     0.701       0.173  0.435  1.130
9300 g0     logit    0.028       0.004  0.021  0.037
9301 sigma   log    78.566      6.379 67.025 92.095

```

9302    The point estimates (MLEs) of density are uniformly lower than the Bayesian estimates  
 9303    (posterior means) shown in Table 9.2. We expect some difference in this direction due

9304 to small-sample skew of the posterior. In addition, there may be slight differences due  
 9305 to the fact that **secr** multi-session model assumes that the  $N_t$  have a Poisson prior, but  
 9306 the implementation in **JAGS** using data augmentation is based on a binomial prior. The  
 9307 estimated  $\sigma$  is very similar between the **JAGS** analysis and **secr**.

### 9.3 SINGLE-CATCH TRAPS

9308 The classical animal trapping experiment is based on a physical trap which captures a  
 9309 single animal and holds that individual until subsequent molestation by a biologist. This  
 9310 type of observation model – the “single-catch” trap – was the original situation considered  
 9311 in the context of spatial capture-recapture by Efford (2004). Nowadays, capture-recapture  
 9312 data are more often obtained by other methods (DNA from hair snares, or scat sampling,  
 9313 camera traps etc...) but nevertheless the single-catch traps are still widely used in small  
 9314 mammal studies (Converse et al., 2006b; Converse and Royle, 2012) and other situations.

9315 The single-catch model is basically a multinomial model but one in which the number  
 9316 of available traps is reduced as each individual is captured. As such, the constraints on the  
 9317 joint likelihood for the sample of  $n$  encounter histories are very complicated. As a result,  
 9318 at the time of this writing, there has not been a formal development of either likelihood or  
 9319 Bayesian analysis of this model and applications of SCR models to single-catch systems  
 9320 have used the independent multinomial model as an approximation (see below).

9321 Nevertheless, we can make some progress to describing the basic observation model  
 9322 formally. In particular, if we imagine that all of the individuals captured queued up at  
 9323 the beginning of the capture session to draw a number indicating their order of capture,  
 9324 then there is a nice conditional structure resulting from a “removal process” operating on  
 9325 the traps. The first individual captured has the multinomial observation model:

$$\mathbf{y}_1 \sim \text{Multinomial}(\boldsymbol{\pi}_1)$$

9326 whereas the 2nd individual captured also has a multinomial encounter probability model  
 9327 but with the trap which captured the first individual removed. We might express this as:

$$\mathbf{y}_2 \sim \text{Multinomial}(\boldsymbol{\pi}_2)$$

9328 where

$$\pi_{2j} = \frac{(1 - y_{1j}) * \exp(\alpha_0 - \alpha_1 d_{ij}^2)}{\sum_j (1 - y_{1j}) * \exp(\alpha_0 - \alpha_1 d_{ij}^2)}$$

9329 and so on for  $i = 3, 4, \dots, n$ . In a certain way, this model is a type of local behavioral  
 9330 response model but where the response is to other individuals being captured. Evidently,  
 9331 the **order of capture** is relevant to the construction of these multinomial cell probabilities.  
 9332 More generally, the *time* of capture of an individual in any trapping interval will  
 9333 affect the encounter probability of subsequently captured individuals, but we think that  
 9334 order of capture might lead to a practical approximation to the single-catch process (this  
 9335 is how we simulate the data in our function **simScSCR**). In the simulation of single catch  
 9336 data, we randomly ordered the population of individuals for each sample occasion, and  
 9337 then cycled through them, turning off each trap if an individual was captured in it.

9338 **9.3.1 Inference for single-catch systems**

9339 For the single-catch model, we argued that the observations have a multinomial type of  
 9340 observation model, but the multinomial observations have a unique conditional dependence  
 9341 structure among them owing to the “removal” of traps as they fill-up with individuals.  
 9342 Thus, competition for single-catch traps renders the independence assumptions for the  
 9343 independent multinomial model invalid. However, as Efford et al. (2009a) noted, we  
 9344 expect “bias to be small when trap saturation (the proportion of traps occupied) is low.  
 9345 Trap saturation will be higher when population density is high...” relative to trap density,  
 9346 or when net encounter probability is high. Efford et al. (2009a) did a limited simulation  
 9347 study and found essentially no effective bias and concluded that estimators of density  
 9348 from the misspecified independent multinomial model are robust to the mild dependence  
 9349 induced when trap saturation is low. Naturally then, we expect that the Poisson model  
 9350 could also be an effective approximation under the same set of circumstances.

9351 In the **R** package **scrbook** we provide a function for simulating data from a single-catch  
 9352 system (function **simScSCR**) and fitting the misspecified model (**example(simScSCR)**) in  
 9353 **JAGS** so that you can evaluate the effectiveness of this misspecified model for situations  
 9354 that interest you.

9355 **9.3.2 Analysis of Efford's possum trapping data**

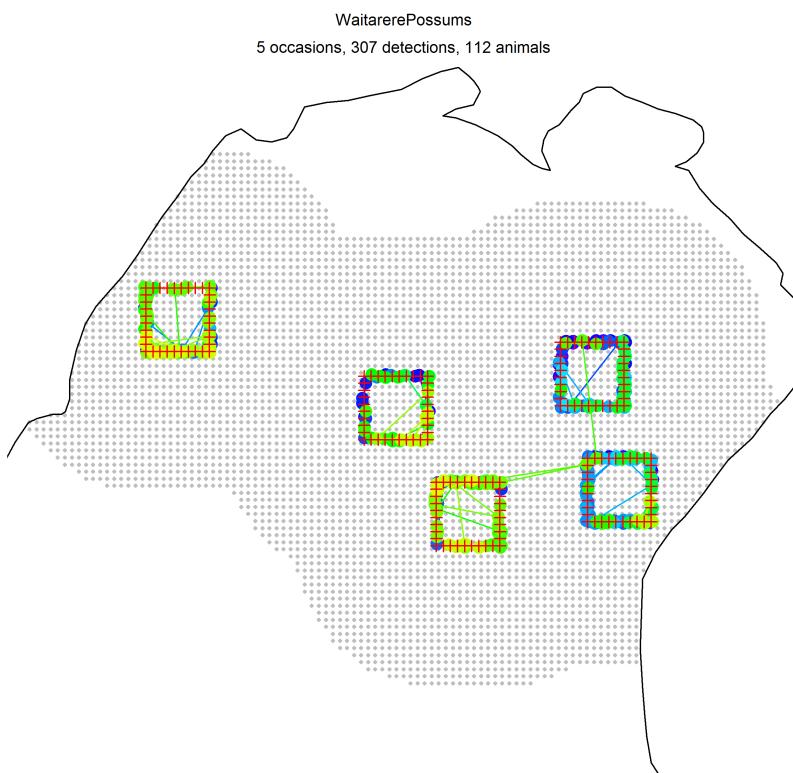
9356 We provide an analysis here of data from a study of brushtail possums in New Zealand.  
 9357 The data are available with the **R** package **secr** (Efford et al., 2009a); see the help file  
 9358 **?possum** after loading the **secr** package. Originally the data were analyzed by Efford et al.  
 9359 (2005), and a detailed description of the data set is available in the help file, from which  
 9360 we summarize:

9361       *Brushtail possums (*Trichosurus vulpecula*) are an unwanted invasive species in New  
 9362 Zealand. Although most abundant in forests, where they occasionally exceed densities  
 9363 of 15/ha, possums live wherever there are palatable food plants and shelter.*

9364 To load the possum data, execute the following commands:

9365 > library(secr)  
 9366 > data(possum)

9367 The study area encompasses approximately 300 ha, and 180 live traps were organized in 5  
 9368 distinct grids, shown in Fig. 9.2. Each square arrangement of traps consisted of 36 traps  
 9369 with a spacing of 20 m. Thus the squares are 180 m on a side. Individuals were captured,  
 9370 tagged, and released over 5 days during April, 2002. A noteworthy aspect of this study is  
 9371 that it involves replicated grids selected in some fashion from within a prescribed region.  
 9372 From an analysis standpoint, we could adopt the use of the multi-session models which we  
 9373 used previously to analyze the ovenbird data. This would be useful if we had covariates  
 9374 at the trapping grid level that we wanted to model. Alternatively, we could pool the data  
 9375 from all of the grids and analyze them jointly as if they were based on a single trapping  
 9376 grid (with 180 traps) which is clearly a reasonable view in this case. In doing this sort of  
 9377 pooling, there is an implicit assumption that  $N_t$  ( $t$  indexing trapping grid in this case) is  
 9378 Poisson distributed, with constant mean (Royle, 2004a; Royle et al., 2012c) which we also  
 9379 address in Chapt. 14.



**Figure 9.2.** Trapping grids used in possum study from Efford et al. (2005), data are contained in the R package `secr` (Efford, 2011a), refer to the help file `?possum` for additional details of this study.

9380     The data file **possumCH** contains 112 encounter histories, and we analyze those here  
 9381     although the last 8 of those are recaptures treated as new individuals<sup>2</sup>. The encounter  
 9382     process is not strictly a single-catch multinomial process because, as noted in the **possum**  
 9383     help file “One female possum was twice captured at two sites on one day, having entered  
 9384     a second trap after being released; one record in each pair was selected arbitrarily and  
 9385     discarded.” which is a similar situation to what might happen in bird mist net studies, as  
 9386     a bird might fly into a net upon release from another. By discarding the two extra-capture  
 9387     events, we can satisfactorily view these data as single-catch data, for which **secr** uses the  
 9388     independent multinomial likelihood (M. Efford, pers. comm.). If multiple, same-session  
 9389     captures were common, then it might be worth developing a model for  $n_{ik}$  = the number  
 9390     of captures of individual  $i$  during sample occasion  $k$ , in order to make use of all captures.

9391     For our Bayesian analysis here, we used a rectangular state-space which doesn’t ac-  
 9392     count for any geographic boundaries of the survey region, but we note that a habitat mask  
 9393     is included in **secr** and it could be used in a Bayesian analysis. Whether or not we use the  
 9394     mask is probably immaterial as long as we understand the predictions of  $N$  or  $D$  over the  
 9395     water don’t mean anything biological and we probably wouldn’t report such predictions.  
 9396     The **JAGS** model specification is based on that of the ovenbird analysis given previously,  
 9397     and so we don’t reproduce the model here. The **R/JAGS** script is called **SCRpossum**,  
 9398     which is in the **scrbook** package. The results are summarized in Table 9.3.

**Table 9.3.** Results of fitting the independent multinomial observation model to the possum trapping data. Strictly speaking, the trapping device is a “single-catch” trap, and the model represents an intentional misspecification. Density is reported in individuals per ha ( $D_{ha}$ ). Posterior summaries were obtained using **JAGS** with 3 chains, each with 2000 iterations, discarding the first 1000 as burn-in, to yield a total of 3000 posterior samples.

Parameter	Mean	SD	2.5%	50%	97.5%	Rhat	n.eff
$N$	235.407	17.435	204.000	235.000	270.000	1.009	340
$D_{ha}$	1.549	0.115	1.343	1.547	1.777	1.009	340
$\alpha_0$	-0.935	0.167	-1.270	-0.934	-0.605	1.007	870
$\alpha_1$	0.000	0.000	0.000	0.000	0.000	1.001	2800
$\sigma$	52.020	2.675	47.067	51.933	57.585	1.001	2800
$\psi$	0.783	0.062	0.666	0.782	0.903	1.008	340

9399     The estimated density (posterior mean) is about 1.53 possums/ha. To obtain the **secr**  
 9400     results for the equivalent null model, we execute the following command

```
9401 > secr.fit( capthist = possumCH, trace = F )
9402 which produces (edited) summary output:
9403 [... some output deleted ...]
9404
9405 Fitted (real) parameters evaluated at base levels of covariates
9406   link estimate SE.estimate      lcl      ucl
9407 D      log    1.6988930  0.17352645  1.3913904  2.0743547
```

<sup>2</sup>M. Efford, personal communication

---

```

9408 g0    logit  0.1968542  0.02256272  0.1563319  0.2448321
9409 sigma   log 51.4689114  2.59981905  46.6204139  56.8216500
9410
9411 [... some output deleted ...]

```

9412 As we've discussed previously, there are many reasons for why there might be differences  
9413 between Bayesian and likelihood estimates. But even among likelihood estimates – any  
9414 time you run a model there is some numerical integration going on which requires some  
9415 specific choices of how to do the integration (see Chapt. 6). For now we just observe that  
9416 the estimated density is certainly in the ballpark (compared to those in Table. 9.3), and  
9417 so too is the estimated  $\sigma$ .

## 9.4 ACOUSTIC SAMPLING

9418 The last decade has seen an explosion of technology that benefits the study of animal  
9419 populations. This includes DNA sampling methods that allow for identification from  
9420 hair or scat, camera trapping and identification software that allow efficient sampling  
9421 of many mammals, and the resulting statistical technology that helps us to make sense  
9422 of such data (Borchers and Efford, 2008; Royle and Young, 2008; Efford et al., 2009b;  
9423 Gopalaswamy et al., 2012b; Sollmann et al., 2013; Chandler and Royle, 2013). One other  
9424 extremely promising technology area is that of acoustic sampling using microphones or  
9425 recording devices. That is, instead of having cameras record encounters, or humans pick  
9426 up scat, we can establish an array of (usually) electronic recording devices which, instead of  
9427 establishing a visual identity of individuals, record a vocal expression of each individual. In  
9428 this context, Efford et al. (2009b) referred to audio recorders as “signal strength proximity  
9429 detectors” to distinguish them from other types of proximity detections, including camera  
9430 traps, which are *visual* proximity detector. Using audio records, the spatial pattern of the  
9431 *signal strength* at the different audio recorders or microphones can be used for inference  
9432 about density (Dawson and Efford, 2009; Efford et al., 2009b) in the same way as the  
9433 spatial pattern of detections is used in the types of SCR models we have discussed so far.  
9434 The basic technical formulation of these models comes from Efford et al. (2009b), and it  
9435 was applied to field study of birds by Dawson and Efford (2009). In that study, recording  
9436 devices were organized in groups of 4 (in a square pattern), with an array of  $5 \times 15$  such  
9437 clusters of 4, separated by 100 m (300 total recorder locations). This data set, called  
9438 **signalCH**, is provided with the **secr** package along with some sample analyses and help  
9439 files. See Efford and Dawson (2010), a version of the document **secr-sound.pdf** (that  
9440 also comes with the **secr** package) which you can access directly from the main help file  
9441 (**?secr**).

9442 Our development here mostly follows Efford et al. (2009b), but we change some nota-  
9443 tion to be consistent with our previous material. Let  $S(\mathbf{x}, \mathbf{u})$  be the strength of a signal  
9444 emanating from signal location  $\mathbf{u}$ , as recorded by a device at location  $\mathbf{x}$ . Just as ordinary  
9445 SCR models represent a model of *encounter frequency* as a function of distance, in acoustic  
9446 models, the acoustic SCR model is a model of sound attenuation as a function of distance.  
9447 In particular, the acoustic models assumes that  $S$  (or a suitable transformation) declines  
9448 with distance  $d$  from the origin of the sound, to the recording device. In the context of  
9449 spatial sampling of animals, the origin is the actual location of some individual animal,

9450 and the recording device is something we nailed to a tree, or mounted on a post. For example, a model of sound attenuation used by Dawson and Efford (2009) is the following:

$$9452 \quad S(\mathbf{x}, \mathbf{u}) = \alpha_0 + \alpha_1 d(\mathbf{x}, \mathbf{u}) + \epsilon \quad (9.4.1)$$

9453 where  $\epsilon \sim \text{Normal}(0, \sigma_s^2)$ . In many standard situations,  $S$  will be measured in decibels, which can be any value on the real line. In the conduct of acoustic sampling and the development of custom models for your own situation, it would probably be helpful to know something about sound dynamics and signal processing. In this model, the parameters  $\alpha_0$ ,  $\alpha_1$  and  $\sigma_s^2$  are to be estimated. We abbreviate the set of parameters by  $\boldsymbol{\theta}$  for short.

9454 The basic structure of an acoustic SCR study is not really much different from ordinary  
 9455 SCR studies. Just as ordinary SCR models require that individuals be encountered at  $> 1$   
 9456 trap, these acoustic models require that individuals be heard at  $> 1$  recorder. Therefore,  
 9457 the acoustic signals (calls or vocalizations) must be reconcilable and, in fact, reconciled  
 9458 successfully by the investigator. In practice, this would require associating signals that  
 9459 occur at the same instant with the same individual (or making a decision one way or the  
 9460 other). Further, if individuals are actively moving during the sample period (that recorders  
 9461 are functioning) then individuals might be double-counted, thereby biasing estimates of  
 9462 density. In general, the models produce an estimate of density of sources, and how that is  
 9463 interpreted depends on whether individuals are stationary or mobile, and other things. In  
 9464 particular, if multiple survey occasions are used (e.g., on different days), then modeling  
 9465 movement of individuals would be essential in order to interpret estimates of density  
 9466 meaningfully. Models that allow some movement should be possible (see Sec. 9.4.3 below,  
 9467 and Chaps. 15 and 16).

#### 9472 9.4.1 The signal strength model

9473 We assert that an individual is detected if  $S$  exceeds a threshold,  $c$ . The reason for introducing  
 9474 this threshold  $c$  is that sound recorders will always record some background sound,  
 9475 and so effective use of the acoustic SCR models requires specification of the threshold of  
 9476 measured signal below which the record is censored (non-detection occurs) because the  
 9477 recorded sound is assumed to be background noise. So we assert that an individual is  
 9478 detected if  $S > c$  which occurs with probability  $\Pr(S > c)$ , the encounter probability. To  
 9479 expand on and formalize this, let  $S_{ij}$  be the observed value of  $S$  for animal  $i$  at detector  
 9480  $j$ . The encounter probability is  $\Pr(S_{ij} > c)$  which is  $\Pr(S_{ij} > c) = 1 - \Pr(S_{ij} < c)$ , so  
 9481 that, if we standardize the variate we have

$$1 - \Pr\left(\frac{(S_{ij} - \mathbb{E}(S))}{\sigma_s} < \frac{(c - \mathbb{E}(S))}{\sigma_s}\right)$$

9482 This probability calculation requires evaluation of the CDF of a standard normal variate  
 9483 say,  $\eta = (S_{ij} - \mathbb{E}(S))/\sigma_s$ , being less than  $\gamma(\boldsymbol{\theta}) = (c - \mathbb{E}(S))/\sigma_s$ , which is a function of all  
 9484 the parameters  $\alpha_0$ ,  $\alpha_1$ ,  $\sigma_s^2$  and also the individual location  $\mathbf{u}$  and trap location  $\mathbf{x}$ . We'll  
 9485 identify it by  $\gamma(\boldsymbol{\theta}, \mathbf{x}, \mathbf{u})$  when we need to be explicit about those things. We can compute  
 9486  $\Pr(S_{ij} > c) = 1 - \Pr(\eta < \gamma(\boldsymbol{\theta}, \mathbf{x}, \mathbf{u}))$  easily using any software package including **R** which  
 9487 has a standard function, **pnorm**, for computing the normal cdf. To be more precise, we'll  
 9488 use the **Phi()** to represent the normal cdf. Therefore, an individual is encountered whenever  
 9489  $S_{ij} > c$  which happens with probability  $\Pr(S_{ij} > c) = 1 - \Phi(\gamma(\boldsymbol{\theta}, \mathbf{x}, \mathbf{u}))$ .

9490 Naturally this quantity should depend on *where* an individual is located at the time  
 9491 of recording – what we call it’s instantaneous location, say  $\mathbf{u}$ , to distinguish it from it’s  
 9492 home-range center  $\mathbf{s}$  (but we outline a model below that contains both  $\mathbf{u}$  and  $\mathbf{s}$ ), and  
 9493 also the trap  $\mathbf{x}$ , so we index the quantity  $\gamma$  by those two quantities, in addition to the  
 9494 parameters  $\alpha_0$ ,  $\alpha_1$  and  $\sigma_s$ . The probability of detection is therefore

$$p_{ij} = p(\alpha_0, \alpha_1, \sigma | \mathbf{x}_j, \mathbf{u}_i) = 1 - \Phi(\gamma(\cdot))$$

9495 where  $\mathbf{u}_i$  is the instantaneous location of individual  $i$  and  $\mathbf{x}_j$  is the location of trap  $j$ .  
 9496 We’ll suppose here that the random variables  $\mathbf{u}_i$  have state-space  $\mathcal{U}$ <sup>3</sup>.

9497 How do we interpret this probability? Well, two things have to happen for an individual  
 9498 to be encountered by a trap: (1) it has to vocalize; (2) the microphone has to record a  
 9499 signal  $> c$ . These two things together are a product of biological and environmental factors  
 9500 which could include time of day, wind direction and speed, or maybe rain, humidity and  
 9501 other things. The bottom line is a lot of factors are balled up in whether or not the  
 9502 microphone records a sound greater than the threshold.

9503 The observations from an acoustic survey are the signal strength measurements, and  
 9504 the likelihood of the observed signal strength from individual  $i$  at detection device  $j$  can  
 9505 be specified by noting that the likelihood is the normal pdf for the observed signal *if* the  
 9506 signal strength is  $> c$  and, otherwise, the contribution to the likelihood is  $\Phi(\gamma(\cdot))$  (see Eq.  
 9507 8 of Efford et al. (2009b)):

$$\Pr(S_{ij} | \mathbf{u}_i) = \Phi(\gamma(\cdot))^{1 - I(S_{ij} > c)} \text{Normal}(S_{ij}; \alpha_0, \alpha_1, \sigma_s, \mathbf{x}_j, \mathbf{u}_i)^{I(S_{ij} > c)}$$

9508 We can use this as the basis for constructing the binomial-form of the likelihood as  
 9509 we did in Chapt. 6, which involves the number of individuals not encountered,  $n_0$ . The  
 9510 probability that an individual is *not* captured is equal to the probability that its signal  
 9511 strength doesn’t exceed  $c$  at any microphone. The probability of not being captured at a  
 9512 microphone  $\mathbf{x}_j$  is:

$$1 - p_{\mathbf{u},j} = \Phi(\gamma(\cdot))$$

9513 and therefore the probability of not being captured at any microphone is:

$$\Pr(\text{all } S_{\mathbf{u},j} < c | \mathbf{u}) = \prod_{j=1}^J (1 - p_{\mathbf{u},j}) = \prod_{j=1}^J \Phi(\gamma(\cdot, \mathbf{x}_j, \mathbf{u}))$$

9514 and therefore the marginal probability of not being captured is

$$\pi_0 = [\text{all } S_{\mathbf{u},j} < c | \boldsymbol{\alpha}] = \int_{\mathcal{U}} \left\{ \prod_{j=1}^J \Phi(\gamma(\boldsymbol{\theta}, \mathbf{x}_j, \mathbf{u})) \right\} d\mathbf{u}$$

9515 which can be used to construct the binomial form of the likelihood as we did in Chapt. 6  
 9516 (see Eq. 6.2.1).

---

<sup>3</sup>We use  $\mathcal{U}$  here to avoid confusion with definition of signal strength,  $S$ . However,  $\mathcal{U}$  is the same state-space as  $\mathcal{S}$  in the rest of the book

9517 **9.4.2 Implementation in secr**

9518 Fitting acoustic encounter models in **secr** is no more difficult than other SCR models.  
 9519 There is a handy manual (**secr-sound.pdf**) with examples (Efford and Dawson, 2010)  
 9520 which comes with the **secr** package. The basic process is that **make.capthist** will make a  
 9521 **capthist** object from a 3-dimensional encounter array – which is a binary array indicating  
 9522 whether each individual was detected or not at each recorder/microphone. In the case  
 9523 of signal strength data, **secr** handles the case where # occasions = 1, i.e., the recorders  
 9524 obtained data for a single sample occasion, but this is not a general requirement of the  
 9525 model for signal strength data (see next section). The “signal” attribute of the **capthist**  
 9526 object contains the signal strength in decibels. The best way to include the signal attribute  
 9527 is to use **make.capthist** in the usual way, providing it with the encounter data and  
 9528 trap data and, in addition, the variable “*c*utval” (which is *c* in our notation above) and  
 9529 then provide the signal strength data as an extra column of the **capthist** object. See  
 9530 **?make.capthist** for details.

9531 **9.4.3 Implementation in BUGS**

9532 We don’t know of any Bayesian applications of acoustic SCR models, although we imagine  
 9533 that implementation of such models in the **BUGS** engines should be achievable. It seems  
 9534 easy enough to write down a general hierarchical model that would accommodate sampling  
 9535 on repeated occasions. Let  $\mathbf{s}_i$  be the home range center, and let  $\mathbf{u}_{ik}$  the instantaneous  
 9536 location of individual  $i$  during sample occasion  $k$  (see Chapt. 15 for similar models). The  
 9537 model for  $\mathbf{u}_{ik}$  can be specified conditional on  $\mathbf{s}_i$ . For example, we could assume that  $\mathbf{u}_{ik}$   
 9538 are bivariate normal draws with mean  $\mathbf{s}_i$  and some variance  $\sigma_u^2$ . Then, conditional on  $\mathbf{u}_{ik}$   
 9539 an individual produces a signal according to the signal attenuation model (Eq. 9.4.1), or  
 9540 perhaps some other model. Then we generate the binary encounter data by truncating the  
 9541 observed signal at  $c$ . This general model then is an example of an SCR model in which  
 9542 parameters of a movement model are identifiable (see Sec. 2.6) because there is direct  
 9543 information about movement outcomes from the sampling method, unlike other types of  
 9544 encounter methods (e.g., camera traps) for which animal locations are restricted to a set of  
 9545 fixed, pre-determined points where traps are located. Other types of SCR methods allow  
 9546 for movement information too, including some of the search-encounter models (Chapt.  
 9547 15).

9548 Instead of developing a Bayesian version of this model here, we leave it to the reader  
 9549 to explore simulating data and devising a Bayesian implementation of the acoustic model  
 9550 in one of the **BUGS** engines. Note that for a single occasion, you can simulate the data  
 9551 using the two stage model (having both  $\mathbf{s}$  and  $\mathbf{u}$ ) or you can simulate  $\mathbf{u}$  uniformly without  
 9552 dealing with  $\mathbf{s}$  in the model. The kernel of the **BUGS** model specification should resemble  
 9553 the following snippet:

```
9554 model {
  9555   # Ignoring loops and data augmentation
  9556   u[i,1] ~ dunif(xlim[1], xlim[2])
  9557   u[i,2] ~ dunif(ylim[1], ylim[2])
  9558   mu[i,j] <- alpha0 + alpha1*d[i,j]
  9559   ####
```

---

```

9560  ##### JAGS has this T() truncation feature
9561  S[i,j] ~ dnorm(mu[i,j], 1/sigma^2)T(c,Inf)
9562  #####
9563  gamma[i,j] <- (c - mu[i,j])/sigma
9564  p[i,j] <- 1 - pnorm(gamma[i,j], 0, 1) # JAGS has pnorm() function
9565  y[i,j] ~ dbern(p[i,j])
9566 }
```

#### 9567 9.4.4 Other types of acoustic data

9568 Efford and Dawson (2010) noted that various other types of acoustic data might arise  
 9569 for which SCR-like models would be useful<sup>4</sup>. For example, we could measure the *time of*  
 9570 *arrival* of a vocal queue of some sort at multiple recorders to estimate the number and  
 9571 origin of  $N$  queues. Another example is that where we measure *direction* to a queue from  
 9572 multiple devices and do, effectively, a type of statistical triangulation to the multiple but  
 9573 unknown number of sources. This has direct relevance to types of double or multiple-  
 9574 observer sampling that people do in field studies of birds. Normally 2 observers stand  
 9575 in close proximity and record birds, reconciling their detections after data collection.  
 9576 An SCR-based formulation of the double-observer method has two observers (or more)  
 9577 standing some distance apart, e.g., 50 or 100 meters, and marking individual birds on a  
 9578 map (or at least a direction) and a time of detection. The SCR/double-observer method  
 9579 could be applied to such data.

## 9.5 SUMMARY AND OUTLOOK

9580 In this chapter we extended SCR models to accommodate alternative models for the  
 9581 observation process, including Poisson and multinomial models. Along with the binomial  
 9582 model described in Chapt. 5, this sequence of models will accommodate a substantial  
 9583 majority of contemporary spatial capture-recapture problems, including the 4 main types  
 9584 of encounter data: binary encounters, multinomial trials from “multi-catch” and “single-  
 9585 catch” (Efford, 2004, 2011a; Royle and Gardner, 2011) trap systems, and Poisson encounter  
 9586 frequency data from devices that can record multiple encounters of the same individual  
 9587 at a device. We summarize the standard observation models and the corresponding **secr**  
 9588 terminology in Table 9.4. What we refer to as search-encounter (or area-search) models  
 9589 (see Chapt. 15) are distinct from most of the other classes in that the observation location  
 9590 can also be random (in contrast to traps, where the location is fixed by design). This  
 9591 auxiliary data is informative about an intermediate process related to movement (Royle  
 9592 and Young, 2008).

9593 There is a need for other types of encounter models that arise in practice. We identify  
 9594 a few of them here, although we neglect a detailed development of them at the present  
 9595 time or, in some cases, put that off until later chapters: (1) Removal systems – Sometimes  
 9596 traps kill individuals and SCR models can handle that. This can be viewed as a kind of  
 9597 open model, with mortality only, and we handle such models (in part) in Chapt. 16; (2)  
 9598 There are models for which only specific summary statistics are observable (Chandler and

---

<sup>4</sup>Some of the following is also related to material presented by D.L. Borchers at the ISEC 2012 conference in Norway.

**Table 9.4.** Different observation models, where we discuss them in this book, and what the corresponding `secr` terminology is

observation model	Where in this book?	<code>secr</code> name
Bernoulli	Chapt. 5	<code>proximity</code>
Poisson	Sec. 9.1	<code>count</code>
Multinomial (ind)	Sec. 9.2	<code>multi-catch</code>
Multinomial (dep)	Sec. 20.2.8	<code>single-catch</code>
Acoustic	Sec. 9.4	<code>signal</code>
Search-encounter	Chapt. 15	<code>polygon</code> (in part)

9599 Royle, 2013; Sollmann et al., 2013) which we cover in Chaps. 18 - 19; (3) We can have  
 9600 multiple observation methods working together as in Gopalaswamy et al. (2012b).

9601 There remains much research to be done to formalize models for certain observation  
 9602 systems. For example, while we think one will usually be able to analyze single-catch  
 9603 systems using the multi-catch model, or even the Bernoulli model if encounter probability  
 9604 is sufficiently low, a formalization of the single-catch model would be a useful development  
 9605 and, we believe, it should be achievable using one or another of the **BUGS** engines. In  
 9606 addition, classical “trapping webs” (Anderson et al., 1983; Wilson and Anderson, 1985a;  
 9607 Jett and Nichols, 1987; Parmenter and MacMahon, 1989; Link and Barker, 1994) have  
 9608 been around for quite some time and it seems like they are amenable to formulation as  
 9609 a type of SCR model although we have not pursued that development simply because  
 9610 trapping webs are rarely used in practice.

9611  
9612

# 10

9613

## SAMPLING DESIGN

9614 Statistical design is recognized as an important component of animal population studies  
9615 (Morrison et al., 2008; Williams et al., 2002). Many biologists have probably been in a  
9616 situation where some problem with their data could be traced back to a flaw in study  
9617 design. Commonly, design is thought of in terms of number of samples to take, when to  
9618 sample, methods of capture, desired sample size (of individuals), power of tests, and related  
9619 considerations. In the context of spatial sampling problems, where populations of mobile  
9620 animals are sampled by an array of traps or devices, there are a number of critical design  
9621 elements. Two of the most important ones are the spacing and configuration of traps  
9622 (or sampling devices) within the array. For traditional capture-recapture, conceptual and  
9623 heuristic design considerations have been addressed by a number of authors (e.g., Nichols  
9624 and Karanth, 2002, Chapt. 11), but little formal analysis focused on spatial design of  
9625 arrays has been carried out. Bondrup-Nielsen (1983) investigated the effect of trapping  
9626 grid size (relative to animal home range area) on capture-recapture density estimates  
9627 using a simulation study and some authors have addressed trap spacing and configuration  
9628 by sensitivity “re-analysis” (deleting traps and reanalyzing; Wegge et al., 2004; Tobler  
9629 et al., 2008). The scarcity of simulation-based studies looking at study design issues is  
9630 surprising, as it seems natural to evaluate prescribed designs by Monte Carlo simulation in  
9631 terms of their accuracy and precision. In the past few years, however, a growing number of  
9632 simulation studies addressing questions of study design in the context of spatial capture-  
9633 recapture have come out (e.g., Marques et al. (2011); Sollmann et al. (2012); Efford and  
9634 Fewster (2012); Efford (2011b)), the results of which we will discuss throughout this  
9635 chapter.

9636 In this chapter we recommend a general framework for evaluating design choices for  
9637 SCR studies using Monte Carlo simulation of specific design scenarios based on trade-offs  
9638 between available effort, funding, logistics and other practical considerations – what we  
9639 call *scenario analysis*. Many study design related issues can be addressed with preliminary  
9640 field studies that will give you an idea of how much data you can expect to collect with a  
9641 unit of effort (a camera trap day or a point count survey, for example). But it is also always  
9642 useful to perform scenario analysis based on simulation before conducting the actual field

9643 survey not only to evaluate the design in terms of its ability to generate useful estimates,  
9644 but also so that you have an expectation of what the data will look like as they are being  
9645 collected. This gives you the ability to recognize some pathologies and possibly intervene  
9646 to resolve issues before they render a whole study worthless. Suppose you design a study  
9647 to place 40 camera traps based on your expectations of parameter values you obtained  
9648 from a careful review of the literature, and simulation studies suggest that you should  
9649 get 3-5 captures of individuals per night of sampling. In the field you find that you're  
9650 realizing 0 or 1 captures per night and therefore you have the ability to sit down and  
9651 immediately question your initial assumptions and possibly take some remedial action in  
9652 order to salvage your project, your PhD thesis and, hopefully, your career. Simulation  
9653 evaluation of design *a priori* is therefore a critical element of any field study.

9654 While we recommend scenario analysis as a general tool to understand your *expected*  
9655 *data* before carrying out a spatial capture-recapture study, it is possible to develop some  
9656 heuristics and even analytic results related to the broader problem of model-based spatial  
9657 design (Müller, 2007) using an explicit objective function based on the inference objective.  
9658 We outline an approach in this chapter where we identify a variance criterion, namely, the  
9659 variance of an estimator of  $N$  for the prescribed state-space. We show that this depends  
9660 on the configuration of trap locations, and we provide a framework for optimizing the  
9661 variance criterion over the design space (the collection of all possible designs of a given  
9662 size). While there is much work to be done on developing this idea, we believe that it  
9663 provides a general solution to any type of design problem where the space of candidate  
9664 trap locations is well-defined.

## 10.1 GENERAL CONSIDERATIONS

9665 Many biologists have experience with the design of natural resource surveys from a classical  
9666 perspective (Thompson, 2002; Cochran, 2007), a key feature of which involves sampling  
9667 space. That is, we identify a sample frame comprised of spatial units and we sample  
9668 those units randomly (or by some other method, such as generalized random tessellation  
9669 stratified (GRTS) sampling (Stevens Jr and Olsen, 2004)) and measure some attribute.  
9670 The resulting inference applies to the attribute of the sample frame. There are some  
9671 distinct aspects of the design of SCR studies which many people struggle with in their  
9672 attempts to reconcile SCR design with classical survey design problems. We discuss some  
9673 of these here.

### 9674 10.1.1 Model-based not design-based

9675 Inference in classical finite-population sampling is usually justified by “design-based” ar-  
9676 guments. This means that properties of estimators (bias, variance) are evaluated over  
9677 realizations of the *sample*. The sample is random, but the attribute being observed is not,  
9678 for the specific sample that is chosen. For example, imagine we have a landscape gridded  
9679 off into 900 1 km  $\times$  1 km grid cells, from which we draw a sample of 100 to measure an  
9680 attribute such as “percent developed” which we aim to use in a habitat model. In the  
9681 classical design-based view, the attribute (percent developed) is a static quantity for each  
9682 of the 900 grid cells and theory tells us that, by taking a random sample, we can expect to  
9683 obtain estimators (e.g., of the mean of all 900 grid cells) with good statistical properties,

9684 where the expectation is with respect to the sample of 100 grid cells. For example, if we  
9685 repeatedly draw samples of size 100 then, over many such samples, the expected value of  
9686 the estimator may be unbiased. Classical design-based sampling does not tell us anything  
9687 about the specific 100 sample units that we obtained in our sample. However, in the SCR  
9688 modeling framework, properties of our estimators are distinctly model-based. We evaluate  
9689 estimators (usually) or care only about a *fixed* sample of spatial locations, averaged over  
9690 realizations of the underlying process and data we might generate. Although sometimes  
9691 we might condition on the data for purposes of inference (if we have our Bayesian hat on),  
9692 the probability model for the data is fundamental to inference, and the spatial sample of  
9693 trap locations is always fixed.

9694 **10.1.2 Sampling space or sampling individuals?**

9695 A fundamental question in any sampling problem is what is the sample frame – or the  
9696 population we are hoping to extrapolate too. In the context of capture-recapture studies,  
9697 it is tempting to think of the sample frame as being spatial (the space within “the study  
9698 area”, tiled into quadrats perhaps). Clearly SCR models involve a type of spatial sampling  
9699 – we have to identify spatial locations for traps, or arrays of traps. However, unlike  
9700 conventional natural resource sampling the attribute we measure is *not* directly relevant  
9701 to the *sample location*, such as where we place a trap and, therefore, it may not be  
9702 sensible to think of the sample frame as being comprised of spatial units. On the other  
9703 hand, capture-recapture studies clearly obtain a sample of *individuals* and SCR models are  
9704 models of *individual* encounter and space use. Therefore, it is more natural to think of the  
9705 sample frame as a list of  $N$  individuals, determined by the definition of the state-space,  
9706 or a subset of the state-space, i.e., the study-area, but the number  $N$  is unknown. The  
9707 purpose of the SCR study is to draw a sample of these  $N$  individuals and learn about an  
9708 individual attribute – namely, where that individual lives. *That* is the sampling context of  
9709 SCR models. SCR models link the observed data (encounter histories) to this individual  
9710 attribute via a model (with parameters) which we need to “fit”. Once we fit that model,  
9711 we usually use it to make a prediction or estimate of the attribute for individuals that did  
9712 not appear in the sample.

9713 Spatial sampling in SCR studies is important, but only as a device for accumulating  
9714 individuals in the sample from which we can learn about their inclusion probability. That  
9715 is, we’re not interested in any sample unit attribute directly but, rather, we use spatial  
9716 units as a means for sampling individuals and obtaining individual level encounter histo-  
9717 ries. It makes sense in this context that we should want to choose a set of spatial sample  
9718 units that provides an adequate sample size of individuals, perhaps as many as possible.  
9719 The key technical consideration as it relates to spatial sampling and SCR is that arbitrary  
9720 selection of sample units has a side-effect that it induces unequal probabilities of inclusion  
9721 into the sample (i.e., an individual exposed to more traps is more likely to be included  
9722 into the sample than an individual exposed to few traps) and so we must also learn about  
9723 these unequal probabilities of sample inclusion as we obtain our sample.

9724 The fact that SCR sampling induces unequal probabilities of sampling is consistent  
9725 with the classical sampling idea of Horvitz-Thompson estimation which has motivated  
9726 capture-recapture models similar to SCR (Huggins, 1989; Alho, 1990). In the Horvitz-  
9727 Thompson framework, the sample inclusion probabilities are usually fixed and known.

9728 However, in all real animal sampling problems they are unknown because we never know  
9729 precisely where each individual lives and therefore cannot characterize its encounter prob-  
9730 ability. Therefore, we have to estimate the sample inclusion probabilities using a model.  
9731 SCR models achieve this effect formally, using a fully model-based approach based on a  
9732 model that accounts for the organization of individual activity centers and trap locations.  
9733 This notion of Horvitz-Thompson estimation suggests that perhaps we should consider  
9734 designing SCR studies based on the Horvitz-Thompson variance estimator as a design  
9735 criterion. We discuss this a little bit later in this chapter.

9736 **10.1.3 Focal population vs. state-space**

9737 In SCR models we make a distinction between the focal population – the population of  
9738 individuals we care about – and those of the state-space, which we are required to prescribe  
9739 in order to fit SCR models. These are not the same thing. The geographic scope of the  
9740 population of inference is the region within which animals live that you care about in your  
9741 study – let's call this “the study area”. This is often prescribed for political reasons or  
9742 legal reasons (e.g. a National Park). To initiate a study, or perhaps motivating the study,  
9743 you have to draw a line on a map to delineate a study area, although often it is difficult  
9744 to draw this line, and where you draw it is not so much a statistical/SCR issue. On the  
9745 other hand, you need to prescribe the state-space to define and fit an SCR model. This  
9746 is the region that contains individuals that you *might* capture. This is different from the  
9747 study area in most cases. To design a study, you need a well-defined study area, but the  
9748 state-space will also be relevant to efficient distribution of traps, and other considerations.

9749 It is helpful to think about this distinction operationally. We define our study area *a priori*.  
9750 As a conceptual device, we might think of this as the area that, given an infinite  
9751 amount of resources, we might wall off so that we can study a real closed population.  
9752 This “study area” should exist independent of any model or estimator of some population  
9753 quantity, i.e., the subject-matter context should determine what the study area is. Given  
9754 a well-defined study area, we use some method to arrange data collecting devices within  
9755 this study area. The method of arrangement can be completely arbitrary but, naturally,  
9756 we want to choose arrangements of traps that are better in terms of obtaining statistical  
9757 information from the data we wind up collecting.

9758 Lets face it – it's quite a nuisance that animals move around and this makes the idea of  
9759 a spatial study area kind of meaningless in terms of management in most cases. Wherever  
9760 you draw a line on a map, there will be animals who live mostly beyond that line that will  
9761 sometimes be subjected to observation in your study. One of the benefits of SCR models  
9762 is they formalize the exposure and contribution of these individuals to your study. That  
9763 is a good thing. Thus, you can probably be a bit sloppy or practical in your definition of  
9764 “the study area” and not worry too much.

9765 With these general concepts of spatial sampling and the sampling of individuals in  
9766 mind, we can now turn our attention to more specific aspects of study design in SCR  
9767 surveys, namely the spatial arrangement of detectors. We discuss some general concepts,  
9768 and then focus on a couple of specific case studies that apply to the Bernoulli observation  
9769 model or passive detection devices. The general concepts are surely relevant to other SCR  
9770 models, and we suspect that the specific case studies are relevant as well.

## 10.2 STUDY DESIGN FOR (SPATIAL) CAPTURE-RECAPTURE

9771 The importance of adequate trap spacing and overall configuration of the trapping array  
9772 has long been discussed in the capture-recapture literature. A heuristic based on recog-  
9773 nizing the importance of typical home range sizes (Dice, 1938, 1941) and thus being able  
9774 to obtain information about home range size from the trap array is that traps should  
9775 be spaced such that the array of available traps exposes as many individuals as possible  
9776 but, at the same time, individuals should be captureable in multiple traps. Thus, good  
9777 designs should generate a high sample size  $n$  (i.e., the number of individuals captured)  
9778 and a large number of spatial recaptures. These two considerations form a trade-off in  
9779 building designs. On one hand, having a lot of traps very close together should produce  
9780 the most spatial recaptures but produce very few unique individuals captured (assuming  
9781 that studies are limited in the total number of sampling devices they can deploy). On the  
9782 other hand, spreading the traps out as much as possible, in a nearly systematic or regular  
9783 design, should yield the most unique individuals, but probably few spatial recaptures. We  
9784 will formalize this trade-off later, when we consider formal model-based design of SCR  
9785 studies.

9786 Traditional CR models require that all individuals in the study area have a probability  
9787  $> 0$  of being captured, which means that the trap array must not contain “holes” large  
9788 enough to contain an animal’s entire home range (Otis et al., 1978). The reason why  
9789 “holes” cause a problem in non-spatial models is that they induce heterogeneity in capture  
9790 probability. If an animal’s home range lies in or partially in a hole, then it will have a  
9791 different probability of being captured than an individual whose home ranges is peppered  
9792 with traps. As a consequence, trap spacing is recommended to be on the same order  
9793 as the radius of a typical home range (e.g., Dillon and Kelly (2007)). For example,  
9794 imagine a camera trap study implemented in South America with the objective to survey  
9795 populations of both jaguars (*Panthera onca*) and the much smaller ocelots (*Leopardus*  
9796 *pardalis*). Ocelots also have much smaller home ranges and therefore should require closer  
9797 trap spacing than the large wide-ranging jaguars. The “no holes” assumption entails  
9798 some strong restrictions with respect to study design. Although we need not cover an  
9799 area systematically with traps, there has to be some consistent coverage of the entire  
9800 area of interest. Often, this is achieved by dividing the study area into grid cells, the  
9801 size of which approximates an average home range (or possibly the smallest home range  
9802 recorded for the study species in the study area or a similar area; e.g. Wallace et al.  
9803 (2003)), and then place (at least) one trap within each cell. In many field situations,  
9804 especially when dealing with large mammals and accordingly large study areas, achieving  
9805 this consistent coverage can be extremely challenging or even impossible. Depending on  
9806 local environmental conditions, parts of the study area can be virtually inaccessible to  
9807 humans, because of dense vegetation cover, or unsuitable for setting up detectors, because  
9808 of flooding. Even when accessible, setting up traps in difficult habitat conditions can  
9809 consume disproportional amounts of time, manpower and other resources. Moreover, even  
9810 when the trap spacing does not result in holes, the problem of spatial heterogeneity in  
9811 capture probability will still exist because individuals with home ranges near the borders  
9812 of the trap array will have a different probability of being captured than individuals that  
9813 spend all their time within the trap array.

9814 Where approaches such as MMDM (mean maximum distance moved) are used in  
9815 combination with traditional CR models to obtain density estimates (see Chapt. 4), trap

spacing also has a major effect on movement estimates, since it determines the resolution of the information on individual movement (Parmenter et al., 2003; Wilson and Anderson, 1985a). If trap spacing is too wide, there is little or no information on animal movement because most animals will only be captured at one trap (Dillon and Kelly, 2007). In addition, only a trapping grid that is large relative to individual movement can capture the full extent of such movements, and researchers have suggested that the grid size should be at least four times that of individual home ranges to avoid positive bias in estimates of density (Bondrup-Nielsen, 1983). This recommendation originated in small mammal trapping, and it should be relatively easy to follow when dealing with species covering home ranges < 1 ha. However, translated to large mammal research, this can entail having to cover several thousands of square kilometers – a logistical and financial challenge that few projects could realistically tackle.

Though closely related, the requirements in terms of spatial study design for SCR models differ distinctly from those for traditional CR. For one, holes in the study area are of no concern in SCR studies. As a practical matter, some animals within the study area might have vanishingly small probability of being included in the sample, i.e.,  $p \approx 0$ . The nice thing about SCR models is that  $N$  is explicitly tied to the state-space, and not the traps which expose animals to encounter. Within an SCR model, extending inference from the sample to individuals that live in these holes represents an extrapolation (prediction of the model outside the range of the data), but one that the model is capable of producing because we have explicit declarations, in the model, that it applies to any area within the state-space (the state-space is a part of the model!), even to areas where we can't capture individuals because we happened to not put a trap near them. Conversely, ordinary capture-recapture models only apply to individuals that have encounter probability that is consistent with the model being considered. Presumably, the existence of a hole in the trap array would introduce individuals with  $p = 0$ , which is not accommodated in those models. This alone allows for completely new and much more flexible study designs in SCR studies, as compared to traditional CR, such as linear designs, “hollow grids” (detectors trace the outline of a square), or small clusters of grid spread out over larger landscapes (Efford et al., 2005, 2009a; Efford and Fewster, 2012).

Whereas traditional CR studies are concerned with the number of individuals and recaptures and with satisfying the model assumption of all individuals having some probability of being captured, in spatial capture-recapture we are looking at an additional level of information: We need spatially dispersed captures and recaptures. It is not enough to recapture an individual – we need to recapture at least some individuals at several traps. Therefore, in general, design of SCR studies boils down to obtaining three bits of information: total unique individuals captured, total number of recaptures informative about baseline encounter rate, and spatial recaptures, informative about  $\sigma$ .

Most SCR design choices wind up trading these three things against each other to achieve some optimal (or good) mix. So, for example, if we sample a very small number of sites a huge number of times then we can get a lot of recaptures but only very few spatial ones, and few unique individuals etc. This need for spatial recaptures may appear as an additional constraint on study design, but actually, SCR studies are much less restricted than traditional CR studies, because of the way animal movement is incorporated into the model:  $\sigma$  is estimated as a specified function of the ancillary spatial information collected in the survey and the capture frequencies at those locations. This function is able to

make a prediction across distances even when these are latent, including distances larger than the extent of the trap array. When there is enough data across at least some range of distances, the model will do well at making predictions at unobserved distances. The key here is that there needs to be ‘enough data across some range of distances’, which induces some constraint on how large our overall trap array must be to provide this range of distances (e.g., Marques et al., 2011; Efford, 2011b). We will review the flexibility of SCR models in terms of trap spacing and trapping grid size in the following section.

### 10.3 TRAP SPACING AND ARRAY SIZE RELATIVE TO ANIMAL MOVEMENT

Using a simulation study, Sollmann et al. (2012) investigated how trap spacing and array size relative to animal movement influence SCR parameter estimates and we will summarize their study here. They simulated encounter histories on an  $8 \times 8$  trap array with regular spacing of 2 units, using a binomial observation model with Gaussian hazard encounter model, across a range of values for the scale parameter  $\sigma^*$ . We refer to the scale parameter as  $\sigma^*$  here, because Sollmann et al. (2012) use a slightly different parametrization of SCR models, in which  $\sigma^*$  corresponds to  $\sigma \times \sqrt{2}$ .

In Sec. 5.4 we pointed out that under the Gaussian (or half-normal) detection model  $\sigma$  can be converted into an estimate of the 95% home range or “use area” around  $s_i$ . Based on this transformation, values for  $\sigma^*$  were chosen so that there was a scenario where the trap array was smaller than a single individual’s home range, i.e. trap spacing was small relative to individual movements ( $\sigma^* = 5$ ), a scenario where spaces between traps were large enough to contain entire home ranges ( $\sigma^* = 0.5$ ), and two intermediate scenarios and where sigma was smaller ( $\sigma^* = 1$  unit) and larger ( $\sigma^* = 2.5$  units) than the trap spacing, respectively.  $N$  was 100, the baseline trap encounter rate  $\lambda_0$  was 0.5 (on the cloglog scale) for all four scenarios and trap encounters were generated over 4 occasions. Table 10.1 shows the results as the average over 100 simulations.

All model parameters were estimated with relatively low bias (< 10%) and high to moderate precision (relative root mean squared error, RRMSE < 25%) for all scenarios of  $\sigma^*$ , except  $\sigma^* = 0.5$  units, under which model parameters were mostly not estimable (therefore excluded from Table 10.1). Data for the latter case mostly differed from the other scenarios in that fewer animals were captured and very few of the captured animals were recorded at more than 1 trap (Table 10.2). For  $\sigma^* = 0.5$ , abundance ( $N$ ) was not estimable in 88% of the simulations, and when estimable, was underestimated by approximately 50%. This shows that a wide trap spacing that is considerably too large relative to animal movement may be problematic in SCR studies.

Estimates (posterior means) of  $N$  were least biased and most precise under the  $\sigma^* = 2.5$  scenario, and in general, all parameters were estimated best under the  $\sigma^* = 2.5$  or the  $\sigma^* = 5$  scenario. All estimates had the highest relative bias and the lowest precision under the  $\sigma^* = 1$  scenario. These results clearly demonstrate that SCR models can successfully handle a range of trap spacing to animal movement ratios, and even when using a trapping array smaller than an average home range: at  $\sigma^* = 5$ , the home range of an individual was approximately 235 units<sup>2</sup>, while the trapping grid only covered 196 units<sup>2</sup>. Still, the model performed very well.

An important consideration in this simulation study is that all but the  $\sigma^* = 0.5$  units

**Table 10.1.** Mean, relative root mean squared error (RRMSE) of the mean, mode, 2.5% and 97.5% quantiles, relative bias of mean (RB) and 95% Bayesian credible interval (BCI) coverage for spatial capture-recapture parameters across 100 simulations for four simulation scenarios, define by the input value of movement parameter  $\sigma^*$ .  $N$  = number of individuals in the state space;  $\lambda_0$  = baseline trap encounter rate.

Scenario	Mean	rrmse	Mode	2.5%	97.5%	RB	BCI
$\sigma^* = 1 (\sigma = 0.71)$							
$N$	108.497	0.172	104.099	78.977	143.406	0.085	96
$\lambda_0$	0.518	0.248	0.477	0.303	0.752	0.035	94
$\sigma^*$	1.008	0.093	0.990	0.857	1.195	0.008	94
$\sigma^* = 2.5 (\sigma = 1.77)$							
$N$	100.267	0.105	98.456	82.086	121.878	0.003	97
$\lambda_0$	0.507	0.118	0.500	0.409	0.623	0.014	92
$\sigma^*$	2.501	0.046	2.491	2.267	2.690	< 0.001	92
$\sigma^* = 5 (\sigma = 3.54)$							
$N$	102.859	0.137	100.756	77.399	130.020	0.029	88
$\lambda_0$	0.505	0.075	0.501	0.435	0.580	0.011	93
$\sigma^*$	5.023	0.039	5.001	4.687	5.431	0.005	97

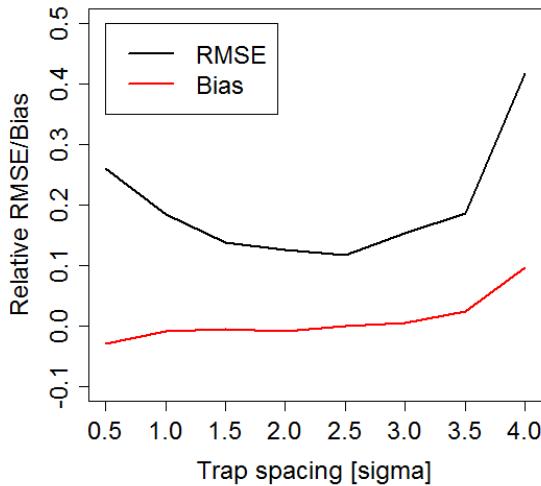
**Table 10.2.** Summary statistics of 100 simulated data sets for four simulation scenarios, defined by the input value of movement parameter  $\sigma^*$ . Individual detection histories were simulated on an  $8 \times 8$  trap array with regular trap spacing of 2 units.

Scenario	Inds. captured	Total captures	Inds. recaptured	Inds. captured at > 1 trap
$\sigma^* = 0.5$	18.29 (3.84)	25.38 (5.86)	5.52 (2.03)	0.72 (0.95)
$\sigma^* = 1.0$	37.70 (13.44)	69.35 (26.05)	19.48 (7.68)	11.87 (5.43)
$\sigma^* = 2.5$	44.19 (4.67)	231.78 (33.98)	36.60 (4.76)	35.21 (4.73)
$\sigma^* = 5.0$	40.51 (5.15)	427.77 (79.09)	33.09 (4.63)	32.60 (4.76)

9904 scenarios provided reasonably large amounts of data, including 20+ individuals being  
9905 captured on the trapping grid. When dealing with real-life animals that are often territorial  
9906 and may have lower trap encounter rates, a very small grid compared to an individual's  
9907 home range may result in the capture of few to no individuals. In that case, the sparse  
9908 data will limit the ability of the model to estimate parameters (Marques et al., 2011),  
9909 which is true of most models.

9910 To further explore the effects of trap spacing and movement on bias and precision of  
9911 estimates of  $N$ , we expanded the simulation study of Sollmann et al. (2012): we considered  
9912 a regular 7 grid, with trap spacing ranging from  $0.5 \times \sigma$  to  $4 \times \sigma$ , with a state-space that  
9913 had variable size so that the buffer around the traps was constant in units of  $\sigma$ . For  
9914 each trap spacing scenario we simulated and analyzed 500 data sets and calculated the  
9915 RRMSE and relative bias for the estimates of  $N$ . Figure 10.1 shows the results of this set  
9916 of simulations. We see that there is clearly an optimal trap spacing, especially in terms  
9917 of precision, which is highest at a trap spacing of  $1.5 - 2.5 \times \sigma$ . Efford (2012) reported  
9918 similar results and highlighted the trade-off between the number of individuals captured  
9919 and the number of spatial recaptures – intuitively, the former goes up with an increase in

trap spacing, whereas the latter goes down. In summary, in small trap spacing scenarios, the small sample size leads to imprecise estimates, whereas in large trap spacing scenarios, lack of spatial recaptures leads to imprecise and biased estimates.



**Figure 10.1.** Relative bias and RRMSE of estimates of  $N$  from an SCR model for a range of trap spacing scenarios.

### 10.3.1 Black bears from Pictured Rocks National Lakeshore

To see how trap array size influences parameter estimates from spatial capture-recapture models in the real world, Sollmann et al. (2012) also looked at a black bear data set from Pictured Rocks National Lakeshore, Michigan, collected using 123 hair snares distributed over an area of  $440 \text{ km}^2$  along the shore of Lake Superior in May-July 2005 (Belant et al., 2005). The SCR model for the bear data included sex-specific encounter rate parameters, and an occasion-specific baseline encounter rate. This was motivated by a) the lower average number of detections for male bears, b) the decreasing number of detections over time in the raw data, and c) the fact that male black bears are known to move over larger areas than females (e.g., Gardner et al., 2010b; Koehler and Pierce, 2003).

To address the impact of a smaller trap array on the parameter estimates, models fitted to the full data set were compared to models fitted to data subsets. The first subset retained only those 50% of the traps closest to the grid center. In the second, only the southern 20% of the traps were retained 10.3.

Reducing the area of the trap array by 50% created a grid polygon of  $144 \text{ km}^2$ , which was smaller than an estimated male black bear home range and only 50% larger than a

**Table 10.3.** Posterior summaries of SCR model parameters for black bears, modified from Sollmann et al. (2012).

	Mean (SE)	Mode	2.5%	97.5%
<b>Full data set</b>				
$D$	10.556 (1.076)	10.448	8.594	12.792
$\sigma^*$ (males)	7.451 (0.496)	7.323	6.579	8.495
$\sigma^*$ (females)	2.935 (0.143)	2.939	2.671	3.226
<b>50% of traps</b>				
$D$	12.648 (1.838)	12.205	9.307	16.713
$\sigma^*$ (males)	5.354 (0.511)	5.248	4.472	6.473
$\sigma^*$ (females)	3.318 (0.277)	3.262	2.841	3.910
<b>20% of traps</b>				
$D$	6.752 (1.611)	5.953	4.000	10.218
$\sigma^*$ (males)	9.881 (3.572)	7.566	5.121	18.447
$\sigma^*$ (females)	2.686 (0.391)	2.657	2.121	3.404

9939 female black bear home range – approximately 260 km<sup>2</sup> and 100 km<sup>2</sup>, respectively, when  
 9940 converting estimates of  $\sigma^*$  to home range size. Table 10.3 shows that this did not greatly  
 9941 influence model results, compared to the full data set.

9942 Removing 80% of the traps and thereby reducing the area of the trap array to 64  
 9943 km<sup>2</sup> – well below the average black bear home range – had a great effect on sample size  
 9944 (only 25 of the original 83 individuals sampled) and parameter estimates. Particularly,  
 9945 male black bear movement was overestimated and imprecise. The combination of the low  
 9946 baseline trap encounter rate of males and the considerable reduction in sample size led to  
 9947 a low level of information on male movement: 5 of the 12 males were captured at one trap  
 9948 only. Although they moved over smaller areas, owing to their higher trap encounter rate,  
 9949 females were, on average, captured at more traps (3.4 traps per individual compared to 2.6  
 9950 for males) so that their movement estimate remained relatively accurate. Overestimated  
 9951 male movements and female trap encounter rates resulted in an underestimate of density  
 9952 of almost 40%. This effect is contrary to what we would expect to see in non-spatial  
 9953 CR models, where a trapping grid that is small relative to animal movement leads to  
 9954 underestimated movement (MMDM) and overestimated density (Bondrup-Nielsen, 1983;  
 9955 Dillon and Kelly, 2007; Maffei and Noss, 2008). While this example again demonstrates  
 9956 the ability of SCR models to deal with a range of trapping grid sizes, it also clearly shows  
 9957 that your study design needs to consider the amount of data you can expect to collect.  
 9958 As an alternative to simulation studies, Efford et al. (2009b) provide a mathematical  
 9959 procedure to determine the expected number of individuals captured and recaptures for a  
 9960 given detector array and set of model parameters.

## 10.4 SAMPLING OVER LARGE AREAS

9961 Trap spacing is an essential aspect of design of SCR studies. However, it is only the most  
 9962 important aspect if one can uniformly cover a study area with traps. In many practical  
 9963 situations, where the study area is large relative to effort that can be expended, one has  
 9964 to consider other strategies which deviate from a strict focus on trap spacing. There are  
 9965 two general strategies that have been suggested for sampling large areas which we think

9966 are useful in practice, either by themselves or combined: Sampling based on *clusters* of  
9967 traps and sampling based on *rotating* groups of traps over the landscape.

9968 Karanth and Nichols (2002) describe 3 strategies for moving traps to achieve coverage  
9969 of a larger study area, geared toward traditional capture-recapture analysis. Suppose that  
9970 sampling the entire area of interest requires sampling  $G$  sites, then the 3 strategies are:

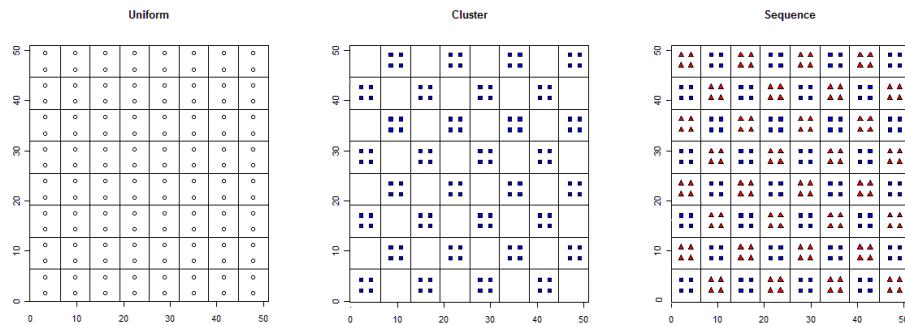
- 9971 (1) For every day/sampling occasion, randomly choose  $x$  out of your  $G$  sites, where  $x$  is  
9972 the number of trapping devices you have at hand. Obviously, this requires that it be  
9973 relatively easy to move traps around.
- 9974 (2) Move blocks of traps that are close to each other in space daily. For example, if you  
9975 divide your total study area into 4 blocks, sample block 1 for a day, then move traps  
9976 to block 2 for a day, and so forth, and repeat until each block has been sampled for a  
9977 sufficient amount of time.
- 9978 (3) If moving blocks of traps daily is too challenging logistically, then you can sample  
9979 each block for a certain number of days/occasions before moving cameras to the next  
9980 block. In this fashion, you only need to move traps to each block once.

9981 In traditional CR we collapse data across traps and assume all individuals in the study  
9982 area have some probability  $> 0$  of being detected. For our data that means that, under  
9983 scenario (2) the first occasion is defined as the time it takes to sample all 4 blocks once, the  
9984 second occasion consists of the second round of sampling all blocks, etc. Under scenario  
9985 (3), we have to combine data from day 1 in each of the blocks to form occasion 1, data  
9986 from day 2 in each of the blocks forms occasion 2, and so on. Especially scenario 3 makes  
9987 modeling time-dependent detection difficult, since occasion 1 does no longer refer to an  
9988 actual day or continuous time interval. We do not have that problem in SCR, where  
9989 accounting for sampling effort at each trap is straight forward, as we first demonstrated  
9990 for the wolverine example in Sec. 5.9. Because we are dealing with detection at the trap  
9991 level, even for design (3) in a spatial framework, we can still look at variation in detection  
9992 over time. As such, we don't think that one of the above designs is superior for SCR  
9993 models than the other, but rather, all of them may produce adequate SCR data, as long  
9994 as overall sample size requirements are met.

9995 Efford and Fewster (2012) looked at the performance of different spatial study designs  
9996 for abundance estimation from traditional and spatial capture-recapture models, including  
9997 a clustered design, where groups of detectors are spaced throughout the larger region of  
9998 interest. They found that in a spatial framework this design performed well, although  
9999 there were indications of a slight positive bias in estimates of  $N$ . Such a clustered design  
10000 enables researchers to increase area coverage without having to increase the number of  
10001 traps. Efford and Fewster (2012) note that distribution of clusters has to be spatially  
10002 representative – for example, systematic with a random origin. The issue of spatially  
10003 representative designs is not limited to SCR and an extensive treatment of the topic can  
10004 be found in the distance sampling literature (Buckland et al., 2001). Further, the authors  
10005 stress that, if distances among clusters are large and individuals are unlikely to show up  
10006 in several clusters, then the method relies on spatial recaptures *within* clusters, meaning  
10007 that spacing of detectors within clusters has to be appropriate to the movements of the  
10008 species under study. A clustered type of design is also suggested by Efford et al. (2009b)  
10009 for acoustic detectors (see Chapt. 9.4) with small groups of such detectors (e.g.,  $2 \times 2$ )  
10010 being distributed in a probabilistic fashion across the region of interest.

10011 In practice, employing both of these strategies – clustering and rotating traps – might  
 10012 be necessary or advantageous. Sun (in prep) used a simulation study to investigate different  
 10013 trap arrangements (Fig. 10.2) for a black bear study based on hair snares distributed  
 10014 over a 2625-km<sup>2</sup> study area. She simulated detection data of bears for 3 trap arrangements  
 10015 including a regular (uniform) coverage of traps, clusters of 4 traps each with a gap between  
 10016 clusters, and a design in which the clusters were moved mid-way through the study to fill  
 10017 the gap (a sequential or “rotating” design). She found that the precision and accuracy of  
 10018 estimates of  $N$  generally decreased when changing from a uniform to a clustered to a ro-  
 10019 tating design, although the loss of efficiency was relatively small when using the clustered  
 10020 design. The result seems to support that cluster designs can be effective with relatively  
 10021 little loss of efficiency.

10022 Further research on optimal detector configurations, especially for large scale studies, is  
 10023 called for (Efford and Fewster, 2012). More generally, work on formalizing and generalizing  
 10024 these ideas of spatial study design is needed. We believe the model-based spatial design  
 10025 approach that we introduce below is one possible way to do that.



**Figure 10.2.** Three designs evaluated by Sun (in prep). The left panel shows uniform coverage of the area with traps (hair snares) equally spaced and static for the duration of the period. The central panel shows clusters of 4 traps in close proximity, with larger gaps between clusters. The right panel shows a design in which all grid cells are sampled by the cluster of 4 traps, but in a sequential (in time) manner.

## 10.5 MODEL-BASED SPATIAL DESIGN

10026 A point we have stressed in previous chapters is that SCR models are basically glorified  
 10027 versions of generalized linear models (GLMs) with a random effect that represents a latent  
 10028 spatial attribute of individuals, the activity center or home range center. This formulation  
 10029 makes analysis of the models readily accessible in freely available software and also allows  
 10030 us to adapt and use concepts from this broad class of models to solve problems in spatial  
 10031 capture recapture. In particular, we can exploit well-established model-based design con-  
 10032 cepts (Kiefer, 1959; Box and Draper, 1959, 1987; Fedorov, 1972; Sacks et al., 1989; Hardin

and Sloane, 1993; Fedorov and Hackl, 1997) to develop a framework for designing spatial trapping arrays for capture-recapture studies. Müller (2007) provides a recent monograph level treatment of the subject that is very accessible.

In the following sections, we adapt these classical methods for constructing optimal designs to obtain the configuration of traps (or sampling devices) in some region (the design space,  $\mathcal{X}$ ), that minimizes some appropriate objective function based on the variance of model parameters,  $\alpha$ , or  $N$ , for a prescribed state-space. We show that this criterion – based on the variance of an estimator of  $N$  – represents a formal compromise between minimizing the variance of the MLEs of the detection model parameters and obtaining a high expected probability of capture. Intuitively, if our only objective was to minimize the variance of parameter estimates than all of our traps should be in one or a small number of clusters where we can recapture a small number of individuals many times each. Conversely, if our objective was only to maximize the expected probability of encounter then the array should be highly uniform so as to maximize the number of individuals being exposed to capture.

### 10.5.1 Statement of the design problem

Let  $\mathcal{X}$ , the *design space*, denote some region within which sampling could occur and let  $\mathbf{X} = \mathbf{x}_1, \dots, \mathbf{x}_J$  denote the *design*, the set of sample locations (e.g., of camera traps), normally we just call these “traps.” The design space  $\mathcal{X}$  must be prescribed (a priori). Operationally, we could equate  $\mathcal{X}$  to the study area itself (which is of management interest) but, in practical cases, there will generally be parts of the study area that we cannot sample. Those areas need to be excluded from  $\mathcal{X}$ . While  $\mathcal{X}$  may be continuous, in practice it will be sufficient to represent  $\mathcal{X}$  by a discrete collection of points which is what we do here. This is especially convenient when the geometry of  $\mathcal{X}$  is complicated and irregular, which would be the case in most practical applications. The technical problem addressed subsequently is how do we choose the locations  $\mathbf{X}$  in a manner that produces the “optimal” (lowest variance) for estimating population size or density, or some other quantity of interest.

As usual, we regard the population of  $N$  individual “activity centers” as the outcome of a point process distributed uniformly over the state-space  $\mathcal{S}$ . The relevance and importance of  $\mathcal{S}$  has been established repeatedly in this book, as it defines a population of individuals (i.e., activity centers) and, in practice, it is not usually the same as  $\mathcal{X}$  due to the fact that animals move freely over the landscape and the location of traps is typically restricted by policies, ownership, logistics and other considerations. The objective we pursue here is: Given (1)  $\mathcal{X}$ , (2) a number of design points,  $J$ ; (3) the state-space  $\mathcal{S}$ , (4) an SCR model, and (5) a design criterion  $Q(\mathbf{X})$ , we want to choose *which*  $J$  design points we should select in order to obtain the *optimal* design under the chosen model, where the optimality is with respect to  $Q(\mathbf{X})$ .

What types of functions make reasonable objective functions,  $Q(\mathbf{X})$ ? We will describe some possible choices for  $Q(\mathbf{X})$  below, but it makes sense that they should relate to the variance of estimators of one or more parameters of the SCR model.

We motivate the basic ideas of model-based design with a simple model that proves to be an effective caricature of the SCR model that we'll use shortly. Suppose  $\mathbf{s}$  is the activity center of a single individual, and  $\mathbf{s}$  is known with certainty. Then, for an array

of traps  $\mathbf{X}$  we measure a response variable, lets say the strength of an acoustic signal, that has a normal distribution. So we have this response variable that has a normal linear model of the form:

$$\mathbf{y} = \mathbf{M}(\mathbf{X}, \mathbf{s})'\boldsymbol{\alpha} + \text{error}$$

In our notation here,  $\mathbf{M}(\mathbf{X}, \mathbf{s})$  is some design matrix where, in the context of SCR models, it has 2 columns (for the basic model): A column of 1's, and then a column of distance from each trap  $\mathbf{x}_j$  to the activity center  $\mathbf{s}$ . The design matrix is therefore, for a single individual, a matrix of dimension  $J \times 2$ .

The inference objective here is to estimate the parameters  $\boldsymbol{\alpha}$ . The variance-covariance matrix of  $\hat{\boldsymbol{\alpha}}$  is, suppressing the dependence on  $\mathbf{X}$  for notational convenience,

$$\text{Var}(\boldsymbol{\alpha}, \mathbf{X}) = (\mathbf{M}(\mathbf{s})'\mathbf{M}(\mathbf{s}))^{-1}$$

Note that the design points  $\mathbf{x}_j$  appear explicitly (in the 2nd column of  $\mathbf{M}$ ). In considering design for estimation in such models it is natural to choose design points, corresponding to values of  $\mathbf{x}$ , such that the variance of  $\hat{\boldsymbol{\alpha}}$  is minimized. Of course,  $\boldsymbol{\alpha}$  is a vector, and so the “variance” is a matrix (at least  $2 \times 2$ ) so we have to work with suitable scalar summaries of that matrix, such as the trace (sum of the diagonals) or a function of the determinant, etc..

For a population of  $N$  individuals, if we know *all*  $N$  values of  $\mathbf{s}$ , the design matrix  $\mathbf{M}$  has the same basic structure but with  $N$  versions stacked-up on top of one another, producing a larger  $N * J \times 2$  design matrix. The 2nd column of that matrix contains the information about trap locations (the 1st column is still just a column of 1s). Therefore, we could easily find the design  $\mathbf{X}$  that optimizes some function of the variance-covariance matrix of the model parameters.

All of this is fine and good if we happen to know the activity centers for each individual. However, this is not a realistic formulation. When  $\mathbf{s}$  is unknown, it might make sense to consider minimizing the expected (spatially averaged) variance:

$$E_{\mathbf{s}} \{ \text{Var}(\boldsymbol{\alpha}, \mathbf{X}) \} = \sum_{\mathbf{s} \in S} (\mathbf{M}'(\mathbf{s})\mathbf{M}(\mathbf{s}))^{-1}.$$

However, this is not the expected variance based on sampling a population of  $N$  individuals, just for a single individual having unknown  $\mathbf{s}$ . Because of the matrix inverse in this expression, it is not sufficient to use a variance criterion that weighs this variance by  $N$ . As an alternative, we can maximize the expected *information*, the inverse of the variance-covariance matrix, which is probably more appealing from an analytic point of view. The information matrix for the data based on a single individual, with known  $\mathbf{s}$ , is:  $\mathcal{I}(\boldsymbol{\alpha}, \mathbf{X}) = (\mathbf{M}'(\mathbf{s})\mathbf{M}(\mathbf{s}))$ . For a population of  $N$  individuals, let  $\mathbf{M}_i$  be the design matrix for the individual with activity center  $\mathbf{s}_i$ . Then, the total information for all  $N$  individuals is:

$$\mathcal{I}(\boldsymbol{\alpha}, \mathbf{X}) = \sum_{i=1}^N (\mathbf{M}'_i(\mathbf{s}_i)\mathbf{M}_i(\mathbf{s}_i))$$

The information matrix depends on the design  $\mathbf{X}$  through the  $N$  individual matrices  $\mathbf{M}_1, \dots, \mathbf{M}_N$ . Now, because we don't know  $\mathbf{s}_i$  we can compute the integrated information,

10112 over all possible values of  $\mathbf{s}_i$ , and for each  $\mathbf{s}_i$ , which is an  $N$ -fold summation:

$$E_{\mathbf{s}_1, \dots, \mathbf{s}_N} \mathcal{I}(\boldsymbol{\alpha}, \mathbf{X}) = \sum_{i=1}^N \sum_{s \in \mathcal{S}} (\mathbf{M}'_i(\mathbf{s}_i) \mathbf{M}_i(\mathbf{s}_i))$$

10113 which is just  $N$  copies of the integrated (spatially averaged) information:

$$E_{\mathbf{s}_1, \dots, \mathbf{s}_N} \mathcal{I}(\boldsymbol{\alpha}, \mathbf{X}) = N \sum_{s \in \mathcal{S}} (\mathbf{M}'_i(\mathbf{s}_i) \mathbf{M}_i(\mathbf{s}_i)).$$

10114 It therefore seems sensible to base design of SCR studies on some criterion that is  
 10115 a function of this expected information matrix. E.g., find the design that maximizes  
 10116 the diagonals, or the determinant, or minimizes the trace of the *inverse* (the variance-  
 10117 covariance matrix based on  $N$  individuals). This can be done for any number of design  
 10118 points  $\mathbf{x}_1, \dots, \mathbf{x}_J$  using standard exchange algorithms (see Müller, 2007, Chapt. 3) and  
 10119 we discuss this below in Sec. 10.5.5. However, our SCR models are not normal linear  
 10120 models but, rather, more like Poisson or binomial GLMs. We see in the next section that  
 10121 we can adapt these ideas for such models.

### 10122 10.5.2 Model-based Design for SCR

10123 Following our development of the normal linear model above, suppose for the moment  
 10124 that we know  $\mathbf{s}$  for a single individual. In this case, its vector of counts of encounter in  
 10125 each trap  $\mathbf{y}$  are either binomial or Poisson counts, and the linear predictor has this form:

$$g(\mathbb{E}(\mathbf{y})) = \alpha_0 + \alpha_1 \|\mathbf{x} - \mathbf{s}\|^2. \quad (10.5.1)$$

10126 for the Gaussian encounter probability model, or any other model could be used. In vector  
 10127 form, we write this as:

$$g(\mathbb{E}(\mathbf{y})) = \mathbf{M}' \boldsymbol{\alpha}$$

10128 where  $\mathbf{M}$  is the  $J \times 2$  design matrix where the 2nd column contains the squared pairwise  
 10129 distances between each individual  $i$  and trap  $j$ , and thus it depends on both  $\mathbf{X}$  and  $\mathbf{s}$ .

10130 The asymptotic formula for  $\text{Var}(\boldsymbol{\alpha})$  can be cooked up for any type of GLM. As an  
 10131 example (we use this below), for the Poisson GLM, the asymptotic variance-covariance  
 10132 matrix of  $\hat{\boldsymbol{\alpha}}$ , considering a single individual having location  $\mathbf{s}$ , is<sup>1</sup>

$$\text{Var}(\hat{\boldsymbol{\alpha}} | \mathbf{X}, \mathbf{s}) = (\mathbf{M}(\mathbf{s})' \mathbf{D}(\boldsymbol{\alpha}, \mathbf{s}) \mathbf{M}(\mathbf{s}))^{-1}. \quad (10.5.2)$$

10133 This is a function of the design  $\mathbf{X}$  as well as  $\mathbf{s}$  both of which are balled-up in the re-  
 10134 gression design matrix  $\mathbf{M}$ , and the matrix  $\mathbf{D}$  which is a diagonal matrix having elements  
 10135  $\text{Var}(y_j | \mathbf{s}) = \exp(\mathbf{m}' \boldsymbol{\alpha})$  for  $y_j$  = the frequency of encounter in trap  $j$  and where  $\mathbf{m}'$  is the  
 10136  $j^{th}$  row of  $\mathbf{M}(\mathbf{s})$ . We can compute the expected information under the Poisson model with  
 10137 known  $N$  using this modified formulation. These ideas are meant to motivate technical  
 10138 concepts related to model-based design, where we know  $N$ , and therefore have a convenient  
 10139 variance or information expression to work with. However, in all real capture-recapture  
 10140 applications we won't know  $N$ , and so we need to address that issue, which we do in the  
 10141 next section.

<sup>1</sup> This is basic GLM theory that derives from the fact that the Poisson is a member of the natural exponential family of distributions, e.g., see McCullagh and Nelder (1989) or Agresti (2002).

---

**10142 10.5.3 An Optimal Design Criterion for SCR**

10143 There are a number of appealing directions to pursue for deriving a variance-based criterion  
 10144 upon which to devise designs for capture recapture studies. For one, we could formulate  
 10145 the objective function based on the variance of the Huggins-Alho estimator (Sec. 4.5) of  
 10146  $N$ . We find that these expressions depend on individual sample inclusion probabilities  
 10147 (if  $s$  is close to traps, the individual has a high probability of being encountered and  
 10148 *vice versa*), and hence the specific trap locations, and parameters of the model. These  
 10149 variance expressions provide a natural design criterion. On the other hand, we find that  
 10150 the calculus is a bit tedious at the present time. As an alternative, we devise a variance  
 10151 criterion based on the conditional estimator of  $N$  having the form

$$\tilde{N} = \frac{n}{\hat{\bar{p}}}$$

10152 where  $\hat{\bar{p}}$  is the MLE of the marginal probability that an individual appears in the sample  
 10153 of  $n$  unique individuals, and it depends on the MLE of the parameters of the encounter  
 10154 probability model,  $\hat{\alpha}$ . We elaborate on the precise form of  $\hat{\bar{p}}$  and the variance of its MLE  
 10155 below. The variance of this estimator is:

$$\text{Var}\left(\frac{n}{\hat{\bar{p}}}\right) = n^2 * \text{Var}\left(\frac{1}{\hat{\bar{p}}}\right)$$

10156 An important thing to note is that this estimator, and its variance, are *conditional* on the  
 10157 sample size of individuals,  $n$ . We never set out, in capture-recapture, to obtain a sample  
 10158 of  $n$  individuals ( $n$  is always a stochastic outcome) and so we need to “uncondition” on  $n$ .  
 10159 Fortunately, this is a simple proposition using standard rules of expectation and variance,  
 10160 which produce the following expression:

$$\text{Var}(\tilde{N}(\alpha)) = N\bar{p}\{(1 - \bar{p}) + N\bar{p}\} \left( \frac{\text{Var}(\hat{\bar{p}})}{\bar{p}^4} \right) \quad (10.5.3)$$

10161 The key thing to note about this as a criterion: (1) It depends on  $\bar{p}$ , the marginal proba-  
 10162 bility of encounter. Clearly the variance decreases as  $\bar{p}$  increases. In general, the form of  $\bar{p}$   
 10163 depends on the SCR model being used. We will provide an example below. Obviously,  $\bar{p}$   
 10164 will depend on the parameter values  $\alpha$ . (2) The criterion depends on  $\text{Var}(\hat{\bar{p}})$ . So, designs  
 10165 that estimate  $\bar{p}$  well should be preferred. This will also depend on the parameters  $\alpha$  and  
 10166 *also* the variance of the MLE,  $\hat{\alpha}$ . Based on these considerations, we suggest a number  
 10167 of appealing criteria for constructing spatial designs for capture-recapture studies. For  
 10168 convenience we label them  $Q_1$  -  $Q_4$ :

- 10169 (1)  $Q_1 = \text{Trace}(\mathbf{V}_\alpha)$  where  $\mathbf{V}_\alpha$  is the variance-covariance matrix of the MLE of  $\alpha$ . De-  
 10170 signs which minimize this criterion are those which are good for estimating the param-  
 10171 eters of the encounter probability model.
- 10172 (2)  $Q_2$  is the variance expression in Eq. 10.5.3. Using this criterion, we should prefer  
 10173 designs that minimize the variance for estimating  $N$ .
- 10174 (3)  $Q_3 = 1 - \bar{p}$ . Designs which minimize this criterion are those which maximize the  
 10175 average capture probability. These should maximize  $n$ .
- 10176 (4)  $Q_4 = \text{Var}(\hat{\bar{p}})$ . We should prefer designs which provide good estimates of  $\bar{p}$ .

10177 To make use of any of these criteria in a particular design problem, we need to decide on  
 10178 values of  $N$ , and the model parameters for computing  $\bar{p}$ , and then think about optimizing  
 10179 the criterion over all possible designs (see below).

10180 **10.5.4 Too much math for a Sunday afternoon**

10181 Here we discuss calculation  $\bar{p}$  and variance expressions required to compute the design  
 10182 criteria above.

10183 **Characterizing  $\bar{p}$**

10184 In SCR models, an individual with activity center  $\mathbf{s}_i$  is captured if it is captured in *any*  
 10185 trap and therefore, under the Bernoulli (passive detector) observation model,

$$\bar{p}(\mathbf{s}_i, \mathbf{X}) = 1 - \prod_{j=1}^J (1 - p_{ij}(\mathbf{x}_j, \mathbf{s}_i))$$

10186 where  $p_{ij}$  here is the Gaussian (or other) encounter probability model that depends on  
 10187 distance between traps and activity centers, say  $d_{ij}$  for the distance between individual  
 10188 activity center  $\mathbf{s}_i$  and trap  $\mathbf{x}_j$ . Under the Poisson observation model, with a Gaussian  
 10189 hazard model:

$$\bar{p}(\mathbf{s}_i, \mathbf{X}) = 1 - \exp(-\lambda_0 \sum_j \exp(\alpha_1 * d(\mathbf{x}_j, \mathbf{s}_i)^2))$$

10190 where here we emphasized that this is conditional on  $\mathbf{s}_i$  and also the design – the trap  
 10191 locations  $\mathbf{x}_j$ . The *marginal* probability of encounter, averaging over all possible locations  
 10192 of  $\mathbf{s}$  is:

$$\bar{p}(\mathbf{X}) = 1 - \int_{\mathbf{s}} \bar{p}(\mathbf{s}_i, \mathbf{X}) d\mathbf{s}. \quad (10.5.4)$$

10193 It is important to note that this can be calculated directly *given* the design  $\mathbf{X}$ , and  
 10194 parameters of the model. This is handy because we see that it is used in the variance  
 10195 formulae given subsequently and therefore it is used directly in evaluating any of the  
 10196 criteria described above.

10197 **Characterizing  $\text{Var}(\hat{p})$**

10198 Developing an expression for  $\text{Var}(\hat{p})$  depends on the observation model. We work here with  
 10199 the Poisson observation model, and we do that because the technical argument that follows  
 10200 is somewhat easier for that case compared to the Bernoulli model for passive detection  
 10201 devices (but see Huggins (1989) and Alho (1990) for additional context). We first express  
 10202 the integral in Eq. 10.5.4 as a summation over a fine mesh of points so that:

$$\bar{p}(\mathbf{X}) = \sum_{\mathbf{s}} 1 - \bar{p}(\mathbf{s}_i, \mathbf{X})$$

10203 which under the Poisson observation model is, in a simplified notation:

$$\bar{p}(\mathbf{X}) = \sum_{\mathbf{s}} \left\{ 1 - \exp\left(-\sum_j \exp(\alpha_0 + \alpha_1 d(\mathbf{x}_j, \mathbf{s})^2)\right) \right\}$$

10204 The MLE of  $\bar{p}(\mathbf{X})$  has us plug in the MLE of the parameters of the model, in this case  
 10205  $\hat{\lambda}_0 = \exp(\hat{\alpha}_0)$  and  $\hat{\alpha}_1$ . To compute the variance of the MLE of  $\bar{p}$ , we note that the variance  
 10206 operator can move inside the summation over  $\mathbf{s}$ , and the subtraction from 1 doesn't count  
 10207 anything, so we have

$$\text{Var}(\hat{p}(\mathbf{X})) = \sum_{\mathbf{s}} \text{Var} \left( \exp \left( - \sum_j \exp(\hat{\alpha}_0 + \hat{\alpha}_1 d(\mathbf{x}_j, \mathbf{s})^2) \right) \right)$$

10208 A few applications of the delta approximation and some arm-waving yields the following  
 10209 expression for the variance of  $\hat{p}$ :

$$\text{Var}(\hat{p}(\mathbf{X})) = \sum_{\mathbf{s}} \left( \exp(-\hat{\lambda}_{\mathbf{s}}) \right) \left( \sum_j \hat{\lambda}(\mathbf{x}_j, \mathbf{s})^2 (\text{Var}(\hat{\alpha}_0) + d(\mathbf{x}_j, \mathbf{s})^4 \text{Var}(\hat{\alpha}_1)) \right)$$

10210 where  $\lambda(\mathbf{x}, \mathbf{s}) = \exp(\alpha_0 + \alpha_1 d(\mathbf{x}, \mathbf{s})^2)$  and  $\lambda_{\mathbf{s}} = \sum_{j=1}^J \lambda(\mathbf{x}_j, \mathbf{s})$ .

### 10211 Characterizing $\text{Var}(\hat{\alpha})$

10212 The big picture is this: For a given design  $\mathbf{X}$ , we can compute  $\text{Var}(\hat{p}(\mathbf{X}))$  – this is just a  
 10213 calculation involving sums over all points in the state-space and design points – provided  
 10214 we know the variance of the estimator of  $\boldsymbol{\alpha}$ ,  $\text{Var}(\hat{\boldsymbol{\alpha}})$  and the parameters of the model.  
 10215 However, it is not so easy to write down the analytic form of this matrix. Some calculus  
 10216 would have to be done on the conditional likelihood (e.g., from Borchers and Efford (2008))  
 10217 to figure out the asymptotic form of this matrix. For our purposes, we think it might  
 10218 suffice<sup>2</sup> to approximate the matrix, using the analogous result from a Poisson or binomial  
 10219 GLM assuming that  $N$  is known, since we have convenient formulas for those (see Eq.  
 10220 10.5.2).

10221 The approximate variance given by Eq. 10.5.2 is conditional on the collection of  
 10222 activity centers,  $\mathbf{s}_1, \dots, \mathbf{s}_N$ . To resolve this, we take the approach outlined previously to  
 10223 compute the *expected* information obtained from a particular realization of  $N$  individuals,  
 10224 and invert that result. In particular, the total information for all  $N$  individuals is

$$\mathcal{I}(N) = \mathbf{M}'_1 \mathbf{D}_1 \mathbf{M}_1 + \dots + \mathbf{M}'_N \mathbf{D}_N \mathbf{M}_N$$

10225 We can compute the expected information over *all* elements of the state-space, which is  
 10226 just  $N$  times the average information of a single individual:

$$\mathbb{E}(\mathcal{I}(N)) = N \sum_{\mathbf{s}} \mathbf{M}(\mathbf{s})' \mathbf{D}(\mathbf{s}) \mathbf{M}(\mathbf{s}).$$

### 10227 Putting it all together

10228 For a single design,  $\mathbf{X}$ , we need to compute this expected information quantity, invert it  
 10229 to get the variance of  $\hat{\boldsymbol{\alpha}}$ , and then either use that variance matrix in the calculation of  
 10230 criterion  $Q_1$ , or else evaluate some other quantities along the way to computing the other  
 10231 criteria. We can compute  $\bar{p}$  (which is  $Q_3$ ) for a given design without doing any variance  
 10232 calculations. If we use  $\text{Var}(\hat{\boldsymbol{\alpha}})$  along with  $\bar{p}$ , we can compute  $\text{Var}(\hat{p})$ , which is  $Q_4$ . We can  
 10233 combine all of these things together and compute  $\text{Var}(N)$  for a given  $\mathbf{X}$ . This gives us  $Q_2$ .

---

<sup>2</sup>Warning: But we don't know. No warranty is implied.

---

**10234 10.5.5 Optimization of the criterion**

10235 To compute spatial designs that optimize a given criterion, we need to come up with a  
 10236 ballpark guess of the model parameters so that the criterion can be evaluated for any  
 10237 design. i.e., what are the values of  $\alpha$  and  $N$  we expect in our study? If we do that,  
 10238 and specify the state-space  $\mathcal{S}$  then, we can, in theory, optimize the variance criterion  
 10239 over all possible configurations of  $J$  traps. In formulating the optimization problem note  
 10240 that we have  $J$  sample locations corresponding to rows of  $\mathbf{X}$ . The problem is therefore a  
 10241  $2J$  dimensional optimization problem which, for  $J$  small, could be solved using standard  
 10242 numerical optimization algorithms as exist in almost every statistical computation environ-  
 10243 ment. However,  $J$  will almost always be large enough so as to preclude effective use of  
 10244 such algorithms. This is a common problem in experimental design, and spatial sampling  
 10245 in general, for which sequential exchange or swapping algorithms have been fairly widely  
 10246 adopted (e.g., Wynn, 1970; Fedorov, 1972; Mitchell, 1974; Meyer and Nachtsheim, 1995;  
 10247 Nychka et al., 1997; Royle and Nychka, 1998). The basic idea is to pose the problem as a  
 10248 sequence of 1-dimensional optimization problems in which the objective function is opti-  
 10249 mized over 1 or several coordinates at a time. In the present case, we consider swapping  
 10250 out  $\mathbf{x}_j$  for some point in  $\mathcal{X}$  that is nearby  $\mathbf{x}_j$  (e.g., a 1st order neighbor). Beginning with  
 10251 an initial design, chosen randomly or by some other method, the objective function is eval-  
 10252 uated for all possible swaps (at most 4 in the case of 1st order neighbors) and whichever  
 10253 point yields the biggest improvement is swapped for the current value. The algorithm is  
 10254 iterated over all  $J$  design points and this continues until convergence is achieved. Such  
 10255 algorithms may yield local optima and optimization for a number of random initial designs  
 10256 can yield incremental improvements. We implemented such a swapping algorithm in **R**,  
 10257 and it is available as a function in the **scrbook** package with the function **SCRdesign**. The  
 10258 algorithm operates on a discrete representation of  $\mathcal{S}$  (an arbitrary matrix of coordinates).  
 10259 For each point in the design,  $\mathbf{X}$ , only the nearest neighbors (the number is specified by  
 10260 the user) are considered for swapping into the design during each iteration. For example,  
 10261 to compute **ndesigns** = 10 putative optimal designs (each based on a random start) of  
 10262 size  $J$  = 11, we execute the function as follows:

```

 10263 > des<-SCRdesign(S,X,ntraps=11,ndesigns=10,nn=15,sigma=1)
  
```

10264 Where the state-space **S**, the candidate set **X** are provided as matrices, **nn** is the number of  
 10265 nearest neighbors to inspect for each design point change, and **sigma** is the scale parameter  
 10266 of, in this case, a Gaussian hazard model. See the help file **SCRdesign** for examples and  
 10267 analysis of the output.

10268 While swapping algorithms are convenient to implement, and efficient at reducing  
 10269 the criterion in very high dimensional problems, they do not always yield the global  
 10270 optimum. In practice, as in the examples below, it is advisable to apply the algorithm to  
 10271 a large number of random starting designs. Our experience is that essentially meaningless  
 10272 improvements are realized after searching through a few dozen random starts.

10273 The design criteria we developed above bear a striking resemblance to design criteria  
 10274 used to construct so-called space-filling designs (Nychka et al., 1997). Such criteria are  
 10275 based on inter-point distances, and space-filling designs seek to optimize some function  
 10276 of distance alone, instead of a variance-based objective function. The benefit of this  
 10277 approach is that one doesn't have to specify the model to produce a design, and space-  
 10278 filling designs have been shown to provide reasonable approximations to designs based on

10279 variance criteria under flexible statistical models (Nychka et al., 1997). This similarity  
 10280 suggests that perhaps certain distance-based design criteria might be suitable for SCR  
 10281 models. A version of a swapping algorithm used to optimize a space-filling criterion is  
 10282 implemented in the **R** package **fields**.

10283 **10.5.6 Illustration**

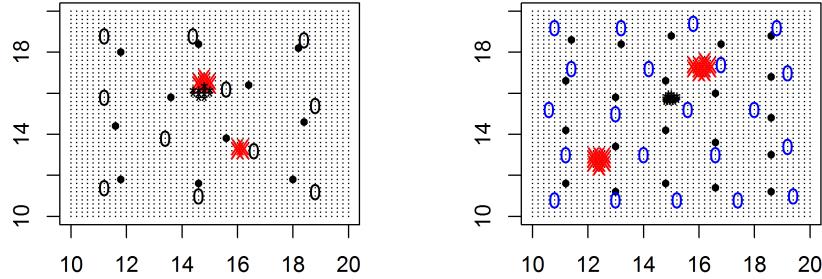
10284 Because the algorithm operates on a discrete version of  $\mathcal{S}$ , it is trivial to apply to situations  
 10285 in which the state-space is arbitrary in extent and geometry. However, we consider a  
 10286 simplified situation here in order to illustrate the calculation of optimal designs and how  
 10287 they look for an idealized situation.

10288 Consider designing a camera trapping study for a square state-space on  $[9, 21] \times [9, 21]$   
 10289 and with  $\mathcal{X}$  being the smaller square  $[10, 20] \times [10, 20]$ . For this illustration we assumed  
 10290  $\alpha_0 = \log(\lambda_0) = -1.7$  and  $\sigma = \sqrt{2}$  so that  $\alpha_1 = 1/(2\sigma^2) = 1/4$ .

10291 Designs of size 11 and 21 were computed using 10 random starting designs. We found  
 10292 the optimal design using each of the 4 criteria we described above. To refresh your memory  
 10293  $Q_1$  is the trace of the variance-covariance matrix of  $\hat{\alpha}$ ,  $Q_2$  is the variance of  $\hat{N}$ ,  $Q_3$  is  $1 - \bar{p}$   
 10294 (so the design that minimizes this criterion obtains the highest possible  $\bar{p}$ ), and  $Q_4$  is the  
 10295 variance of  $\hat{p}$ . The putative optimal designs (henceforth “best”) are shown in Fig. 10.3.  
 10296 There are a few points of some interest here.

10297 The designs are not completely regular but obviously have a systematic look to them.  
 10298 For the  $J = 11$  designs, the  $Q_1$  design is slightly more compact, with an average closest  
 10299 neighbor distance of 2.59 units vs 3.03 units for the  $Q_3$  design. The two designs are qual-  
 10300 itatively similar, providing roughly uniform coverage of the design space  $\mathcal{X}$ . Conversely,  
 10301 the other two criteria produce designs that are highly clustered. Criterion  $Q_2$  which is  
 10302 optimal for  $\hat{N}$ , produces 2 clusters of traps, 7 traps in one and 4 traps in the other. Finally,  
 10303 designs which are optimal for the criterion  $Q_4$ , the variance of estimating  $\bar{p}$ , produce a  
 10304 single cluster of traps that is roughly centrally located in the design space. This makes  
 10305 sense, because the very dense cluster of traps provides a large number of recaptures near  
 10306 the origin  $d = 0$ , which, intuitively, provides the most information about estimating pa-  
 10307 rameter of the encounter probability model. For the  $J = 21$  designs, we have an average  
 10308 closest neighbor distance of 1.87 for  $Q_1$  and 2.19 for  $Q_3$  but, qualitatively, the structure  
 10309 of the designs is similar to  $J = 11$ . The best design for estimating  $N$  (the criterion  $Q_2$ )  
 10310 produces 2 clusters, but just with more traps. While these illustrations make sense to us,  
 10311 we’re not entirely convinced of the implication that 2 clusters of traps should be optimal  
 10312 with  $J = 21$  total traps. However, it is clear what is going on here is that the tight clusters  
 10313 are providing good information about estimating  $\bar{p}$  and, by spreading the two clusters out,  
 10314 the expected sample size,  $n$ , is maximized.

10315 While the designs for  $Q_1$  and  $Q_3$  are not exactly uniform, they are very regular looking  
 10316 which we should expect given the regularity of both  $\mathcal{S}$  and  $\mathcal{X}$  in this case. One thing to  
 10317 note is that the trap spacing varies depending on  $J$  even though  $\sigma$  is the same, so optimal  
 10318 trap spacing should not be viewed as a static thing depending only on the model. Because  
 10319 the designs are not exactly regular, the average closest neighbor distance is not exactly  
 10320 the same as the trap spacing of a regular grid design.



**Figure 10.3.** Best designs for each of 4 design criteria, produced using the exchange algorithm with 15 nearest-neighbors and based on 10 random starting designs. The left panel shows the best  $J = 11$  point designs, and the right panel shows the best  $J = 21$  point designs. The solid black dots correspond to the best design for  $Q_1$ , 0 marks the design for  $Q_3$ , "X" for  $Q_2$  and the tightly clustered "\*" corresponds to  $Q_4$ .

### 10.5.7 Density covariate models

Many capture-recapture studies will involve one or more landscape or habitat covariates that are thought to affect density, with the idea of using the methods described in Chapt. 11 for modeling and inference. We imagine that it should be possible to extend the model-based framework described previously to accommodate uncertainty due to having to estimate  $\beta$ , and this could be included as a feature of the design criterion.

In this case, we can think of the captures in a trap being Poisson random variables with mean  $\mu(\mathbf{x}, \mathbf{s}) * D(\mathbf{s})$  and we think the same arguments as given above can be used to devise design criteria and optimize them. However, in this case we might not only care about estimating  $N$  but also (or instead) inference about the parameters  $\beta$ . Thus, we might choose designs that are good for  $N$  or perhaps only good for estimating  $\beta$  or perhaps both. Intuitively, we think these two design objectives conflict with one another to some extent. Model-based approaches should favor areas of higher density, but the design points need to realize variation in the landscape covariates too.

## 10.6 TEMPORAL ASPECTS OF STUDY DESIGN

The spatial configuration of traps is one of the most important aspects of sampling design for capture-recapture studies. Indeed, as we discussed in the previous section, design under SCR models can be thought of as being analogous to classical model-based spatial design, and the concepts and methods from that field can be brought to bear on the design

of capture-recapture studies. However, there are other aspects of sampling design that should be considered in capture-recapture studies, including the frequency or length of temporal samples. We discuss some of these issues here, although without a detailed or formal analysis.

### 10.6.1 Total sampling duration and population closure

All the models we have discussed so far are *closed population* SCR models, i.e., models that assume that the population remains constant during our survey. Traditionally, two different levels of closure have been considered in the capture-recapture literature – demographic and geographic closure (see also Chapt. 4). Demographic closure refers to the absence of births and deaths, while geographic closure refers to the absence of immigration and emigration during a study. In traditional capture-recapture, the geographic closure assumption prohibited (in theory, not in field praxis, of course) any movement off the trapping grid. Kendall (1999) explored a range of scenarios of closure violation, focusing on different kinds of movement in and out of the study area, and found that several of these scenarios caused biased abundance estimates from traditional capture-recapture models.

As discussed in Chapt. 5, one main objective of SCR models is to relax the geographic closure assumption – the model explicitly allows for movements of individuals about their activity centers, which may have them off the trapping grid for parts of the time, even if the activity center itself is on the grid. SCR models do, however, assume no permanent emigration or immigration from the state-space. The interpretation of demographic closure remains the same in SCR models as it is in traditional CR models.

We have not explored effects of closure violation on SCR abundance and density estimates. Conceptually, we expect estimates to be biased high when births or immigrations happen during our study. For one, the total number of individuals at the study site during the course of the study would be higher than at any particular point in time and correspond to a *cumulative* number of individuals in our study area. Further, because some individuals are not available for detection for the entire study (they only become available when they are recruited) we would expect detection to be underestimated, potentially leading to further positive bias in estimates of abundance. Death or emigrations during a study do not inflate the number of individuals actually on the study area, but as animals die and become unavailable for detection, we can again imagine a negative bias in baseline detection and, consequently, some positive bias in  $N$ .

To avoid such bias in population estimates, closed population models should typically be applied to short surveys, where short is relative to the life history of the species under study. For example, for small mammals, that might mean a few days, whereas for large, long-lived species with a slow population turnover, several weeks or even a couple of months can still be considered short. In practice, we have no means of ever guaranteeing a closed population – even if we sample animals for a day, one of the individuals we record may be eaten by a predator later that day, or a dispersing individual may arrive just as we turn our backs. On the other hand, we are faced with the need to collect sufficient data, which, especially for elusive species, pushes us to sample over longer rather than shorter time periods. If we do not have enough sampling devices to cover the entire area of interest at once, rotating study designs (see sec. 10.4) can require even longer sampling to accumulate sufficient captures and recaptures. So clearly, in temporal study design we

have to strive for a compromise between collecting enough data while still approximating a closed population. For some species we may be able to avoid seasons where violation of demographic closure is particularly likely – for example migration seasons in migratory birds, or specific breeding seasons (or collective suicide season in lemmings). But for many species such biological seasons might be less clear cut. For example, in warm climates tigers and other large cats can breed year round (Nowak, 1999). As a consequence, guidelines as to what time frame adequately approximates a closed population are generally vague and arm-wavy. Unfortunately, we do not have much more to offer on the subject of how to decide on the length of a study, other than to urge you to think about the biology of your study species *before the study* and choose a time window that seems appropriate for that purpose.

### 10.6.2 Diagnosing and dealing with lack of closure

Once a field study has been conducted, you may wonder whether the collected data contain any evidence that the closure assumption has been met or violated. Relatively few tests for population closure in traditional capture-recapture have been developed, mostly due to the fact that behavioral variation in detection is indistinguishable from violation of demographic closure (Otis et al., 1978; White et al., 1982). Otis et al. (1978) developed a test for population closure that can handle heterogeneity in detection probability, but does not perform well in the presence of time or behavioral variation in  $p$ . Stanley and Burnham (1999) developed a closure test for model  $M_t$  (time variation in detection), which works well when there is permanent emigration and a large number of individuals migrate. Both tests are implemented in the program **CloseTest** Stanley and Richards (2013).

There are no specific population closure tests for SCR models, for the same reasons that violation of other model assumptions cannot necessarily be distinguished from a lack of population closure. If you are worried that closure might have been violated in your study, one approach of dealing with this problem is to fit an open population model. You can subdivide your study into several periods and fit a spatial version of Pollock's robust design capture-recapture model, which can estimate population size/density for each of these periods (in this context also called primary periods) using models of demographic closure. Alternatively, we may consider fully dynamic models which contain explicit parameters of survival and recruitment (Chapt. 16). These models can be quite time consuming, and if you wanted a faster check you could alternatively fit a spatial Cormack-Jolly-Seber model that only estimates survival. The magnitude of the survival estimates gives you some partial information about population closure in your study – if survival is close to 1 at least there is little evidence of losses of individuals, either through permanent emigration or death. These and other open population models are presented in detail in Chapt. 16. Finally, if your data are too sparse to fit a full-blown open population model, you can subdivide your study into  $t = 1, 2, \dots, T$  primary periods and estimate abundance separately for each period's worth of data, possibly sharing the detection parameters across periods, if you can safely assume they remain constant. You can do that by either letting  $N_t$  be independent from each other, or by specifying an underlying distribution for all  $N_t$  in a multi-session framework as described in Chapt. 14.

## 10.7 SUMMARY AND OUTLOOK

10425 Design of capture-recapture studies in the context of *spatial* models is an important prob-  
10426 lem, but solutions to this problem are mostly *ad hoc* or incomplete at the present time.  
10427 As a general rule, we always recommend scenario analysis by Monte Carlo simulation  
10428 (Efford and Fewster, 2012; Sollmann et al., 2012; Sun, in prep). This takes a lot of time  
10429 but it guarantees forward progress, or at least not doing the dumbest from among several  
10430 dumb things. We discussed some examples from the literature that assess trap spacing  
10431 and evaluate trap clustering and rotating coverage strategies for sampling large areas. The  
10432 nice thing about simulation studies is that we can simulate data for any complex situation  
10433 we desire, even if we can't fit the model effectively. Thus, we can always characterize  
10434 worst-case situations under pathological model misspecifications.

10435 When designing a spatial capture-recapture study for a single species, trap spacing and  
10436 the size of the array can (and should) be tailored to the spatial behavior of that species to  
10437 ensure adequate data collection. However, some trapping devices like camera traps may  
10438 collect data on more than one species and researchers may want to analyze these data,  
10439 too. Independent of the trapping device used, study design will in most cases face a limit  
10440 in terms of the number of traps available or logically manageable. As a consequence,  
10441 researchers need to find the best compromise between trap spacing and the overall grid  
10442 area.

10443 Particularly for large mammal research, SCR models have much more realistic require-  
10444 ments in terms of area coverage than non-spatial CR models. In the latter, density esti-  
10445 mates can be largely inflated with small trapping grids relative to individual movement  
10446 (Maffei and Noss, 2008) – covering at least 4 times the average home range is recom-  
10447 mended. Further, we need consistent coverage of the entire study area, as all individuals  
10448 in the population of interest must have some probability  $> 0$  to be captured.

10449 In contrast, SCR models work well in study areas on the scale of an individual's home  
10450 range (as long as sufficient data is collected) (Sollmann et al., 2012; ?; Marques et al.,  
10451 2011), and they provide unbiased estimates for sampling designs that do not expose all  
10452 individuals in the sampled population to detectors, i.e., that have "holes" (Efford and  
10453 Fewster, 2012). These results, however, should not encourage researchers to design non-  
10454 invasive trap arrays based on minimum area requirements and with a minimum number  
10455 of detectors. Study design should still strive to expose as many individuals as possible  
10456 to sampling and obtain adequate data on individual movement. Large amounts of data,  
10457 both individuals and recaptures, do not only improve precision of parameter estimates  
10458 (Sollmann et al., 2012; Efford et al., 2004), they also allow including potentially important  
10459 covariates (such as gender or time effects in the black bear example – see also Chapt. 7)  
10460 into SCR models to obtain density estimates that reflect the actual state of the studied  
10461 population.

10462 Beyond the traditional grid-based sampling design, the flexibility of SCR models allows  
10463 for different spatial detector arrangements such as linear (see ? for an example), or  
10464 dispersed clusters of detectors. How well these different designs perform, comparatively,  
10465 remains to be explored.

10466 The other general strategy for constructing spatial designs is a formal model-based  
10467 strategy in which we seek the configuration of design points (trap locations)  $\mathbf{x}_1, \dots, \mathbf{x}_J$   
10468 which are optimal for some formal information-based objective function. This is a standard  
10469 approach in classical sampling and experimentation, yet it has not gained widespread use

in ecology. In our view, model-based design under SCR models has great potential due to its coherent formulation and flexibility. On this topic, we have just barely scratched the surface here, showing how to formulate a criterion that is a function of the design, and then optimizing the criterion over all possible designs. Our cursory analysis of model-based design in a single situation did reveal an important aspect of design that has not been discussed in the literature. That is, the optimal spacing of traps in an array depends on the *density* of traps in the state-space. In our analysis, the spacing of 11 and 21 trap optimal designs was quite different. Therefore, this should be considered in practical SCR design exercises.

Conceptually, the information in SCR studies comes in two parts: Recaptures of individuals at different traps (spatial recaptures) and the total sample size of individuals. Maximizing both of these things as objectives induces an explicit trade-off in the construction of capture-recapture designs. We need designs that are good for estimating  $\bar{p}$  and also designs that obtain a high sample size of  $n$ . Designs that are extremely good only for one or the other will produce bad SCR designs – estimators of density with low precision – or designs in which  $N$  is not estimable due to a lack of spatial recaptures. One possible exception is when telemetry data are available (or other auxiliary data). In Chapt. 13 we discuss SCR models that integrate auxiliary information on resource selection obtained by telemetry. Telemetry data are directly informative about the coefficient of the distance term ( $\sigma$  or  $\alpha_1$ ) and, in fact, can be estimated from telemetry data alone. It stands to reason that, when telemetry data are available, this should affect considerations related to trap spacing. Conceivably even, one might be able to build SCR designs that don't yield any formal spatial recaptures because all of the information about  $\sigma$  is provided by the telemetry data. We have done limited evaluations of the trap spacing problem in the presence of telemetry data, and the results suggest that, while efficient designs have a larger trap spacing than without telemetry data, the realization of some spatial recaptures is important even when telemetry data are available. With the **R** code we provide in Chapt. 13, you should be able to carry out your own custom evaluation of these types of design problems.



10499

## Part III

10500

---

10501

# Advanced SCR Models



10502  
10503

---

# 11

10504  
10505

## MODELING SPATIAL VARIATION IN DENSITY

10506 Underlying every SCR model is a spatial point process that describes the number and  
10507 distribution of animal activity centers. Spatial point processes are characterized by two  
10508 key elements: a spatial domain, or state-space  $\mathcal{S}$ , and an intensity function which returns  
10509 the expected density of points at any location in  $\mathcal{S}$ . If the intensity is constant throughout  
10510  $\mathcal{S}$ , the point process is said to be homogeneous. Thus far we have focused our attention  
10511 on homogeneous point processes whose realized values are the locations of the  $N$  activity  
10512 centers. When a Poisson prior is placed on  $N$ , the model is known as a homogeneous  
10513 Poisson point process, which is the classic model of “complete spatial randomness.” A  
10514 similar model, that we often use in conjunction with data augmentation and MCMC,  
10515 places a binomial prior on  $N$ . This is also a model of spatial randomness, and in this  
10516 chapter we will compare and contrast the two.

10517 The spatial randomness assumption is often viewed as restrictive because ecological  
10518 processes such as habitat selection can result in non-uniform distributions of organisms.  
10519 We have argued, however, that this assumption is less restrictive than may be recognized  
10520 because a homogeneous point process actually allows for infinite possible “point patterns”,  
10521 or realized configurations of activity centers. Furthermore, given enough data, the uniform  
10522 prior will have very little influence on the estimated locations of activity centers.  
10523 Nonetheless, a homogeneous point process does not allow one to model population density  
10524 using covariates, which is an important objective in much ecological research. For  
10525 example, even when assuming a homogeneous point process for the activity centers, an  
10526 estimated density surface may strongly suggest that individuals are more abundant in one  
10527 habitat than another; however, such results do not provide the basis for formally testing  
10528 hypotheses about spatial variation in density, and they could not be used to make pre-  
10529 dictions about habitat-specific abundance in other regions. A more direct approach is to  
10530 replace the homogeneous model with an inhomogeneous model in which the point process  
10531 intensity is allowed to vary spatially.

10532 In this chapter, we cover methods for fitting inhomogeneous Poisson and binomial  
10533 point process models so that density can be modeled as a function of covariates in much

10534 the same way as is done in generalized linear models. The covariates we consider differ  
 10535 from those covered in previous chapters, which were typically attributes of the animal  
 10536 (e.g. sex or age) or the trap (e.g. baited or not) and were used to model movement  
 10537 or encounter rate. In contrast, here we wish to model covariates that are defined at all  
 10538 points in  $\mathcal{S}$ , and so we will refer to them as state-space covariates or density covariates.  
 10539 These may include continuous covariates such as elevation, or categorical covariates such  
 10540 as habitat type. Typically, these state-space covariates are formatted as raster images  
 10541 with a prescribed resolution and extent.

10542 One thing to keep in mind when modeling density is that the SCR definition of density  
 10543 is different than what is perhaps a more common definition of density in ecology. In SCR  
 10544 models, density is defined as the number (or expected number) of *activity centers* in  
 10545 some region, whereas in other ecological studies, density is often defined as the number  
 10546 of *individuals* in some region at some instant in time. The latter definition is closer to  
 10547 the quantity being estimated in distance sampling studies. So which definition is better?  
 10548 Does it make more sense to contemplate activity centers or individuals at an instant in  
 10549 time? From our perspective, either definition may suffice for a given objective, but we  
 10550 note that there exists a formal relationship between the two since an activity center is  
 10551 the *average* of an individual's locations during some time period. As such, an activity  
 10552 center may be a better descriptor of an individual's preferences than is a location during  
 10553 a single instant in time. Moreover, with SCR models we can model both the distribution  
 10554 of activity centers (as we will do in this chapter) as well as the distribution of individuals  
 10555 during specific instances in time, as is demonstrated in Chapt. 15.

10556 Inhomogeneous Poisson point process models were discussed in the original formulation  
 10557 of SCR models by Efford (2004) and were described in more detail by Borchers and Efford  
 10558 (2008). We will show that an inhomogeneous point process with a binomial prior on  $N$  is  
 10559 quite similar to the Poisson model, but is more easily implemented in MCMC algorithms.  
 10560 To do so, we will define the data augmentation parameter  $\psi$  in terms of the point process  
 10561 intensity function, and we will replace the uniform prior on the activity centers with a  
 10562 prior that is also derived from the intensity function. Development of this prior, which  
 10563 does not have a standard form, is a central component of this chapter. First we begin  
 10564 with a review of homogeneous point process models.

## 11.1 HOMOGENEOUS POINT PROCESS REVISITED

10565 The homogeneous Poisson point process is *the* model of complete spatial randomness and  
 10566 is often used in ecology as a null model to test for departures from randomness (Cressie,  
 10567 1991; Diggle, 2003; Illian et al., 2008). The Poisson model asserts that the number of points  
 10568 in  $\mathcal{S}$  is Poisson distributed:  $N \sim \text{Poisson}(\mu|\mathcal{S}|)$  where  $\mu > 0$  is the intensity parameter  
 10569 and  $|\mathcal{S}|$  is the area of the state-space. The intensity parameter  $\mu$  is the density of points,  
 10570 and thus multiplying the intensity by the area of some region yields the expected number  
 10571 of points in that region. As with all homogeneous point process models, the  $N$  points are  
 10572 distributed uniformly, which implies that they do not interact with each other in any way  
 10573 – for example, they neither attract nor repel one another.

10574 Unlike the Poisson point process, the binomial point process assumes that  $N$  is fixed,  
 10575 not random. The distinction is illustrated by this simple R code that generates realizations  
 10576 from Poisson and binomial point processes in the unit square ( $\mathcal{S} = [0, 1] \times [0, 1]$ ):

---

```

10577 > Area <- 1                                # Area of unit square
10578 > muP <- 4                                # intensity
10579 > nP <- rpois(1, muP*Area)               # number of points: random
10580 > PPP <- cbind(runif(nP), runif(nP))    # Poisson point pattern
10581 > nB <- 4                                # number of points: fixed
10582 > muB <- nB/Area                          # intensity
10583 > BPP <- cbind(runif(nB), runif(nB))    # binomial point pattern

```

10584 Both of these models are homogeneous because the intensity parameter is constant ( $\mu = 4$   
 10585 in both cases) and the locations of  $N$  the points are mutually independent and uniformly  
 10586 distributed. with each other. The key distinction is that  $N$  is random in the former and  
 10587 fixed in the latter.

10588 Another difference between the Poisson and binomial models is that if the state-space is  
 10589 divided into  $K$  disjunct regions, the number of points in each region  $n(B_k) : k = 1, \dots, K$ ;  
 10590 are independent and identically distributed (i.i.d.) under the Poisson model but not under  
 10591 the binomial model. In the Poisson case, the counts are  $n(B_k) \sim \text{Poisson}(\mu|B_k|)$ , where  
 10592  $|B_k|$  is the area of the region  $B_k$ . For the binomial model,  $n(B_k) \sim \text{Binomial}(N, \pi(B_k))$   
 10593 where  $\pi(B_k)$  is the proportion of the state-space in  $B_k$ ; however, these counts are not i.i.d.  
 10594 because the number of points in one region is informative about the number of points in  
 10595 another region. For example, if  $N = 10$  and if there are 7 points outside the region  $B_1$ ,  
 10596 then we can say with certainty that  $B_1 = 10 - 7 = 3$ .

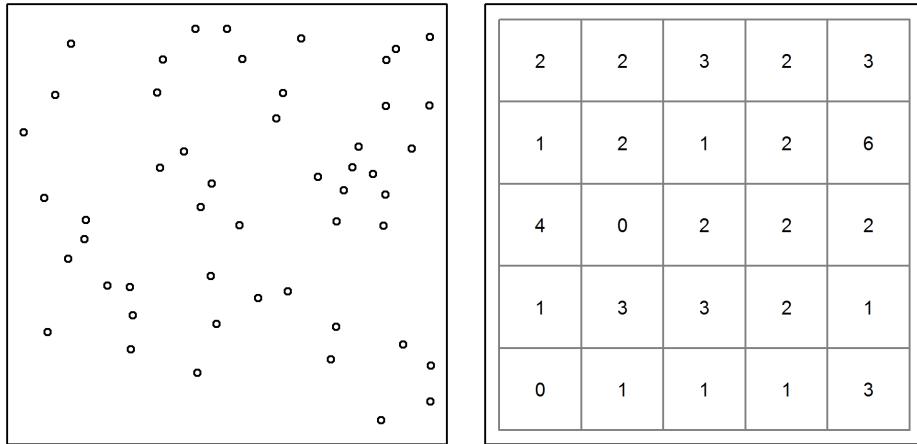
10597 Fig. 11.1 is meant to further illustrate the characteristics of the binomial model. The  
 10598 left panel shows a point pattern realized from a homogeneous binomial point process with  
 10599  $N = 50$ . The right panel shows the same realization, except that the state-space has  
 10600 been discretized into 25 equally-sized disjunct regions, or pixels, and the counts in each  
 10601 pixel are shown. Since the pixels are the same size, we have that  $\pi(B_k) = 1/25$ , and  
 10602 the expected number of point in each pixel is  $\mathbb{E}(n(B_k)) = N\pi(B_k) = 50/25 = 2$ , which  
 10603 happens to be the empirical mean in this instance. However, as previously stated, these  
 10604 counts are not independent realizations from a binomial distribution since  $\sum_k n(B_k) =$   
 10605  $N$ . Rather, the model for the entire vector is multinomial:  $\{n(B_1), n(B_2), \dots, n(B_k)\} \sim$   
 10606  $\text{Multinomial}(N, \{p(B_1), p(B_2), \dots, p(B_K)\})$  (Illian et al., 2008). If you need a refresher on  
 10607 the multinomial distribution, refer to Sec. 2.2.3, and consider the following R code, which  
 10608 generates counts similar to those seen in Fig. 11.1:

```

10609 > n.Bk <- rmultinom(1, size=50, prob=rep(1/25, 25))
10610 > matrix(n.Bk, 5, 5)
10611      [,1] [,2] [,3] [,4] [,5]
10612 [1,]    2    2    2    2    1
10613 [2,]    2    4    0    5    0
10614 [3,]    0    3    2    4    1
10615 [4,]    1    2    1    4    1
10616 [5,]    4    2    4    1    0

```

10617 The dependence among counts has virtually no practical consequence when the number  
 10618 of pixels is large. For example, if there are 100 pixels, the number of points in one  
 10619 pixels carries very little information about the expected number of points in another  
 10620 pixel. However, if there are only 2 pixels, then clearly the number of points in one pixel  
 10621 allows one to determine how many points will occur in the remaining pixel.



**Figure 11.1.** Homogeneous binomial point process with  $N=50$  points represented in continuous and discrete space.

10622 The discrete representation of space shown in Fig. 11.1 is not only helpful for under-  
 10623 standing the properties of a point process, it is also of practical importance when fitting  
 10624 SCR models because spatial covariates are almost always represented as rasters, i.e. grids  
 10625 with predetermined extent and resolution. In such cases, the definition of the prior for the  
 10626 point locations can be changed from the probability that a point occurs at some location  
 10627 in space to the probability that it occurs in some pixel of the raster. As we will explain  
 10628 in Sec. 11.4.2, this typically involves changing the prior from a uniform distribution to a  
 10629 multinomial or categorical distribution.

10630 Having sketched out the basic characteristics of homogeneous Poisson and binomial  
 10631 point process models, we will now review their relevance to SCR models before moving  
 10632 on to the inhomogeneous models. In a SCR model with a homogeneous point process,  
 10633 the intensity parameter  $\mu$  is interpreted as population density, and  $N$  is interpreted as  
 10634 population size (i.e. the number of activity centers in  $\mathcal{S}$ ). These interpretations are true  
 10635 regardless of whether we consider the Poisson model or the binomial model, but since  $N$   
 10636 is always unknown, one might wonder why we are discussing the binomial model at all.

10637 In our work, we typically adopt the binomial model simply because it is easy to  
 10638 implement using MCMC and data augmentation. And while  $N$  is truly unknown, we  
 10639 use an upper bound,  $M$ , which is fixed. Thus, the standard point process we use in  
 10640 Bayesian analyses can be regarded in two ways. First, it is a binomial point process with  
 10641  $M$  points. Second, in terms of  $N$ , it is a thinned binomial point process, where  $\psi$  is  
 10642 the thinning parameter. With this in mind, the only real difference between the Poisson  
 10643 and binomial models, as implemented in SCR contexts, is that in the former, we have  
 10644  $N \sim \text{Poisson}(\mu|\mathcal{S}|)$ , and in the latter we have  $N \sim \text{Binomial}(M, \psi)$ . In other words, we  
 10645 just have a different prior on  $N$ , and when using MCMC, the binomial prior is much more

convenient because it fixes the size of the parameter space and makes it easy to extend the model in each of the ways discussed in this book. It is also worth remembering that the Poisson distribution is the limit of the binomial distribution when  $M$  is very large and  $\psi$  is very small (Chapt. 2), and thus the two models are much more similar than may appear.

You might have noticed that the intensity parameter  $\mu$  was not shown for the binomial prior  $N \sim \text{Binomial}(M, \psi)$ . Instead, we see the data augmentation parameter  $\psi$ , which has been used throughout this book, but without much mention of the point process intensity. What then is the relationship between  $\psi$  and  $\mu$ ? As first discussed in Chapt.5, under data augmentation, the expected value of  $N$  is  $\mathbb{E}[N] = M\psi$ . But, from this chapter, we also know that the expected value of  $N$  can be written in terms of  $\mu$  as  $\mathbb{E}[N] = \mu|\mathcal{S}|$ . Therefore,  $\psi = \mu|\mathcal{S}|/M$  and hence we can directly estimate  $\mu$  rather than  $\psi$  if we want, as will be demonstrated in the next section where the objective is to model  $\mu$  as a function of spatially-referenced covariates. First, consider the following **R** code, which illustrates some the concepts we just covered:

```

10646 > Area <- 1                      # Area of state-space
10647 > M <- 100                         # Data augmentation size
10648 > mu <- 10                          # Intensity (points per area)
10649 > psi <- (mu*Area)/M                # Data augmentation parameter (thinning rate)
10650 > N <- rbinom(1, M, psi)            # Realized value of N under binomial prior
10651 > cbind(runif(N), runif(N))        # Point pattern from thinned binomial model
10652 [,1]      [,2]
10653 [1,] 0.52779588 0.84306878
10654 [2,] 0.11529168 0.80635046
10655 [3,] 0.06777632 0.66072116
10656 [4,] 0.18694649 0.56761245
10657 [5,] 0.30176929 0.03159091
10658 [6,] 0.84352724 0.89691452
10659 [7,] 0.52766808 0.08871199
10660 [8,] 0.73007529 0.63184825
10661 [9,] 0.01119023 0.69807029

```

## 11.2 INHOMOGENEOUS POINT PROCESSES

The principal difference between homogeneous and inhomogeneous point processes is that the intensity parameter  $\mu$  is allowed to vary spatially in the inhomogeneous model. Thus, rather than  $\mu$  being a fixed constant, it is now a function defined at each point  $\mathbf{x} \in \mathcal{S}$ . A vast number of options exist for modeling spatial variation in the intensity of a point process (Cox, 1955; Stoyan and Penttinen, 2000; Illian et al., 2008), but here we focus on modeling  $\mu$  as a function of spatially-referenced covariates and a vector of regression coefficients  $\beta$ ; a function we will denote  $\mu(\mathbf{x}, \beta)$ . To be clear,  $\mu(\mathbf{x}, \beta)$ , is a function that returns the expected density of activity centers at location  $\mathbf{x}$ , given the covariate values at  $\mathbf{x}$ <sup>1</sup>. Since the intensity must be positive, and because the natural logarithm is the

<sup>1</sup>The use of  $\mathbf{x}$  to denote any point in the state-space could cause confusion because we use  $\mathbf{x}_j$  as the location of a trap, but it is standard notation, and the distinction should be evident by the context.

10686 canonical link function of the Poisson generalized linear model (McCullagh and Nelder,  
 10687 1989), it is natural to consider the following model:

$$\log(\mu(\mathbf{x}, \boldsymbol{\beta})) = \beta_0 + \sum_{v=1}^V \beta_v C_v(\mathbf{x}) \quad (11.2.1)$$

10688 which says that there are  $V$  covariates and  $\beta_v$  is the regression coefficient for covariate  
 10689  $C_v(\mathbf{x})$ . This covariate,  $C_v(\mathbf{x})$ , could be any variable defined at all points in the state-  
 10690 space, such as habitat type or elevation. Eq. 11.2.1 should look familiar because it is  
 10691 the standard linear predictor used in Poisson regression. As with other GLMs, one could  
 10692 consider alternative link functions.

10693 Recall from the previous section that for a homogeneous point process, the expected  
 10694 number of points in the state-space was simply the intensity parameter multiplied by area:  
 10695  $\mathbb{E}(N) = \mu|\mathcal{S}|$ . But now that we are regarding the intensity as a function, rather than a  
 10696 scalar, this equation is not very useful. So what is  $\mathbb{E}(N)$  for an inhomogeneous point  
 10697 process? Contemplating a discrete state-space is useful for figuring this out. Imagine  
 10698 that the state-space is represented as a raster with many tiny pixels. In this case, we  
 10699 will associate  $\mathbf{x}$  with pixel ID, i.e.  $\mathbf{x}$  just references some pixel with  $V$  covariates values  
 10700 associated with it. The expected number of individuals in this pixel, say  $\mathbb{E}(n(\mathbf{x}))$ , can  
 10701 intuitively be found by evaluating the intensity function (Eq. 11.2.1) and multiplying it  
 10702 by the area of the pixel. In other words, we compute the expected number of individuals  
 10703 in a pixel by multiplying the expected value of density for that pixel by the area of the  
 10704 pixel. If we do this for each pixel in the state-space, then summing up these values gives  
 10705 us what we are after, the expected value of  $N$ . Specifically,  $\mathbb{E}(N) = \sum_{\mathbf{x} \in \mathcal{S}} \mathbb{E}(n(\mathbf{x}))$ . As  
 10706 the area of the pixels approaches zero, such that we move from discrete space back to  
 10707 continuous space, the summation must be replaced with an integration of the form:

$$\mathbb{E}(N) = \int_{\mathcal{S}} \mu(\mathbf{x}, \boldsymbol{\beta}) d\mathbf{x}. \quad (11.2.2)$$

10708 Together, Eqs. 11.2.1 and 11.2.2 describe a model for spatial variation in density as well  
 10709 as population size. The key task in fitting such inhomogeneous point process models is to  
 10710 estimate the  $\boldsymbol{\beta}$  parameters.

10711 We have now described an approach for modeling the point process intensity, yet in  
 10712 order to define the likelihood or to develop an MCMC algorithm for the inhomogeneous  
 10713 model, we need to specify the prior distribution for the activity centers. Recall that under  
 10714 the homogeneous point process, the prior was  $\mathbf{s}_i \sim \text{Uniform}(\mathcal{S})$ , for  $i = 1, \dots, N$ , or  
 10715 equivalently:

$$[\mathbf{s}_i] = 1/|\mathcal{S}| \quad (11.2.3)$$

10716 where, as before,  $|\mathcal{S}|$  is the area of the state-space. This simply indicates that an activity  
 10717 center is just as likely to occur at any location as another. However, if animals exhibit  
 10718 habitat selection or simply occur in one region more often than another, it would be  
 10719 preferable to replace this prior with one describing the spatial variation in density. Clearly  
 10720 this prior should be determined in some way by the spatially-varying intensity function  
 10721  $\mu(\mathbf{x}, \boldsymbol{\beta})$ . Since the integral of a probability density function (pdf) must be unity, we can  
 10722 convert  $\mu(\mathbf{x}, \boldsymbol{\beta})$  into a pdf by dividing it by a normalizing constant. In this case, the

normalizing constant is found by integrating  $\mu(\mathbf{x}, \boldsymbol{\beta})$  over the entire state-space. The probability density function of the new prior is therefore:

$$[\mathbf{s}_i | \boldsymbol{\beta}] = \frac{\mu(\mathbf{s}_i, \boldsymbol{\beta})}{\int_S \mu(\mathbf{x}, \boldsymbol{\beta}) d\mathbf{x}} \quad (11.2.4)$$

Substituting the uniform prior with this new distribution allows us to fit inhomogeneous binomial point process models to spatial capture-recapture data.

As a practical matter, note that the integral in the denominator of Eq. 11.2.4 is evaluated over space, and since we always regard space as two-dimensional (the state-space is planar), this is a two-dimensional integral that can be approximated using the methods discussed in Chapt. 9, which include Monte Carlo integration and Gaussian quadrature. Alternatively, if our state-space covariates are in raster format, i.e. they are in discrete space, the integral can be replaced with a summation over all the pixels in the raster,

$$[\mathbf{s}_i | \boldsymbol{\beta}] = \frac{\mu(\mathbf{s}_i, \boldsymbol{\beta})}{\sum_{\mathbf{x} \in S} \mu(\mathbf{x}, \boldsymbol{\beta})} \quad (11.2.5)$$

where  $s$  is now defined as “pixel ID” rather than a point in space.

Although the discrete space approach is standard practice, it is technically unjustified because covariate values must be known for all points in space, and a raster is simply a set of spatially-referenced covariate values at an evenly-spaced subset of points (the pixel centers). This same problem is present anytime that we have a sample of the spatial covariates, rather than a function defining their value for all points in space. In such cases, it may be necessary to interpolate the values of the covariates for points in space where they were not measured. One option would be to use a Kriging interpolator, as demonstrated by Rathbun (1996). Another option is to sample the spatial covariates using probabilistic sampling methods, which allow for design-based estimators of their values for the entire study area (Rathbun et al., 2007). Either option could be implemented within maximum likelihood or MCMC estimation methods; however, we do not demonstrate them here because it seems likely that they will be inconsequential in most cases where the raster data are of high resolution, such that the loss of information is negligible when going from continuous space to discrete space. Furthermore, the validity of this assertion, and the level of resolution required to adequately approximate continuous space can often be assessed by checking the consistency of the parameter estimates among varying levels of resolution, as was demonstrated in Chapt. 5.

We now have all the tools needed to fit inhomogeneous point process models. Likelihood-based inference for inhomogeneous Poisson point process models was described by Borchers and Efford (2008) and reviewed in Chapt. 6. Another example is demonstrated in the next section, but first we focus on the binomial model that we favor when conducting Bayesian inference. In the previous section we noted that the data augmentation parameter  $\psi$  can be expressed in terms of the intensity parameter  $\mu$ . The same is true for inhomogeneous models. Specifically, rather than  $\mathbb{E}(N) = \psi M$  as before, we use the expected value of  $N$  shown in Eq. 11.2.2 which results in

$$\psi = \frac{\int_S \mu(\mathbf{x}, \boldsymbol{\beta}) d\mathbf{x}}{M} \quad (11.2.6)$$

Note that the data augmentation limit  $M$  must be high enough so that it is greater than the numerator – i.e., the expected value of  $N$  must be less than  $M$ .

10762 In the next sections we walk through a few examples, building up from the simplest  
 10763 case where we actually observe the activity centers as though they were data. In the second  
 10764 example, we fit the inhomogeneous model to simulated data in which density is a function  
 10765 of a single continuous covariate. The next example shows an analysis in discrete space  
 10766 using both **secr** (Efford, 2011a) and **JAGS** (Plummer, 2003), and in the final example,  
 10767 we model the intensity of activity centers for a real dataset collected on jaguars (*Panthera*  
 10768 *onca*) in Argentina.

### 11.3 OBSERVED POINT PROCESSES

10769 In SCR models, the points (activity centers) are not directly observed, but in other contexts  
 10770 they are. Examples include the locations of disease outbreaks, the locations of trees  
 10771 in a forest, or the locations of radio-tracked animals. In such cases, it is straightforward  
 10772 to fit inhomogeneous point process models and estimate the parameters  $\beta$  from  
 10773 Eq. 11.2.1, as we will do in the following example.

Suppose we knew the locations of  $N$  animal activity centers, perhaps as the result of an extensive telemetry study. If we assume  $N$  is Poisson distributed and the points are mutually independent of one another, we can fit the inhomogeneous Poisson point process model. The likelihood of this model has two components:  $[\{\mathbf{s}_1, \dots, \mathbf{s}_N\}|N]$  and  $[N]$ . The pdf of the first part is given by Eq. 11.2.4, and with the Poisson assumption we have:

$$\begin{aligned}\mathcal{L}(\beta|\{\mathbf{s}_1, \dots, \mathbf{s}_N\}) &= [\{\mathbf{s}_1, \dots, \mathbf{s}_N\}|N][N] \\ &= \left\{ \prod_{i=1}^N \frac{\mu(\mathbf{s}_i, \beta)}{\int_S \mu(\mathbf{x}, \beta) d\mathbf{x}} \right\} \frac{e^{-\int_S \mu(\mathbf{x}, \beta) d\mathbf{x}} \int_S \mu(\mathbf{x}, \beta) d\mathbf{x}^N}{N!}.\end{aligned}$$

10774 This can be simplified by noting that the denominator in the first component of the model  
 10775 cancels with the corresponding piece in the numerator of the second component. And,  
 10776 since  $N$  is observed and thus does not depend on the parameters,  $N!$  can be omitted as  
 10777 well. After log-transforming the remaining pieces, we have the log-likelihood often seen in  
 10778 textbooks, such as Diggle (2003, pg. 104):

$$\ell(\beta|\{\mathbf{s}_i\}) = \sum_{i=1}^N \log(\mu(\mathbf{s}_i, \beta)) - \int_S \mu(\mathbf{x}, \beta) d\mathbf{x}.$$

10779 Having arrived at the likelihood we could choose a prior distribution for  $\beta$  and obtain the  
 10780 posterior distribution using Bayesian methods, or we can find the maximum likelihood  
 10781 estimates (MLEs) using standard numerical methods as is demonstrated below.

10782 First, we simulate some data under the model  $\mu(\mathbf{x}, \beta) = \beta_0 + \beta_1 \text{ELEV}(\mathbf{x})$ , where  
 10783  $\text{ELEV}(\mathbf{x})$  is a spatial covariate, say elevation, and  $\beta_0 = -6$  and  $\beta_1 = 1$ . It is worth  
 10784 emphasizing that a spatial covariate must be defined at any location in the state-space,  
 10785 as is true of the following covariate `elev.fn`:

```
10786 > elev.fn <- function(x) {           # spatial covariate
  10787 +   x <- matrix(x, ncol=2)          # Force x to be a matrix
  10788 +   (x[,1] + x[,2] - 100) / 40.8 # Returns (standardized) "elevation"
  10789 + }
```

```

10790 > # intensity function
10791 > mu <- function(x, beta0, beta1) exp(beta0 + beta1*elev.fn(x=x))
10792 > beta0 <- -6 # intercept of intensity function
10793 > beta1 <- 1 # effect of elevation on intensity
10794 > # Next line computes integral
10795 > EN <- cuhre(2, 1, mu, beta0=beta0, beta1=beta1,
10796 +           lower=c(0,0), upper=c(100,100))$value

```

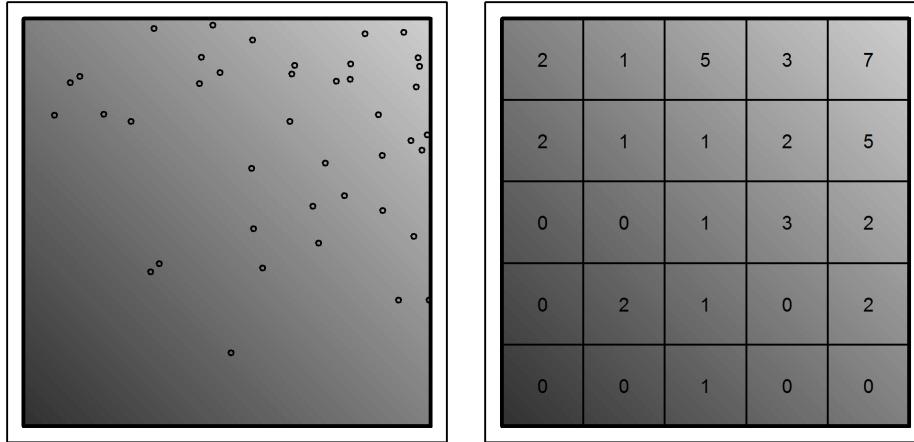
10797 The function `elev.fn` returns the value of elevation at any location  $x$ . The standardization bit is not necessary, but helps with the model fitting below. The next lines of the  
 10798 code define the intensity function  $\mu(x, \beta)$  in terms of elevation and the regression coefficients.  
 10799 The last line uses the `cuhre` function in the `R2Cuba` package (Hahn et al., 2010) to  
 10800 compute the expected value of  $N$  in a  $[0, 100] \times [0, 100]$  square state-space, which is the  
 10801 two-dimensional integral of Eq. 11.2.4. This integral could also be computed using a fine  
 10802 grid of points as we have done in previous chapters, but it is useful to gain familiarity  
 10803 with more efficient integration functions in **R**.

10804 The **R** code above demonstrates how to obtain the expected value of  $N$  given a spatial  
 10805 covariate and the coefficients defining the intensity function. Now we need to generate a  
 10806 realized value of  $N$  and distribute the  $N$  points in proportion to the intensity function.  
 10807 This is not as simple as it was to simulate data from a homogeneous point process because  
 10808 the points are no longer uniformly distributed within the state-space. Instead one must  
 10809 resort to methods such as rejection sampling, which involves simulating data from a stan-  
 10810 dard distribution and then accepting or rejecting each point using probabilities defined  
 10811 by the distribution of interest. For more information, readers should consult an accessible  
 10812 text such as Robert and Casella (2010). In our example, we simulate from a uniform dis-  
 10813 tribution and then accept or reject using the (scaled) probability density function  $[s_i | \beta]$   
 10814 (Eq. 11.2.4). The following **R** commands demonstrate the use of rejection sampling to  
 10815 simulate an inhomogeneous point process for the elevation covariate depicted in Fig. 11.2.

```

10817 > set.seed(31025)
10818 > beta0 <- -6 # intercept of intensity function
10819 > beta1 <- 1 # effect of elevation on intensity
10820 > # Next line computes integral, which is expected value of N
10821 > EN <- cuhre(2, 1, mu, beta0=beta0, beta1=beta1,
10822 +           lower=c(0,0), upper=c(100,100))$value
10823 > EN
10824 [1] 39.96634
10825 > N <- rpois(1, EN) # Realized N
10826 > s <- matrix(NA, N, 2) # This matrix will hold the coordinates
10827 > elev.min <- elev.fn(c(0,0))
10828 > elev.max <- elev.fn(c(100, 100))
10829 > Q <- max(c(exp(beta0 + beta1*elev.min),
10830 +               exp(beta0 + beta1*elev.max)))
10831 > counter <- 1
10832 > while(counter <= N) {
10833 +   x.c <- runif(1, 0, 100); y.c <- runif(1, 0, 100)
10834 +   s.cand <- c(x.c,y.c)

```



**Figure 11.2.** An example of a spatial covariate, say elevation, and a realization from an inhomogeneous Poisson point process with  $\mu(\mathbf{x}, \boldsymbol{\beta}) = \exp(\beta_0 + \beta_1 \text{ELEV}(\mathbf{x}))$  where  $\beta_0 = -6$  and  $\beta_1 = 1$ .

```

10835 +   pr <- mu(s.cand, beta0, beta1) #/ EN
10836 +   if(runif(1) < pr/Q) {
10837 +     s[counter,] <- s.cand
10838 +     counter <- counter+1
10839 +   }
10840 + }
```

10841 Similar methods are also implemented in the **R** package **spatstat** (Baddeley and Turner, 2005).

10843 The 41 simulated points are shown in Fig 11.2. High elevations are represented by  
 10844 light gray and low elevations are darker. The density of points in apparently higher in  
 10845 lighter regions suggesting that these simulated animals prefer high elevations. Given these  
 10846 points, we will now estimate  $\beta_0$  and  $\beta_1$  by minimizing the negative-log-likelihood using  
 10847 **R**'s **optim** function.

```

10848 > nll <- function(beta) {
10849 +   beta0 <- beta[1]
10850 +   beta1 <- beta[2]
10851 +   EN <- cuhre(2, 1, mu, beta0=beta0, beta1=beta1,
10852 +                     lower=c(0,0), upper=c(100,100))$value
10853 +   -(sum(beta0 + beta1*elev.fn(s)) - EN)
10854 + }
10855 > starting.values <- c(-10, 0)
10856 > fm <- optim(starting.values, nll, hessian=TRUE)
```

```

10857 > cbind(Est=fm$par, SE=sqrt(diag(solve(fm$hessian)))) # estimates and SEs
10858   Est      SE
10859 [1,] -5.9335547 0.2204693
10860 [2,]  0.9545532 0.1771507

```

Maximizing the Poisson likelihood took a fraction of a second, and we obtained estimates of  $\hat{\beta}_0 = -5.93$  and  $\hat{\beta}_1 = 0.95$ , which are very close to the data-generating values. The 95% confidence interval for  $\hat{\beta}_1$  is [0.61, 1.3] and since it does not include zero, the null hypothesis that  $\beta_1 = 0$ , i.e. that there is no effect of elevation on density, can be rejected. In addition to testing hypotheses, these results can be used to predict population size in new regions or create predicted density surface maps by plugging the parameter estimates into Eqs. 11.2.1 and 11.2.2.

You might wonder if the results would differ if we assumed a binomial rather than a Poisson distribution for  $N$ . This can be checked using the following code:

```

10870 > nllBin <- function(beta, M=100) {
10871   +   beta0 <- beta[1]
10872   +   beta1 <- beta[2]
10873   +   EN <- cuhre(2, 1, mu, beta0=beta0, beta1=beta1,
10874   +                     flags=list(verbose=0),
10875   +                     lower=c(0,0), upper=c(100,100))$value
10876   +   N <- nrow(s)
10877   +   psi <- EN/M
10878   +   -(sum(beta0 + beta1*elev.fn(s) - log(EN)) +
10879   +     dbinom(N, M, psi, log=TRUE))
10880   + }
10881 > cbind(Est=fmBin$par, SE=sqrt(diag(solve(fmBin$hessian)))) # est and SE
10882   Est      SE
10883 [1,] -5.9339490 0.1965479
10884 [2,]  0.9545742 0.1771962

```

which indicates that the MLEs are almost identical, and supports the claim that the prior on  $N$  has little influence in SCR models. Notice, however, that the standard error for  $\beta_0$  is smaller under the binomial model than it was under the Poisson model – a difference that will dissipate as  $M$  tends toward infinity.

This example demonstrates that if we had the data we wish we had, i.e. if we knew the coordinates of the activity centers, we could easily estimate the parameters governing the underlying point process and make inferences about spatial variation in density and abundance. Unfortunately, in virtually all animal ecology studies, the locations of the  $N$  animals, or the  $N$  activity centers, cannot be directly observed. Thus we need extra information to estimate the locations of these unobserved points, which in the case of SCR, comes from the locations where each animal is captured.

## 11.4 FITTING INHOMOGENEOUS POINT PROCESS SCR MODELS

### 11.4.1 Continuous space

In this example, we will use the same set of points simulated in the previous section to generate spatial capture-recapture data. Specifically, we overlay a grid of 49 traps on the

map shown in Fig. 11.2 and simulate capture histories conditional on the activity centers. Then, we will attempt to estimate the activity center locations as though we did not know where they were, as is the case in real applications. We will also estimate  $\beta_0$  and  $\beta_1$  as before and see how the estimates compare when the points are not actually observed. The following **R** code simulates encounter histories under a Poisson observation model (see Chapt. 9), which could be appropriate in camera trapping studies or when using other methods in which animals could be detected multiple times at a trap during a single occasion.

```

10897 > xsp <- seq(20, 80, by=10); len <- length(xsp)
10900 > X <- cbind(rep(xsp, each=len), rep(xsp, times=len)) # traps
10901 > ntraps <- nrow(X); nooccasions <- 5
10902 > y <- array(NA, c(N, ntraps, nooccasions)) # capture data
10903 > sigma <- 5 # scale parameter
10904 > lam0 <- 1 # basal encounter rate
10905 > lam <- matrix(NA, N, ntraps)
10906 > set.seed(5588)
10907 > for(i in 1:N) {
10908 +   for(j in 1:ntraps) {
10909 +     # The object "s" was simulated in previous section
10910 +     distSq <- (s[i,1]-X[j,1])^2 + (s[i,2] - X[j,2])^2
10911 +     lam[i,j] <- exp(-distSq/(2*sigma^2)) * lam0
10912 +     y[i,j,] <- rpois(nooccasions, lam[i,j])
10913 +   }
10914 + }
```

Now that we have a simulated capture-recapture dataset  $y$ , we can simulation the posterior distributions of the model parameters using MCMC. A commented Gibbs sampler written in **R** is available in the accompanying **R** package **scrbook** (see **?scrIPP**). This function is not meant to be an all purpose tool for fitting SCR models using MCMC. Instead, it is presented so that interested readers can better understand the computational aspects of the problem and can modify it for their purposes. The function can be used as SO:

```

10930 > fm1 <- scrIPP(y, X, M=150, 10000, xlims=c(0,100), ylims=c(0,100),
10931 +   space.cov=elev.fn, tune=c(0.4, 0.2, 0.3, 0.3, 7))
10932 > plot(mcmc(fm1$out))
```

which requests 10000 posterior samples and estimates the effect of the spatial covariate, elevation, on density. The argument **space.cov** accepts any spatial covariate that returns a real value for any location in the rectangular state-space defined by the **xlims** and **ylims** arguments. Currently, the function places uniform priors on the parameters  $\sigma$ ,  $\lambda_0$ ,  $\beta_0$  and  $\beta_1$ , although this could easily be modified. The **tune** argument specifies the tuning parameters used in the Metropolis-within-Gibbs steps of the algorithm. These should be chosen using trial and error to achieve an acceptance rate of between 0.4 and 0.6, roughly. See Chapt. 17 for more details about MCMC.

Results of the analysis are shown in Fig. 11.3 and Table 11.1. Fig. 11.3 displays the trace plots of the Markov chains as well as the posterior distributions for three parameters.

10943 The chains appear to converge rapidly but may need to be run longer to reduce Monte  
 10944 Carlo error. Summaries of the posterior distributions are presented in Table 11.1. The  
 10945 posterior means for  $\beta_0$  and  $\beta_1$  are quite similar to MLEs from the analysis in the previous  
 10946 section in which we assumed no observation error. However, we see that the confidence  
 10947 intervals are wider. With respect to the other parameters in the model, we see that all of  
 10948 the data generating parameter fall within the 95% credible intervals. One thing to note  
 10949 is that, although the point estimates for the expected and realized values of  $N$  are quite  
 10950 similar, the posterior for the realized value of  $N$  is more precise. This is to be expected  
 10951 because the uncertainty associated with the realized value of  $N$  is entirely determined by  
 10952 the sampling error. That is, if we could perfectly detect all of the individuals in  $S$ , there  
 10953 would be no uncertainty about  $N$ . In contrast, the variance for expected value of  $N$  is  
 10954 composed both process error and sampling error. See Chapt. 5 and Efford and Fewster  
 10955 (2012) for additional discussion on the difference between realized and expected values of  
 10956 abundance.

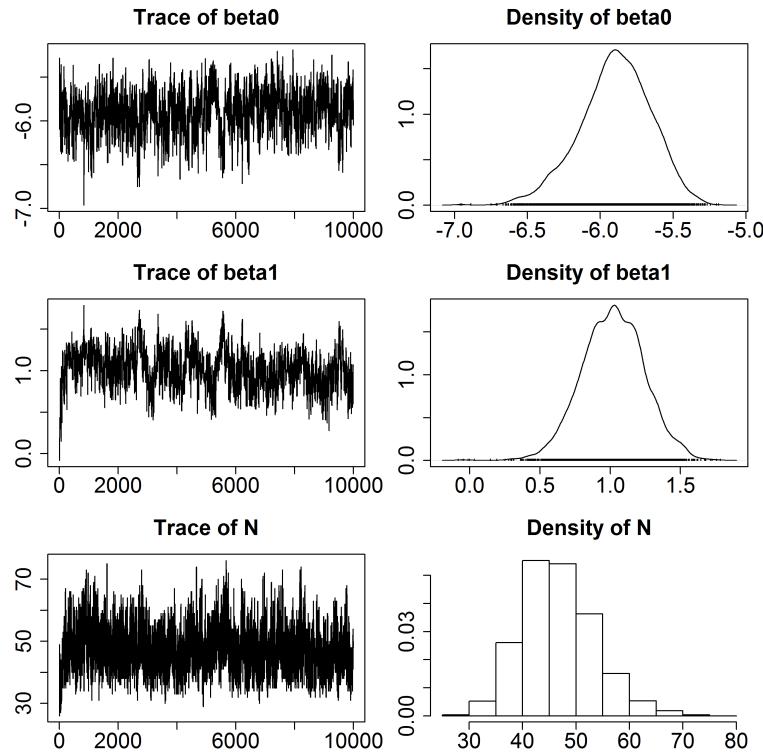
10957 Fitting continuous space inhomogeneous point process models is somewhat difficult in  
 10958 **BUGS** because the “IPP” prior  $[s_i|\beta]$ , unlike the uniform prior, is not one of the available  
 10959 distributions that comes with the software. It is possible to add new distributions in  
 10960 **BUGS**, but it is somewhat cumbersome. **secr** allows users to fit continuous space models  
 10961 using linear or polynomial functions of the easting and northing coordinates, but it does  
 10962 not accept truly continuous covariates that are functions of space. However, these are  
 10963 not really important limitations because discrete space versions of the model are straight-  
 10964 forward, and virtually all spatial covariates are, or can be, defined as such.

**Table 11.1.** Summary of posterior distributions from SCR model with inhomogeneous point process.

Parameter	Mean	SD	2.5%	97.5%
$\sigma = 5$	5.232	0.310	4.681	5.858
$\lambda_0 = 1$	0.802	0.119	0.595	1.049
$\beta_0 = -6$	-5.856	0.254	-6.376	-5.393
$\beta_1 = 1$	0.985	0.209	0.575	1.378
$N = 41$	47.615	8.041	35.000	66.000
$E(N) = 39.9$	47.551	10.992	29.837	71.332

### 10965 11.4.2 Discrete space

10966 To fit inhomogeneous point process models using covariates in discrete space, i.e. in raster  
 10967 format, we follow the same steps as outlined in Chapt. 9 – we define  $s_i$  as pixel ID,  
 10968 and we use the categorical distribution as a prior. This effectively changes the problem  
 10969 from estimating the coordinates of an activity center, to estimating the pixel in which an  
 10970 activity center is located. As pixel size approaches zero, these two become equivalent. A  
 10971 good example is found in (Mollet et al., In review). Here we present an analysis of the  
 10972 simulated data shown in the Fig. 11.2. The spatial covariate, let’s call it forest canopy  
 10973 height (CANHT), was simulated using using the code shown on the help page `ch11` in  
 10974 `scrbook`. The points are the number of activity centers in each pixel, generated from



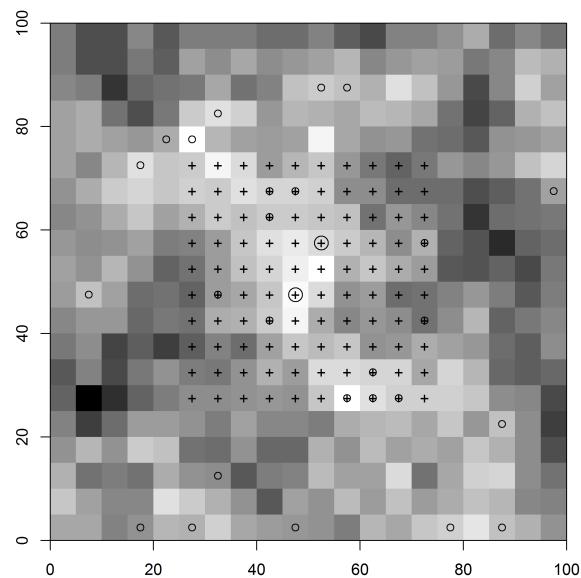
**Figure 11.3.** Trace plots and posterior distributions from MCMC analysis of SCR model with inhomogeneous point process. Analysis was conducted using the `scrIPP` function in the accompanying **R** package `scrbook`.

10975 a single realization of the inhomogeneous point process model with intensity  $\mu(\mathbf{x}, \boldsymbol{\beta}) =$   
 10976  $\exp(\beta_0 + \beta_1 \text{CANHT}(\mathbf{x})) \times \text{pixelArea}$ , where  $\beta_0 = -6$  and  $\beta_1 = 1$ .

10977 The **BUGS** description of the model is shown in panel 11.1. The vector `probs[]` is  
 10978 the prior probability defined by Eq. 11.2.5, which is the probability that an individual's  
 10979 activity center is located at pixel  $\mathbf{x}$ . `grid` is the matrix of coordinates for each pixel.

10980 This model can also be fit in `secr`, which refers to the raster data as a "habitat mask".  
 10981 The habitat mask is essentially a `data.frame` with attributes. The `data.frame` itself has  
 10982 2 columns for the coordinates of each of the pixel centers. The attributes of the object  
 10983 include information such as the area of the pixels and the spacing between pixel centers.  
 10984 If there are covariates, these too are stored as an attribute of the habitat mask, and are  
 10985 formatted as a `data.frame` with 1 row per pixel and 1 column per covariate. Once the  
 10986 data have been formatted correctly, fitting the model in `secr` is as simple as:

```
10987 > secr1 <- secr.fit(ch, model=D~canht, mask=msk)
```



**Figure 11.4.** Simulated activity centers in discrete space. The spatial covariate, canopy height, is highest in the lighter areas and density increases with canopy height. A single activity center is shown as a small circle, and larger circles represent two activity centers in a pixel. Trap locations are shown as crosses.

**Table 11.2.** Comparison of **secr** and **JAGS** results. Point estimates from the Bayesian analysis are posterior means. Intervals are lower and upper 95% CIs.

Parameter	Truth	Software	Mean	SD	2.5%	97.5%
$\lambda_0$	1.00	<b>JAGS</b>	1.04	0.087	0.88	1.22
	1.00	<b>secr</b>	1.08	0.089	0.92	1.27
$\sigma$	10.00	<b>JAGS</b>	10.16	0.373	9.46	10.94
	10.00	<b>secr</b>	9.84	0.350	9.18	10.55
$\beta_1$	1.00	<b>JAGS</b>	1.20	0.350	0.50	1.88
	1.00	<b>secr</b>	1.09	0.316	0.47	1.71
$N$	30.00	<b>JAGS</b>	26.63	2.585	23.00	33.00
	30.00	<b>secr</b>	28.19	3.037	24.49	37.39
$\mathbb{E}(N)$	32.30	<b>JAGS</b>	26.39	5.048	17.25	36.96
	32.30	<b>secr</b>	28.19	6.117	18.52	42.93

10988 where `D~canht` indicates that we want to model density as a function of canopy height,  
 10989 which is defined in the `msk` object. R code to format the data and fit the models using  
 10990 **secr** and **JAGS** is available in `scrbook`, found by issuing the command: `help(ch11secr-jags)`.

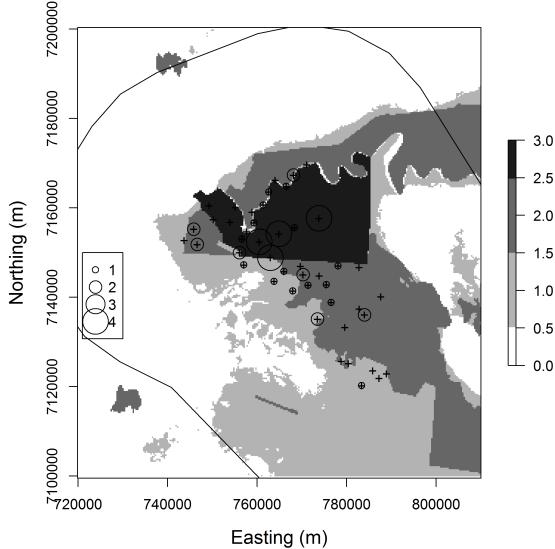
10991 Results of fitting the model in **JAGS** and **secr** are shown in Table 11.2 and are similar  
 10992 as expected. The differences that do exist are likely due to the differences in Bayesian and  
 10993 frequentist estimation methods, as discussed in Chapt 3. Either answer may be “more  
 10994 correct” depending upon one’s criteria for correctness!

## 11.5 ARGENTINA JAGUAR STUDY

10995 Estimating density of large felines has been a priority for many conservation organizations,  
 10996 but few robust methodologies existed before the advent of SCR. Distance sampling is not  
 10997 feasible for such rare and cryptic species, and traditional capture-recapture methods yield  
 10998 estimates that are highly sensitive to the subjective choice of the effective survey area.  
 10999 SCR models provide a powerful alternative because density can be estimated directly and  
 11000 data can be collected using non-invasive methods such as camera traps or hair snares.

11001 In this example, we demonstrate how readily density can be estimated for a globally  
 11002 imperiled species using SCR. Furthermore, we show how inhomogeneous point process  
 11003 models can be used to test important hypotheses regarding the factors affecting density.  
 11004 The data come from an 8-year camera-trapping study designed to assess the impacts  
 11005 of poaching on jaguar density in Argentina, near the borders of Brazil and Paraguay.  
 11006 Additional information about the study is presented in Paviolo et al. (2008) and Paviolo  
 11007 et al. (2009). The expected effect of poaching is a decline in jaguar density due to the direct  
 11008 removal of individuals and the depletion of its main prey species. To conserve jaguars and  
 11009 related species, protected areas have been established and three levels of protection are  
 11010 recognized, as depicted in Fig. 11.5. The dark gray area is the Iguazú National Park that is  
 11011 patrolled regularly by law enforcement officials. The medium gray areas are not protected  
 11012 and rarely patrolled. Finally, the light gray areas are large soybean monocultures, cities  
 11013 and dams which provide no suitable habitat for jaguars

11014 To test for differences in density between the three regions, we modeled the point  
 11015 process intensity parameter as a function of protection status (PROTECT), which we



**Figure 11.5.** Jaguar detections at 46 camera trap stations. The three levels of protection status are no protection (light gray), some protection (gray), and Iguazú National Park (dark gray). Non-habitat (soybean monocultures) is shown in white.

treated as an ordinal variable:

$$\mu(\mathbf{x}, \boldsymbol{\beta}) = \exp(\beta_0 + \beta_1 \text{PROTECT}(\mathbf{x})) \times \text{pixelArea}.$$

We predicted that  $\beta_1$  would be greater than zero, indicating that jaguar density increases with protection status. In addition to modeling spatial variation in density, we also modeled the scale parameter of the half-normal (or Gaussian) encounter model as sex-specific because male cats typically have larger home ranges than females (Sollmann et al., 2011). Since sex is an individual-specific covariate, and not observed for the individuals that were not captured, a prior distribution is required for the sex of uncaptured individuals. We used a Bernoulli prior with probability 0.5 to describe our uncertainty about sex ratio. Another equivalent option is to augment the data with an equal number of males and females and let the MCMC algorithm determine which of these individuals are actually members of the population.

An additional unique aspect of this study is the highly irregular state-space. Unlike in the examples of simulated data, the geometry of this state-space is not a simple rectangular region. Instead, it is the area south of the Iguazú River, which runs along the northern border of the park shown in dark green in Fig. 11.5, and it excludes the large soybean monocultures. Fitting models in highly convoluted spatial regions raises the question: How does one integrate Eq. 11.2.4 over this irregular space? Earlier we used the function `cuhre`

**Table 11.3.** Summaries of posterior distributions from the model of jaguar density.  $\sigma$  is the scale parameter of the half-normal detection function.  $\lambda_0$  is baseline encounter rate,  $\beta_1$  is the effect of protection status on jaguar density,  $\rho$  is the sex-ratio,  $N$  is population size. The last three parameters are the density estimates (jaguars/100 km<sup>2</sup>) for the three levels of protection.

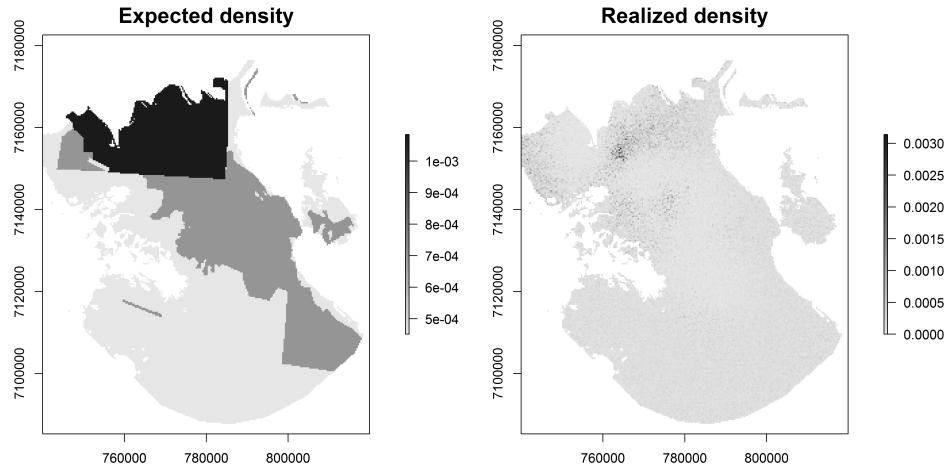
	Mean	SD	2.5%	97.5%
$N$	35.819	7.9749	23.0000	54.0000
$D_{\text{low}}$	0.906	0.3265	0.3813	1.6682
$D_{\text{med}}$	0.770	0.2841	0.2698	1.4392
$D_{\text{high}}$	1.370	0.3069	0.8315	1.9955
$\sigma_{\text{female}}$	5501.204	876.8774	4142.2756	7578.5692
$\sigma_{\text{male}}$	6452.570	915.3623	4970.3215	8505.5219
$\lambda_0$	0.006	0.0016	0.0034	0.0098
$\psi$	0.355	0.0937	0.1998	0.5638
$\beta_0$	-4.686	0.2602	-5.2346	-4.2129
$\beta_1$	0.174	0.3500	-0.5104	0.8649
Sex Ratio	0.489	0.0550	0.3824	0.6000

in R for the two-dimensional integration, but its `lower` and `upper` arguments essentially assume that the state-space is square. There are methods of transforming the state-space that might allow us to work around this problem, but once again we find that it is most convenient to work in discrete space and sum over all the pixels defining  $\mathcal{S}$ .

We fit the model to data from a single year in which 46 camera stations were operational, each consisting of a pair of cameras placed along roads or small trails. Forty-five detections of 16 jaguars (8 males and 8 females) were made over a 95-day sampling period. The mean number of sampling days at each camera station was 48.2. The raw capture data shown in Fig. 11.5 suggest that the highest number of captures was in the national park, but there were also several traps in the park with no captures. Furthermore, few cameras were placed far from the protected areas, making it somewhat difficult to detect differences in density. R code to fit the model is available in `scrbook` on the help page `jaguarDataCh11`. Parameter estimates are shown in Table 11.3.

The results indicate that efforts to protect jaguars by reducing poaching in protected areas are not working as well as hoped for. The posterior probability that  $\beta_1 > 0$  was only 0.69, and the posterior mean of realized density was only 51% higher in the national park than in the unprotected area. Fig. 11.6 shows the estimated density surfaces. The first map is the expected density in each of the three values, which was computed by plugging in the posterior mean values of  $\beta_0$  and  $\beta_1$  into the log-linear intensity function. The second map is the realized density surface – the conditional-on- $N$  probability distribution of the number of activity centers in each pixel of the rasterized state-space. The expected values would be used if we were interested in making inferences about other areas or time periods, whereas the realized map is the best description of the system during the study period.

We note that there is room for improvement in our analysis, and our results should be considered preliminary. The political boundaries used to demarcate protected areas are not as concrete as we might like. In reality poaching pressure is likely higher near remote park boundaries than in well-guarded park interiors. One option for addressing this would be to use a continuous measure of poaching pressure such as distance from the nearest



**Figure 11.6.** Estimated density (activity centers / pixel) surfaces from the analysis of the jaguar data.

town, or some other accessibility metric. It would also be worthwhile to model density separately for each sex because many of the detections outside of the park were of males, and thus it is possible that the sexes use habitat differently (Conde et al., 2010). Other extensions warranting investigation include treating PROTECT as a categorical rather than ordinal variable, and assessing the effects of roads and trails on jaguar movement using the methods described in Chapt. 12. Developing models for these extensions could be readily accomplished by modifying the fitting functions found in the **R** package **scrbook**.

## 11.6 SUMMARY AND OUTLOOK

One of the distinguishing features of spatial capture-recapture models is that they allow for inference about spatial variation in density without relying on ad hoc approaches for determining the amount of area surveyed. The approach described in this chapter involves modeling the locations of activity centers as outcomes of an inhomogeneous point process with intensity determined by covariates defined at all locations in the state-space. Covariate effects can be evaluated in exactly the same way as is done in generalized linear models, making it easy to interpret the results.

All the examples in this section included a single state-space covariate, but this was for simplicity only. Including multiple covariates poses no additional challenges. Similarly, additional model structure such sex-specific encounter rate parameters or behavioral responses can be accommodated and fit using **secr**, **BUGS**, or by extending the functions in **scrbook**. It is also possible to consider covariates that affect both density and ecological distance as will be described in the next chapter. The ramifications of this are enormous for applied ecological research and conservation efforts because researchers can use capture-recapture data to identify areas where both density and landscape connectiv-

11084      ity are high (Royle et al., 2013). Addressing such questions is simply not possible using  
11085      standard, non-spatial capture-recapture methods.

11086      Although we focused on modeling the point process intensity as a function of covariates,  
11087      other options for fitting inhomogeneous models exist (Illian et al., 2008). Cox processes are  
11088      models in which the point process intensity is a function of spatial random effects. Such  
11089      methods are useful for accommodating overdispersion, but it seems unlikely that most SCR  
11090      datasets could support such complexity. Gibbs processes are another important class of  
11091      models that are distinguished by the interactions of points. Although little work has been  
11092      done on such models in the context of SCR studies (Reich et al., 2012), we expect they  
11093      will receive more attention because they can be used to model processes such territoriality  
11094      (points repel one another) or aggregation (points attract one another). Neyman-Scott  
11095      processes are another option for modeling aggregation or clustering, and could be useful  
11096      for studying gregarious species.

---

```

model{
  sigma ~ dunif(0, 20)
  lam0 ~ dunif(0, 5)
  beta0 ~ dunif(-10, 10)
  beta1 ~ dunif(-10, 10)
  for(j in 1:nPix) {
    mu[j] <- exp(beta0 + beta1*CANHT[j])*pixArea
    probs[j] <- mu[j]/EN
  }
  EN <- sum(mu[]) # Expected value of N, E(N)
  psi <- EN/M
  for(i in 1:M) {
    z[i] ~ dbern(psi)
    s[i] ~ dcat(probs[])
    x0g[i] <- grid[s[i],1]
    y0g[i] <- grid[s[i],2]
    for(j in 1:ntraps) {
      dist[i,j] <- sqrt(pow(x0g[i]-traps[j,1],2) + pow(y0g[i]-traps[j,2],2))
      lambda[i,j] <- lam0*exp(-dist[i,j]*dist[i,j]/(2*sigma*sigma)) * z[i]
      y[i,j] ~ dpois(lambda[i,j])
    }
  }
  N <- sum(z[]) # Realized value of N
}

```

---

Panel 11.1: **BUGS** model specification for the inhomogeneous point process model in discrete space. A nearly equivalent formulation would involve omitting  $\beta_0$  and modeling the expected number of activity centers as  $\mathbb{E}(N) = M\psi$  with  $\psi \sim \text{Uniform}(0, 1)$ .



11097  
11098

# 12

11099

## MODELING LANDSCAPE CONNECTIVITY

11100 Every spatial capture-recapture model that we have considered so far has expressed en-  
11101 counter probability as a function of the Euclidean distance between individual activity  
11102 centers  $s$  and trap locations  $x$ . As a practical matter, models based on Euclidean distance  
11103 imply circular, symmetric, and stationary home ranges of individuals, and these are not  
11104 often biologically realistic. While these simple encounter probability models will often be  
11105 sufficient for practical purposes, especially in small data sets, sometimes developing more  
11106 complex models of the detection process as it relates to space usage of individuals will  
11107 be useful. Animals may not judge distance in terms of Euclidean distance but, rather,  
11108 according to the configuration of habitat patches, quality of local habitat, perceived mor-  
11109 tality risk, and other considerations. Together, the degree to which these factors facilitate  
11110 or impede movement determines landscape connectivity (Tischendorf and Fahrig, 2000),  
11111 which is widely recognized to be an important component of population viability (With  
11112 and Crist, 1995; Compton et al., 2007). Moreover, because encounter probability and  
11113 the distance metric upon which it is based represent outcomes of individual movements  
11114 about their home range, ecologists might have explicit hypotheses about how environmen-  
11115 tal variables affect the distance metric, and it is therefore desirable to incorporate these  
11116 hypotheses directly into SCR models so that they may be formally evaluated statistically.

11117 Although much theory has been developed to predict the effects of decreasing con-  
11118 nectivity, few empirical studies have been conducted to test these predictions due to the  
11119 paucity of formal methods for estimating connectivity parameters (Cushman et al., 2010;  
11120 Hanks and Hooten, in press). Instead, ecologists often rely on expert opinion or *ad hoc*  
11121 methods of specifying connectivity values, even in important applied settings (Adriaensen  
11122 et al., 2003; Beier et al., 2008; Zeller et al., 2012). In addition, no methods are available  
11123 for simultaneously estimating population density and connectivity parameters, in spite of  
11124 theory predicting interacting effects of density and connectivity on population viability  
11125 (Tischendorf et al., 2005; Cushman et al., 2010). In this chapter, following Royle et al.  
11126 (2013), we provide a framework for modeling landscape connectivity using SCR models,  
11127 by parameterizing models for encounter probability based on “ecological distance”. A  
11128 natural candidate framework for modeling ecological distance is the least-cost path which

is used widely in landscape ecology for modeling connectivity, movement and gene flow (Adriaensen et al., 2003; Manel et al., 2003; McRae et al., 2008). In practical applications, variables that influence landscape connectivity, or the effective cost of moving across the landscape, include things like highways (e.g., Epps et al., 2005), elevation (Cushman et al., 2006), ruggedness (Epps et al., 2007), snow cover (Schwartz et al., 2009), distance to escape terrain (Shirk et al., 2010), range limitations (McRae and Beier, 2007), or distance from urban areas, highways, human disturbance or other factors that animals might avoid.

Royle et al. (2013) provided an SCR framework based on least-cost path for modeling landscape connectivity. They parameterized encounter probability based not on Euclidean distance but, rather, on the least-cost path between an individual's activity center and a trap location. This is parameterized in terms of one or more parameters that relate the *resistance* of the landscape to explicit covariates. In this way, SCR models can explicitly accommodate landscape structure and account for connectivity of the landscape. Using this methodological extension of SCR models, it is possible to make formal statistical inferences about movement and connectivity from capture-recapture studies that generate sparse individual encounter history data without subjective prescription of resistance or cost surfaces. While we believe there should be much ecological interest in developing SCR models that account for landscape connectivity, it is also important for obtaining more accurate estimates of density; under simple models of landscape connectivity, incorrectly fitting the basic model SCR0 produces substantial bias in estimates of  $N$  and hence density (Royle et al., 2013).

## 12.1 SHORTCOMINGS OF EUCLIDEAN DISTANCE MODELS

In the standard SCR models, encounter probability is modeled as a function of Euclidean distance. For example, using the binomial observation model (Chapt. 5), let  $y_{ij}$  be individual- and trap-specific binomial counts with sample size  $K$  and probabilities  $p_{ij}$ . The Gaussian model is

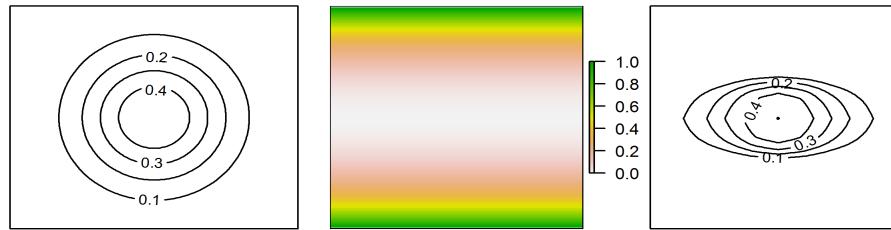
$$p_{ij} = p_0 \exp(-d_{ij}^2/(2\sigma^2)) \quad (12.1.1)$$

where  $d_{ij} = ||\mathbf{x}_j - \mathbf{s}_i||$  is Euclidean distance. As usual, we will sometimes adopt the log-scale parameterization based on  $\log(p_{ij}) = \alpha_0 + \alpha_1 d_{ij}^2$  where  $\alpha_0 = \log(p_0)$  and  $\alpha_1 = -1/(2\sigma^2)$ .

The main problem with the Euclidean distance metric in this encounter probability model is that it is unaffected by habitat or landscape structure, and it implies that the space used by individuals is stationary and symmetric, which may be unreasonable assumptions for some species. By stationary we mean in the formal sense of invariance to translation. That is, the properties of an individual home range centered at some point  $\mathbf{s}$  are exactly the same as any other point say  $\mathbf{s}'$ . As an example, if the common detection model based on a bivariate normal probability distribution function is used, then the implied space usage by *all* individuals, no matter their location in space or local habitat conditions, is symmetric with circular contours of usage intensity.

In the framework of Royle et al. (2013), SCR models explicitly incorporate information about the landscape so that a unit of distance is variable depending on identified covariates, say  $C(\mathbf{x})$ . Thus, where an individual lives on the landscape, and the state of the surrounding landscape, will determine the character of its usage of space. In particular, they suggest distance metrics, based on least-cost path, that imply irregular, asymmet-

ric and non-stationary home ranges of individuals. As an example, Fig. 12.1 shows a typical symmetric home range (left panel), and a compressed home range (right panel) resulting from the effect of an environmental variable (center panel) on an animal's movement behavior. We might think of the environmental variable as representing an elevation gradient of a valley and so, for a species that avoids high elevation, space usage will be concentrated in flatter terrain at lower elevations and therefore producing the elliptical home range shape.



**Figure 12.1.** A symmetric home range (left), a habitat variable (center) such as representing an elevation gradient, and a non-symmetric home range (right) resulting from the cost imposed on movement by the habitat variable.

## 12.2 LEAST-COST PATH DISTANCE

We adopt a cost-weighted distance metric here which defines the effective distance between points by accumulating pixel-specific costs determined using a cost function defined by the user. The idea of cost-weighted distance to characterize animal use of landscapes is widely used in landscape ecology for modeling connectivity, movement and gene flow (Beier et al., 2008). For reasons of computational tractability we consider a discrete landscape defined by a raster of some prescribed resolution. The distance between any two points  $\mathbf{x}$  and  $\mathbf{x}'$  can be represented by a sequence of line segments connecting neighboring pixels, say  $\mathbf{l}_1, \mathbf{l}_2, \dots, \mathbf{l}_m$ . Then the cost-weighted distance between  $\mathbf{x}$  and  $\mathbf{x}'$  is

$$d(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{m-1} \text{cost}(\mathbf{l}_i, \mathbf{l}_{i+1}) \|\mathbf{l}_i - \mathbf{l}_{i+1}\| \quad (12.2.1)$$

where  $\text{cost}(\mathbf{l}_i, \mathbf{l}_{i+1})$  is the user-defined cost to move from pixel  $\mathbf{l}_i$  to neighboring pixel  $\mathbf{l}_{i+1}$  in the sequence. Given the cost of each pixel, it is a simple matter to compute the cost-weighted distance between any two pixels, along *any* path, simply by accumulating the incremental costs weighted by distances. In the context of spatial capture-recapture models (and, more generally, landscape connectivity) we are concerned with the *minimum* cost-weighted distance, or the *least-cost path*, between any two points which we will denote by  $d_{lcp}$ , which is the sequence  $\mathcal{P} = (\mathbf{l}_1, \mathbf{l}_2, \dots, \mathbf{l}_m)$  that minimizes the objective function

11193 defined by Eq. 12.2.1. That is,

$$d_{lcp}(\mathbf{x}, \mathbf{x}') = \min_{\mathcal{P}} \sum_{i=1}^{m-1} \text{cost}(\mathbf{l}_i, \mathbf{l}_{i+1}) \|\mathbf{l}_i - \mathbf{l}_{i+1}\| \quad (12.2.2)$$

11194 The least-cost path distance can be calculated in many geographic information systems  
11195 and other software packages, including the **R** package **gdistance** (van Etten, 2011) which  
11196 we use below.

11197 The key ecological aspect of least-cost path modeling is the development of models  
11198 for pixel-specific cost. A natural approach is to model cost as a function of one or more  
11199 covariates defined on every pixel of the according raster. For example, using a single  
11200 covariate  $C(\mathbf{x})$  we define the cost of moving from some pixel  $\mathbf{x}$  to neighboring pixel  $\mathbf{x}'$  as

$$\log(\text{cost}(\mathbf{x}, \mathbf{x}')) = \alpha_2 \left( \frac{C(\mathbf{x}) + C(\mathbf{x}')}{2} \right) \quad (12.2.3)$$

11201 Thus, if  $\alpha_2 = 0$  then substituting  $\text{cost}(\mathbf{x}, \mathbf{x}') = \exp(0) = 1$  into Eq. 12.2.2 will produce the  
11202 ordinary Euclidean distance between points. Here we assume the covariate  $C$  is positive-  
11203 valued, and we constrain  $\alpha_2 \geq 0$  so as to avoid negative costs. While not necessarily  
11204 problematic from a mathematical standpoint, negative costs are unrealistic biologically.

11205 The use of least-cost path models to model landscape connectivity has been around  
11206 for a long time. And, although  $\alpha_2$  is rarely known, conservation biologists design link-  
11207 ages that require this resistance value as input (see Beier et al., 2008, and articles cited  
11208 therein). However, formal inference (e.g., estimation) of parameters is not often done. In-  
11209 stead, in many existing applications of least-cost path analysis, the parameter  $\alpha_2$  is fixed  
11210 by the investigator, or based on expert opinion (Beier et al., 2008), although recently  
11211 researchers have begun to define costs based on resource selection functions<sup>1</sup>, animal  
11212 movement (Tracy, 2006; Fortin et al., 2005), or genetic distance data (e.g., Gerlach and  
11213 Musolf (2000); Epps et al. (2007); Schwartz et al. (2009)).

11214 To formalize the use of cost-weighted distance in SCR models, we substitute Eq. 12.2.2  
11215 for Euclidean distance in the expression for encounter probability (Eq. 12.1.1) and maxi-  
11216 mize the resulting likelihood (see below). In doing so, we can directly estimate parameters  
11217 of the least-cost path model, evaluate how landscape covariate influence connectivity, and  
11218 test explicit hypotheses about these things using only individual level encounter history  
11219 data from capture-recapture studies.

### 11220 12.2.1 Example of Computing Cost-weighted distance

11221 As an example of the cost-weighted distance calculation consider the following landscape  
11222 comprised of 16 pixels with unit spacing identified as follows, along with the pixel-specific  
11223 cost:

<small>11224</small>	<small>pixel ID</small>	<small>Cost</small>
<small>11225</small>	4 8 12 16	100 1 1 1
<small>11226</small>	3 7 11 15	100 100 1 1

<sup>1</sup>We address the integration of resource selection models based on telemetry data with SCR models in Chapt. 13.

```
11227      2 6 10 14          100 100 100 1
11228      1 5   9 13         100 100   1 1
```

11229 We assume the scale is such that the distance between neighboring pixels in any cardinal  
 11230 direction is 1 unit, and the distance between neighbors on a diagonal is  $\sqrt{2}$  units. We  
 11231 assigned low cost of 1 to “good habitat” pixels (or pixels we think of as “highly connected”  
 11232 by virtue of being in good habitat) and, conversely, we assign high cost (100) to “bad  
 11233 habitat”. This simple cost raster is shown in Fig. 12.2. The **R** commands for creating  
 11234 this simple example are as follows (which can be run using the **R** script **SCRed** – see the  
 11235 help file for that):

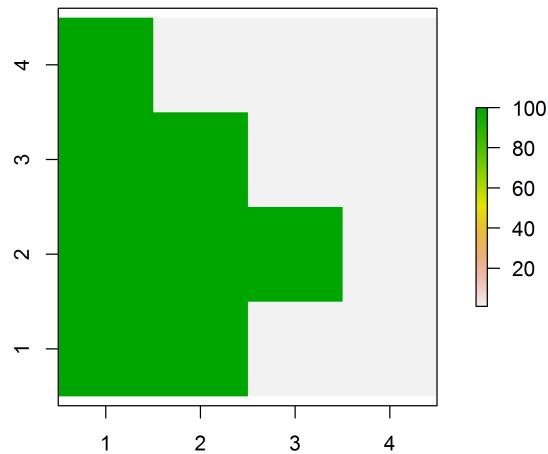
```
11236 > library(raster)
11237 > library(gdistance)
11238 > r <- raster(nrows=4,ncols=4)
11239 > projection(r) <- "+proj=utm +zone=12 +datum=WGS84" # Sets the projection
11240 > extent(r) <- c(.5,4.5,.5,4.5) #sets the extent of the raster
11241 > costs1 <- c(100,100,100,100,1,100,100,100,1,1,100,1,1,1,1,1)
11242 > values(r) <- matrix(costs1,4,4,byrow=FALSE) #assign the costs to the raster
11243 > par(mfrow=c(1,1))
11244 > plot(r)
```

11245 This produces Fig. 12.2.

11246 For this simple case we can easily compute the shortest cost-weighted distance between  
 11247 any pixels “by eye”. For example, the shortest cost-weighted distance between pixels 5  
 11248 and 9 in this example is 50.5 units:  $1 * (100 + 1)/2 = 50.5$ , the shortest distance between  
 11249 pixels 4 and 8 is also 50.5, while the shortest cost-distance between 4 and 12 is 51.5. What  
 11250 is the shortest distance between 7 and 16? Suppose an individual at pixel 7 can move  
 11251 diagonal (which has distance  $\sqrt{2}$ ) and pay  $\sqrt{2}(100 + 1)/2$ , and then move once to the  
 11252 right to pay 1 additional unit cost, for a total of 72.4. However, if the individual instead  
 11253 moved one unit to the right, to pixel 11, and then diagonally, the total cost is 51.914  
 11254 which is the minimum cost-weighted distance in getting from pixel 7 to 16. These two  
 11255 ways of moving from 7 to 16 have the same Euclidean distance, but different cost-weighted  
 11256 distances according to our cost function.

11257 The least-cost path distances can be computed with just a few **R** commands, and  
 11258 these commands can be inserted directly into the likelihood construction for an ordinary  
 11259 spatial capture-recapture model. The **R** package **gdistance** calculates least-cost path using  
 11260 Dijkstra’s algorithm (Dijkstra, 1959) (from the **igraph** package (Csardi and Nepusz,  
 11261 2006)). To compute the least-cost path, or the minimum cost-weighted distances between  
 11262 every pixel and every other pixel, we make use of the helper function **transition**, which  
 11263 calculates the cost of moving between neighboring pixels. It operates on the inverse-scale  
 11264 (“conductance”), and so the **transitionFunction** argument is given as  $1/\text{mean}(x)$ . The  
 11265 function **geoCorrection** modifies this object depending on the projection of the coor-  
 11266 dinate system (e.g., it corrects for curvature of the earth’s surface if longitude/latitude  
 11267 coordinates are used). The result is fed into the function **costDistance** to compute the  
 11268 pair-wise distance matrix. For that, we define the center points of each raster, here these  
 11269 are just integers on  $[1, 4] \times [1, 4]$ . The commands altogether are as follows:

```
11270 > tr1 <- transition(r,transitionFunction=function(x) 1/mean(x),directions=8)
```



**Figure 12.2.** A  $4 \times 4$  raster depicting a binary cost surface, with cost = 1 (white) or 100 (shaded) to represent ease of movement across a pixel.

```

11271 > tr1CorrC <- geoCorrection(tr1,type="c",multpl=FALSE,scl=FALSE)
11272 > pts <- cbind( sort(rep(1:4,4)),rep(4:1,4))
11273 > costs1 <- costDistance(tr1CorrC,pts)
11274 > outD <- as.matrix(costs1)

```

11275 Now we can look at the result and see if it makes sense to us. Here we produce the  
 11276 first 5 columns of this distance matrix to illustrate a couple of examples of calculating the  
 11277 minimum cost-weighted distance between points:

```

11278 > outD[1:5,1:5]
11279      1       2       3       4       5
11280 1 0.0000 100.0000 200.0000 205.2426 100.0000
11281 2 100.0000 0.0000 100.0000 200.0000 141.4214
11282 3 200.0000 100.0000 0.0000 100.0000 126.1604
11283 4 205.2426 200.0000 100.0000 0.0000 105.2426
11284 5 100.0000 141.4214 126.1604 105.2426 0.0000

```

11285 An interesting case is that between point 1 and 4. Note that simply taking the shortest  
 11286 Euclidean distance, weighted by cost, produces a cost-weighted distance of  $100 \times 1$  to

move from pixel 1 to pixel 2, and similarly from 2 to 3 and 3 to 4, producing a total cost-weighted distance of 300. However, the actual *least-cost path* has cost-weighted distance 205.2426. See if you can figure out the shortest path by inspection.

The key point here is that, once we can compute this distance matrix, we can use it as the distance matrix in computing the encounter probability between activity centers and traps, and we can use our existing MLE technology (Chapt. 6) to fit models that are based on ecological distance.

### 12.3 SIMULATING SCR DATA USING ECOLOGICAL DISTANCE

Royle et al. (2013) simulated capture-recapture data such that landscape connectivity was governed by a cost function having a single covariate, and they considered two hypothetical covariate landscapes (Fig. 12.3). The landscape here is a  $20 \times 20$  pixel raster, with extent =  $[0.5, 4.5] \times [0.5, 4.5]$ . For example, think of each pixel as representing, say, a  $1 \times 1$  km grid cell with something like “percent developed” or “trail/road density” representing the covariate. For sampling by capture-recapture, imagine that 16 camera traps are established at the integer coordinates  $(1, 1), (1, 2), \dots, (4, 4)$ . The two covariates were constructed as follows (see `?make.EDcovariates` for the R commands): First is an increasing trend from the NW to the SE (“systematic covariate”), where  $C(\mathbf{x})$  is defined as  $C(\mathbf{x}) = \text{row}(\mathbf{x}) + \text{col}(\mathbf{x})$  and  $\text{row}(\mathbf{x})$  and  $\text{col}(\mathbf{x})$  are just the row and column, respectively, of the raster. This might mimic something related to distance from an urban area or a gradient in habitat quality due to land use, or environmental conditions such as temperature or precipitation gradients. In the second case, the covariate was generated using spatially correlated noise to emulate a typical patchy habitat covariate (“patchy covariate”) such as tree or understory density.

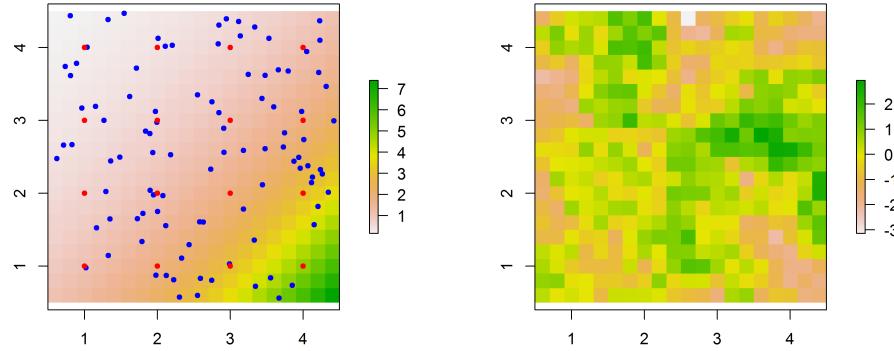
For both covariates we use a cost function in which transitions from pixel  $\mathbf{x}$  to  $\mathbf{x}'$  is given by:

$$\log(\text{cost}(\mathbf{x}, \mathbf{x}')) = \alpha_2 \left( \frac{C(\mathbf{x}) + C(\mathbf{x}')}{2} \right)$$

where  $\alpha_2 = 1$  for simulating the observed data. Remember that with  $\alpha_2 = 0$  the model reduces to one in which the cost of moving across each pixel is constant, and therefore Euclidean distance is operative. In the left panel of Fig. 12.3, a sample realization of  $N = 100$  activity centers is shown. While encounter probability is assumed to be related to landscape connectivity according to the single-variable cost function, individual activity centers are assumed to be uniformly distributed, although we can modify this assumption (See Sec. 12.8 below).

When distance is defined by the cost-weighted distance metric given by Eq. 12.2.2 then individual space-usage varies spatially in response to the landscape covariate(s) used in the distance metric. As a consequence, home range contours are no longer circular, as in SCR models based on Euclidean distance. For example, using one of the covariates we use in our simulation study below (Fig. 12.3, right panel) with a Gaussian encounter probability model but having distance metric defined by Eq. 12.2.2, produces home ranges such as those shown in Fig. 12.4.

To simulate data, we have to load the `scrbook` package and call the function `make.EDcovariates` to generate our raster covariates (see the help file for how that is done). We process the covariate into a least-cost path distance matrix, and then simulate observed encounter

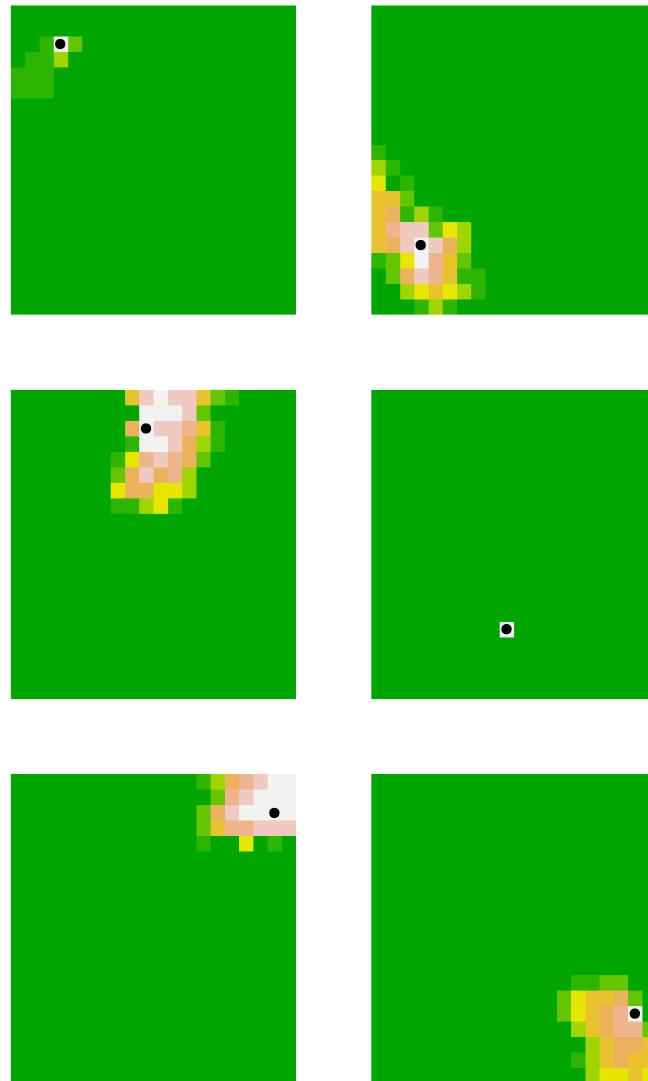


**Figure 12.3.** Two covariates (defined on a  $20 \times 20$  grid) used in simulations. Left panel shows a covariate with systematic structure meant to mimic distance from some feature, and the right panel shows a “patchy” covariate. A hypothetical realization of  $N = 100$  activity centers (blue dots) is superimposed on the left figure, along with 16 trap locations.

11328 data using standard methods which we have used many times previously in this book.  
 11329 The complete set of **R** commands is:

```

11330 ### Grab a covariate
11331 > library(scrbook)
11332 > set.seed(2013)
11333 > out <- make.EDcovariates()
11334 > covariate <- out$covariate.patchy
11335
11336 ### prescribe some settings
11337 > N <- 200
11338 > alpha0 <- -2
11339 > sigma <- .5
11340 > alpha1 <- 1/(2*sigma*sigma)
11341 > alpha2 <-1
11342 > K <- 5
11343 > S <- cbind(runif(N,.5,4.5),runif(N,.5,4.5))
11344
11345 # make up some trap locations
11346 > xg <- seq(1,4,1); yg<-4:1
11347 > traplocs <- cbind( sort(rep(xg,4)),rep(yg,4))
11348 > points(traplocs,pch=20,col="red")
11349 > ntraps <- nrow(traplocs)
```



**Figure 12.4.** Typical home ranges for 6 individuals based on the cost surface shown in the right panel of Fig. 12.3 with  $\alpha_2 = 1$ . The black dot indicates the home range center and the pixels around each home range center are shaded according to the probability of encounter, if a trap were located in that pixel.

```

11350
11351 ### make a raster and fill it up with the "cost"
11352 > r <- raster(nrows=20,ncols=20)
11353 > projection(r) <- "+proj=utm +zone=12 +datum=WGS84"
11354 > extent(r) <- c(.5,4.5,.5,4.5)
11355 > cost <- exp(alpha2*covariate)
11356
11357 ### compute least-cost path distance
11358 > tr1 <- transition(cost,transitionFunction=function(x) 1/mean(x),directions=8)
11359 > tr1CorrC <- geoCorrection(tr1,type="c",multpl=FALSE,scl=FALSE)
11360 > D <- costDistance(tr1CorrC,S,traplocs)
11361 > probcap <- plogis(alpha0)*exp(-alpha1*D*D)
11362
11363 # now generate the encounters of every individual in every trap
11364 # discard uncaptured individuals
11365 > Y <- matrix(NA,nrow=N,ncol=ntraps)
11366 > for(i in 1:nrow(Y)){
11367 +   Y[i,] <- rbinom(ntraps,K,probcap[i,])
11368 + }
11369 > Y <- Y[apply(Y,1,sum)>0,]

```

## 12.4 LIKELIHOOD ANALYSIS OF ECOLOGICAL DISTANCE MODELS

11370 Throughout much of this book we rely on Bayesian analysis by MCMC mostly using  
 11371 **BUGS**, but sometimes (as in Chapt. 17) developing our own implementations. However,  
 11372 occasionally we prefer to use likelihood estimation, such as when we can compare a set  
 11373 of models directly by likelihood either to do a direct hypothesis test of a parameter, or  
 11374 to tabulate a bunch of AIC values. For the class of models that use least-cost path, we  
 11375 also prefer likelihood methods not because they have any conceptual or methodological  
 11376 benefit, but simply because they are more computationally efficient to implement (Royle  
 11377 et al., 2013).

11378 There are no technical considerations in adapting our formulation of maximum likeli-  
 11379 hood estimation (Borchers and Efford, 2008) from Chapt. 6 for the class of models based  
 11380 on least-cost path (see the appendix in Royle et al. (2013) for complete details). Likeli-  
 11381 hood analysis is really just a straightforward adaptation in which we replace the Euclidean  
 11382 distance with least-cost path. Consider the Bernoulli model in which the individual- and  
 11383 trap-specific observations have a binomial distribution conditional on the latent variable  
 11384  $\mathbf{s}_i$ :

$$y_{ij} | \mathbf{s}_i \sim \text{Binomial}(K, p_{\boldsymbol{\alpha}}(d_{lcp}(\mathbf{x}_j, \mathbf{s}_i; \boldsymbol{\alpha}_2); \boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1)) \quad (12.4.1)$$

11385 where we have indicated the dependence of  $p$  on the parameters  $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2)$ , and also  
 11386  $d_{lcp}$  which itself depends on  $\boldsymbol{\alpha}_2$ , and the latent variable  $\mathbf{s}_i$ . We note that the only difference  
 11387 between likelihood analysis of this model and the standard Bernoulli model, is the use of  
 11388  $d_{lcp}$  here. For the random effect we have  $\mathbf{s}_i \sim \text{Uniform}(\mathcal{S})$ , we can easily compute the  
 11389 integrated (marginal) likelihood of an encounter history. The likelihood is given in the  
 11390 **scrbook** package as the function **intlik3ed**. The help file provides an example of its usage  
 11391 and for simulating data. To use this function the cost covariate  $C(\mathbf{x})$  has to be of class

**Table 12.1.** Summary output of fitting models based on Euclidean and least-cost path distance to simulated data using the `intlik3ed` function (see `?intlik3ed`). Data were simulated based on the least-cost path model using the “patchy” covariate shown in Fig. 12.3.

Distance metric	-loglik	$\alpha_0$	$\alpha_1$	$\log(n_0)$	$\alpha_2$
True value		-2	2	4.644	1
Euclidean	133.495	-1.885	1.247	3.549	—
Least-cost path (truth)	70.119	-1.780	2.471	4.459	0.046

11392 `RasterLayer` which requires packages `sp` and `raster` to manipulate.

#### 11393 12.4.1 Example of SCR with least-cost path

11394 Now we use the **R** function `nlm` along with our `intlik3ed` function to obtain the MLEs  
 11395 of the model parameters for the data simulated in Sec. 12.3. We’ll do that for both the  
 11396 standard Euclidean distance and then for the ecological distance based on the “patchy”  
 11397 covariate using the following commands:

```
11398 > frog1<-nlm(intlik3ed,c(alpha0,alpha1,3)),hessian=TRUE,y=Y,K=K,X=traplocs,  

  11399   distmet="euclid",covariate=covariate,alpha2=1)  

  11400  

  11401 > frog2<-nlm(intlik3ed,c(alpha0,alpha1,3,-.3),hessian=TRUE,y=Y,K=K,X=traplocs,  

  11402   distmet="ecol",covariate=covariate,alpha2=NA)
```

11403 The summary output for the two model fits is shown in Table 12.1. The model based  
 11404 on least-cost path (the data generating model) appears to be much preferred in terms  
 11405 of negative log-likelihood. The output parameter order is  $\alpha_0, \alpha_1, \log(n_0)$ , and  $\log(\alpha_2)$   
 11406 (remember, we want to keep  $\alpha_2$  positive, so its logarithm is estimated). The data gener-  
 11407 ating parameter values were  $\alpha_0 = -2$ ,  $\alpha_1 = 2$  and  $\log(\alpha_2) = 0$ . The simulated sampling  
 11408 produced a sample of 96 individuals and so the number of individuals not captured is  
 11409  $n_0 = 104$ , and  $\log(n_0) = 4.64$ . We see that the MLEs of the least-cost path model are  
 11410 pretty close whereas they are not so close under the misspecified model based on Euclidean  
 11411 distance.

## 12.5 BAYESIAN ANALYSIS

11412 While implementation of these ecological distance SCR models is reasonably straightfor-  
 11413 ward, the model cannot be fitted in the **BUGS** engines because least-cost path distance  
 11414 cannot be computed. It would be possible to fit the models in **BUGS** if the parameter  $\alpha_2$   
 11415 was fixed. In that case, one could compute the least-cost distance matrix ahead of time  
 11416 and reference the required elements for a given `s`. Alternatively, it would be possible to  
 11417 write a custom MCMC routine using the methods we present in Chapt. 17, although we  
 11418 have not yet developed our own MCMC implementation of SCR models with ecological  
 11419 distance metrics.

## 12.6 SIMULATION EVALUATION OF THE MLE

11420 Royle et al. (2013) carried-out a limited simulation study to evaluate the general statistical  
 11421 performance of the density estimator under this new model, the effect of mis-specifying  
 11422 the model with a normal Euclidean distance metric, and evaluate the general bias and  
 11423 precision properties of the MLE using the systematic and patchy landscapes shown in Fig.  
 11424 12.3. Their results showed extreme bias in estimates of  $N$  when the misspecified Euclidean  
 11425 distance is used, and only negligible small-sample bias of 3-5% in the MLE of  $N$  using  
 11426 the least-cost distance which becomes negligible as the expected sample size increases  
 11427 (either due to increasing  $K$ , or larger population sizes). The performance of estimating  
 11428 the other parameters, including the cost parameter  $\alpha_2$  mirrors the results for estimating  
 11429  $N$ . We reproduce a subset of the results from Royle et al. (2013) in Table 12.6 in order  
 11430 to highlight some key points.

**Table 12.2.** Simulation results for estimating population size  $N$  for a prescribed state-space with  $N = 100$  or  $N = 200$  and various levels of replication ( $K$ ) using the “patchy” landscape shown in Fig. 12.3. For each simulated data set, the SCR model was fitted by maximum likelihood with standard Euclidean distance (“euclid”), or least-cost path (“lcp”), which was the true data-generating model. The summary statistics of the sampling distribution reported are the mean, standard deviation (“SD”) and quantiles (0.025, 0.50, 0.975).

		N=100				
		mean	SD	0.025	0.50	0.975
$K = 3$						
euclid		78.68	18.12	49.40	76.34	125.47
lcp		110.96	28.65	69.55	106.98	181.84
$K = 5$						
euclid		77.85	11.55	59.17	77.44	101.14
lcp		104.44	15.79	78.38	101.47	139.55
$K = 10$						
euclid		78.01	5.26	68.00	77.96	87.81
lcp		100.42	7.56	86.72	100.34	115.47
		N=200				
$K = 3$						
euclid		154.34	33.74	107.00	146.34	221.43
lcp		208.77	49.29	141.68	197.89	325.77
$K = 5$						
euclid		153.39	15.57	129.31	149.54	185.38
lcp		200.91	20.78	164.42	200.47	246.46
$K = 10$						
euclid		156.27	8.51	142.17	156.05	174.55
lcp		198.45	11.44	180.06	198.04	219.52

## 12.7 DISTANCE IN AN IRREGULAR PATCH

11431 We provide another illustration of how to employ ecological distance calculations in SCR  
 11432 models. This example is meant to mimic a situation where we have something like a hard  
 11433 habitat boundary such as a habitat corridor or park unit or some other block of relatively  
 11434 homogeneous good-quality habitat for some species. This particular system (shown in  
 11435 Fig. 12.5) could be habitat surrounded by a suburban wasteland of McDonuts and Beer-  
 11436 Marts, much less hospitable habitat for most species. For our purposes, we suppose that  
 11437 individuals live within the buffered “f-shaped” region, although we could also imagine the  
 11438 negative of the situation in which individuals live outside of the region, so that the polygon  
 11439 represents a barrier (a lake) or bad habitat (an urban area) or similar. We describe the  
 11440 steps for creating this landscape shortly, so that you can use a similar process to generate  
 11441 more relevant landscapes for your own problems.

11442 In this case we’re not going to estimate any parameters of the cost function (though  
 11443 you could adapt the analyses of the previous sections to do that) but instead we’re going  
 11444 to use ecological distance ideas only to constrain movement within (or to avoid) landscape  
 11445 features. Note that, normally, distance “as the crow flies” would not be suitable for  
 11446 irregular habitat patches such as that shown in Fig. 12.5.

### 11447 12.7.1 Basic Geographic Analysis in R

11448 In practical applications our landscape will contain polygons which delineate good or bad  
 11449 habitat or other important characteristics of the landscape. These might exist as GIS  
 11450 shapefiles or merely as a text file with coordinates defining polygon boundaries. To work  
 11451 with polygons in the context of SCR models we need to create a raster, overlay the polygon  
 11452 and assign values to each pixel depending on whether pixels are in the polygon or not,  
 11453 or how far they are from polygon boundaries. These operations are relatively easy to do  
 11454 within a GIS system but we need to be able to do them in **R** in order to compute the  
 11455 least-cost paths needed in the likelihood evaluation. Some additional geographic analyses  
 11456 have been discussed in Sec. 17.7 where we talked about reading in the shapefile and doing  
 11457 SCR analyses with it.

11458 Often we will have GIS shapefiles that define polygons but, here, we create a set of  
 11459 polygons by buffering and joining some line segments. In the **R** package **scrbook**, we  
 11460 provide a function **make.seg** which allows you to make such line segments given a specific  
 11461 trap region. To use **make.seg** we first create a plot region and then call **make.seg** which  
 11462 has a single argument being the number of points used to define the line segment. The user  
 11463 will click on the visual display until the required number of points has been obtained by  
 11464 **make.seg**. In the following set of commands we generate two line segments, **l1** consisting  
 11465 of 9 points and **l2** consisting of 5 points, and these reside in a geographic region enclosed  
 11466 by  $[0, 10] \times [0, 10]$ :

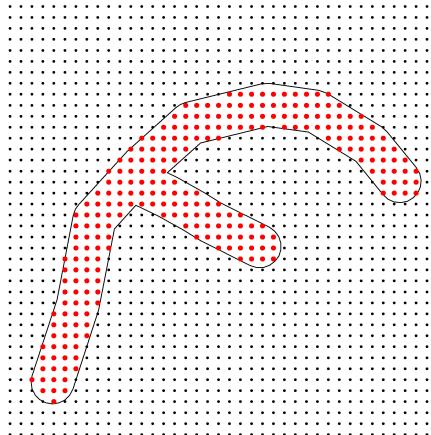
```
11467 > library(scrbook)
11468 > library(sp)
11469 > plot(NULL, xlim=c(0,10), ylim=c(0,10))
11470 > l1 <- make.seg(9)
11471 > plot(l1)
11472 > l2 <- make.seg(5)
```

```
11473 > plot(11)
11474 > lines(12)
```

11475 We used this function as above to create a habitat corridor composed of line segments  
 11476 of class **SpatialLines** from the **R** package **sp**. The corridor can be loaded from **scrbook**  
 11477 by typing the command **data(fakecorridor)**. This data list has 2 line files in it (11 and  
 11478 12) and a trap locations file (**traps**). We use some functions from the **R** packages **sp** and  
 11479 **rgeos** to join and buffer (by 0.5 units) the two segments. The commands are as follows  
 11480 and the result is shown in Fig. 12.5.

```
11481 > data(fakecorridor)
11482 > library(sp)
11483 > library(rgeos)
11484
11485 > buffer <- 0.5
11486 > par(mfrow=c(1,1))
11487 > aa <- gUnion(11,12)
11488 > plot(gBuffer(aa,width=buffer),xlim=c(0,10),ylim=c(0,10))
11489 > pg <- gBuffer(aa,width=buffer)
11490 > pg.coords <- pg@polygons[[1]]@Polygons[[1]]@coords
11491
11492 > xg <- seq(0,10,,40)
11493 > yg <- seq(10,0,,40)
11494
11495 > delta <- mean(diff(xg))
11496 > pts <- cbind(sort(rep(xg,40)),rep(yg,40))
11497 > points(pts,pch=20,cex=.5)
11498
11499 > in pts <- point.in.polygon(pts[,1],pts[,2],pg.coords[,1],pg.coords[,2])
11500 > points(pts[in.pts==1,],pch=20,col="red")
```

11501 In this example, we're not going to estimate parameters of the cost function. Instead,  
 11502 the point is to compute ordinary Euclidean distance but restricted by the boundaries of  
 11503 the corridor (or patch geometry in general) and thus not distance "as the crow flies." To  
 11504 do this, we imagine that animals will tend to severely avoid leaving the buffered habitat  
 11505 zone. Therefore, we assign **cost** = 1 if a pixel is within the buffer, and **cost** = 10000 if a  
 11506 pixel is outside of a buffer. Therefore the cost to move to a neighboring pixel outside of the  
 11507 buffered area is 5000.5 compared to the cost of 1 to move to a neighboring pixel inside the  
 11508 buffer. With this cost specification, we can compute the least-cost path distance matrix  
 11509 one time and modify our likelihood code to accept the distance matrix as input. We give  
 11510 that likelihood in the package **scrbook** as the function **intlik3edv2**. We note also that  
 11511 this function accepts a habitat mask in the form of a vector of 0's and 1's that define any  
 11512 potential state-space restrictions. i.e., 1 if the pixel is an element of the state-space and 0  
 11513 if it is not, and so additional modifications to the geometry of the region could be made.  
 11514 However, in the analysis of this simulated data set, we define the state-space to be the  
 11515 buffered corridor system. Here we simulate a population of  $N = 200$  individuals in the  
 11516 corridor system and so we restrict our state-space accordingly for purposes of fitting the



**Figure 12.5.** A fake wildlife corridor or reserve. The boundary outlines a polygon of suitable habitat surrounded by suburban development.

model. However we encourage you to refit the model without the state-space restriction (for fitting the model only) and then compare the results. The code for doing all of this is in the help file for `intlik3edv2`, which contains the likelihood function and sample **R** script (`?intlik3edv2`).

```

11517  model. However we encourage you to refit the model without the state-space restriction
11518  (for fitting the model only) and then compare the results. The code for doing all of this
11519  is in the help file for intlik3edv2, which contains the likelihood function and sample R
11520  script (?intlik3edv2).
11521  ### Define the cost structure
11522  > cost <- rep(NA,nrow(pts))
11523  > cost[in pts==1]<-1      # low cost to move among pixels but not 0
11524  > cost[in pts!=1]<-10000 # high cost
11525
11526  ### Stuff costs into a raster
11527  > library("raster")
11528  > r <- raster(nrows=40,ncols=40)
11529  > projection(r) <- "+proj=utm +zone=12 +datum=WGS84"
11530  > extent(r) <- c(0-delta/2,10+delta/2,0-delta/2,10+delta/2)
11531  > values(r) <- matrix(cost,40,40,byrow=FALSE)
11532
11533  # check what it looks like
11534  > plot(r)

```

```

11535 > points(pts,pch=20,cex=.4)
11536
11537 # compute ecological distances:
11538 > library("gdistance")
11539 > tr1 <- transition(r,transitionFunction=function(x) 1/mean(x),directions=8)
11540 > tr1CorrC <- geoCorrection(tr1,type="c",multpl=FALSE,scl=FALSE)
11541 > costs1 <- costDistance(tr1CorrC,pts)
11542 > outD <- as.matrix(costs1)

```

11543 In the next block of code we simulate some data and then fit a model to the simulated  
 11544 data. Note that the object `traps` is loaded with `data(fakecorridor)` along with the data  
 11545 which define the f-shaped patch in Fig. 12.5:

```

11546 > library(scrbook)
11547 > traplocs <- traps$loc
11548 > trap.id <- traps$locid
11549 > ntraps <- nrow(traplocs)
11550
11551 > set.seed(2013)
11552 > N <- 200
11553 > S.possible <- (1:nrow(pts))[in pts==1]
11554 > S.id <- sample(S.possible,N,replace=TRUE)
11555 > S <- pts[S.id,]
11556
11557 > Dtraps <- outD[trap.id,]
11558 > Deuclid <- e2dist(pts[trap.id,],pts)
11559
11560 > alpha0 <- -1.5
11561 > sigma <- 1.5
11562 > alpha1 <- 1/(2*sigma*sigma)
11563 > K <- 10
11564
11565 > probcap <- plogis(alpha0)*exp(-alpha1*D*D)
11566 > Y <- matrix(NA,nrow=N,ncol=ntraps)
11567 > for(i in 1:nrow(Y)){
11568 +   Y[i,] <- rbinom(ntraps,K,probcap[i,])
11569 > }
11570 > Y <- Y[apply(Y,1,sum)>0,]
11571
11572 > frog1 <- nlm(intlik3edv2,c(-2.5,2,log(4)),hessian=TRUE,y=Y,K=K,X=traplocs,
11573 +           S=pts,D=Dtraps,inpoly=in.pts)
11574 > frog2 <- nlm(intlik3edv2,c(-2.5,2,log(4)),hessian=TRUE,y=Y,K=K,X=traplocs,
11575 +           S=pts,D=Deuclid,inpoly=in.pts)

```

11576 These two models fit, with the correctly specified ecological distance, constrained by  
 11577 the patch boundaries, and that with the ordinary (misspecified) Euclidean distance are  
 11578 summarized in Table 12.3. We find little difference between the two models. In particu-  
 11579 lar, 150 individuals were captured and so truth (the number of uncaptured individuals)

**Table 12.3.** Summary output of fitting models to simulated data in which movement is restricted by the habitat corridor shown in Fig. 12.5. The two models fitted were those based on distance constrained by the corridor boundary (“constrained”) and a misspecified model based on ordinary Euclidean distance which is “as the crow flies”, and cuts through some boundaries. See `?fakecorridor` for the **R** commands to fit these models.

Distance	neg. LL	$\alpha_0$	$\alpha_1$	$\log(n_0)$
constrained	-21.892	-1.338	0.332	4.353
Euclidean	-21.128	-1.307	0.382	4.212

is  $\log(n_0) = 3.9$ . The correct model produces only a slightly more accurate estimate, and it is favored by only 0.7 negative log-likelihood units. Therefore, for this single instance, the results are not too different. This is primarily because the distance between individuals, and traps that they are likely to be captured in, is well-approximated by Euclidean distance.

## 12.8 ECOLOGICAL DISTANCE AND DENSITY COVARIATES

Habitat characteristics that affect spatial variation in density can also affect home range size and movement behavior. For example, a species that occurs at high density in a forest may be reluctant to venture from a forest patch into an adjacent field. Thus, even if a trap placed in a field is located very close to an animal’s activity center, the probability of capture may be very low. In this case, forest cover is a covariate of both density and encounter probability, and we could model it as such by combining the methods described in this chapter with those described in Chapt. 11.

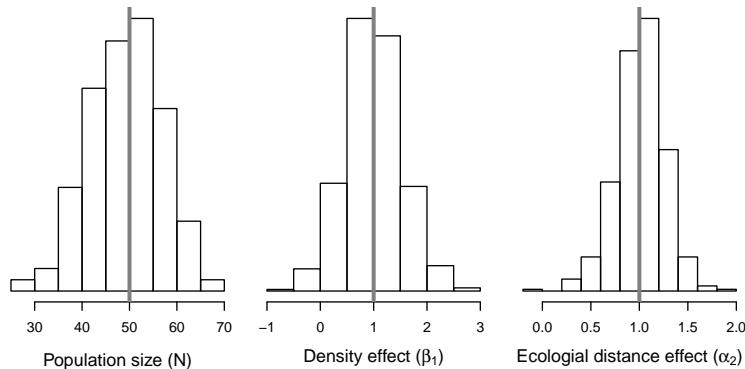
To demonstrate, we continue with our analysis of the data shown in Fig 11.4.2. Once again, we suppose that density increases with canopy height, but this time, we also allow home range size to decrease as density increases. This commonly-observed phenomenon can be explained by numerous factors such as intra-specific competition (Sillett et al., 2004) or optimal foraging behavior (Tufto et al., 1996; Saïd and Servanty, 2005).

A question that arises is: Is it possible to estimate the effect of the covariate on density ( $\beta_1$ ) and  $\alpha_2$  using standard SCR data? In other words, can we model spatial variation in density and connectivity at the same time, using standard SCR data? Currently, it is not possible to model least-cost distance using **JAGS** or **secr**, so we wrote our own function, **scrDED**, to fit the model using maximum likelihood. An example analysis is provided on the help page for the function in our **R** package **scrbook**. We briefly note here that the function requires the capture history data, the trap locations, and the raster data formatted using the **raster** package (Hijmans and van Etten, 2012). The linear model for the intensity parameter  $\mu(s, \beta)$  and the least-cost distance function  $lcd(\theta)$  are specified using **R**’s formula interface. A simple function call is

```
11607 > fm <- scrDED(y, traplocs=X, den.formula=~elev, dist.formula=~elev,
11608 + rasters=elev.raster)
```

To assess the possibility of estimating both  $\beta$  and  $\alpha_2$ , we conducted a small simulation study, generating 500 datasets from the model with both parameters set to 1, which

11611 corresponds to the conditions described above. The results indicate that it is possible to  
 11612 estimate both parameters (Fig 12.6).



**Figure 12.6.** Histograms of parameter estimates from 500 simulations under the model in which both density and ecological distance are affected by the same covariate, canopy height. The vertical lines indicate the data-generating value.

## 12.9 SUMMARY AND OUTLOOK

11613 Almost all published applications of SCR models to date have been based on models for  
 11614 the encounter probability that are functions of the Euclidean distance between individual  
 11615 activity centers and traps. The obvious limitations of such models are that Euclidean  
 11616 distance is unaffected by landscape or habitat structure and implies stationary, isotropic  
 11617 and symmetrical home ranges. These are standard criticisms of the basic SCR model  
 11618 which we have seen many times in referee reports, or heard in discussions with colleagues.  
 11619 However, this should not be seen as criticism that is inherent to the basic conceptual  
 11620 formulation of SCR models because, as we have shown here, one can modify the Euclidean  
 11621 distance metric to accommodate more realistic formulations of distance that allow for  
 11622 inference to be made about landscape connectivity, and model “distance” as a function  
 11623 of local habitat characteristics. As such, effective distance between individual home range  
 11624 centers and traps varies depending on the local landscape.

11625 How animals use space and therefore how distance to a trap is perceived by individuals  
 11626 is not something that can ever be known. We can only ever conjure up models to  
 11627 describe this phenomenon and fit those models to limited data on a sample of individuals  
 11628 during a limited amount of time. Here we have shown that there is hope to estimate con-  
 11629 nectivity parameters that describe how animals use space, from capture-recapture data  
 11630 alone, thereby allowing for irregular home range geometry that is influenced by landscape  
 11631 structure.

11632 In the presence of functional landscape connectivity, misspecification of the model by  
 11633 an ordinary SCR model based on Euclidean distance produces biased estimates of model

parameters (Royle et al., 2013). This is expected because the effect is similar to failing to model heterogeneity, i.e., if we mis-specify “model  $M_h$ ” (Otis et al., 1978) with “model  $M_0$ ” (Otis et al., 1978) then we will expect to under-estimate  $N$ . So the effect of mis-specifying the ecological distance metric with a standard homogeneous Euclidean distance has the same effect. In our view, this bias is not really the most important reason to consider models of ecological distance. Rather, inference about the structure of ecological distance is fundamental to many problems in applied and theoretical ecology related to modeling landscape connectivity, corridor and reserve design, population viability analysis, gene flow, and other phenomena. Models based on least-cost path distance allow investigators to evaluate landscape factors that influence movement of individuals over the landscape from non-invasively collected capture-recapture data. Therefore SCR models based on ecological distance metrics might aid in understanding aspects of space usage and movement in animal populations and, ultimately, in addressing conservation-related problems such as corridor design.



11648  
11649

# 13

---

11650  
11651  
11652

## INTEGRATING RESOURCE SELECTION WITH SPATIAL CAPTURE-RECAPTURE MODELS

11653 In Chapt. 5 we briefly discussed the notion of how SCR encounter probability models relate  
11654 to models of space usage. When using symmetric and stationary encounter probability  
11655 models, SCR models imply that space usage is a decreasing function of distance from an  
11656 individual's home range center. This is not a very realistic model in most applications.  
11657 In this chapter, we extend SCR models to incorporate models of resource selection, such  
11658 as when one or more explicit landscape covariates are available which the investigator  
11659 believes might affect how individual animals use space within their home range. This is  
11660 what Johnson (1980) called *third-order* selection – a term emphasizing the hierarchical  
11661 nature of resource selection.

11662 An appealing feature of SCR models is that they provide a mechanism for modeling  
11663 multiple levels of the resource selection hierarchy. For instance, Johnson (1980) de-  
11664 fined *second-order* selection as the process determining the location of home ranges on  
11665 a landscape, which is exactly the process being modeled using the methods presented in  
11666 Chapt. 11. Thus, SCR provides a way of studying the density and distribution of home  
11667 range centers, while at the same time allowing for inferences about the use of resources  
11668 within home ranges.

11669 Our treatment follows Royle et al. (2012a) who integrated a standard family of resource  
11670 selection models based on auxiliary telemetry data into the capture-recapture model for en-  
11671 counter probability. They argued that SCR models and resource selection models (Manly  
11672 et al., 2002) are based on the same basic underlying model of space usage. The important  
11673 distinction between SCR and RSF studies is that, in SCR studies, encounter of individuals  
11674 is imperfect (i.e., “ $p < 1$ ”) whereas, with RSF data obtained by telemetry, encounter is  
11675 perfect. SCR and telemetry data can therefore be combined in the same likelihood by  
11676 formally recognizing this distinction in the model.

11677 There are two important motives for considering a formal integration of RSF models

with capture-recapture. The first is to integrate models of resource use by individuals with models of population size or density. There is relatively little in the literature on this topic, although Boyce and McDonald (1999) describe a procedure where (an estimate of) population size is used to scale resource selection functions to produce a population density surface. The second reason is because this allows for the integration of auxiliary data from telemetry studies with capture-recapture data. Telemetry studies are extremely common in animal ecology for studying movement and resource selection, and capture-recapture studies frequently involve a simultaneous telemetry component. Telemetry data has been widely used in conjunction with capture-recapture data using standard non-spatial models. For example, White and Shenk (2001) and Ivan (2012) suggested using telemetry data to estimate the probability that an individual is exposed to capture-recapture sampling. However, their estimator requires that individuals are telemetry-tagged in proportion to this unknown quantity, which seems impossible to achieve in many studies. In addition, they do not directly integrate the telemetry data with the capture-recapture model so that common parameters are jointly estimated. Sollmann et al. (in revision) and Sollmann et al. (2013) used telemetry data to directly inform the parameter  $\sigma$  from the bivariate normal SCR model in order to improve estimates of density, although these models do not include an explicit resource selection component.

Formal integration of capture-recapture with telemetry data for the purposes of modeling resource selection has a number of immediate benefits. For one, telemetry data provide direct information about  $\sigma$  (Sollmann et al., 2013, in revision). As a result, this leads to improved estimates of model parameters, and also has design consequences (see Sec. 10.7). In addition, active resource selection by animals induces a type of heterogeneity in encounter probability, which is misspecified by standard SCR encounter probability models. Animals that use more space due to the configuration of habitat or landscape features, stand to be exposed to more traps than animals that use less space. As a result, estimates of population size or density under models that do not account for resource selection can be biased (Royle et al., 2012a). Finally, because the resource selection model translates directly to a model for encounter probability for spatial capture-recapture data, the implication of this is that it allows us to estimate resource selection model parameters directly from SCR data, i.e., *absent* telemetry data. This fact should broaden the practical relevance of spatial capture-recapture not just for estimating density, but also for directly studying movement and resource selection.

### 13.1 A MODEL OF SPACE USAGE

Assume that the landscape is defined in terms of a discrete raster of one or more covariates, having the same dimensions and extent. Let  $\mathbf{x}_1, \dots, \mathbf{x}_G$  identify the center coordinates of  $G$  pixels that define a landscape, organized in the matrix  $\mathbf{X}_{G \times 2}$ . Let  $C(\mathbf{x})$  denote a covariate defined for every pixel  $\mathbf{x}$ . We suppose that individual members of a population wander around space in some manner related to the covariate  $C(\mathbf{x})$ .

As a biological matter, use is the outcome of individuals moving around their home range (Hooten et al., 2010), i.e., where an individual is at any point in time is the result of some movement process. However, to understand space usage, it is not necessary to entertain explicit models of movement, just to observe the outcomes, and so we don't elaborate further on what could be sensible or useful models of movement, but we imagine existing

methods of hierarchical or state-space models are suitable for this purpose (Ovaskainen, 2004; Jonsen et al., 2005; Forester et al., 2007; Ovaskainen et al., 2008; Patterson et al., 2008; Hooten et al., 2010; McClintock et al., 2012). We consider explicit movement models in the context of SCR models later chapters of this book (Chaps. 15 and 16). Here we adopt more of a phenomenological formulation of space usage as follows: If an individual appears in pixel  $\mathbf{x}$  at some instant, this is defined as a decision to “use” pixel  $\mathbf{x}$ . Thus, over any prescribed time interval, the percentage of time an individual spends in each pixel is theoretically knowable. Or, if we sample some number of points during that interval, say  $R$ , then the frequency of use decisions is, conceivably, observable by some omnipotent accounting mechanism (e.g., telemetry that doesn’t malfunction). In this case, let  $m_{ij}$  be the *true* use frequency of pixel  $j$  by individual  $i$  – i.e., the number of times individual  $i$  used pixel  $j$ . We assume the vector of use frequencies  $\mathbf{m}_i = (m_{i1}, \dots, m_{iG})$  has a multinomial distribution:

$$\mathbf{m}_i \sim \text{Multinomial}(R, \boldsymbol{\pi}_i)$$

where  $R = \sum_j m_{ij}$  is the total number of “use decisions” made by individual  $i$  and

$$\pi_{ij} = \frac{\exp(\alpha_2 C(\mathbf{x}_j))}{\sum_x \exp(\alpha_2 C(\mathbf{x}))}$$

for each  $j = 1, 2, \dots, G$  pixels. This is a standard RSF model (Manly et al., 2002) used to model telemetry data. In particular, this is “protocol A” of (Manly et al., 2002) where all available landscape pixels are censused (i.e., known without error), and used pixels are sampled randomly for each individual. The parameter  $\alpha_2$  is the effect of the landscape covariate  $C(\mathbf{x})$  on the relative probability of use. Thus, if  $\alpha_2$  is positive, the relative probability of use increases as the covariate increases.

In practice, we don’t get to observe  $m_{ij}$  for all individuals but, instead, only for a small subset which we capture and telemeter. For the telemetered individuals, we assume they use resources according to the same RSF model as the population as a whole. To extend this model to make it more realistic, and consistent with the formulation of SCR models, let  $\mathbf{s}$  denote the center of an individual’s home range and let  $d_{ij} = \|\mathbf{x}_j - \mathbf{s}_i\|$  be the distance from the home range center of individual  $i$ ,  $\mathbf{s}_i$ , to pixel  $j$ ,  $\mathbf{x}_j$ . We modify the space usage model to accommodate that space use will be concentrated around an individual’s home range center:

$$\pi_{ij} = \frac{\exp(-\alpha_1 d_{ij}^2 + \alpha_2 C(\mathbf{x}_j))}{\sum_x \exp(-\alpha_1 d_{ij}^2 + \alpha_2 C(\mathbf{x}))} \quad (13.1.1)$$

The parameters  $\alpha_1$ ,  $\alpha_2$  and the activity centers  $\mathbf{s}$  can be estimated directly from telemetry data, using standard likelihood methods based on the multinomial likelihood (Johnson et al., 2008b). Normally this model is expressed in terms of the scale parameter  $\sigma$ ,  $\alpha_1 = 1/(2\sigma^2)$ , and the multinomial model Eq. 13.1.1 can be understood as a compound model of space usage governed by distance-based “availability” according to a Gaussian kernel, and also “use”, conditional on availability (Johnson et al., 2008b; Forester et al., 2009). In other words, the model suggests a kind of distance-based availability in which a pixel is less available to an individual if it is located further away from  $\mathbf{s}_i$ .

Eq. 13.1.1 resembles standard SCR encounter probability models that we have used previously, but here the model includes an additional covariate  $C(\mathbf{x})$  (see Chapt. 9). In

11759 particular, under this model for space usage or resource selection, if we have no covariates  
 11760 at all, or if  $\alpha_2 = 0$ , then the probabilities  $\pi_{ij}$  are directly proportional to the SCR model  
 11761 for encounter probability, *if we have a trap in every pixel*. Therefore, setting  $\alpha_2 = 0$ , the  
 11762 probability of use for pixel  $j$  is:

$$p_{ij} \propto \exp(-\alpha_1 d_{ij}^2).$$

11763 Clearly, whatever function of distance we use in the RSF model implies an equivalent  
 11764 model of space usage (Sec. 5.4) as an SCR model for encounter probability. In particular,  
 11765 for whatever model we choose for  $p_{ij}$  in an ordinary SCR model, we can modify the  
 11766 distance component in the RSF function in Eq. 13.1.1 to be consistent with that model  
 11767 by setting:

$$\pi_{ij} \propto \exp(\log(p_{ij}) + \alpha_2 C(\mathbf{x}_j))$$

11768 (see Forester et al. (2009)).

11769 One difference between this multinomial observation model for resource use data and  
 11770 those that we have considered in previous chapters is that it includes the normalizing  
 11771 constant  $\sum_x \exp(-\alpha_1 d_{ij}^2 + \alpha_2 C(\mathbf{x}_j))$ , which ensures that the use distribution is a proper  
 11772 probability density function. In that sense, the model has the same form as the multinomial  
 11773 SCR model described in Chapt. 9 except that, here, the probability density of use  
 11774 locations is distributed over the whole state space  $\mathcal{S}$ , not just the subset of locations where  
 11775 we have traps. In a sense, we view telemetry data as a perfect sampling of space, equivalent  
 11776 to having a trap in each pixel, and the number of captures (uses by an individual) is  
 11777 fixed by design.

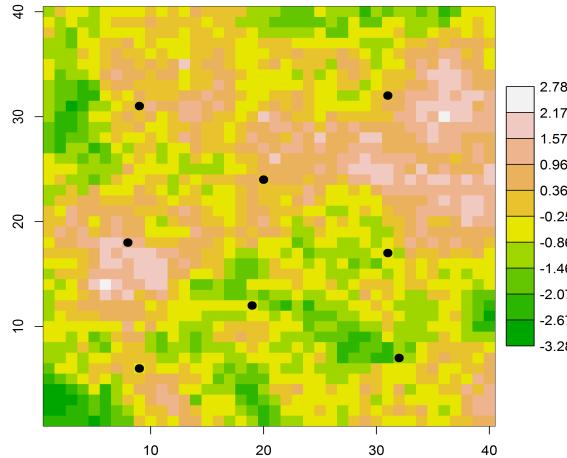
### 11778 13.1.1 A simulated example

11779 For a simulated landscape (shown in Fig. 13.1), Royle et al. (2012a) depicted some  
 11780 typical space usage patterns under the model described above, which we reproduce here  
 11781 in Fig. 13.2. The covariate in this case was simulated using a kriging model of correlated  
 11782 random noise with the following R commands:

```
11783 > set.seed(1234)
11784 > gr <- expand.grid(1:40,1:40)
11785 > Dmat<-as.matrix(dist(gr))
11786 > V <- exp(-Dmat/5)
11787 > C <- t(chol(V))%*%rnorm(1600)
```

11788 The resulting covariate vector  $\mathbf{C}$  is multivariate normal with mean 0 and variance-covariate  
 11789 matrix  $\mathbf{V}$  which, here, has pairwise correlations which decay exponentially with distance.  
 11790 The use densities shown in Fig. 13.2 were simulated with  $\alpha_1 = 1/(2\sigma^2)$ , with  $\sigma = 2$ , and  
 11791 the coefficient on  $C(\mathbf{x})$  set to  $\alpha_2 = 1$ . The resulting space usage densities – or “home  
 11792 ranges” – exhibit clear non-stationarity in response to the structure of the underlying  
 11793 covariate, and they are distinctly asymmetrical. We note that if  $\alpha_2$  were set to 0, the 8  
 11794 home ranges shown here would be proportional to a bivariate normal kernel with  $\sigma = 2^1$ .  
 11795 The commands for the kriging model, and those to produce Fig. 13.1 are in the package  
 11796 `scrbook` (see `?RSF_example`).

<sup>1</sup>This is why we have always referred to the similar-looking model for encounter probability as the Gaussian or bivariate normal model, instead of half-normal.



**Figure 13.1.** A typical habitat covariate reflecting habitat quality or hypothetical utility of the landscape to a species under study. Home range centers for 8 individuals are shown with black dots.

### 13.1.2 Poisson model of space use

A natural way to motivate the multinomial model of space usage is to assume that individuals make a sequence of resource selection decisions so that the outcomes  $m_{ij}$  are *independent* Poisson random variables:

$$m_{ij} \sim \text{Poisson}(\lambda_{ij})$$

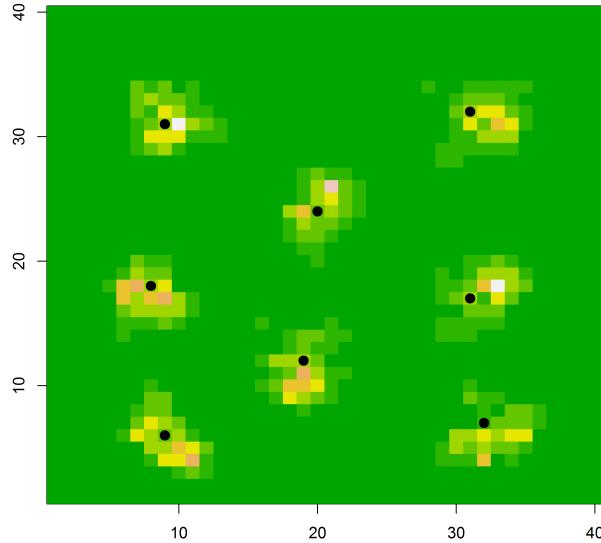
where

$$\log(\lambda_{ij}) = a_0 - \alpha_1 d_{ij}^2 + \alpha_2 C(\mathbf{x}_j).$$

In this case, the number of visits to any particular cell is affected by the covariate  $C(\mathbf{x})$  but has a baseline rate,  $\exp(a_0)$ , related to the amount (in an expected value sense) of movement occurring over some time interval. This is an equivalent model to the multinomial model given previously in the sense that, if we condition on the total sample size  $R = \sum_j m_{ij}$ , then the vector  $\mathbf{m}_i$  has a multinomial distribution with probabilities given by Eq. 13.1.1 (see also Chapt. 9).

In practice, we never observe “truth”, i.e., the actual use frequencies  $m_{ij}$ . Instead, we observe a sample of the actual use outcomes by an individual. As formulated in Sec. 5.4, we assume a binomial (“random”) sampling model:

$$y_{ij} \sim \text{Binomial}(m_{ij}, p_0).$$



**Figure 13.2.** Space usage patterns of 8 individuals under a space usage model that contains a single covariate which is shown in Fig. 13.1. The plotted value is the multinomial probability  $\pi_{ij}$  for pixel  $j$  under the model in Eq. 13.1.1.

11811 We can think of these counts as arising by thinning the underlying point process (here,  
 11812 aggregated into pixels) where  $p_0$  is the thinning rate of the point process. In this case,  
 11813 the marginal distribution of the observed counts  $y_{ij}$  is also Poisson but with mean

$$\log(\mathbb{E}(y_{ij})) = \log(p_0) + a_0 - \alpha_1 d_{ij}^2 + \alpha_2 C(\mathbf{x}_j).$$

11814 Thus, the space-usage model (RSF) for the thinned counts  $y_{ij}$  is the same as the space-  
 11815 usage model for the original variables  $m_{ij}$ . This is because if we remove  $m_{ij}$  from the  
 11816 conditional model by summing over its possible values, then the vector  $\mathbf{y}_i$  is *also* multi-  
 11817 nominal with cell probabilities

$$\pi_{ij} = \frac{\lambda_{ij}}{\sum_j \lambda_{ij}}$$

11818 where any constant (the intercept term  $a_0$  and thinning rate  $p_0$ ) cancel from the numer-  
 11819 ator and denominator. Thus, the underlying multinomial RSF model applies to the true  
 11820 unobserved count frequencies  $\mathbf{m}_i$  and also those produced from thinning or sampling,  $\mathbf{y}_i$ .

### 13.2 INTEGRATING CAPTURE-RECAPTURE DATA

11821 The key to combining RSF data with SCR data is to note that the Poisson model of space  
 11822 usage given above is exactly our Poisson encounter probability model from Chapt. 9, only  
 11823 with a spatial covariate  $C(\mathbf{x})$ , and some arbitrary intercept off-set related to the sampling  
 11824 rate by the telemetry device. We've used exactly this model for our SCR data (Chapt. 7),  
 11825 but with a different intercept,  $\alpha_0$ , unrelated to the intercept of the Poisson use model for  
 11826 telemetry described above but, rather, to the efficiency of the capture-recapture encounter  
 11827 device. In other words, we view camera traps (or other devices) located in some pixel  $\mathbf{x}$   
 11828 (or multiple pixels) as being equivalent to being able to turn on a type of (less perfect)  
 11829 telemetry device only in that pixel. Therefore, data from a camera trapping are Poisson  
 11830 random variables for every pixel  $j$  where a trap is located:

$$y_{ij} | \mathbf{s}_i \sim \text{Poisson}(\lambda_{ij})$$

11831 with

$$\log(\lambda_{ij}) = \alpha_0 - \alpha_1 d_{ij}^2 + \alpha_2 C(\mathbf{x}_j).$$

11832 The parameters  $\alpha_1$  and  $\alpha_2$  are shared with the multinomial model for the telemetry data.

11833 Alternatively, the SCR study can produce binary encounters depending on the type of  
 11834 sampling being done, where  $y_{ij} = 1$  if the individual  $i$  visited the pixel containing a trap  
 11835 and was detected, then we imagine that  $y_{ij}$  is related to the latent variable  $m_{ij}$  being the  
 11836 event  $m_{ij} > 0$ , which occurs with probability

$$p_{ij} = 1 - \exp(-\lambda_{ij}) \quad (13.2.1)$$

11837 and then the observed encounter frequencies for individual  $i$  and trap  $j$ , from sampling  
 11838 over  $K$  occasions are binomial:

$$y_{ij} | \mathbf{s}_i \sim \text{Binomial}(K, p_{ij})$$

11839 A key point here is that if resource selection is happening, then it appears as a covariate  
 11840 on encounter rate (or encounter probability) in the same way as ordinary covariates which  
 11841 we discussed in Chapt. 7.

11842 To construct the likelihood for SCR data when we have direct information on space  
 11843 usage from telemetry data, we regard the two samples (SCR and RSF) as independent  
 11844 of one another, and we form the likelihood for each set of observations as a function of  
 11845 the same underlying parameters. The joint likelihood then is the product of the two  
 11846 components.

11847 In particular, let  $\mathcal{L}_{scr}(\alpha_0, \alpha_1, \alpha_2, N; \mathbf{y})$  be the likelihood for the SCR data in terms of  
 11848 the basic encounter probability parameters and the total (unknown) population size  $N$ ,  
 11849 and let  $\mathcal{L}_{rsf}(\alpha_1, \alpha_2; \mathbf{m})$  be the likelihood for the RSF data based on telemetry which, be-  
 11850 cause the sample size of telemetered individuals is fixed, does not depend on  $N$ . Assuming  
 11851 independence of the two datasets, the joint likelihood is the product of these two pieces:

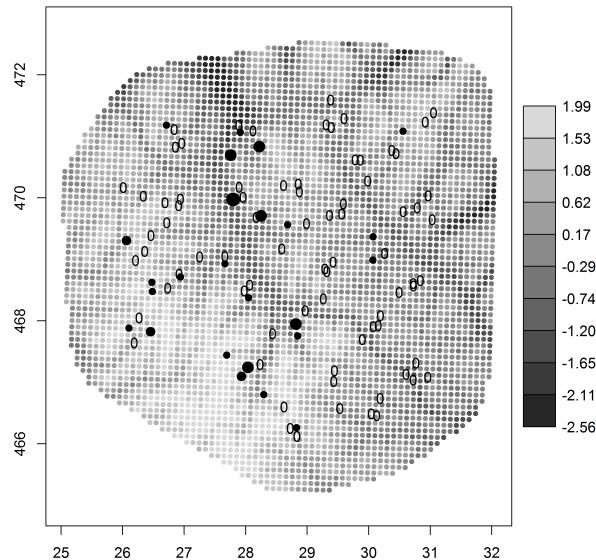
$$\mathcal{L}_{rsf+scr}(\alpha_0, \alpha_1, \alpha_2, N; \mathbf{y}, \mathbf{m}) = \mathcal{L}_{scr}(\alpha_0, \alpha_1, \alpha_2, N; \mathbf{y}) \times \mathcal{L}_{rsf}(\alpha_1, \alpha_2; \mathbf{m}),$$

11852 where the  $\mathcal{L}_{scr}$  is the standard integrated likelihood (Chapt. 6), and the RSF likelihood  
 11853 contribution is the multinomial telemetry likelihood having cell probabilities Eq. 13.1.1.  
 11854 The R code for maximizing the joint likelihood was given in the supplement to Royle  
 11855 et al. (2012a), and we include a version of this in the `scrbook` package, see `?intlik3rsf`,  
 11856 which also shows how to simulate data and fit the combined SCR+RSF model.

### 13.3 SW NEW YORK BLACK BEAR STUDY

11857 Royle et al. (2012a) applied the integrated SCR+RSF model to data from a study of  
 11858 black bears (*Ursus americanus*) in a region of approximately 4,600 km<sup>2</sup> in southwestern  
 11859 New York. These data come from a research project by C. Sun (Sun, in prep) at Cornell  
 11860 University, and it is a different data set than our Fort Drum bear study data set which  
 11861 we've analyzed in previous chapters. The data can be loaded from the **scrbook** package  
 11862 with the command **data(nybears)**. We reproduce the findings of Royle et al. (2012a) in  
 11863 this section.

11864 The data are based on a noninvasive genetic capture-recapture study using 103 hair  
 11865 snares in June and July, 2011. Hair snares were baited and scented and checked weekly  
 11866 for hair (Sun, in prep). The study yielded relatively sparse encounter histories of 33 indi-  
 11867 viduals with a total of 14 recaptures and 27 individuals captured 1 time only. Telemetry  
 11868 data were collected on 3 telemetry-collared individuals, which produced locations for each  
 11869 bear approximately once per hour. Telemetry locations were thinned to once per 10 hours  
 11870 to produce movement outcomes that might be more independent. This produced 195  
 11871 telemetry locations used in the RSF component of the model. Elevation was used as the  
 11872 covariate for this model, a standardized version of which is shown in Fig. 13.3 along with  
 11873 the number of individuals captured at each hair snare site.



**Figure 13.3.** Elevation (standardized), hair snare locations are marked by the number of individuals captures at each site. The largest size solid mark corresponds to 4 individuals captured, the smallest to 1 individual. Hair snares that produced no individuals are given by "0".

11874 There are a number of models that could be fitted to these data based on the combination  
11875 of SCR and RSF data as well as the elevation covariate. The models fit here are  
11876 based on the Gaussian hazard trap encounter/space usage model, including an ordinary  
11877 SCR model with no covariates or telemetry data, the SCR model with elevation affecting  
11878 either  $\lambda_0$  or density  $D(\mathbf{x})$  (Chapt. 11), and models that use telemetry data. The 6 models  
11879 fitted were:

11880 Model 1, SCR: ordinary SCR model  
11881 Model 2, SCR+p(C): ordinary SCR model with elevation as a covariate on baseline  
11882 encounter probability  $\lambda_0$ .  
11883 Model 3, SCR+D(C): ordinary SCR model with elevation as a covariate on density only.  
11884 Model 4, SCR+p(C)+D(C): ordinary SCR model with elevation as a covariate on both  
11885 baseline encounter probability and density.  
11886 Model 5, SCR+p(C)+RSF: SCR model including data from 3 telemetered individuals.  
11887 Model 6, SCR+p(C)+RSF+D(C): SCR model including telemetered individuals and  
11888 with elevation as a covariate on density.

11889 Parameter estimates for the six models are given in Table 13.1 (reproduced from Royle  
11890 et al. (2012a), see also the help file `?nybears`). It is tempting to want to compare these  
11891 different models by AIC but, because models 5 and 6 involve additional data, they cannot  
11892 be compared with models 1-4.

11893 By looking at Table 13.1, it is clear based on the negative log likelihood for just Models  
11894 1-4, that those containing an elevation effect on density are preferred (Model 3 and 4).  
11895 The parameter estimates indicate a positive effect of elevation on density, which seems to  
11896 be consistent with the raw capture data shown in Fig. 13.3. Despite this strong effect of  
11897 elevation, the estimates of  $N$  under each of these models only ranged from 93 – 103 bears  
11898 for the 4600 km<sup>2</sup> state-space, and so estimated density is pretty consistent across models.  
11899 If we consider not just density, but space usage (i.e., looking at the parameter  $\alpha_2$ ), the  
11900 effect of elevation is negative. Thus, elevation, appears to affect density and space usage  
11901 differently. It was suggested that density operates at the second-order scale of resource  
11902 selection and “....is largely related to the spacing of individuals and their associated home  
11903 ranges across the landscape. On the other hand, our RSF was defined based on selection of  
11904 resources within the home range (third-order).” (Royle et al., 2012a) The positive effect of  
11905 density on elevation is consistent with some other studies on black bears (e.g. Frary et al.,  
11906 2011), and the negative effect of elevation on space usage can be attributed to seasonal  
11907 variation in food availability, usage of corridors, or environmental conditions.

11908 Models 5 and 6 include the additional telemetry data, thus the negative log-likelihoods  
11909 are not directly comparable to the first 4 models, but we can still make a few important  
11910 observations. First is that the parameter estimates under these two models are consistent  
11911 with Model 4 in that elevation had a strong effect on both density and space usage. In  
11912 comparing models 5 and 6, the latter model which includes elevation as an effect on density  
11913 reduces the negative log-likelihood by 5 units. Additionally, including the telemetry data  
11914 reduces the standard errors (SE) of the density and space usage parameters and as we  
11915 would expect, the incorporation of telemetry data also reduces the SE for  $\sigma$ . The increased  
11916 precision for the estimated population size ( $N$ ) is negligible with the use of telemetry data  
11917 in this case. However, that may be different if more telemetry information were available.  
11918 Model 6 (SCR+p(C)+RSF+D(C)), was used to produce maps of density (Fig. 13.4) and

**Table 13.1.** Summary of model-fitting results for the black bear study. Parameter estimates are for the intercept ( $\alpha_0$ ), logarithm of  $\sigma$ , the scale parameter of the Gaussian hazard encounter model,  $\beta$  is the coefficient of elevation on density, and the total population size  $N$  of the state-space. Standard errors are in parentheses. The SCR data are based on  $n = 33$  individuals, and the telemetry data are based on 3 individuals.

model	$\alpha_0$	$\log(\sigma)$	$\alpha_2$	$N$	$\beta$	-loglik
SCR(elev)	-2.860 (0.390)	-1.117 (0.139)	0.175 (0.248)	95.8 (22.99)		122.738
SCR	-2.729 (0.345)	-1.122 (0.140)	—	93.9 (22.06)		122.990
SCR+D(elev)	-2.715 (0.353)	-1.133 (0.139)	—	94.2 (21.90)	1.247 (0.408)	118.007
SCR(elev)+D(elev)	-2.484 (0.391)	-1.157 (0.142)	-0.384 (0.276)	103.5 (26.56)	1.571 (0.463)	117.075
SCR(elev)+RSF	-3.068 (0.272)	-0.814 (0.036)	-0.281 (0.118)	81.6 (17.65)		1271.739
SCR(elev)+RSF+D(elev)	-3.070 (0.272)	-0.810 (0.037)	-0.371 (0.124)	89.1 (20.55)	1.273 (0.411)	1266.700

space usage (Fig. 13.5) showing the effect of elevation on both components of the model. The map of space usage shows the relative probability of using a pixel  $\mathbf{x}$  relative to one having the mean elevation, given a constant distance to the individual's activity center.

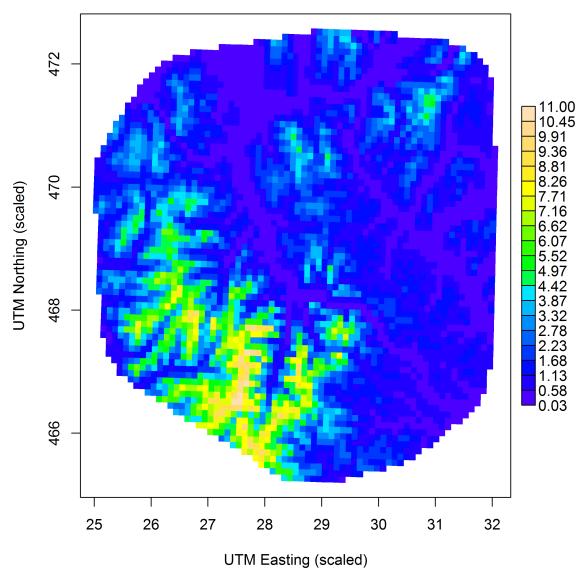
### 13.4 SIMULATION STUDY

Using the simulated landscape shown in Fig. 13.1, Royle et al. (2012a) presented results of a simulation study considering populations of  $N = 100$  and  $N = 200$  individuals exposed to encounter by a  $7 \times 7$  array of trapping devices, with  $K = 10$  sampling occasions, using the Gaussian hazard model (Eq. 13.2.1) with

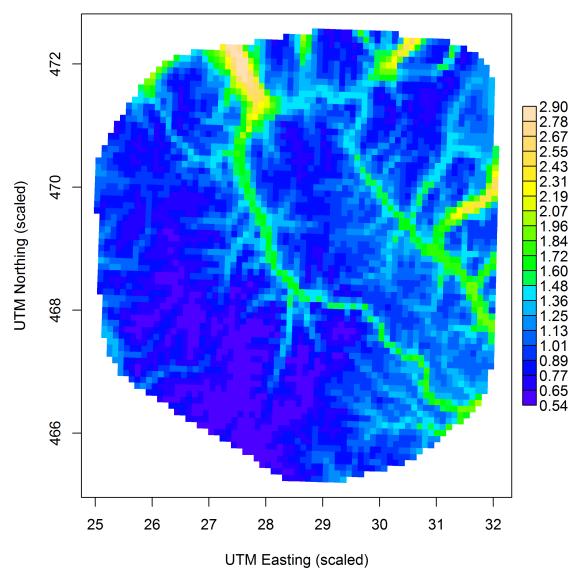
$$\log(\lambda_{ij}) = -2 - \frac{1}{2\sigma^2} d_{ij}^2 + 1 * C(\mathbf{x}_j).$$

where  $\sigma = 2$ . They looked at the effect of misspecification of the resource selection model with an ordinary model SCR0 (i.e. no habitat covariates affecting the trap encounter model), and the performance of the MLEs, under SCR+telemetry designs having 2, 4, 8, 12, and 16 telemetered individuals (with 20 independent telemetry fixes *per* individual). Three models were fitted: (i) the SCR only model, in which the telemetry data were not used; (ii) the integrated SCR/RSF model which combined all of the data for jointly estimating model parameters; and (iii) the RSF only model which just used the telemetry data alone (and therefore the parameters  $\alpha_0$  and  $N$  are not estimable). An abbreviated version of the results from Royle et al. (2012a) is summarized in Table 13.2. We provide an **R** script (see `?RSFsim`) that can be modified for further analysis and exploration.

One thing we see is a pretty dramatic negative bias in estimating  $N$  if the model SCR0 is fitted (interestingly, there is much less bias in estimating  $\sigma$ ). Overall, though, when either the SCR model with covariate or the joint SCR+RSF model is fitted, the MLEs exhibit little bias for the parameter values simulated here. In terms of RMSE, there is



**Figure 13.4.** Predicted density of black bears (per 100 km<sup>2</sup>) in southwestern New York study area.



**Figure 13.5.** Relative probability of use of pixel  $x$  compared to a pixel of mean elevation, at a constant distance from the activity center.

only a slight  $\approx$  5-10% reduction in RMSE of the estimator of  $N$  when we have at least 2 telemetered individuals. Thus, estimating  $N$  benefits only slightly from the addition of telemetry data, which is because information about the intercept,  $\alpha_0$ , comes only from the capture-recapture data. However, there is a large improvement in precision (50-60%) for estimating the scale parameter  $\sigma$ . While this doesn't translate much into improved estimation of  $N$ , it suggests that it should be relevant to the design of SCR studies for which trap spacing is one of the main considerations (Chapt. 10). In terms of study design these results also suggest that, perhaps, spatial recaptures are not needed if some telemetry data are available (in Chapt. 19, in the context of mark-resight models, we show a case study of raccoons where additional telemetry data allows estimating model parameters in spite of a very low number of spatial recaptures (Sollmann et al., 2013)). The resource selection parameter  $\alpha_2$  is well-estimated even *without* telemetry data. The fact that parameters of resource selection can be estimated from ordinary capture-recapture data should have considerable practical relevance in the study of animal populations and landscape ecology. For the highest sample size of telemetered individuals ( $n = 16$ ), the RMSE for estimating this parameter only decreases from about 0.09 to 0.07.

**Table 13.2.** This table summarizes the sampling distribution of the MLE of model parameters for models fitted to data generated under a resource selection model. The models fitted include the misspecified model, which is a basic model SCR0 (with no covariate), the SCR model with the covariate on encounter probability, and the SCR model including the covariate and a sample of telemetered individuals ( $n$  is the number of individuals telemetered). Data were simulated with  $N = 200$  individuals,  $\alpha_2 = 1$  and  $\sigma = 2$ .

	$\hat{N}$	RMSE	$\hat{\alpha}_2$	RMSE	$\hat{\sigma}$	RMSE
$n=2$						
SCR+C(x)	199.11	14.28	0.99	0.09	2.00	0.090
SCR+RSF	199.11	13.80	0.99	0.09	2.00	0.079
SCR0	161.48	39.98	—	—	1.84	0.180
$n=4$						
SCR only	199.67	13.87	1.00	0.09	2.00	0.090
SCR/RSF	199.65	13.59	1.00	0.09	2.00	0.072
SCR0	161.32	40.00	—	—	1.83	0.191
$n=8$						
SCR only	199.24	15.49	0.99	0.10	2.01	0.093
SCR/RSF	199.55	14.17	0.99	0.08	2.00	0.063
SCR0	161.46	40.06	—	—	1.84	0.184
$n=12$						
SCR only	200.41	15.16	0.99	0.10	2.00	0.086
SCR/RSF	200.95	13.04	1.00	0.08	2.00	0.051
SCR0	162.40	38.95	—	—	1.84	0.185
$n=16$						
SCR only	199.16	15.62	1.00	0.09	2.00	0.095
SCR/RSF	199.63	13.38	1.00	0.07	2.00	0.052
SCR0	160.93	40.44	—	—	1.84	0.190

### 13.5 RELEVANCE AND RELAXATION OF ASSUMPTIONS

11956 In constructing the combined likelihood for RSF and SCR data, we assumed the data from  
11957 capture-recapture and telemetry studies were independent of one another. This implies  
11958 that whether or not an individual enters into one of the data sets has no effect on whether  
11959 it enters into the other data set. We cannot foresee situations in which violation of this  
11960 assumption should be problematic or invalidate the estimator under the independence  
11961 assumption. In some cases it might so happen that some individuals appear in *both* the  
11962 RSF and SCR data sets. In this case, ignoring that information should entail only an  
11963 incremental decrease in precision because a slight bit of information about an individuals  
11964 activity center is disregarded.

11965 Our model pretends that we do not know anything about the telemetered individuals  
11966 in terms of their encounter history in traps. In principle it should not be difficult to admit  
11967 a formal reconciliation of individuals between the two lists. In that case, we just combine  
11968 the two conditional likelihoods before we integrate  $s$  from the conditional likelihood. This  
11969 would be almost trivial to do if *all* individuals were reconcilable (or none, as in the case  
11970 we have covered here). But, in general, we think you will often have an intermediate case,  
11971 i.e., either none will be or at most a subset of telemetered guys will be known and there  
11972 will be some individuals of unknown mark status. In that case, basically a type of marking  
11973 uncertainty or misclassification, is clearly more difficult to deal with (see Chapt. 19 for  
11974 some additional context).

11975 We developed the model in a discrete landscape which regarded potential trap locations  
11976 and the covariate  $C(\mathbf{x})$  as being defined on the same set of points. In practice, trap  
11977 locations may be chosen independent of the definition of the raster and this does not pose  
11978 any challenge or novelty to the model as it stands. In that case, the covariate(s) need to be  
11979 defined at each trap location. The model should be applicable also to covariates that are  
11980 naturally continuous (e.g., distance-based covariates) although, in practice, it will usually  
11981 be sufficient to work with a discrete representation of such covariates.

11982 The multinomial RSF model for telemetry data assumes independent observations of  
11983 resource selection. This would certainty be reasonable if telemetry fixes are made far apart  
11984 in time (or thinned). However, as noted by Royle et al. (2012a), the independence as-  
11985 sumption is *not* an assumption of spatially independent movement outcomes in geographic  
11986 space. Active resource selection should probably lead to the appearance of spatially de-  
11987 pendent outcomes, regardless of how far apart in time the telemetry locations are. Even  
11988 if resource selection observations are dependent, use of the independence model probably  
11989 yields unbiased estimators while under-stating the variance. Development of integrated  
11990 SCR+RSF models that accommodate more general models of movement is needed.

### 13.6 SUMMARY AND OUTLOOK

11991 How animals use space is of fundamental interest to ecologists and is important in the  
11992 conservation and management of many species. Investigating space use is normally done  
11993 using telemetry and models referred to as resource selection functions (Manly et al., 2002)  
11994 but in all of human history, animal resource selection has *never* been studied using capture-  
11995 recapture models. Instead, essentially all applications of SCR models have focused on  
11996 density estimation. It is intuitive, however, that space usage or resource selection should  
11997 affect encounter probability and thus it should be highly relevant to density estimation in

1198 SCR applications, and, vice versa, SCR applications should yield data relevant to resource  
1199 selection questions. The development in this chapter shows clearly that these two ideas  
1200 can be unified within the SCR methodological framework so that classical notions of  
1201 resource selection modeling can be addressed simultaneous to modeling of animal density.  
1202 What we find is that if animal resource selection is occurring, this can be modeled as  
1203 covariate on encounter probability, with or without the availability of auxiliary telemetry  
1204 data. If telemetry data do exist, we can estimate parameters jointly by combining the two  
1205 likelihood components – that of the SCR data and that of the telemetry data.

1206 Active resource selection by individuals induces a type of heterogeneous encounter  
1207 probability, and this induces (possibly severe) bias in the estimated population size for  
1208 a state-space when default symmetric encounter probability models are used. As such,  
1209 it is important to account for resource selection when relevant covariates are known to  
1210 influence resource selection patterns. Aside from properly modeling this selection-induced  
1211 heterogeneity, integration of RSF data from telemetry with SCR models achieves a number  
1212 of useful advances: First, it leads to an improvement in our ability to estimate density, and  
1213 also an improvement in our ability to estimate parameters of the RSF function. As many  
1214 animal population studies have auxiliary telemetry information, the incorporation of such  
1215 information into SCR studies has broad applicability to many studies. It seems possible  
1216 even to estimate density now, with no spatial recaptures, provided telemetry data are  
1217 available. Secondly, the integrated model allows for the estimation of RSF model param-  
1218 eters directly from SCR data *alone*. This establishes clearly that SCR models *are* explicit  
1219 models of resource selection. In our view, this greatly broadens the utility and importance  
1220 of capture-recapture studies beyond their primary historical use of estimating density or  
1221 population size. Finally, we note that telemetry information provide direct information  
1222 about the home range shape parameter,  $\sigma$  in our analyses above, and its estimation is  
1223 greatly improved with even moderate amounts of telemetry data (see also Sollmann et al.  
1224 (2013) and Sollmann et al. (in revision). This should have some consequences in terms  
1225 of the design of capture-recapture studies (Chapt. 10), especially as it relates to trap  
1226 spacing.

1227 Simultaneously conducting telemetry studies with capture-recapture is extremely com-  
1228 mon in field studies of animal populations. However, the simultaneous, integrated analysis  
1229 of the two sources of data is uncommon. The new class of integrated SCR/RSF models  
1230 based on Royle et al. (2012a) allows researchers to model how the landscape and habitat  
1231 influence the movement and space use of individuals around their home range, using non-  
1232 invasively collected capture-recapture data that can be augmented with telemetry data.  
1233 This should improve our ability to understand, and study, aspects of space usage and  
1234 it might, ultimately, aid in addressing conservation-related problems such as reserve or  
1235 corridor design. This should greatly expand the relevance and utility of spatial capture-  
1236 recapture beyond its use for density estimation.



12037  
12038

---

# 14

12039  
12040

## STRATIFIED POPULATIONS: MULTI-SESSION AND MULTI-SITE DATA

12041 In this chapter, we describe SCR models for situations when we have multiple distinct  
12042 sample groups, strata or “sessions” (the term used in **secr**) each with a population size  
12043 parameter  $N_g$ , for group  $g$ . Such “stratified” populations are commonplace in capture-  
12044 recapture studies, especially in the context where the strata represent distinct spatial  
12045 regions, yet most SCR applications have been based on models that are distinctly single-  
12046 population models. This is done either by analyzing separate data sets one-at-a-time,  
12047 producing many, if not dozens, of independent estimates of abundance, or by pooling data  
12048 from multiple study areas. A standard example that arises frequently is that in which  
12049 multiple habitat patches (often refuges, parks or reserves) are sampled independently  
12050 with the goal of estimating the population size of some focal species in each reserve.  
12051 If there are parameters that can be shared across sessions or groups, it makes sense to  
12052 combine the data together into a single model that permits the sharing of information  
12053 about some parameters, but provides individual estimates of abundance for each land  
12054 unit. A similar situation is that in which a number of replicate trap arrays are located  
12055 within a landscape, sometimes for purposes of evaluating the effects of management actions  
12056 or landscape structure on populations. This is a common situation in studies of small  
12057 mammals (Converse et al., 2006a,b; Converse and Royle, 2012), or in mist-netting of birds  
12058 (DeSante et al., 1995), but there are examples of large-scale monitoring of carnivores and  
12059 other species too, e.g., tigers (Jhala et al., 2011).

12060 In previous chapters, we’ve analyzed data for a number of examples that have a natural  
12061 stratification or group structure. In Chapt. 9, we analyzed the ovenbird data as an  
12062 example of a multi-catch (independent multinomial) model, where we used year as the  
12063 stratification variable, and the possum data set (illustrating the single-catch situation) in  
12064 which the group structure arose from the use of 5 distinct trap arrays. In Chaps. 7 and 8  
12065 we fitted models with sex-specificity of parameters using multi-session models, where the  
12066 stratification variable in that case was sex. In this chapter, we focus on Bayesian analysis  
12067 of stratified SCR models using data augmentation (Converse and Royle, 2012; Royle et al.,  
12068 2012c). The technical modification of data augmentation to deal with such models is that

12069 it is based on a model for the joint distribution of the stratum-specific population sizes,  
 12070  $N_g$ , *conditioned* on their total. This results in a multinomial distribution for all  $N_g$ ,  
 12071 which we can analyze in some generality using data augmentation. As a practical matter,  
 12072 specification of this multinomial distribution for the  $N_g$  parameters *induces* a distribution  
 12073 for an individual covariate, say  $g_i$ , which is “group membership”. This is extremely handy  
 12074 to analyze by MCMC in the various **BUGS** engines that you are familiar with by now,  
 12075 and the flexibility of model specification in **BUGS** is why we focus a whole chapter here on  
 12076 Bayesian analysis by data augmentation. However, we have noted previously that the **R**  
 12077 package **secr** fits a class of multi-session models which we have already seen (Sec. 6.5.4),  
 12078 and we used **secr** to analyze several case studies using the multi-session models including  
 12079 the ovenbird (Sec. 9.2.4) and the possum data (Sec. 9.3.2), and models with sex-specific  
 12080 parameters in Chaps. 7 and 8.

12081 In the stratified population models considered here, an individual is assumed to be  
 12082 a member of a single stratum, so that the population sizes  $N_g$  for the  $g$  strata are inde-  
 12083 pendent of one another. However, stratified or multi-session SCR models are also directly  
 12084 relevant when the stratification index is time, either involving distinct periods within a  
 12085 biological season, or even across years. In this case, individuals might belong to multiple  
 12086 of the strata, but, the models discussed in this chapter do not acknowledge that explicitly.  
 12087 Unlike the case in which the strata represent spatial units, with temporally defined strata,  
 12088 we imagine a fully dynamic, or demographically open model for  $N$  might be appropriate  
 12089 – one that involves survival and recruitment. We deal with those models specifically in  
 12090 Chapt. 16. However, the stratified models covered here can be thought of as a primitive  
 12091 type of model for open systems in which the population sizes are assumed to be *inde-  
 12092 pendent* across temporal strata, and so we might still find them useful in cases where the  
 12093 strata are temporal periods or sessions.

## 14.1 STRATIFIED DATA STRUCTURE

12094 We suppose that  $g = 1, 2, \dots, G$  strata (or groups), having sizes  $N_g$ , and state-spaces  $\mathcal{S}_g$ ,  
 12095 are sampled using some capture-recapture method producing sample sizes of  $n_g$  unique  
 12096 individuals and encounters  $y_{ijk}$  for individual  $i = 1, 2, \dots, \sum_{g=1}^G n_g$ . Right now we won’t  
 12097 be concerned with the details of every type of capture-recapture observation model so,  
 12098 for context, and to develop some technical notions, we consider a Bernoulli encounter  
 12099 model in which individual and trap-specific encounter frequencies are binomial counts:  
 12100  $y_{ij} \sim \text{Binomial}(K, p_{ij})$ . Let  $g_i$  be a covariate (integer-valued,  $1, \dots, G$ ) indicating the  
 12101 group membership of individual  $i$ . This covariate is *observed* for the sample of captured  
 12102 individuals but not for individuals that are never captured.

12103 To illustrate the prototypical data structure for stratified SCR data, we suppose that  
 12104 a population comprised of 4 groups is sampled  $K = 5$  times. Then, a plausible data set  
 12105 has the following structure:

```
12106   individual (i) : 1 2 3 4 5 6 7 8 9 10
12107   total   encounters (y) : 1 1 3 1 1 2 2 4 1 1
12108       group (g)    : 1 1 1 2 3 3 3 3 4 4
```

12109 This data set indicates three individuals were captured in group 1 (captured 1, 1, and  
 12110 3 times), a single individual was captured in group 2, four individuals were captured in  
 12111 group 3, and two individuals were captured in group 4.

A key idea discussed shortly is that the assumption of certain models for the collection of abundance variables  $N_g$  implies a specific model for the group membership variable  $g_i$ . Then, the data from all groups can be pooled, and analyzed as data from a single population with the appropriate model on  $g_i$ , without having to deal with the  $N_g$  parameters in the model directly. In this way, we can easily build hierarchical models for stratified populations, using an *individual* level parameterization of the model. Obviously this is important for SCR models as they all possess at least one individual level random effect in the form of the activity center  $\mathbf{s}$ . In the context of stratified or multi-session type models, the “population membership” variable  $g_i$  is a *categorical* type of individual covariate (Huggins, 1989; Alho, 1990; Royle, 2009b). Before considering SCR models specifically, in the next section we talk a little bit about the technical formulation of data augmentation for stratified populations in the context of ordinary closed population models.

## 14.2 MULTINOMIAL ABUNDANCE MODELS

One of the key ideas to Bayesian analysis of stratified population models is that we make use of multinomial models for allocating individuals into strata or sessions. We do this because it allows us to analyze the models by data augmentation (Converse and Royle, 2012; Royle and Converse, in review), and it has a natural linkage to the Poisson model, which is commonly used throughout ecology to model variation in abundance.

To motivate the technical framework, consider sampling  $g = 1, 2, \dots, G$  groups having unknown sizes  $N_g$ , and we wish to impose model structure on the group-specific population size variables using a Poisson distribution:

$$N_g \sim \text{Poisson}(\lambda_g) \quad (14.2.1)$$

with

$$\log(\lambda_g) = \beta_0 + \beta_1 C_g \quad (14.2.2)$$

where  $C_g$  is some measured attribute for group  $g$ . We could generalize this a bit by considering a random effect in Eq. 14.2.2, producing over-dispersed population sizes  $N_g$ . For the special case of adding log-gamma noise, this results in negative binomial models for  $N_g$ .

To develop a data augmentation scheme for this group-structured model, let’s think about doing data augmentation on each population *individually*, by assuming that

$$N_g \sim \text{Binomial}(M_g, \psi)$$

where  $\psi \sim \text{Uniform}(0, 1)$  as usual. A key point is that we allow  $M_g$  to be population specific but  $\psi$  is constant. We could do this multi-population data augmentation by just picking each  $M_g$  to be some large integer (as we always do by data augmentation; see Sec. 9.2.4). However, we want to pick  $M_g$  in a way that induces the correct structure on  $N_g$ . If we want to enforce our Poisson model on  $N_g$  from above, we naturally choose  $M_g$  to be Poisson also, in which case the marginal distribution of  $N_g$  is also Poisson, but with mean  $\psi \exp(\beta_0 + \beta_1 C_g)$ . Here, clearly  $\psi$  and  $\beta_0$  are confounded (see below for more discussion). Regardless, for multiple groups that we want to model jointly, the key point is that we impose the structure that we desire for  $N_g$ , on the super-population parameters  $M_g$ . To implement this model at the individual level we need to get rid of the

12149  $M_g$  parameters (which is the entire motivation of data augmentation in the first place).  
 12150 So we condition on the “total super-population” size  $M_T = \sum_g M_g$  (in a sense, this is  
 12151 the super-super-population!). Then, the vector  $\mathbf{M} = (M_1, \dots, M_G)$  has a multinomial  
 12152 distribution:

$$\mathbf{M}|M_T \sim \text{Multinomial}(M_T; \boldsymbol{\pi}) \quad (14.2.3)$$

12153 where  $\pi_g = \lambda_g / \sum_g \lambda_g$ . This is handy because we can implement this model, e.g., in  
 12154 **BUGS**, by introducing a variable  $g_i$  for each  $i = 1, 2, \dots, M_T$  which is the “group mem-  
 12155 bership” of each individual in the super-super-population. Then, conditional on  $g_i$ , an  
 12156 individual is either “real”, or a pseudo-individual, according to the binary data augmen-  
 12157 tation variable  $z_i$ . As specified in **BUGS** pseudo-code, the model is:

```
12158     psi ~ dunif(0,1)
12159     for(g in 1:G){
12160         pi[g] <- lambda[g]/sum(lambda[])
12161     }
12162     g[i] ~ dcat(pi[1:G])
12163     z[i] ~ dbern(psi)
```

12164 This produces a vector of population size parameters  $\mathbf{N} = (N_1, \dots, N_G)$  which are ap-  
 12165 proximately, for large  $M_T$ , independent Poisson random variables.

12166 When we apply data augmentation to the multinomial joint distribution, the  $\psi$  pa-  
 12167 rameter takes the place of  $N_T$ , the total population size (across all groups or strata).  
 12168 In addition, by constructing the model conditional on the total,  $N_T$ , we lose information  
 12169 about the intercept  $\beta_0$ <sup>1</sup> but this is recovered in the data augmentation parameter  $\psi$ . Thus,  
 12170 one of these parameters has to be fixed. We can set  $\beta_0 = 0$  or else we can fix  $\psi$  (see Chapt.  
 12171 11). The constraint can be specified by noting that, under the binomial data augmenta-  
 12172 tion model  $\mathbb{E}(N_T) = \psi M_T$  and, under the Poisson model,  $\mathbb{E}(N_T) = \sum_g \exp(\beta_0 + \beta_1 C_g)$   
 12173 and so we can set

$$\psi = \frac{1}{M_T} \sum_g \exp(\beta_0 + \beta_1 C_g).$$

12174 The linkage of  $\beta_0$  and  $\psi$  was also discussed in Chapt. 11 in the context of building spatial  
 12175 models for density. In that case,  $\beta_0$  was the intercept of the intensity function and one  
 12176 could choose to estimate either  $\beta_0$  or the data augmentation parameter  $\psi$ .

### 12177 14.2.1 Implementation in **BUGS**

12178 The **BUGS** implementation of data augmentation for structured populations is straight-  
 12179 forward. For each individual in the super-super-population we introduce a latent variable  
 12180  $g_i$  to indicate which *population* the individual belongs to, and we introduce a second  
 12181 variable  $z_i$  to indicate whether the individual is alive or not. So, the latent structure for  
 12182 the  $M_g$  variables and the binomial sampling of those super-population sizes is equivalently  
 12183 represented by the latent variable pair  $(g_i, z_i)$  where  $g_i$  is categorical with prior probabili-  
 12184 ties  $\pi_s$  and  $z_i \sim \text{Bernoulli}(\psi)$ . In particular, the multinomial assumption for the latent

<sup>1</sup> A technical argument is that the total  $N_T$  is the sufficient statistic for  $\beta_0$  in the multinomial model and so, by conditioning on the total,  $\beta_0$  is no longer a free parameter.

variables  $M_g$  is formulated in terms of “group membership” for each individual in the super-super-population of size  $M_T$  according to:

$$g_i \sim \text{Categorical}(\boldsymbol{\pi})$$

with  $\boldsymbol{\pi} = (\pi_1, \dots, \pi_G)$  and  $\pi_g = \lambda_g / (\sum_g \lambda_g)$ . The binomial sampling is described by the binary variables  $z_1, \dots, z_{M_T}$  such that

$$z_i \sim \text{Bernoulli}(\psi)$$

where  $\psi$  is constrained as noted in the previous section. The **BUGS** model specification for this individual-level formulation of the model is shown in Panel 14.1 for an ordinary closed population model (model  $M_0$ ). This actually shows two equivalent formulations. In the left panel we have  $\psi$  and  $\beta_0$  as free parameters. The right panel shows the equivalent model but recognizing the constraint between  $\psi$  and  $\beta_0$ . Running these models using the `multisession.sim` function, you can verify that the two parameters are not uniquely estimable. In particular, using the model (representation 1) in the left-hand side of Panel 14.1, you will see that draws of  $\beta_0$  appear to be draws from the prior distribution, uninformed by the data, supporting the point we made previously that  $\psi$  and  $\beta_0$  are not uniquely informed by the data.

### 14.2.2 Groups with no individuals observed

In practical settings, when the groups represent small populations, it will sometimes happen that some groups have no encountered individuals or even that  $N_g = 0$  for some groups. This is dealt with implicitly in the development of the model shown in Panel 14.1 in the sense that the *prior* for  $N_g$  has the proper dimension (namely,  $G$  multinomial cells of non-zero probability) and thus some posterior mass may occur on non-zero values of  $N_g$  even if the *data* contain no representatives of group  $g$ . You can try this out to verify for yourself.

### 14.2.3 The group-means model

Under the Poisson model for group abundance  $N_g$ , even with a constant mean  $\lambda$ , each stratum or group may have a different realized population size, and this comes at the low price of a single parameter in the model ( $\lambda$  or, equivalently, the data augmentation parameter  $\psi$ ). Thus, for a single parameter in this group-structure model, we are able to realize variation in the  $N_g$  parameters. In a sense, this is a benefit of the group structure in which  $N_g$  are regarded as random variables.

To accommodate more flexibility than afforded by the single-parameter Poisson model, there are a couple of choices: (1) We could allow the mean to be group specific such as:  $N_g \sim \text{Poisson}(\lambda_g)$  where each  $\lambda_g$  is its own free parameter, independent of each others. This produces a model with  $G$  distinct “fixed” parameters, and effectively renders the Poisson assumption irrelevant as it doesn’t induce any “Bayesian shrinkage” (Sauer and Link, 2002) or impose any group structure on the population sizes  $N_g$ . It should provide estimates that are effectively the same as analyzing each data set independently, or using the independent binomial prior that we introduced in Chapt. 9, where some information

---

Implementation 1	Implementation 2
<pre> model {   # This will show that psi and b0   #   are confounded.   p ~ dunif(0,1)   beta0 ~ dnorm(0,.1)   beta1 ~ dnorm(0,.1)   psi ~ dunif(0,1)   for(j in 1:G){     log(lam[j]) &lt;- beta0+beta1*C[j]     gprobs[j]&lt;-lam[j]/sum(lam[1:G])   }   for(i in 1:M){     g[i] ~ dcat(gprobs[])     z[i] ~ dbern(psi)     mu[i] &lt;- z[i]*p     y[i] ~ dbin(mu[i],K)   }   N &lt;- sum(z[1:M]) } </pre>	<pre> model {   # This version constrains psi with   #   the intercept parameter   p ~ dunif(0,1)   beta0 ~ dnorm(0,.1)   beta1 ~ dnorm(0,.1)   psi &lt;- sum(lam[])/M   for(j in 1:G){     log(lam[j]) &lt;- beta0+beta1*C[j]     gprobs[j]&lt;-lam[j]/sum(lam[1:G])   }   for(i in 1:M){     g[i] ~ dcat(gprobs[])     z[i] ~ dbern(psi)     mu[i] &lt;- z[i]*p     y[i] ~ dbin(mu[i],K)   }   N &lt;- sum(z[1:M]) } </pre>

---

Panel 14.1: BUGS model specification for a capture-recapture model with constant encounter probability and Poisson subpopulation sizes,  $N_g$ , with mean depending on a single covariate  $C[j]$ . Two versions of the model: The first one describes the model in terms of the intercept  $\beta_0$  and DA parameter  $\psi$ , which are confounded. The required constraint is indicated in the specification under Implementation 2.

12222 might be borrowed from the different groups for estimating the encounter probability  
 12223 parameters. Under this model, we constraint one of the  $\lambda_g$  parameters to be 0, and  $N_g$   
 12224 for that group is taken up by the data augmentation parameter  $\psi$ ; (2) Alternatively, we  
 12225 could identify specific fixed covariates which might explain variation across groups. Each  
 12226 additional covariate adds only 1 additional fixed parameter to the model; (3) A flexible  
 12227 formulation that provides something of an intermediate model, between that of a constant  
 12228  $\lambda$  and independent group specific  $\lambda_g$ 's, is that in which we put a prior on  $\lambda_g$ . For example,  
 12229 if we assume

$$\lambda_g \sim \text{Gamma}(a, b)$$

12230 this corresponds to imposing a Dirichlet compound-multinomial model on the population  
 12231 size vector, or, marginally, a negative binomial model on  $N_g$ . See Takemura (1999) for  
 12232 some discussion of such models relevant to data augmentation. For this model, we impose  
 12233 the constraint  $b = 1$  to account for conditioning on the total population size  $N_T$  to use  
 12234 data augmentation.

---

**14.2.4 Simulating stratified capture-recapture data**

It is helpful, as always, to simulate some data in order to understand the model. Suppose we cracked the conservation lotto jackpot and obtained funding to carry out a camera trapping study of some flashy carnivore in 20 forest patches or reserves, using a 5 x 5 array of traps. Here we will consider an ordinary closed population model, model  $M_0$ , and we suppose there is some forest level covariate, say  $\text{Dist}$  = disturbance regime, perhaps measured by an index of trail density or something. We imagine a model for patch-level population size such as the following:

$$N_g \sim \text{Poisson}(\lambda_g)$$

$$\log(\lambda_g) = \beta_0 + \beta_1 \text{Dist}_g$$

We simulate some population sizes and encounter data under this model as follows:

```

12244 > set.seed(2013)
12245 > G <- 20                                # G = 20 groups or strata
12246 > beta0 <- 3                             # Abundance model parameters
12247 > beta1 <- .6
12248 > p <- .3                               # Encounter probability
12249 > K <- 5                                # Sample occasions for capture-recapture
12250 > Dist <- rnorm(G)                      # Simulate covariate
12251 > lambda <- exp(beta0+beta1*Dist)    # Simulate population sizes
12252 > N <- rpois(G,lambda=lambda)

12253
12254 > y <- NULL                            # Simulate model M0 data
12255 > for(g in 1:G){
12256   + if(N[g]>0)
12257   +   y <- c(y, rbinom(N[g],K,p))
12258   + }
12259 > g<- rep(1:G,N)
12260
12261 > ## Now keep the group id and encounter frequency only for
12262 > ## individuals that are captured
12263 > g<-g[y>0]
12264 > y<-y[y>0]
```

That's it! We just simulated a population size model and capture-recapture data for the populations inhabiting  $G = 20$  forest patches (the "groups" in this situation). To fit this model, we need to augment the  $\mathbf{g}$  and  $\mathbf{y}$  data objects, and then we can run the model in **JAGS** or **WinBUGS** using the code given in Panel 14.1. See the help file `?multisession.sim` for doing this analysis with these simulated data.

### 14.3 OTHER APPROACHES TO MULTI-SESSION MODELS

The multinomial super-population model allows for the joint modeling of a collection of population sizes using data augmentation. However, as we demonstrated in Sec. 9.2.4, we can analyze the models by putting independent binomial priors on each  $N_g$  and doing

the data augmentation independently for each population by itself. This is not any more or less difficult than the multinomial formulation but, we imagine, it could be slightly less efficient computationally. In this case we could build in among-group structure by modeling the DA parameter  $\psi$  as being variable for each subject, as a function of group-specific variables (see Hendriks et al., 2013, for an example). For example, if  $C_g$  is the value of some covariate for group  $g$ , then we could have  $z_i \sim \text{Bernoulli}(\psi_i)$  with

$$\text{logit}(\psi_i) = \beta_0 + \beta_1 C_{g_i}$$

This implies a binomial model for the stratum population sizes:

$$N_g \sim \text{Binomial}(M, \psi_g).$$

If  $M$  is large then the  $N_g$  are approximately independent Poisson random variables with means  $\psi_g M$ .

As we noted in Chapt. 6, the multi-session models in **secr** are based on a Poisson prior for  $N_g$  with mean  $\Lambda_g$ , and then among group structure is modeled in the parameter  $\Lambda_g$ . In our view, either model (binomial based on data augmentation, or Poisson) is satisfactory for any application of capture-recapture to stratified populations. The main advantage of the formulation we provided here over that implemented in **secr** is we have quite a bit more flexibility in specifying models of all sorts, either in the population size model for  $N_g$ , or for the capture-recapture model. For example, Royle and Converse (in review) fitted a model having random group effects on encounter probability and abundance (i.e., extra-Poisson variation).

## 14.4 APPLICATION TO SPATIAL CAPTURE-RECAPTURE

Although we developed the implementation of Bayesian models for stratified populations using ordinary closed population models, the underlying ideas are completely general and can be applied equally to spatial capture-recapture models without any novel considerations. We already discussed (Chapt. 4) that SCR models are ordinary closed population models but with an individual covariate which is the activity center  $s_i$ , and the observation model has to be defined for each trap. With this in mind, it should be obvious how the **BUGS** specification in Panel 14.1 can be modified to accommodate a group-structured SCR situation. Specifically, we include the prior distribution for  $s_i$  and the observation model that relates  $s_i$  to the probability of encounter for individual  $i$  and trap  $j$ , as we've done so many times in previous chapters.

### 14.4.1 Multinomial (“multi-catch”) observations

We discuss Bayesian analysis of the multi-session model using data augmentation in the context of a multinomial observation model such as for a multi-catch sampling situation<sup>2</sup>. For context, we return to the ovenbird data set, from the **R** package **secr**, which we introduced in Chapt. 9. Another example can be found in Royle and Converse (in review),

<sup>2</sup>This might be slightly confusing that we are considering multinomial observation models *and* multinomial models for group-specific abundance parameters  $N_g$ , but we will take care to be clear about this along the way.

12306 who applied the model to a small mammal trapping problem which involved replicate  
 12307 “single-catch” arrays of traps, in a study of the effects of forest management practices on  
 12308 small-mammal densities. The ovenbird data is a type of multi-catch observation model  
 12309 where the group index variable is “year” and, in our earlier analyses, we analyzed the  
 12310 data set using independent binomial priors for  $N_g$  within data augmentation in **JAGS**,  
 12311 as well as with a Poisson prior in **secr** using the multi-session models. We mirror the  
 12312 **secr** analysis here, but using the data augmentation formulation leading to a multinomial  
 12313 distribution for  $N_g$  we introduced above.

12314 To refresh your memory about the multinomial observation model, let  $\mathbf{y}_{ik} = (y_{i1k}, y_{i2k}, \dots, y_{iJk}, y_{i,J+1,k})$   
 12315 be the spatial encounter history for individual  $i$ , during sample occasion  $k$  where the last  
 12316 element  $y_{i,J+1,k}$  corresponds to “not captured”. For mist nets, an individual can be cap-  
 12317 tured in at most one trap. Then, the vector  $(y_{i1k}, y_{i2k}, \dots, y_{iJk}, y_{i,J+1,k})$ , contains a single  
 12318 1 and the remaining values are 0. This  $(J + 1) \times 1$  vector  $\mathbf{y}_{ik}$  is a multinomial trial:

$$\mathbf{y}_{ik} \sim \text{Multinomial}(n = 1; \boldsymbol{\pi}_{ik})$$

12319 where  $\boldsymbol{\pi}_{ik}$  is a  $(J + 1) \times 1$  vector where each element represents the probability of being  
 12320 encountered in a trap (for elements  $1, \dots, J$ ) or not captured at all (element  $J + 1$ ).

12321 For the multinomial observation model, the encounter probability vector is a func-  
 12322 tion of distance between trap locations and individual activity centers, modeled on the  
 12323 multinomial logit scale. The Gaussian encounter probability model is:

$$\text{mlogit}(\pi_{ij}) = \eta_{ij} = \alpha_0 - \alpha_1 \text{dist}(\mathbf{x}_j, \mathbf{s}_i)^2 \quad (14.4.1)$$

12324 where  $\alpha_1 = 1/(2\sigma^2)$  and  $\sigma$  is the scale parameter of the Gaussian model. Then,

$$\boldsymbol{\pi}_{ij} = \exp(\eta_{ij}) / [1 + \sum_j \exp(\eta_{ij})]$$

12325 for each  $j = 1, 2, \dots, J$ , and the last cell corresponding to the event “not captured” is:

$$\pi_{i,J+1} = 1 - \sum_{j=1}^J \pi_{ij}$$

12326 There are no novel technical considerations in order to model covariates of any kind.  
 12327 For example, in many studies we are concerned with a behavioral response to physical  
 12328 capture. This is typical in small-mammal trapping studies, and also in mist-net studies  
 12329 of birds where individuals exhibit net avoidance after first capture. For this, let  $C_{ik}$  be  
 12330 a covariate of previous encounter (i.e.,  $C_{ik} = 0$  before the occasion of first capture, and  
 12331  $C_{ik} = 1$  thereafter), then we include this covariate in our multinomial observation model  
 12332 as follows:

$$\text{mlogit}(\pi_{ijk}) = \eta_{ijk} = \alpha_0 - \alpha_1 \text{dist}(\mathbf{x}_j, \mathbf{s}_i)^2 + \alpha_2 C_{ik}$$

12333 We note that, in this case, the multinomial probabilities depend not only on individual  
 12334 and trap, but also on sample occasion.

**Table 14.1.** Posterior summaries for the Bayesian stratified population (“multi-session”) model fitted to the ovenbird data. Results are based on 3 chains, each with 5000 iterations (first 1000 discarded), for a total of 12000 iterations saved.

	Mean	SD	2.5%	50%	97.5%	Rhat
D[1]	0.883	0.191	0.562	0.868	1.308	1.002
D[2]	0.972	0.200	0.624	0.954	1.418	1.001
D[3]	1.146	0.224	0.758	1.125	1.638	1.001
D[4]	0.836	0.183	0.538	0.819	1.247	1.001
D[5]	0.705	0.167	0.428	0.685	1.088	1.001
N[1]	72.208	15.596	46.000	71.000	107.000	1.002
N[2]	79.478	16.367	51.000	78.000	116.000	1.001
N[3]	93.725	18.327	62.000	92.000	134.000	1.001
N[4]	68.399	14.952	44.000	67.000	102.000	1.001
N[5]	57.665	13.659	35.000	56.000	89.000	1.001
alpha0	-3.465	0.159	-3.779	-3.465	-3.155	1.004
alpha1	0.000	0.000	0.000	0.000	0.000	1.009
beta0[1]	4.250	0.244	3.754	4.257	4.710	1.001
beta0[2]	4.349	0.233	3.872	4.356	4.786	1.001
beta0[3]	4.516	0.220	4.059	4.522	4.930	1.001
beta0[4]	4.194	0.248	3.697	4.202	4.664	1.001
beta0[5]	4.013	0.275	3.456	4.022	4.524	1.001
psi	0.371	0.051	0.281	0.367	0.482	1.001
sigma	77.918	6.314	66.963	77.240	91.583	1.009

#### 12335 14.4.2 Reanalysis of the Ovenbird data

12336 Here we use Bayesian analysis by data augmentation to fit a model that approximates the  
 12337 Poisson model with expected value  $\mathbb{E}(N_g) = \lambda_g$  where we model effects on the log-mean  
 12338 scale according to:

$$\log(\lambda_g) = \beta_0 + \beta_1 C_g.$$

12339 We considered only two models here: A model with year-specific abundance, and a model  
 12340 with a linear trend in density over time, so  $C_g \equiv \text{Year}$ . However, using the Kuo and  
 12341 Mallick (1998) indicator variable selection idea (see Chapt. 8), the linear trend term was  
 12342 found to have little or no posterior probability, so we do not reproduce analyses of that  
 12343 here (but see the `ovenbird.ms` function for the **R** script). We show the **BUGS** model  
 12344 specification for the year-specific abundance model in Panel 14.2. Note the construction  
 12345 of the multinomial cell probabilities which distribute individuals among years, based on  
 12346 the year-specific mean  $\lambda_t$ . On the log-scale, each of these parameters has a diffuse normal  
 12347 prior: `beta0[t] ~ dnorm(0, 0.01)`. A few lines of model specification that compute the  
 12348 derived population size parameters and density are not shown, but you can look at the **R**  
 12349 script `ovenbird.ms` in `scrbook` to run this analysis, and produce the posterior summaries  
 12350 shown in Table 14.1.

12351 We previously analyzed these data in Sec. 9.2.4 using `secr` and the “one-at-a-time”  
 12352 data augmentation approach (independent binomial priors for  $N_t$ ). To reproduce those  
 12353 results from `secr` for the equivalent model we execute this command:

```
12354 > ovenbird.model.DT<-secr.fit(ovenCH,model=list(D~session),buffer=300)
```

```

model {
  alpha0 ~ dnorm(0,.01)                      # Prior distributions
  sigma ~ dunif(0,200)
  alpha1 <- 1/(2*sigma*sigma)
  psi <- sum(lambda[]) / bigM
  for(t in 1:5){
    beta0[t] ~ dnorm(0,0.01)                  # Year-specific abundances
    log(lambda[t]) <- beta0[t]
    pi[t] <- lambda[t]/sum(lambda[])
  }                                            # Calculate multinomial probs
  for(i in 1:bigM){
    z[i] ~ dbern(psi)
    yrid[i] ~ dcat(pi[])
    S[i,1] ~ dunif(xlim[1],xlim[2])          # Activity centers
    S[i,2] ~ dunif(ylim[1],ylim[2])
    for(j in 1:ntraps){
      d2[i,j] <- pow(pow(S[i,1]-X[j,1],2) + pow(S[i,2]-X[j,2],2),1)
    }
    for(k in 1:K){
      Ycat[i,k] ~ dcat(cp[i,k])
      for(j in 1:ntraps){                      # Construct trap enc. probs.
        lp[i,k,j] <- exp(alpha0 - alpha1*d2[i,j])*z[i]*(1-died[i,k])
        cp[i,k,j] <- lp[i,k,j]/(1+sum(lp[i,k,1:ntraps]))
      }
      cp[i,k,ntraps+1] <- 1-sum(cp[i,k,1:ntraps]) # last cell = not captured
    }
  }
}

```

Panel 14.2: BUGS model specification for a stratified (multi-session) SCR model using data augmentation. This shows a multinomial (“multi-catch”) type of observation model, used to analyze the ovenbird data. Some code to tally up the derived population sizes and density parameters is omitted. See ovenbird.ms script

**Table 14.2.** Estimates for the multi-session model fitted to the ovenbird data using `secr`. The model had a year-specific density parameter, and constant encounter probability parameters.

2005					
	Link	Estimate	SE Estimate	LCL	UCL
D	log	0.920	0.228	0.571	1.484
g0	logit	0.028	0.004	0.021	0.037
sigma	log	78.567	6.379	67.025	92.095
2006					
	Link	Estimate	SE Estimate	LCL	UCL
D	log	0.963	0.238	0.598	1.553
g0	logit	0.028	0.004	0.021	0.037
sigma	log	78.566	6.379	67.025	92.095
2007					
	Link	Estimate	SE Estimate	LCL	UCL
D	log	1.139	0.282	0.706	1.836
g0	logit	0.028	0.004	0.021	0.037
sigma	log	78.566	6.379	67.025	92.095
2008					
	Link	Estimate	SE Estimate	LCL	UCL
D	log	0.832	0.206	0.516	1.341
g0	logit	0.028	0.004	0.021	0.037
sigma	log	78.566	6.379	67.025	92.095
2009					
	Link	Estimate	SE Estimate	LCL	UCL
D	log	0.701	0.173	0.435	1.130
g0	logit	0.028	0.004	0.021	0.037
sigma	log	78.566	6.379	67.025	92.095

12355 Note, small values of `buffer` can produce a warning that it is too small relative to the  
 12356 indicated value of  $\sigma$  (which has posterior mass up to near  $\sigma = 100$ ). The `secr` results  
 12357 are as follows shown in Table 14.2. There are, as always, slight differences between the  
 12358 MLEs shown here and the posterior summaries shown Table 14.1. The absolute difference  
 12359 between the MLEs and the Bayesian posterior means was .037, -.011, -.006, -.004 and  
 12360 -.004 for years 1 to 5, respectively.

## 14.5 SPATIAL OR TEMPORAL DEPENDENCE

12361 The models described here, and including the multi-session formulation used in `secr`,  
 12362 assume that the population sizes  $N_g$  are *independent* (in a limiting sense, under data  
 12363 augmentation). As a practical matter, this precludes the sharing of individuals among  
 12364 populations (i.e., the same individual cannot be captured in multiple groups) which can  
 12365 be violated in a number of situations. First, when the groups represent sampling in distinct  
 12366 time periods (seasons, years) but of the same functional population (a standard “robust  
 12367 design” situation), it is possible that some individuals remain in the population from one  
 12368 time period to the next. In this situation, by disregarding individual identity across groups,  
 12369 the models ignore a slight bit of dependence of  $N_g$  which may entail some incremental loss  
 12370 of efficiency. We imagine this should have little practical effect unless survival probability

12371 is extremely high between the periods. Estimators of parameters obtained by assuming  
12372 independence should be conservative in their statement of precision, but they should be  
12373 unbiased (or, rather, ignoring the dependence should not affect the bias of the estimator  
12374 much if at all).

12375 A second distinct situation is that in which the stratification variable is *spatial*, and  
12376 the strata (e.g., trap arrays or other sampling mechanism) are in relatively close spatial  
12377 proximity to one another so that individuals can sometimes be encountered by more than  
12378 one array (e.g., the possum data, see Fig. 9.2). This case is somewhat easier to deal  
12379 with in the analysis because we can build a model in which the state-space is the joint  
12380 state-space enclosing all of the trapping arrays, and we preserve individual identity in an  
12381 ordinary SCR model, just with a larger array of traps that is the union of the trap arrays  
12382 of all sample groups. This may be impractical when the trap arrays are far apart creating  
12383 only a slight bit of overlap of populations, because, in that case, the combined state-  
12384 space may contain a huge population that one has to deal with in the MCMC (remember  
12385 that increasing  $M$  increases computation time). (Royle et al., 2011a) had this problem  
12386 in an analysis of data from a sample of 1 km quadrats using a search-encounter type  
12387 model (discussed in the following chapter). Even in this case the independent  $N_g$  model  
12388 is probably not too detrimental to inferences that apply to explaining marginal variation  
12389 in  $N_g$ , such as habitat or landscape effects that are modeled on the expected value of  $N_g$ .

## 14.6 SUMMARY AND OUTLOOK

12390 Capture-recapture data are not always collected as single isolated studies but, instead,  
12391 data are often grouped or stratified in some natural way, either because a number of  
12392 distinct trap arrays are used, or sampling occurs in several forest patches, or over time.  
12393 Often this is motivated by specific objectives, e.g., the trap arrays or units represent  
12394 experimental replicates, or sometimes just to derive more valid estimates of density by  
12395 obtaining a representative (ideally, random) sample of space within some region. The fact  
12396 that data are grouped in such a way raises the obvious technical problem of having to  
12397 combine data from multiple arrays, sites or otherwise defined groups in a single unified  
12398 model that accommodates explicit sources of variation in density among these groups.  
12399 This is naturally accomplished by developing an explicit model for variation in  $N$ , e.g., a  
12400 Poisson GLM or similar (Converse and Royle, 2012; Royle et al., 2012c).

12401 In this chapter, we outlined an approach to Bayesian analysis of multi-session models  
12402 using data augmentation Converse and Royle (2012); Royle and Converse (in review). This  
12403 approach gives us one method for building explicit models for  $N_g$  and also gives us great  
12404 flexibility in specifying the encounter model using standard or novel capture-recapture  
12405 modeling considerations. Certain types of multi-session models can be fitted easily in  
12406 **secr** (see Chapt. 9) and we suspect that platform will be satisfactory for many problems  
12407 you encounter. However, as always, we believe the flexible model-building platform of the  
12408 **BUGS** language can be beneficial in many situations.

12409 A common applied context of these multi-session models is when replicate arrays are  
12410 used to address explicit hypotheses about the effects of landscape variation or modification  
12411 on abundance. For example, in studies of forestry practices and their effects on local fauna,  
12412 small mammal grids are used as experimental units, and the “dependent variable” is  $N$   
12413 (or density) of small mammals (or some small mammal focal species) for each trap array,

which is not observable. Thus, hierarchical models are needed to directly address the basic hypotheses of such studies. Another distinct context for the application of multi-session models is when the populations are temporally structured (e.g., the ovenbird data), such as when sampling occurs in distinct seasons or years. In these applications, we view multi-session models as a simplified type of open population model, an open model *without* explicit Markovian dynamics. They are analogous to what is usually referred to as models of random temporary emigration (Kendall et al., 1997; Chandler et al., 2011). The models are not incorrect, just simplified, reduced versions of more general Markovian models, and with fewer parameters to estimate. We cover general Markovian models in Chapt. 16.

12423  
12424

12425  
12426

# 15

---

## MODELS FOR SEARCH-ENCOUNTER DATA

12427 In this chapter we discuss models for search-encounter data. These models are useful in  
12428 situations where the locations of individuals, say  $\mathbf{u}_{ik}$  for individuals  $i$  and sample occasions  
12429  $k$ , are observed directly by searching space (often delineated by a polygon) in some  
12430 fashion, rather than restricted to fixed trap locations. In all the cases addressed in this  
12431 chapter, both detection probability and parameters related to movement can be estimated  
12432 using such models. To formalize this notion a little bit using some of the ideas we've in-  
12433 troduced in previous chapter, most of the SCR models we've talked about in the book  
12434 involve just two components of a hierarchical model, the observation component, which  
12435 we denote by  $[y|\mathbf{s}]$  (e.g., Bernoulli, Poisson, or multinomial), and the process component  
12436 describing the activity center model  $[\mathbf{s}]$ , the point process model for the activity centers.  
12437 The search-encounter models described here involve an additional component for the loca-  
12438 tions conditional on the activity centers. We write this as follows: The observation model  
12439 has the form  $[y|\mathbf{u}]$ , and the process model has two components, a movement model  $[\mathbf{u}|\mathbf{s}]$ ,  
12440 which describes the individual encounter locations conditional on  $\mathbf{s}$ , and the point process  
12441 model  $[\mathbf{s}]$ . Because we can resolve parameters of the  $[\mathbf{u}|\mathbf{s}]$  component, search-encounter  
12442 models are slightly more complicated, and also more biologically realistic. Conversely,  
12443 when we have an array of fixed trap locations, the movement process is completely con-  
12444 founded with the encounter process because the list of potential observation locations is  
12445 prescribed, a priori, independent of any underlying movement process.

12446 A few distinct types of situations exist where search-encounter models come in handy.  
12447 The prototypical, maybe ideal, situation Royle et al. (2011a) is where we have a single  
12448 search path through a region of space from which observations are made (just as in the  
12449 typical distance sampling situation, using a transect). As we walk along the search path,  
12450 we note the location of each individual that is detected, *and their identity* (this is different  
12451 from distance sampling in that sense). Alternatively, we could delineate a search area, and  
12452 conduct a systematic search of that region. An example is that of Royle and Young (2008),  
12453 which involved a plot search for lizards. They assumed the plot was uniformly searched  
12454 which justified an assumption of constant encounter probability,  $p$ , for all individuals

12455 within the plot boundaries. The data set was  $\geq 1$  location observations for each of a  
 12456 sample of  $n$  individuals. The recent paper by Efford (2011a) discussed likelihood analysis  
 12457 of similar models. In the terminology of `secr` such models are referred to as models for  
 12458 *polygon detectors*.

## 15.1 SEARCH-ENCOUNTER DESIGNS

12459 Before we discuss models for search-encounter data, we'll introduce some types of sampling  
 12460 situations that produce individual location data by searching space. We imagine there are  
 12461 a lot more sampling protocols (and variations) than identified here, but these are some  
 12462 of the standard situations that we have encountered over the last few years in developing  
 12463 applications of SCR models. For our purposes here we recognize 4 basic sampling designs,  
 12464 each of which might have variations due to modification of the basic sampling protocol.

### 12465 15.1.1 Design 1: Fixed Search Path

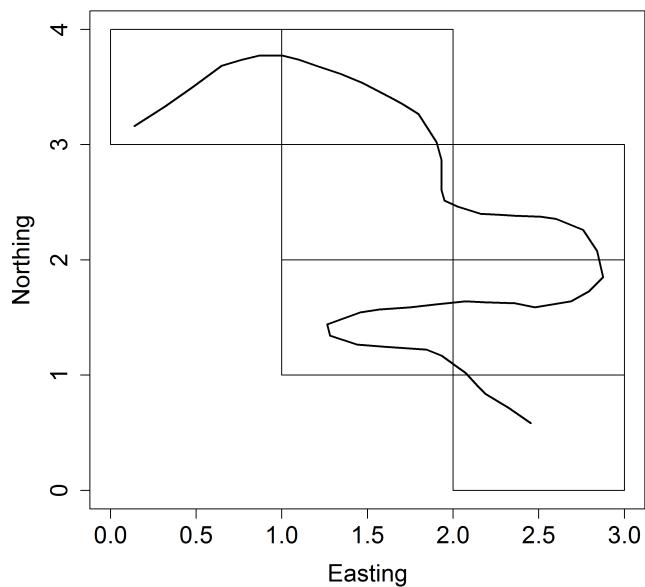
12466 A useful class of models arises when we have a fixed search-path or line, or multiple such  
 12467 lines, in some region (Fig. 15.1) from which individual detections are made. We assume the  
 12468 survey path is laid out *a priori* in some manner that is done independent of the activity  
 12469 centers of individuals and the collection of data does not affect the lines. The purpose  
 12470 of this assumption, in the models described subsequently, is to allow us to assume that  
 12471 the activity centers are uniformly distributed on the prescribed state-space. Alternatively,  
 12472 explicit models could be entertained to mitigate a density gradient or covariate effects (see  
 12473 Chapt. 11). The situation depicted in Fig. 15.1 shows the search path traversing several  
 12474 delineated polygons, although the polygon boundaries may or may not affect the potential  
 12475 locations of individuals (see below).

12476 A number of variations of this fixed search path situation are possible, and these  
 12477 produce slightly different data structures and corresponding modifications to the model,  
 12478 although we do not address all of these from a technical standpoint here:

- 12479     Protocol (1a). We know the search path and record the locations of individuals.
- 12480     Protocol (1b). We record the location of individuals and the location on the search path  
           where we first observed the individual.
- 12482     Protocol (1c). We record the closest perpendicular distance. This is a typical distance  
           sampling situation, and this is a type of hybrid SCR/distance sampling model.

### 12484 15.1.2 Design 2: Uniform search intensity

12485 In the uniform search intensity model (or just “uniform search”), we have one or more  
 12486 well-defined sample areas (polygons), such as a quadrat or a transect, and we imagine that  
 12487 the area is uniformly searched so that encounter probability is constant for all individuals  
 12488 within the search area. This type of sampling method is often called “area search” in the  
 12489 bird literature (Bibby et al., 1992). Sampling produces locations of individuals within the  
 12490 well-defined boundaries of the sample area. The polygon boundaries defining the sample  
 12491 unit are important because they tell us that  $p = 0$  by design outside of the boundary.



**Figure 15.1.** A survey line through parts of 7 quadrats in a hypothetical landscape. An observer travels the transect and identifies individuals in the vicinity of the line, recording their identity and location.

12492 Using the example from the Fig. 15.1, but ignoring the survey line through the plot  
 12493 (pretend it doesn't exist), we imagine that each of the identified quadrats is uniformly  
 12494 searched, which is to say, we assume that each individual within the boundaries of the  
 12495 *quadrat* has an equal probability of being detected. In the context of replicate sampling  
 12496 occasions (e.g., on consecutive days), individuals may move on or off of the plot, and so  
 12497 individuals may have different probabilities of being *available* to encounter, based on the  
 12498 closeness of their activity center to the quadrat boundaries. However, given that they're  
 12499 available, the uniform search model assumes they have constant encounter probability.

## 15.2 A MODEL FOR FIXED SEARCH PATH DATA

12500 In contrast to most of the models described in this book (but see Sec. 9.4), we develop  
 12501 models for encounter probability that depend explicitly on the instantaneous location  $\mathbf{u}_{ik}$ ,  
 12502 for individual  $i$  at sample occasion  $k$ , say  $p_{ik} \equiv p(\mathbf{u}_{ik}) = \Pr(y_{ik} = 1 | \mathbf{u}_{ik})$ . Note that  $\mathbf{u}$  is  
 12503 unobserved for the  $y = 0$  observations and thus we cannot analyze the conditional-on- $\mathbf{u}$   
 12504 likelihood directly. Instead, we regard  $\mathbf{u}$  as random effects and assume a model for them,  
 12505 which allows us to handle the problem of missing  $\mathbf{u}_{ik}$  values (Sec. 15.4.1). We assume  
 12506 that individuals do not move *during* a sampling occasion or, if they do, the individual is  
 12507 not added to the data set twice.

12508 To develop encounter probability models for this problem we cannot just use the  
 12509 previous models because the "trap" is actually a line or collection of line segments (e.g.,  
 12510 Fig. 15.1). Intuitively,  $\Pr(y_{ik} = 1 | \mathbf{u}_{ik})$  should increase as  $\mathbf{u}_{ik}$  comes "close" to the line  
 12511 segments  $\mathbf{X}$ . It seems reasonable to express closeness by some distance metric  $\|\mathbf{u}_{ik} - \mathbf{X}\|$   
 12512 is the distance between locations  $\mathbf{u}_{ik}$  and  $\mathbf{X}$ , and then assume

$$\text{logit}(p_{ik}) = \alpha_0 + \alpha_1 \|\mathbf{u}_{ik} - \mathbf{X}\|.$$

12513 For the case where  $\mathbf{X}$  describes a wandering line, some kind of average distance from  $\mathbf{u}$  to  
 12514 the line might be reasonable; possible alternatives include the absolute minimum distance  
 12515 or the mean over specific segments of the line (within some distance), etc. . . . We could  
 12516 also have a model without an explicit distance component, by assuming that individuals  
 12517 within a certain distance from the search path are encountered with equal probability. In  
 12518 this case, we have only a single parameter  $\alpha_0$  but must also specify the distance limit.

### 12519 15.2.1 Modeling total hazard to encounter

12520 Because the line  $\mathbf{X}$  is not a single point (like a camera trap) we have to somehow describe  
 12521 the total encounter probability induced by the line. A natural approach is to model the  
 12522 total hazard to capture (Borchers and Efford, 2008), which is standard in survival analysis,  
 12523 and also distance sampling (Hayes and Buckland, 1983; Skaug and Schweder, 1999). The  
 12524 individual is detected if encountered at any point along  $\mathbf{X}$ . Naturally, covariates are  
 12525 modeled as affecting the hazard rate and we think of distance to the line as a covariate  
 12526 acting on the hazard. Let  $h(\mathbf{u}_{ik}, \mathbf{x})$  be the hazard of individual  $i$  being encountered by  
 12527 sampling at a point  $\mathbf{x}$  on occasion  $t$ . For example, one possible model assumes, for all  
 12528 points  $\mathbf{x} \in \mathbf{X}$ ,

$$\log(h(\mathbf{u}_{ik}, \mathbf{x})) = \alpha_0 + \alpha_1 * \|\mathbf{u}_{ik} - \mathbf{x}\|. \quad (15.2.1)$$

Additional covariates could be included in the hazard function in the same way as for any model of encounter probability that we've discussed previously. The total hazard to encounter anywhere along the survey path, for an individual located at  $\mathbf{u}_{ik}$ , say  $H(\mathbf{u}_{ik})$ , is obtained by integrating over the surveyed line, which we will evaluate numerically by a discrete sum where the hazard is evaluated at the set of points  $\mathbf{x}_j$  along the surveyed path:

$$H(\mathbf{u}_{ik}) = \exp(\alpha_0) \left\{ \sum_{j=1}^J \exp(\alpha_1 * ||\mathbf{u}_{ik} - \mathbf{x}_j||) \right\} \quad (15.2.2)$$

where  $\mathbf{x}_j$  is the  $j^{th}$  row of  $\mathbf{X}$  defining the survey path as a collection of line segments which can be arbitrarily dense, but should be regularly spaced. Then the probability of encounter on a given sampling occasion is

$$p_{ik} \equiv p(\mathbf{u}_{ik}) = 1 - \exp(-H(\mathbf{u}_{ik})). \quad (15.2.3)$$

Its possible that the search path could vary by sampling occasion, say  $\mathbf{X}_k$ , which can easily be accommodate in the model simply by calculating the total hazard to encounter for each distinct search path.

This is a reasonably intuitive type of encounter probability model in that the probability of encounter is large when an individual's location  $\mathbf{u}_{ik}$  is close to the line in the average sense defined by Eq. 15.2.2, and vice versa. Further, consider the case of a single survey point, i.e.,  $\mathbf{X} \equiv \mathbf{x}$ , which we might think of as a camera trap location. In this case note that Eq. (15.2.3) is equivalent to

$$\log(-\log(1 - p_{ik})) = \alpha_0 + \alpha_1 * ||\mathbf{u}_{ik} - \mathbf{x}||$$

which is to say that distance is a covariate on detection that is linear on the complementary log-log scale, which is similar to the "trap-specific" encounter probability of our Bernoulli encounter probability model (see Chapt. 5). The difference is that, here, the relevant distance is between the "trap" (i.e. the survey lines) and the individual's present location,  $\mathbf{u}_{ik}$ , which is observable. On the other hand, in the context of camera traps, the distance is that between the trap and a latent variable,  $\mathbf{s}_i$ , representing an individual's home range or activity center which is not observed.

A key assumption of this formulation of the model is that encounters at each point along the line,  $\mathbf{x}_j$ , are independent of each other point. Then, the event that an individual is encountered *at all* is the complement of the event that it is not encountered *anywhere* along the line (Hayes and Buckland, 1983). In this case, the probability of not being encountered at trap  $j$  is:  $1 - p(\mathbf{u}_{ik}, \mathbf{x}_j) = \exp(-h(\mathbf{u}_{ik}, \mathbf{x}_j))$  and so the probability that an individual is not encountered at all is  $\prod_j \exp(-h(\mathbf{u}_{ik}, \mathbf{x}_j))$ . The encounter probability is therefore the complement of this, which is precisely the expression given by Eq. 15.2.3.

Any model for encounter probability can be converted to a hazard model so that encounter probability based on total hazard can be derived. We introduced this model above:

$$\log(h(\mathbf{u}_{ik}, \mathbf{x})) = \alpha_0 + \alpha_1 * ||\mathbf{u}_{ik} - \mathbf{x}||.$$

which is usually called the Gompertz hazard function in survival analysis, and it is most often written as  $h(t) = a \exp(b * t)$  in which case  $\log(h(t)) = \log(a) + b * t$ . In the context

of survival analysis,  $t$  is “time” whereas, in SCR models, we model hazard as a function of distance. The Gaussian model has a squared-distance term:

$$\log(h(\mathbf{u}_{ik}, \mathbf{x})) = \alpha_0 + \alpha_1 * \|\mathbf{u}_{ik} - \mathbf{x}\|^2.$$

Borchers and Efford (2008) use this model:

$$h(\mathbf{u}_{ik}, \mathbf{x}) = -\log(1 - \text{expit}(\alpha_0) \exp(\alpha_1 * \|\mathbf{u}_{ik} - \mathbf{x}\|^2))$$

which produces a normal kernel model for *probability of detection* at the point level. i.e.,  $\Pr(y = 1) = 1 - \exp(-h) = h_0 \exp(\alpha_1 * \|\mathbf{u}_{ik} - \mathbf{x}\|^2)$  where  $h_0 = \text{logit}^{-1}(\alpha_0)$ . Another model is:

$$\log(h(\mathbf{u}_{ik}, \mathbf{x})) = \alpha_0 + \alpha_1 * \|\mathbf{u}_{ik} - \mathbf{x}\|$$

which is a Weibull hazard function.

### 15.2.2 Modeling movement outcomes

We have so far described the model for the encounter data in a manner that is conditional on the locations  $\mathbf{u}_{ik}$ , some of which are unobserved. Naturally, we should specify a model for these latent variables – i.e., a movement model – so that we could either do a Bayesian analysis by MCMC (Royle and Young, 2008; Royle et al., 2011a) or compute the marginal likelihood (Efford, 2011a). To develop such a model, we adopt what is now customary in SCR models – we assume that individuals are characterized by a latent variable,  $\mathbf{s}_i$ , which represents the activity center. This leads to some natural models for the movement outcomes  $\mathbf{u}_{ik}$  conditional on the activity center  $\mathbf{s}_i$ . Royle and Young (2008) used a bivariate normal model:

$$\mathbf{u}_{ik} | \mathbf{s}_i \sim \text{BVN}(\mathbf{s}_i, \sigma_{move}^2 \mathbf{I}),$$

where  $\mathbf{I}$  is the  $2 \times 2$  identity matrix. We consider alternatives below. This is a primitive model of individual movements about their home range but we believe it will be adequate in many capture-recapture studies which are often limited by sparse data.

We adopt our default assumption for the activity centers  $\mathbf{s}$ :

$$\mathbf{s}_i \sim \text{Uniform}(\mathcal{S}); \quad i = 1, 2, \dots, N.$$

The usual considerations apply in specifying the state-space  $\mathcal{S}$  – either choose a large rectangle, or prescribe a habitat mask to restrict the potential locations of  $\mathbf{s}$ .

### 15.2.3 Simulation and analysis in JAGS

Here we will simulate a sample data set that goes with the situation described in Fig. 15.1 and then analyze the data in **JAGS**. We begin by defining the state-space containing all of the grid cells in the rectangle  $[-1, 4] \times [-1, 5]$ , which contains 30  $1 \times 1$  cells. The survey line in Fig. 15.1 traverses 7 of those  $1 \times 1$  boxes. We define the total population to be 4 individuals per grid cell ( $1 \times 1$ ). To set this up in **R**, we do this:

```
> xlim <- c(-1, 4)
> ylim <- c(-1, 5)
> perbox <- 4
> N <- 30*perbox # Total of 30 1x1 quadrats
```

12598 The line in Fig. 15.1 is an irregular mesh of points obtained by an imperfect manual  
 12599 point-and-clicking operation, which mimics the way in which GPS points come to us. In  
 12600 order to apply our model we need a regular mesh of points. We can obtain a regular  
 12601 mesh of points from the irregular mesh by using some functions in the packages **rgeos** and  
 12602 **sp**, especially the function **sample.Line**, which produces a set of equally-spaced points  
 12603 along a line. The **R** commands are as follows (the complete script is given in the function  
 12604 **snakeline**):

```
12605 > library(rgeos)
12606 > library(sp)
12607 > line1 <- source("line1.R")
12608
12609 > line1 <- as.matrix(cbind(line1$value$x,line1$value$y))
12610 > points <- SpatialPoints(line1)
12611
12612 > sLine <- Line(points)
12613 > regpoints <- sample.Line(sLine,250,type="regular") # Key step!
```

12614 Next, we set a random number seed, simulate activity centers and set some model parameters  
 12615 required to simulate encounter history data. In the following commands you can see  
 12616 where the regular mesh representation of the sample line is extracted from the **regpoints**  
 12617 object which we just created:

```
12618 > set.seed(2014)
12619 > sx <- runif(N,xlim[1],xlim[2])
12620 > sy <- runif(N,ylim[1],ylim[2])
12621
12622 > sigma.move <- .35
12623 > sigma <- .4
12624 > alpha0 <- .8
12625 > alpha1 <- 1/(2*(sigma^2))
12626 > X <- regpoints@coords
12627 > J <- nrow(X)
```

12628 Next we're going to simulate data which we do in 2 steps: For each individual in the  
 12629 population and for each of  $K$  sample occasions, we simulate the location of the individual  
 12630 as a bivariate normal random variable with mean  $s_i$  and  $\sigma_{move} = 0.35$ . Next, we compute  
 12631 the encounter probability model using Eq. 15.2.3, with the bivariate normal hazard model,  
 12632 and then retain the data objects corresponding to individuals that get captured at least  
 12633 once. All of this goes according to the following commands:

```
12634 > K <- 10 ## Sample occasions = 10
12635 > U <- array(NA,dim=c(N,K,2)) ## Array to hold locations
12636 > y <- pmat <- matrix(NA,nrow=N,ncol=K) ## Initialize
12637 > for(i in 1:N){
12638 +   for(k in 1:K){
12639 +     U[i,k,] <- c(rnorm(1,sx[i],sigma.move),rnorm(1,sy[i],sigma.move))
12640 +     dvec <- sqrt( (U[i,k,1] - X[,1])^2 + (U[i,k,2] - X[,2])^2 )
```

---

```

12641 +      loghaz <- alpha0 - alpha1*dvec*dvec
12642 +      H <- sum(exp(loghaz))
12643 +      pmat[i,k] <- 1-exp(-H)
12644 +      y[i,k] <- rbinom(1,1,pmat[i,k])
12645 >    }
12646 >  }
12647 > Ux <- U[,1]
12648 > Uy <- U[,2]
12649 > Ux[y==0] <- NA
12650 > Uy[y==0] <- NA

```

12651 In the commands shown above, we define matrices,  $U_x$  and  $U_y$ , that hold the observed  
 12652 locations of individuals during each occasion. Note that, if an individual is *not* captured,  
 12653 we set the value to `NA`. We pass these partially observed objects to **JAGS** to fit the model.

12654 Finally, we do the data augmentation and we make up some starting values for the  
 12655 location coordinates that are missing. For these, we cheat a little bit (for convenience and  
 12656 hopefully to improve the efficiency of the MCMC for the simulated data sets) and use the  
 12657 actual activity center values. In practice, we might think about using the average of the  
 12658 observed locations.

```

12659 > ncap <- apply(y,1,sum)
12660 > y <- y[ncap>0,]
12661 > Ux <- Ux[ncap>0,]
12662 > Uy <- Uy[ncap>0,]

12663
12664 > M <- 200
12665 > nind <- nrow(y)
12666 > y <- rbind(y,matrix(0,nrow=(M-nrow(y)),ncol=ncol(y)))
12667 > Namat <- matrix(NA,nrow=(M-nind),ncol=ncol(y))
12668 > Ux <- rbind(Ux,Namat)
12669 > Uy <- rbind(Uy,Namat)
12670 > S <- cbind(runif(M,xlim[1],xlim[2]),runif(M,ylim[1],ylim[2]))
12671 > for(i in 1:nind){
12672 +     S[i,] <- c( mean(Ux[i,],na.rm=TRUE),mean(Uy[i,],na.rm=TRUE))
12673 > }
12674 > Ux.st <- Ux
12675 > Uy.st <- Uy
12676 > for(i in 1:M){
12677 +     Ux.st[i,!is.na(Ux[i,])]<-NA
12678 +     Uy.st[i,!is.na(Uy[i,])]<-NA
12679 +     Ux.st[i,is.na(Ux[i,])]<-S[i,1]
12680 +     Uy.st[i,is.na(Uy[i,])]<-S[i,2]
12681 + }

```

12682 The **BUGS** model specification is shown in Panel 15.1, although we neglect the stan-  
 12683 dard steps showing how to bundle the `data`, `inits`, and farm all of this stuff out to **JAGS**  
 12684 (see the help file for `snakeline` for the complete script). Simulating the data as described  
 12685 above, and fitting the model in Panel 15.1 produces the results in Table 15.1.

```

model {

  alpha0~dunif(-25,25)          # Priors distributions
  alpha1~dunif(0,25)
  lsigma~dunif(-5,5)
  sigma.move<-exp(lsigma)
  tau<-1/(sigma.move*sigma.move)
  psi~dunif(0,1)

  for(i in 1:M){ # Loop over individuals
    z[i]~dbern(psi)
    s[i,1]~dunif(xlim[1],xlim[2])   # Activity center model
    s[i,2]~dunif(ylim[1],ylim[2])
    for(k in 1:K){                 # Loop over sample occasions
      ux[i,k] ~ dnorm(s[i,1],tau)  # Movement outcome model
      uy[i,k] ~ dnorm(s[i,2],tau)
      for(j in 1:J){ # Loop over each point defining line segments
        d[i,k,j]<- pow(pow(ux[i,k]-X[j,1],2) + pow(uy[i,k]-X[j,2],2),0.5)
        h[i,k,j]<-exp(alpha0-alpha1*d[i,k,j]*d[i,k,j])
      }
      H[i,k]<-sum(h[i,k,1:J])       # Total hazard H
      p[i,k]<- z[i]*(1-exp(-H[i,k]))
      y[i,k] ~ dbern(p[i,k])
    }
  }
  # Population size is a derived quantity
  N<-sum(z[])
}

```

Panel 15.1: **BUGS** model specification for the fixed search path model, based on that from Royle et al. (2011a). See the help file `?snakeline` for the **R** code to simulate data and fit this model.

**Table 15.1.** Posterior summary statistics for the simulated fixed search path data. These are based on 3 chains, and a total of 9000 posterior samples. The data generating parameter values were  $N = 100$ ,  $\sigma_{move} = 0.35$ ,  $\sigma = 0.4$ , and  $\alpha_0 = 0.8$ . The parameter  $\alpha_1 = 1/(2\sigma^2)$ .

Parameter	Mean	SD	2.5%	50%	97.5%	Rhat
$N$	117.626	5.675	107.000	117.000	129.000	1.015
$\alpha_0$	1.305	0.494	0.425	1.280	2.387	1.009
$\alpha_1$	3.806	0.423	3.050	3.777	4.733	1.008
$\sigma_{move}$	0.347	0.008	0.332	0.347	0.364	1.023
$\sigma$	0.364	0.020	0.325	0.364	0.405	1.008
$\psi$	0.587	0.044	0.501	0.588	0.673	1.006

#### 12686 15.2.4 Hard plot boundaries

12687 The previous development assumed that locations of individuals can be observed anywhere  
 12688 in the state-space, determined only by the encounter probability model as a function of  
 12689 distance from the search path. However, in many situations, we might delineate a plot  
 12690 which restricts where individuals might be observed (as in the situation considered by  
 12691 Royle and Young (2008)). For such cases we truncate the encounter probability function  
 12692 at the plot boundary, according to:

$$p(\mathbf{u}_{ik}) = (1 - \exp(-H(\mathbf{u}_{ik})))I(\mathbf{u}_{ik} \in \mathcal{X}) \quad (15.2.4)$$

12693 where  $\mathcal{X}$  is the surveyed polygon and the indicator function  $I(\mathbf{u}_{ik} \in \mathcal{X}) = 1$  if  $\mathbf{u}_{ik} \in \mathcal{X}$   
 12694 and 0 otherwise. That is, the probability of encounter is identically 0 if an individual  
 12695 is located *outside* the plot at sample period  $t$ . We demonstrated how to do this in the  
 12696 **BUGS** language below for a model of uniform search intensity (area-search model).

#### 12697 15.2.5 Analysis of other protocols

12698 In the situation elaborated on above (what we called “Protocol 1a”), the sample path is  
 12699 used to locate individuals and, whether or not an individual is encountered, is a function  
 12700 of the total hazard to encounter along the whole line. We think there are a number of  
 12701 variations of this basic design that might arise in practice. A slight variation (what we  
 12702 called “Protocol 1b”) is based on recording location of individuals and also the location  
 12703 on the transect where we observed the individual. The probability of encounter is the  
 12704 probability of encounter prior to the point on the line where the detection takes place  
 12705 (Skaug and Schweder, 1999). This is exactly a distance-sampling observation model, but  
 12706 with an additional hierarchical structure that describes the individual locations about their  
 12707 activity centers. There are no additional novel considerations in analysis of this situation  
 12708 compared to Protocol 1a, and so we have not given it explicit consideration here. Similarly,  
 12709 “Protocol 1c” is a slight variation of this – instead of recording the point on the line where  
 12710 the individual was first detected, we use, instead, the point on the line that has the shortest  
 12711 perpendicular distance. This is a classical distance sampling observation model, and it  
 12712 represents an intentional misspecification of the model but it seems that the effect of this  
 12713 is relatively minor, or, otherwise, we imagine people wouldn’t do it.

### 15.3 UNSTRUCTURED SPATIAL SURVEYS

12714 A common situation in practice is that in which sampling produces a survey path, but  
 12715 the path was not laid out *a priori* but, rather evolves opportunistically during the course  
 12716 of sampling, a situation we'll call an unstructured spatial survey (Thompson et al., 2012;  
 12717 Russell et al., 2012). We imagine that the survey path evolves in response to information  
 12718 about animal presence, which could be both the number of unique individuals or the  
 12719 amount of sign in the local search area. The motivating problem has to do with area  
 12720 searches using dog teams, in which the dogs usually wander around hunting scat, and their  
 12721 search path is based on how they perceive the environment and what they're smelling.  
 12722 This violates the main assumptions that the line is placed *a priori*, independent of density  
 12723 and unrelated to detectability.

12724 The analysis framework implemented by Thompson et al. (2012) and Russell et al.  
 12725 (2012) is based on a heuristic justification wherein the sampling of space is imagined  
 12726 to have been grid-structured, with grid cells that are large enough so that dogs are not  
 12727 influenced by scat or sign beyond the specific cell being searched. Then, we assume the dog  
 12728 applies a consistent search strategy to each cell so that that resulting cell-level detections  
 12729 can be regarded as independent Bernoulli trials with probability  $p_{ij}$  depending on the  
 12730 distance  $\|\mathbf{x}_j - \mathbf{s}_i\|$  between the grid cell with center  $\mathbf{x}_j$ , and individual with activity  
 12731 center  $\mathbf{s}_i$  and the amount of search effort (or length of the search route) within a cell.  
 12732 In other words, we use an ordinary SCR type of model but treating the center point of  
 12733 each cell as an effective "trap". The deficiency with this approach is that some of the  
 12734 "sub-grid" resolution information about movement is lost, so we probably lose precision  
 12735 about any parameters of the movement model when the cells are large relative to a typical  
 12736 home range size. We discuss a couple of examples below.

#### 12737 15.3.1 Mountain lions in Montana

12738 Russell et al. (2012) analyzed mountain lion (*Puma concolor*) encounter history data to  
 12739 assess the status of mountain lions in the Blackfoot Mountains of Montana. The data  
 12740 collection was based on opportunistic searching by hunters with dogs, who tree the lion  
 12741 (Fig. 15.2). Tissue is extracted with a biopsy dart and analyzed in the lab for individual  
 12742 identity. They used  $5 \text{ km} \times 5 \text{ km}$  grid cells for binning the encounters, and the length  
 12743 of the search path in each grid cell as a covariate of effort ( $C_j$ ) that each grid cell was  
 12744 searched. The model is the Gaussian hazard model with baseline encounter probability  
 12745 that depended on sex and effort in each grid cell, on the log scale:

$$\log(\lambda_{0,ij}) = \alpha_0 + \alpha_2 \log(C_j) + \alpha_3 \text{Sex}_i$$

12746 Note for grid cells that were not searched,  $C_j = 0$  and, for those, the constraint  $\lambda_{0,ij} = 0$   
 12747 was imposed so that the probability of encounter was identically 0.

12748 One problem encountered by Russell et al. (2012) in their analysis is the possibility  
 12749 of dependence in encounters because of group structure in the data (usually, juveniles in  
 12750 association with their mother). In this situation, in addition to dependence of encounter,  
 12751 multiple individuals have effectively the same activity center, thus violating a number of  
 12752 assumptions related to the ordinary SCR model. To resolve this problem, the authors  
 12753 made some assumptions about group association and fitted models where group served as  
 12754 the functional individual.



**Figure 15.2.** Mountain lion. Run! Photo credit: Bob Wiesner.

12755 **15.3.2 Sierra National Forest Fisher Study**

12756 Here we consider a more detailed example and provide the data and **R** script for this  
12757 analysis. The data come from an analysis of individual encounter histories of the fisher  
12758 (*Martes pennanti*) by Thompson et al. (2012). The survey area was divided into 15 ap-  
12759 proximately 1,400 ha hexagons (Fig. 15.3), which is roughly the size of a female fisher's  
12760 home range, and each hexagon was surveyed 3 times by sniffer dog teams searching space  
12761 for scat. The dogs were given considerably latitude to determine their route. Thus, the  
12762 search path is not laid out a priori but rather evolves opportunistically, based on what  
12763 the dog senses at a local scale. The authors divided the region into 1 km grid cells (also  
12764 shown in Fig. 15.3).

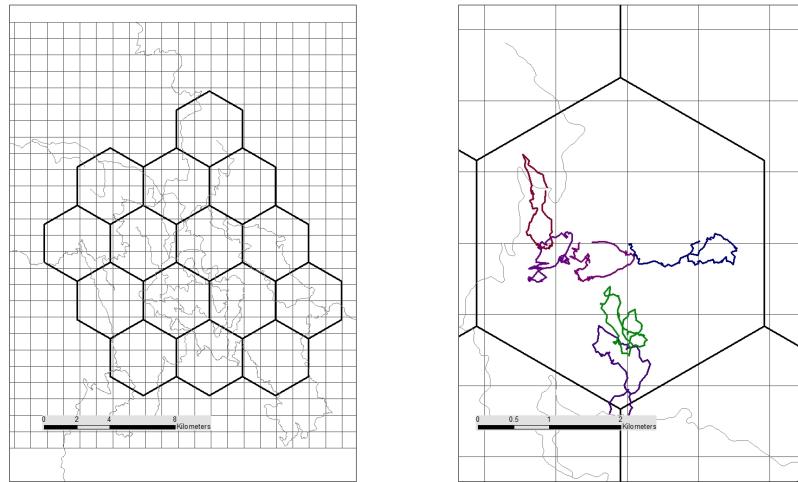
12765 We provide the data from this study in the **scrbook** package, and it can be loaded with  
12766 the command **data(fisher)**. The **R** script **SCRfisher** produces the posterior summary  
12767 statistics shown in Table 15.2. One thing is relatively poor mixing of the Markov chains  
12768 here due to sparse data and a fairly long run is probably necessary.

#### 15.4 DESIGN 2: UNIFORM SEARCH INTENSITY

12769 A special case of a search-encounter type of model arises when it is possible to subject  
12770 a quadrat (or quadrats) to a uniform search intensity. This could be interpreted as an  
12771 exhaustive search, or perhaps just a thorough systematic search of the available habitat.  
12772 The example considered by Royle and Young (2008) involved searching a 9 ha plot for  
12773 horned lizards (Fig. 15.4) by a crew of several people. It was felt in that case that complete

**Table 15.2.** Posterior summary statistics for the fisher study data, based on 30000 posterior samples. Here  $\lambda_0 = \exp(\alpha_0)$ . This example exhibits relatively poor mixing due to sparse data, and the Rhat statistic should be reduced by obtaining a larger posterior sample.

Parameter	Mean	SD	2.5%	50%	97.5%	Rhat
$N$	315.889	230.041	12.000	280.000	738.775	1.133
$\sigma$	4.745	2.909	0.163	4.650	9.704	1.020
$\lambda_0$	0.003	0.033	0.000	0.000	0.016	1.097
$\alpha_1$	0.188	0.170	0.005	0.138	0.641	1.002
$\psi$	0.413	0.300	0.016	0.366	0.964	1.131



**Figure 15.3.** Fisher study area showing the gridding system (left panel). The larger hexagons are approximately the size of a typical female home range. The 1 km grid cells define the SCR model grid, where the center point of each one served as a trap. The right panel shows the GPS trackline of the dog team through one of the grid cells. The total length of the trackline was used as a covariate on encounter probability. Credit: Craig Thompson, U.S. Forest Service

12774 and systematic (i.e., uniform) coverage of the plot was achieved. In general, however, we  
 12775 think you could have a random sample of the plot and approximate that as a uniform  
 12776 coverage – this is kind of a design-based argument justifying the uniform search intensity  
 12777 model (we haven't simulated this situation, but it would be worth investigating).



**Figure 15.4.** A flat-tailed horned lizard showing its typical cryptic appearance in its native environment. Detection of flat-tailed horned lizards is difficult because they do not run when approached. Instead they shuffle under the sand or press down and remain motionless as shown in the picture. The horns are employed only as a last resort if the camouflage fails. *Photo credit: Kevin and April Young*

12778 It is clear that this uniform search intensity model is a special case of the fixed search  
 12779 path model in the sense that the probability of encounter of an individual is a constant  
 1280  $p_0$  if the individual is located in the polygon  $\mathcal{X}$  during sample occasion  $k$ , i.e.,

$$p(\mathbf{u}_{ik}) = p_0 I(\mathbf{u}_{ik} \in \mathcal{X})$$

12781 which resembles Eq. 15.2.4 except replacing the encounter probability function with con-  
 12782 stant  $p_0$ .

12783 Subsequently, we give a simple analysis using simulated data and simple movement  
 12784 models for  $\mathbf{u}$ , including a bivariate normal model and a random walk. For further examples  
 12785 and analyses, we refer you to Royle and Dorazio (2008), who reanalyzed the lizard data  
 12786 from Royle and Young (2008), and Efford (2011b) and Marques et al. (2011).

#### 12787 **15.4.1 Alternative movement models**

12788 As with the general fixed search path model (“Design 1”), we require a model to describe  
 12789 the movement outcomes  $\mathbf{u}_{ik}$ . In the analysis of Royle and Young (2008), a simple bivariate

12790 Gaussian movement model was used, in which

$$\mathbf{u}_{ik} | \mathbf{s}_i \sim \text{Normal}(\mathbf{s}_i, \sigma_{move}^2 \mathbf{I}),$$

12791 However, clearly more general versions of the model can be developed. For example, imagine  
 12792 a situation where the successive surveys of a bounded sample polygon are relatively  
 12793 close together in time so that successive locations of individuals are not well-approximated  
 12794 by the Gaussian movement model, which implies independence of locations. Naturally we  
 12795 might consider using an auto-regressive or random-walk type of model in which the suc-  
 12796 cessive coordinate locations of individual  $i$  behave as follows:

$$\begin{aligned} u_{1,i,k} | u_{1,i,k-1} &\sim \text{Normal}(u_{1,i,k-1}, \sigma_{move}^2) \\ u_{2,i,k} | u_{2,i,k-1} &\sim \text{Normal}(u_{2,i,k-1}, \sigma_{move}^2) \end{aligned}$$

12797 here we use the notation  $u_1$  and  $u_2$  for the easting and northing coordinates, respectively.  
 12798 (and, for clarity, we are using commas in the sub-scripting here when we have to refer to  
 12799 time-lags). In addition, we require that the initial locations have a distribution and, for  
 12800 that, we might begin with a simple model such as the uniformity model:

$$\mathbf{u}_{i,1} \sim \text{Uniform}(\mathcal{S})$$

12801 which effectively takes the place of the model for  $\mathbf{s}_i$  that we typically use. Under this  
 12802 model, individuals don't have an activity center but, rather, they drift through space  
 12803 more-or-less randomly based just on their previous location. See Ovaskainen (2004) and  
 12804 Ovaskainen et al. (2008) for development and applications of similar movement models  
 12805 in the context of capture-recapture data, and also our discussion of a similar model that  
 12806 might arise in acoustic surveys (Sec. 9.4). We could allow for dependent movements  
 12807 about a central location  $\mathbf{s}_i$  using a bivariate auto-regression or similar type of model with  
 12808 parameter  $\rho$ , e.g.,

$$\mathbf{u}_{i,k} | \mathbf{s}_i \sim \text{BVN}(\rho * (\mathbf{u}_{i,k-1} - \mathbf{s}_i), \sigma_{move}^2 \mathbf{I}).$$

12809 We don't have any direct experience fitting these movement models to real capture-  
 12810 recapture data, but we imagine they should prove effective in applications that yield large  
 12811 sample sizes of individuals and recaptures.

### 12812 15.4.2 Simulating and fitting uniform search models

12813 The **R** script `uniform_search`, in the `scrbook` package, we provide a script for simulating  
 12814 and fitting search-encounter data using the iid Gaussian model and also the random walk  
 12815 model. The **BUGS** model specification is shown in Panel 15.2 for the random walk  
 12816 situation. We encourage you to adapt this model and the simulation code for the auto-  
 12817 regression movement model. To fit this model to data, we set up the run with **JAGS** using  
 12818 the standard commands. We did not specify starting values for the missing coordinate  
 12819 locations although we imagine that **JAGS** should perform better if we provide decent  
 12820 starting values, e.g., the last observed location or some other reasonable location. We  
 12821 imagine that resource selection could be parameterized in this movement model as well,  
 12822 perhaps using similar ideas to those described in Chapt. 13.

12823    The following script simulates a population of N individuals and their locations at  
 12824    each of 4 times to see if they are in a square [3,13] or not. This simulates a random walk  
 12825    thing so we imagine that the sampling occasions are close together in time. The initial  
 12826    state is assumed to be uniformly distributed on the state-space which, in this case, is the  
 12827    square  $[0, 16] \times [0, 16]$ . We store the movement outcomes here in a 3-d array U, instead of  
 12828    in two separate 2-d arrays (one for each coordinate) as we did above. The R commands  
 12829    are as follows:

```

12830 > N <- 100
12831 > nocc <- 4
12832 > Sx <- Sy <- matrix(NA,nrow=N,ncol=nocc)
12833 > sigma.move <- .25
12834
12835 # Simulate initial coordinates on the square:
12836 > Sx[,1] <- runif(N,0,16)
12837 > Sy[,1] <- runif(N,0,16)
12838
12839 > for(t in 2:nyear){
12840 +   Sx[,t] <- rnorm(N,Sx[,t-1],sigma.move)
12841 +   Sy[,t] <- rnorm(N,Sy[,t-1],sigma.move)
12842 + }
12843
12844 # Now we generate encounter histories on a search rectangle
12845 #   with sides [3,13]:
12846 > Y <- matrix(0,nrow=N,ncol=nyear)
12847 > for(i in 1:N){
12848 +   for(t in 1:nyear){
12849 +     # IF individual is in the sample unit we can capture it:
12850 +     if( Sx[i,t] > 3 & Sx[i,t]< 13 & Sy[i,t]>3 & Sy[i,t]<13 )
12851 +       Y[i,t] <- rbinom(1,1,.5)
12852 +   }
12853 + }
12854
12855 # Subset data. If an individual is never captured, cannot have him in our data set
12856 > cap<- apply(Y,1,sum) > 0
12857 > Y <- Y[,cap,]
12858 > Sx <- Sx[,cap,]
12859 > Sy <- Sy[,cap,]
12860
12861 > Sx[Y==0] <- NA
12862 > Sy[Y==0] <- NA
12863
12864 ## Data augmentation:
12865 > M <- 200
12866 > Y <- rbind(Y,matrix(0,nrow=(M-nrow(Y)),ncol=nyear))
12867 > Sx <- rbind(Sx,matrix(NA,nrow=(M-nrow(Sx)),ncol=nyear))
12868 > Sy <- rbind(Sy,matrix(NA,nrow=(M-nrow(Sy)),ncol=nyear))
```

---

```

12869
12870 # Make 3-d array of coordinates "U"
12871 > U <- array(NA,dim=c(M,nyear,2))
12872 > U[,,1] <- Sx
12873 > U[,,2] <- Sy

```

---

```

model{
psi ~ dunif(0,1)                                # Prior distributions
tau ~ dgamma(.1,.1)
p0 ~ dunif(0,1)
sigma.move <- sqrt(1/tau)

for (i in 1:M){
  z[i] ~ dbern(psi)
  U[i,1,1] ~ dunif(0,16)                         # Initial location
  U[i,1,2] ~ dunif(0,16)

  for (k in 2:n.occasions){
    U[i,k,1] ~ dnorm(U[i,k-1,1], tau)
    U[i,k,2] ~ dnorm(U[i,k-1,2], tau)
  }
  for(k in 1:n.occasions){
    # Test whether the actual location is in- or outside the
    # survey area. Needs to be done for each grid cell
    inside[i,k] <- step(U[i,k,1]-3) * step(13-U[i,k,1]) *
      step(U[i,k,2]-3) * step(13-U[i,k,2])
    Y[i,k] ~ dbern(mu[i,k])
    mu[i,k] <- p0 * inside[i,k] * z[i]
  }
}
N <- sum(z[])                                     # Population size, derived
}

```

---

Panel 15.2: **BUGS** model specification for the uniform search intensity model similar to Royle and Young (2008) but with a random walk movement model. help file `?uniform_search` in the **R** package `scrbook`.

#### 15.4.3 Movement and Dispersal in Open Populations

In Chapt. 16 we discuss many aspects of modeling open populations, including some aspects of modeling movement and dispersal and the relevance of SCR models to these

12877 problems. However, given the introduction of the uniform search model above, this is  
 12878 clearly relevant to modeling movement and dispersal in open populations. In particular,  
 12879 the model described in Panel 15.2 could easily be adapted to an open population by  
 12880 conditioning on the first, and introducing a latent “alive state” with survival parameter  
 12881  $\phi_t$ . This would be a spatial version of the standard Cormack-Jolly-Seber model (Chapt.  
 12882 16.3)<sup>1</sup>.

## 15.5 PARTIAL INFORMATION DESIGNS

12883 The prototype search-encounter (Design 1) and uniform search (Design 2) cases are ideal  
 12884 in the sense that they produce both precise locations of individuals and also a precise  
 12885 characterization of the manner in which individuals are encountered by sampling space.  
 12886 We have seen a number of studies that, in an ideal world, would have generated data  
 12887 consistent with one of these situations but, for some practical reason or other reason,  
 12888 partial or no spatial information about the search area or the locations of individuals was  
 12889 collected (or retained), and so the models described above could not be used. We imagine  
 12890 (indeed, have encountered) at least 3 distinct situations:

- 12891 (a) The search path is not recorded, but locations of individuals are recorded
- 12892 (b) The search path is recorded, but locations of individuals are not.
- 12893 (c) The search path is not recorded, and the locations are not recorded, just raw sum-  
 12894       maries for prescribed areas or polygons.

12895 For analysis of these search-encounter designs with partial information, we see a num-  
 12896 ber of options of varying levels of formality, depending on the situation (and these are  
 12897 largely untested). For (a) You could always assume uniform search intensity, which might  
 12898 be reasonable if the plots were randomly searched. Otherwise, its validity would depend  
 12899 on the precise manner in which the search activity occurred. For (b) or (c), we could  
 12900 adopt the approach we took in the fisher analysis above, and map the locations to the  
 12901 center of each plot, thinking of the plot as an effective trap, and using the search path  
 12902 length as a covariate. A 4th case with even less information is that in which we don’t  
 12903 record individual identity at all. Instead, we just have total count frequencies in each plot.  
 12904 This model is precisely the one considered by (Chandler and Royle, 2013) and this is the  
 12905 focus of Chapt. 18.

## 15.6 SUMMARY AND OUTLOOK

12906 The generation of spatial encounter history data in ecological studies is widespread. While  
 12907 such data have historically been obtained mostly by the use of arrays of fixed traps (catch  
 12908 traps, camera traps, etc..), in this chapter we showed that SCR models are equally rel-  
 12909 evant to a large class of “search-encounter” problems which are based on organized or  
 12910 opportunistic searches of spatial areas. Standard examples include “area searches” in bird  
 12911 population studies, use of detector dogs to obtain scat samples, from which DNA can  
 12912 be obtained to determine individual identity, or sampling along a fixed search path (or  
 12913 transects) by observers noting the locations of detected individuals (this is common in

<sup>1</sup>Some work related to this is currently being carried out by our colleagues Torbjørn Ergon and Michael Schaub.

12914 sampling for reptiles and amphibians). The latter situation closely resembles distance  
12915 sampling but, with repeated observations of the same individual (on multiple occasions),  
12916 it has a distinct capture-recapture element to it. In a sense, the fixed search path models  
12917 are hybrid SCR-DS models.

12918 Many models for search-encounter data have three elements in common. They contain:  
12919 (1) a model for encounter conditional on locations of individuals; (2) a model that describes  
12920 how these observable animal locations are distributed in space about their activity centers;  
12921 and (3) a model for the distribution of activity centers. We interpret the 2nd model  
12922 component as an explicit movement model, and the existence of this component is distinct  
12923 from most of the other models considered in this book. One of the key conceptual points  
12924 is that, with these search-encounter types of designs, the locations of observations are *not*  
12925 biased by the locations of traps but, rather, locations of individuals can occur anywhere  
12926 within search plots or quadrats, or in the vicinity of a transect or search path. Because we  
12927 can obtain direct observations of location – outcomes of movement – for individuals, it is  
12928 possible to resolve explicit models of movement from search-encounter data. We considered  
12929 the simple case of the independent bivariate normal movement model, and also a random  
12930 walk type model, which can easily be fitted in the **BUGS** engines. We imagine much  
12931 more general movement models can be fitted, although we have had limited opportunities  
12932 to pursue this and in most practical capture-recapture studies, we will probably be limited  
12933 by sparse data in the complexity of the movement models that could be considered.

12934 Search-encounter sampling is fairly common, although we think that many people don't  
12935 realize that it can produce encounter history data that is amenable to the development  
12936 of formal models for density, movement and space usage. We believe that these protocols  
12937 will become more appealing as methods for formal analysis of the resulting encounter  
12938 history data become more widely known. At the same time, search-encounter models will  
12939 increase in relevance in future studies of animal populations because so many new methods  
12940 of obtaining encounter history data can be based on DNA extracted from animal tissue  
12941 or scat, which is easy to obtain by searching space opportunistically. In addition, as the  
12942 cost of obtaining individual identity from scat or tissue decreases, its widespread collection  
12943 and use in capture-recapture models can only increase.



12944  
12945

12946

# 16

---

## OPEN POPULATION MODELS

### 16.1 INTRODUCTION

In the previous chapters we focused mostly on closed population models for estimating density and for inference about spatial variation in density and space usage. However, a thorough understanding of population dynamics requires information about both spatial *and* temporal variation in population density and demographic parameters. In this chapter, we discuss modeling the processes governing spatial and temporal population dynamics, namely survival, recruitment, and movement over larger temporal scales (e.g., migration, dispersal, etc...). The ability to estimate these parameters is critical to both basic and applied ecological research (Knape, 2012). For example, testing hypotheses about life history trade-offs requires accurate estimates of both survival and fecundity (Caswell, 1989; Nichols et al., 1994). Inference about density-dependent population regulation, which has fascinated theoretical ecologists for well over a century, is likewise best accomplished by directly studying the factors affecting survival and fecundity, rather than the more common approach of modeling time series data (Nichols et al., 2000b). A mechanistic understanding of population changes, which is essential for basic ecological and conservation related questions, requires useful models of vital rates. Furthermore, if we know how environmental variables affect demographic parameters, we can make predictions about population changes under different future scenarios. We can also assess the sensitivity of parameters such as population growth rate to variation in survival or fecundity. Although matrix population models are often used for these purposes (Caswell, 1989; Sæther and Bakke, 2000), the same objectives can be accomplished by computing posterior predictive distributions of projected population sizes as part of the MCMC algorithm.

The modeling framework we will develop in this chapter is based on a formulation of the classical Cormack-Jolly-Seber (CJS) and Jolly-Seber (JS) type models (Cormack, 1964; Jolly, 1965; Seber, 1965) that are amenable to modeling individual effects, including individual covariates. There is a long history of use of these models in fisheries, wildlife, and ecology studies (Pollock et al., 1990; Lebreton et al., 1992; Pradel, 1996; Williams et al., 2002; Schwarz and Arnason, 2005; Gimenez et al., 2007). Additionally, there have

12975 been many modifications and developments of the CJS and JS models including dealing  
 12976 with individuals that do not have a well defined home range but instead are moving  
 12977 through the sampled area (transients), dealing with more than one site or state (multi-state  
 12978 models, where states maybe geographic units, reproductive stage, etc.), and addressing  
 12979 individual movement through spatially implicit models.

12980 For the first time, these models can fully integrate the movement of individuals in the  
 12981 vicinity of the trap array with their encounter histories to simultaneously estimate density,  
 12982 survival, and recruitment in a spatial model. For many species, such as those that are rare  
 12983 or not often observed by researchers, this allows inferences to be made about survival and  
 12984 recruitment without having to physically capture individuals. Additionally, another reason  
 12985 for extending SCR models to open populations arises purely from a sampling perspective.  
 12986 Longer time periods are often needed to sample rare or elusive species to ensure that  
 12987 enough captures and recaptures are produced. This extended time frame can quickly lead  
 12988 to violations in the assumption of population closure (see also Chapt. 10). For example,  
 12989 the European wildcat study that was mentioned in Chapt. 7 (see Kéry et al. (2011) for  
 12990 details) was conducted over a year-long period. While the researchers in that study used  
 12991 a closed population model, they did model variation in detection as a function of time  
 12992 to account for seasonal variation in behavior. Another approach would have been to use  
 12993 an open population model to account for possible changes in the population over time  
 12994 (however, the spatial capture recapture open models had not been developed at the time  
 12995 of the wildcat study, so we'll forgive the authors for not having used this more appropriate  
 12996 model).

12997 In this chapter, we present the traditional JS model and the spatial version, demon-  
 12998 strating both with an example of mist-netting of ovenbirds, which was also analyzed in  
 12999 Chapt. 9. Then we review the traditional CJS, multi-state CJS, and then describe the  
 13000 spatial model. In this section, we will use a an example of American shad. Finally, we  
 13001 end by discussing some of the new approaches to dynamics of activity centers including  
 13002 correlated movement and dispersal.

### 13003 **16.1.1 Brief overview of population dynamics**

13004 The most basic formulation of models for population growth stems from an idea originally  
 13005 used in accounting, the balance sheet (see Comroy and Carroll (2009, Chapt. 3) for a more  
 13006 complete description). To gain a mechanistic understanding of population dynamics, it  
 13007 is important to understand four fundamental processes that drive population size: births  
 13008 and immigrants (i.e., population “credits”) and deaths and emigrants (i.e., population  
 13009 “debits”). The population at time  $t + 1$  is a function of these four components:

$$N(t + 1) = N(t) + B(t) + I(t) - D(t) - E(t)$$

13010 where  $N(t)$  is the population size at time  $t$ ,  $B(t)$  and  $I(t)$  are the credits (additions)  
 13011 from births and immigrants at time  $t$ , and  $D(t)$  and  $E(t)$  are the debits (losses) due to  
 13012 deaths and emigration. This balance equation model is known as the “BIDE model”. A  
 13013 simple population growth model under density independence, assuming no immigration  
 13014 or emigration, can be derived as:

$$N(t + 1) = N(t) + N(t)r(t)$$

13015 where  $r(t) = b(t) - d(t)$ . Here,  $b(t)$  and  $d(t)$  are the per-capita birth and death rates  
13016 and thus  $r(t)$  is the per-capita growth rate. Models which are based only on the intrinsic  
13017 population growth rate, ‘ $r$ ’, however, do not retain much information about the underlying  
13018 drivers of the population dynamics. Density-dependent, age structured, stochastic effects  
13019 on growth, spatially structured, and competition models (e.g., Lotka-Volterra) all are  
13020 derivations of the basic BIDE model.

13021 In closed population models, we focus on estimating the population size,  $N$ , but in open  
13022 population models we are interested in the dynamics that arise between years or seasons  
13023 and thus we focus not only on  $N(t)$  but on the processes that drive population changes.  
13024 By taking the basic parameters in the BIDE model and reconceptualizing them, they can  
13025 then be related to the commonly used parameters in JS and CJS models, described in more  
13026 detail throughout this chapter. In the absence of movement, deaths (D) can be estimated  
13027 in the CJS model and both D and B (births) can be estimated in the JS model. However,  
13028 in considering movement, it becomes difficult to distinguish births from immigrants and  
13029 deaths from emigrants because data are usually only collected in one area and when the  
13030 animal leave that area we cannot determine it’s fate.

13031 For example, survival ( $\phi(t)$ ) is defined as the probability of an individual surviving from  
13032 time  $t$  to  $t + 1$ , and often this is called ‘*apparent* survival’ because deaths and emigration  
13033 cannot be separated. Mortality, the probability of dying from time  $t$  to  $t + 1$ , is  $1 - \phi(t)$ .  
13034 Recruitment ( $\gamma$ ) is the probability of a new individual entering the population between  $t$   
13035 to  $t + 1$ , which includes both those born into the population and immigrants. This inability  
13036 to distinguish between the different forms of losses and gains does not allow researchers to  
13037 test specific hypotheses about population dynamics. To address this, Nichols and Pollock  
13038 (1990) applied the robust design to a two age class situation in order to separate estimates  
13039 of recruitment into immigration and *in situ* reproduction. While models that focus on the  
13040 population growth rate tend to lose important information on population dynamics, more  
13041 recent work has been done to estimate the contributions of survival and recruitment to  
13042 the per capita growth rate, ‘ $r$ ’, using capture-recapture data and a reverse-time modeling  
13043 approach (Pradel, 1996; Nichols et al., 2000a). All of these model improvements have  
13044 provided invaluable information in the study of population dynamics, but none explicitly  
13045 incorporate animal movement.

### 13046 16.1.2 Animal movement related to population demography

13047 One issue that arises frequently in traditional open population models is that movement  
13048 can make it difficult to distinguish survival from emigration. For example, we know that  
13049 movement of transients and temporary emigration will affect the estimates of survival,  
13050 causing us to refer to estimates as “apparent survival” (Lebreton et al., 1992). This is  
13051 because an animal that appears in the population for a short period of time and then  
13052 leaves is going to appear as though it has died. Due to this problem, there has been  
13053 a significant amount of work developing models to deal with temporary emigration and  
13054 transients in both closed and open capture-recapture models (Kendall et al., 1997; Pradel  
13055 et al., 1997; Hines et al., 2003; Clavel et al., 2008; Gilroy et al., 2012; Chandler et al.,  
13056 2011). Because movement is modeled directly within the SCR framework, we can better  
13057 understand the impact of animals moving onto and off of the trap array and hence we can  
13058 improve our estimates of survival by combining the traditional CJS and JS models with

13059 the SCR model.

13060 While demographic parameters such as survival rates, population growth, etc. are  
13061 influenced by density (Fowler, 1981; Murdoch, 1994; Saether et al., 2002), it is also likely  
13062 that movement of individuals can influence these parameters. It is generally accepted that  
13063 population structure (i.e., age, stage, or size distribution) can affect both population size  
13064 and growth over time (Caswell and Werner, 1978). We also know that how animals dis-  
13065 tribute themselves in space can directly influence the age or stage structure of a population  
13066 – this can be behavioral, habitat related, or some combination of factors. For example,  
13067 if habitat is limited, some younger members of the population might have trouble finding  
13068 and/or defending a territory. Ultimately, this may lower survival for a certain age class  
13069 in the population directly impacting the population structure.

13070 Dispersal can also affect population structure. In population ecology, dispersal can be  
13071 related with access to reproduction, population regulation, habitat quality, as well as the  
13072 linking of local populations in metapopulation ecology (Clobert et al., 2001; Ovaskainen,  
13073 2004; Ovaskainen et al., 2008). It is known that dispersal may be influenced by density-  
13074 dependence (Matthysen, 2005); for example, competition may cause individuals to be  
13075 more likely to emigrate from an area, or individuals may leave an area in search of a mate  
13076 or partner. We discuss modeling dispersal with capture recapture data a bit further in  
13077 Sec. 16.4 at the end of the chapter.

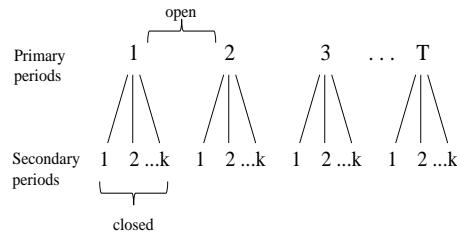
## 16.2 JOLLY-SEBER MODELS

### 13078 16.2.1 Traditional Jolly-Seber models

13079 The JS model was developed as a way to estimate not only detection and abundance, but  
13080 survival and recruitment (new individuals coming into the population) based on capture-  
13081 recapture data (Jolly, 1965; Seber, 1965). There are a number of ways that researchers  
13082 have formulated the JS model (Cooch and White, 2006) and while all are slightly different,  
13083 the resulting estimates of abundance and the driving parameters such as survival and  
13084 some form of recruitment should be equivalent. Commonly used formulations are the  
13085 Link-Barker (Link and Barker, 2005), Pradel-recruitment (Pradel, 1996), Burnham JS  
13086 (Burnham, 1997), and the super-population formulation of Schwarz and Arnason (1996).  
13087 In all of these models, the parameter of interest is recruitment, or how new individuals  
13088 arrive into the population. Therefore one of the main differences between the various  
13089 models is how new entrants into the population are parameterized.

13090 Traditionally, sampling for the JS model included only one data collection event per  
13091 primary occasion and this allowed for the estimation of survival and recruitment. However,  
13092 without repeated visits within a primary occasion, there is not enough data to allow for  
13093 variation in detection and this lead to potentially inaccurate estimates of population size.  
13094 This lead Pollock (1982) to devise the robust design in order to allow for heterogeneity  
13095 in capture probability (by sex, age, social status, etc.) and trap response under the JS  
13096 model. We present the robust design approach as it is more flexible and generalizing to  
13097 the spatial version of the JS model will be much simpler. The basic idea is that there are  
13098 primary occasions (e.g., years, seasons) and we allow the population to be “open” between  
13099 the primary occasions. This means that individuals can enter and leave the population  
13100 (i.e., births, deaths, immigration, emigration can occur) between the primary occasions  
13101 and within a primary occasion, the population is assumed to be closed to these processes.

13102 The standard JS model does not allow for variation in detection probability between in-  
 13103 dividuals or within a primary occasion because only one sample is collected per primary  
 13104 period. However, when multiple samples are taken within a primary occasion (we call  
 13105 these “secondary occasions”), then variation in detection probability can be modeled and  
 13106 thus our estimates of  $N$  can be improved. To that extent, we can envision the data as  
 13107 arising from repeated sampling over seasons or years (or *primary* periods) within which  
 13108 one or more samples (e.g., trap nights) might be taken (*secondary* periods). Fig. 16.1  
 13109 demonstrates the sampling process graphically. Comparing this with all of our previous  
 13110 work, the sample occasions (e.g., trap nights, weeks, etc...) described in the closed popu-  
 13111 lation chapters are called *secondary* sampling occasions in the context of open population  
 13112 models.



**Figure 16.1.** Schematic of the robust design with  $T$  primary sampling periods and  $K$  secondary periods. The populations are considered open between primary periods and closed within each.

13113 We can easily formulate a non-spatial JS model using the robust design. We define  $y_{ikt}$   
 13114 as the encounter history for individual  $i$  at secondary occasion  $k$  during primary occasion  
 13115  $t$ . If we have a Bernoulli encounter process then we can describe the observation model,  
 13116 specified conditional on the “alive state”,  $z(i, t)$ , for individual  $i$  at primary time  $t$ , as:

$$y_{ikt}|z(i, t) \sim \text{Bernoulli}(p_t z(i, t)).$$

13117 (Note: throughout this chapter we will focus on changes in the alive state, so we will  
 13118 index  $z$  using parentheses in order to make the notation easier to read, where  $z(i, t)$  is  
 13119 equivalent to  $z_{it}$ ). Thus, if individual  $i$  is alive at time  $t$  ( $z(i, t) = 1$ ), then the observations  
 13120 are Bernoulli with detection probability  $p$  as before. Conversely, if the individual is not  
 13121 alive ( $z(i, t) = 0$ ), then the observations must be fixed zeros with probability 1. Note  
 13122 our distinct use of the variable  $z$  here as representing the state of individuals (alive/dead)  
 13123 instead of our previous use of  $z$  as the data augmentation variable.

13124 Survival and recruitment in the open population are manifest in a model for the  
 13125 latent state variables  $z(i, t)$  describing individual mortality and recruitment events. An  
 13126 important aspect of the hierarchical formulation of the model that we adopt here is that the

model for the state variables is described conditional on the total number of individuals ever alive during the study (a parameter which we label  $N$ ) based on  $T$  periods, as in Schwarz and Arnason (1996). Data augmentation induces a special interpretation on the latent state variables  $z(i, t)$ . In particular, “not alive” includes individuals that have died, or individuals that have not yet been recruited. Royle and Dorazio (2008) showed that using this formulation simplifies the state model and also allows it to be implemented directly in the **BUGS** language. For example, considering the case  $T = 2$ , the state model is composed of the following two components: First, the initial state is described by:

$$z(i, 1) \sim \text{Bernoulli}(\psi)$$

and then a model describing the transition of individual states from  $t = 1$  to  $t = 2$ :

$$z(i, 2) \sim \text{Bernoulli}(\phi z(i, 1) + \gamma(1 - z(i, 1))).$$

If  $z(i, 1) = 1$ , then the individual may survive to time  $t = 2$  with probability  $\phi$  whereas, if  $z(i, 1) = 0$ , then the “pseudo-individual” may be recruited with probability  $\gamma$ .

We can then generalize this model for  $T > 2$  time periods and allow survival and recruitment to be time dependent. Initialize the model for time  $T = 1$  as we have done above and then the model describing the transition of individual states from  $t$  to  $t + 1$  is:

$$z(i, t + 1) \sim \text{Bernoulli}(\phi_t z(i, t) + \gamma_t(1 - z(i, t))).$$

This parameterization results in  $T - 1$  survival and recruitment parameters. The main difference here from the CJS model, described below, is that we include recruitment and are interested in estimating  $N$  for each  $t$ . Since this state model described above is conditional-on- $N$ , we must deal with the fact that  $N$  is unknown, which is done through data augmentation similar to how we used it in the closed population models.

### 16.2.2 Data augmentation for the Jolly-Seber model

The fundamental challenge in carrying out a Bayesian analysis of this model is that the parameter  $N$  (the total number of individuals alive during the study) is not known. We have discussed and demonstrated data augmentation in many previous chapters; however, with the open population model, we have to take care that two issues are addressed: (1) the data augmentation is large enough to accommodate all potential individuals alive in the population during the entire study and (2) that individuals cannot die and then re-enter the population. Royle and Dorazio (2008) (see also Kéry and Schaub (2012)) describe this formulation for open population models, including the non-spatial JS and robust design models.

To begin, let’s consider the role of recruitment,  $\gamma$ , in the model when we use data augmentation to estimate  $N$ . Data augmentation formally reparameterizes the model, replacing  $N$ , the number of individuals ever alive with  $M$ , where we assume  $N \sim \text{Binomial}(M, \psi)$ . That is, the expected value of  $N$  under the model is equal to  $\psi M$ . As a result of this reparameterization, the recruitment parameters  $\gamma_t$  are also relative to the number of “available recruits” on the data augmented list of size  $M$ , and not directly related to the population size. This can be dealt with by deriving  $N_t$ , and  $R_t$ , the population size and

13164 number of recruits in year  $t$ , as a function of the latent state variables  $z(i, t)$ . For example,  
 13165 the total number of individuals alive at time  $t$  is

$$N_t = \sum_{i=1}^M z(i, t)$$

13166 and the number of recruits is

$$R_t = \sum_{i=1}^M (1 - z(i, t-1)) z(i, t)$$

13167 which is the number of individuals *not* alive at time  $t-1$  but alive at time  $t$ .

13168 In the case of just two primary periods, this process is straightforward. When the  
 13169 number of primary sample occasions is greater than 2, we must formulate the model for  
 13170 recruitment by introducing another latent variable, in order to ensure that an individual  
 13171 can only be recruited once into the population. Here, this formulation of the model uses  
 13172 a set of latent indicator variables, which we label  $A(i, t)$ , which describe the time interval  
 13173  $(t-1, t)$  at which individual  $i$  is recruited into the population. Let  $A(i, t) = 1$  if individual  
 13174  $i$  is recruited in time interval  $(t-1, t)$  otherwise  $A(i, t) = 0$ . To construct the recruitment  
 13175 process we make use of the standard conditional binomial construction of a removal process  
 13176 (Royle and Dorazio 2008). The initial state is given by:

$$A(i, 1) \sim \text{Bernoulli}(\gamma_1)$$

13177 for  $i = 1, 2, \dots, N$ . Then, for  $t > 1$

$$A(i, t) | A(i, t-1) \dots A(i, 1) \sim \text{Bernoulli}\left((1 - \sum_{\tau=1}^{t-1} A(i, \tau)) \gamma_t\right)$$

13178 where  $\gamma_1$  is equivalent to  $\psi$  in the description of the 2 primary occasion version open  
 13179 population model above and  $\tau$  is just a counter for times 1 to  $t-1$ . Each recruitment  
 13180 variable is conditional on whether the individual was ever previously recruited and this  
 13181 construction forces the recruitment variable after initial recruitment to be degenerate  
 13182 (have a sample size of 0). This ensures that an individual cannot be recruited again after  
 13183 initial recruitment. Then, we can describe the state variables  $z(i, t)$  by a 1st order Markov  
 13184 process. For  $t = 1$ , the initial states are fixed:

$$z(i, 1) \equiv A(i, 1)$$

13185 and, for subsequent states, we have

$$z(i, t) | z(i, t-1), A(i, t) \sim \text{Bernoulli}(\phi_t z(i, t-1) + A(i, t)).$$

13186 Thus, if an individual is in the population at time  $t$  (i.e.,  $z(i, t) = 1$ ), then that individual's  
 13187 status at time  $t+1$  is the outcome of a Bernoulli random variable with parameter (survival  
 13188 probability)  $\phi_t$ . If the individual, however, is not in the population at time  $t$  (i.e.,  $z(i, t) =$   
 13189 0), then the outcome is a Bernoulli random variable with probability  $\gamma_t$ , a parameter that  
 13190 is related to *per capita* recruitment. Recall that we use  $A$  to describe if an individual is  
 13191 available for recruitment and this is directly related to  $z$  as described above. We carry

13192 out this process in **JAGS** by using the `sum()` and `step()` functions together to ascertain  
 13193 if a particular individual  $i$  was ever previously alive. The `step()` function is a logical test  
 13194 in **JAGS** for  $x \geq 0$  such that `step( $x \geq 0$ )` returns a 1, otherwise 0. Individuals that  
 13195 were ever previously alive are no longer eligible to be “recruited” into the population. The  
 13196 implementation of this model in **JAGS** is shown in panel 16.1.

---

```
model{

  psi ~ dunif(0,1)
  phi ~ dunif(0,1)
  p.mean ~ dunif(0,1)

  for(t in 1:T){
    N[t] <- sum(z[1:M,t])
    gamma[t] ~ dunif(0,1)
  }

  for(i in 1:M){
    z[i,1] ~ dbern(psi)           # Alive state for the first year
    cp[i,1] <- z[i,1]*p.mean
    Y[i,1] ~ dbinom(cp[i,1], K)   # Y are the number of encounters
    A[i,1] <- (1-z[i,1])

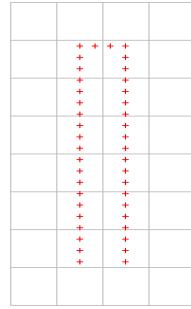
    for(t in 2:T){               # For loop for years 2 to T
      a1[i,t] <- sum(z[i, 1:t])  # Sum over the alive states from 1 to t
      A[i,t] <- 1-step(a1[i,t] - 1)
      # A is the indicator if an individual is available to be recruited
      mu[i,t]<- (phi*z[i,t-1]) + (gamma[t]*A[i,t-1])
      # Alive state at t is dependent on phi and gamma
      z[i,t] ~ dbern(mu[i,t])
      cp[i,t] <- z[i,t]*p.mean
      Y[i,t] ~ dbinom(cp[i,t], K)
    }
  }
}
```

---

Panel 16.1: **JAGS** model specification for the non-spatial Jolly-Seber model using data augmentation.

13197 **Ovenbird mist-netting study**

13198 We now return to the ovenbird data collected using mist-nets at Patuxent Wildlife  
 13199 Research Center. We introduced these data in Chaps. 9 and 14, and they are provided



**Figure 16.2.** Arrangement of the mist nests in the ovenbird study. The nets are arranged in a 600-m by 100m rectangle, spaced 30 m apart.

13200 with the **secr** package (see, Efford et al. (2004); Borchers and Efford (2008)). To refresh  
 13201 your memory: 44 mist nets spaced 30 m apart on the perimeter of a 600-m x 100-m  
 13202 rectangle (see Fig. 16.2) were operated on 9 or 10 non-consecutive days in late May and  
 13203 June for 5 years from 2005-2009.

13204 In Chaps. 9 and 14, we dealt with this dataset as a type of spatial “multi-session”  
 13205 model where abundances in each year,  $N_t$ , were regarded as independent random variables  
 13206 either with a Poisson prior (as implemented in **secr**, or a binomial prior if analyzed using  
 13207 **BUGS** with data augmentation. This is the simplest approach for modeling data collected  
 13208 over multiple years, but it does not allow for inference about demographic processes, as  
 13209 does the JS model.

13210 In the spatial multi-session model (S-MS) we were not interested in individual identity  
 13211 across years; however, we need to maintain the order of individuals across years to estimate  
 13212 the survival and recruitment of individuals into the population. We organize the data set  
 13213 so that each row in the array represents just one individual across all primary periods. For  
 13214 the ovenbird dataset, we can organize the data by creating a master list of all individuals  
 13215 captured during the entire study. From this list, we can assign each individual a unique row  
 13216 in our dataset (in the **R** commands, we do this by using the **unique()** function on the row  
 13217 names for each year of our 3-D array and use **pmatch()** to associate the data to the correct  
 13218 column). The resulting array is individual by secondary occasion by primary occasion,  $M$   
 13219 x  $K$  x  $T$ . The **R** commands to organize the data in a way suitable for fitting a Jolly-Seber  
 13220 type model are included in the **scrbook** package using the function **ovenbirds.js()** and  
 13221 are not shown here. The key difference between our model and organization of the data  
 13222 here and that in Chapt. 9 is that, here, we have to preserve individual identity across  
 13223 years (in the model and data structure).

13224 The data augmentation must be large enough to include individuals alive during any  
 13225 of the time periods and to account for that, we set  $M = 200$ . There were 70 unique  
 13226 individuals captured over the 5 year period. For this example, we hold survival constant  
 13227 but allow recruitment to be time dependent (since  $\gamma$  is essentially a function of the data

13228 augmentation process as described above, it does not make sense to hold recruitment  
 13229 constant and we therefore make it time specific).

13230 To implement the model in Panel 16.1, the following commands are used:

```
13231 # Set initial values for the alive state, z
13232 > zst <- c(rep(1,M/2),rep(0,M/2))
13233 > zst <- cbind(zst,zst,zst,zst,zst)
13234
13235 > inits <-function(){list(z=zst,sigma=runif(1,25,100),gamma=runif(5,0,1))}
13236 > parameters <- c("psi","N","phi", "p.mean", "gamma")
13237 > data <- list (K=10,Y=Ybin,M=M)
13238
13239 > library("rjags")
13240 > out1 <- jags.model("modelNSJS.txt",data,inits,n.chains=3,n.adapt=500)
13241 > out2NSJS <- coda.samples(out1,parameters,n.iter=20000)
```

13242 In this non-spatial JS model,  $N_t$  is estimated to be between about 22 and 33 for each  
 13243 of the 5 years (see Table 16.1 for results). The posterior mean for detection (`p.mean` in the  
 13244 model) was 0.14. We did not include `p.mean` in the table because the SCR models do not  
 13245 have a parameter that directly corresponds to it. Instead, SCR models have a detection  
 13246 function that is related to distance.

#### 13247 **Shortcomings of the traditional JS models**

13248 One of the biggest shortcomings of the non-spatial JS model is that we estimate  $N$  but  
 13249 have no explicit spatial area associated with it (so, in Table 16.1, the density estimate from  
 13250 the non-spatial JS model is listed as NA). Ignoring the spatial information in the data  
 13251 makes the estimation of density an informal process. As we saw in the closed models, the  
 13252 explicit incorporation of spatial information in the model will allow us to make an explicit  
 13253 estimate of density. This improvement should also carry through in our estimation of other  
 13254 demographic parameters such as survival and recruitment as the movement of individuals  
 13255 is directly accounted for in the model.

#### 13256 **16.2.3 Spatial Jolly-Seber models**

13257 To parameterize the spatial JS models, we follow all of the same steps as the non-spatial  
 13258 model but also include the trap location information into our detection function. Basically,  
 13259 we are using the closed population SCR model to estimate the detection parameters and  
 13260 initial population size, and the open component is carried out in the process of how we  
 13261 model the transition of  $z(i,t)$  to  $z(i,t+1)$  which is the same as in the non-spatial JS  
 13262 model. To do so, we describe the Bernoulli observation model, specified conditional on  
 13263  $z(i,t)$ , as has been done throughout the book:

$$y_{ijk|t}|z(i,t) \sim \text{Bernoulli}(p_{ijk}z(i,t)).$$

13264 with

$$p_{ijk} = p_0 * \exp(-\alpha_1 d_{ij}^2) \quad (16.2.1)$$

13265 where  $d_{ij} = ||\mathbf{x}_j - \mathbf{s}_i||$ , the distance between activity center  $\mathbf{s}_i$  and trap  $\mathbf{x}_j$ . As before,  
 13266  $p_0$  is the baseline encounter probability, for an individual with home range center located

precisely at a trap, and  $\alpha_1 = (1/(2\sigma^2))$  where  $\sigma$  is the scale parameter in this Gaussian encounter probability model.

If individual  $i$  is alive at time  $t$  ( $z(i,t) = 1$ ), then the observations are Bernoulli. Conversely, if the individual is not alive ( $z(i,t) = 0$ ), then the observations must be fixed zeros with probability 1. As always, other observation models can be considered in the context of a fully open JS type model, such as the Poisson or multinomial models described in Chapt. 9, and we can consider many alternative models of encounter probability.

We initialize the model for time  $T = 1$  and then model the transition of individual states from  $t$  to  $t + 1$  as:

$$z(i,t+1) \sim \text{Bern}(\phi_t z(i,t) + \gamma_t(1 - z(i,t))).$$

Previously, in sec. 16.2.2, it was described how this formulation of the model uses a set of latent indicator variables  $A(i,t)$  which describes if an individual is recruited into the population during time interval  $(t-1, t)$ . We apply the same approach here, so that, as before,  $A(i,t) = 1$  if individual  $i$  is recruited in time interval  $(t-1, t)$ ; otherwise  $A(i,t) = 0$ .

The number of recruits into the population is calculated based on the alive state of the previous time steps  $(1, 2, \dots, t-1)$  and the current time step ( $t$ ). For example, to estimate the number of recruits from time period 1 to 2, we count those individuals not in the population at time 1 ( $z(i,1) = 0$ ) but alive at time 2 ( $z(i,2) = 1$ ). We can determine if individual  $i$  has entered the population at time  $t = 2$  by using the formula:  $R_{i,2} = (1 - z(i,1))z(i,2)$  and then sum  $R_{i,2}$  over  $M$  to get the total number of recruits. We can do this for all the primary periods in our study, as shown in the **JAGS** code in Panel 16.2.

### Ovenbird mist-netting study

In the previous analysis of the ovenbird data, we did not make use of the spatial location for each net the ovenbirds were captured in. However, there were 44 mist nets operational during each of the sampling occasions. We already organized the data above so that our 3-D encounter histories are set up. The data set is then  $M = 200$  individuals by  $K = 10$  secondary occasions by  $T = 5$  primary occasions. In the non-spatial version, we reduced the data to captured or not-captured; however, the encounter history array, **Yarr**, contains the number of the net that each individual was captured in and contains a 45 if the individual was not captured. The code above describes how the encounter history array is created, so we do not reproduce this piece of code here. To call the model, use the following **R** code which sets the initial values for **z[i,t]**, the parameters to monitor, and calls **JAGS**. The code is also available in the **ovenbirds.js()** function.

```

13300 > zst <- c(rep(1,n),rep(0,M-n))
13301 > zst <- cbind(zst,zst,zst,zst,zst)
13302
13303 > inits <- function(){list(z=zst,sigma=runif(1,25,100),
13304                               gamma=runif(5,0,1), S=Sst,alpha0=runif(1,-2,-1))}
13305 > parameters <- c("psi", "alpha0", "alpha1", "sigma", "N",
13306                               "D", "phi", "gamma", "R")
13307 > data <- list(X=as.matrix(X[[1]]), K=10, Ycat=Yarr,
13308                           M=M, ntraps=ntraps, ylim=ylim, xlim=xlim)

```

```

model {
  psi ~ dunif(0,1)      # Prior distributions
  phi ~ dunif(0,1)
  alpha0 ~ dnorm(0,10)
  sigma ~ dunif(0,200)
  alpha1 <- 1/(2*sigma*sigma)
  A <- ((xlim[2]-xlim[1]))*((ylim[2]-ylim[1]))  # Area of state-space

  for(t in 1:T){
    N[t] <- sum(z[1:M,t])  # Calculate abundance for each year
    D[t] <- N[t]/A          # Calculate density for each year
    R[t] <- sum(R[1:M,t])   # Calculate the recruits for each year
    gamma[t] ~ dunif(0,1)   # Prior for time specific recruitment parameter
  }

  for(i in 1:M){
    z[i,1] ~ dbern(psi)
    R[i,1] <- z[i,1]        # To estimate the number of recruits
    R[i,2] <- (1-z[i,1])*z[i,2]
    R[i,3] <- (1-z[i,1))*(1-z[i,2])*z[i,3]
    R[i,4] <- (1-z[i,1))*(1-z[i,2))*(1-z[i,3])*z[i,4]
    R[i,5] <- (1-z[i,1))*(1-z[i,2))*(1-z[i,3))*(1-z[i,4])*z[i,5]

    for(t in 1:T){
      # Independent activity centers for each year
      S[i,1,t] ~ dunif(xlim[1],xlim[2])
      S[i,2,t] ~ dunif(ylim[1],ylim[2])
      for(j in 1:ntraps){
        d[i,j,t] <- pow(pow(S[i,1,t]-X[j,1],2) + pow(S[i,2,t]-X[j,2],2),1)
      }
      for(k in 1:K){
        for(j in 1:ntraps){
          lp[i,k,j,t] <- exp(alpha0 - alpha1*d[i,j,t])*z[i,t]
          cp[i,k,j,t] <- lp[i,k,j,t]/(1+sum(lp[i,,t]))
        }
        cp[i,k,ntraps+1,t] <- 1-sum(cp[i,k,1:ntraps,t])
        # Here, the last cell indicates not captured
        Ycat[i,k,t] ~ dcat(cp[i,k,,t])
      }
    }
    A[i,1]<-(1-z[i,1])
    for(t in 2:T){           # For loop for years 2 to T
      a1[i,t] <- sum(z[i, 1:t]) # Sum over alive states from 1 to t
      A[i,t] <- 1-step(a1[i,t] - 1)
      # A indicates if individual is available to be recruited at time t
      mu[i,t] <- (phi*z[i,t-1]) + (gamma[t]*A[i,t-1])
      # Alive state at t is dependent on phi and gamma
      z[i,t] ~ dbern(mu[i,t])
    }
  }
}

```

**Table 16.1.** Posterior mean of model parameters for the non-spatial Jolly-Seber model (NS-JS), the spatial Jolly-Seber model (S-JS), and the spatial multi-session model (S-MS) fitted to the ovenbird data set. Density shown in individuals per hectare.

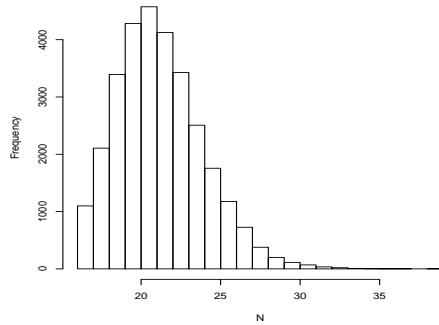
	NS-JS	S-JS	S-MS
D[1]	NA	0.96	0.93
D[2]	NA	1.00	1.00
D[3]	NA	1.10	1.20
D[4]	NA	1.10	0.89
D[5]	NA	0.79	0.76
N[1]	26.5	33	32.4
N[2]	30.2	36	35.8
N[3]	33.1	39	42.1
N[4]	29.5	37	30.8
N[5]	21.7	28	26.2
alpha0	NA	-2.9	-2.88
alpha1	NA	1.2e-04	1.22e-04
sigma	NA	6.4	6.44
gamma[1]	0.50	0.50	NA
gamma[2]	0.09	0.09	NA
gamma[3]	0.11	0.13	NA
gamma[4]	0.13	0.16	NA
gamma[5]	0.07	0.08	NA
phi	0.48	0.53	NA
psi	0.14	0.17	NA
R2	NA	15	NA
R3	NA	19	NA
R4	NA	8.3	NA
R5	NA	8.3	NA

```

13309 > library("rjags")
13310 > out1 <- jags.model("modelJS.txt", data, inits, n.chains=3,
13311   n.adapt=500)
13312 > out2JS <- coda.samples(out1,parameters,n.iter=10000)

```

13313 Our results for density,  $\alpha_0$ , and  $\alpha_1$  are rather similar to those found in the multi-
 13314 season analysis from Chapt. 9. Since all of our parameters including  $\alpha_0$  and  $\alpha_1$  are
 13315 shared between seasons, we would expect these results to be similar between the multi-
 13316 season model and the JS model (see Table 16.1). There are some slight differences in the
 13317 parameter estimates, for example, the density is lower in year 4 in the multi-season model
 13318 than in the JS model. This may be due to a smaller sample size in that year; due to the
 13319 Markovian relationship between abundances, the JS model is able to make use of the data
 13320 more efficiently. Because we have defined the same state space for the spatial JS model
 13321 and multi-season, our estimates of  $N_t$  are directly comparable. However, the estimates
 13322 of  $N_t$  under the non-spatial JS model are not directly comparable as we do not have a
 13323 well-defined effective trapping area. We see from Table 16.1 that  $N_t$  is smallest for the
 13324 non-spatial JS model across all years. This suggests that the actual effective trapping area
 13325 is smaller than our state-space, but we cannot know how much relative to the state-space
 13326 to make useful comparisons between the  $N_t$ s.



**Figure 16.3.** Posterior distribution of  $N_5$  from the spatial JS model for the ovenbird dataset. This figure suggests that there is no truncation of the posterior of  $N_5$  by  $M$ .

In the JS formulation of the model, we also estimate the recruitment for each year, and we can look at our derived values for recruitment (R2, R3, R4, and R5). R2 is the number of new recruits from primary period 1 to 2; R3 is the number of new recruits from primary period 2 to 3; and so forth. R2 and R3 are almost double that of R4 and R5, suggesting that less animals were recruited into the population in the latter years of the study. The density in the last year of the study was lower than previous years. It is good to check your results when you see a pattern like this – the number of recruits declining each year – because this could be an indication that the data augmentation was not large enough. In this example, we checked to make sure that  $M=200$  was sufficiently large by examining the recruitment parameter,  $\gamma$ . If  $\gamma$  is close to 1 during any of the time periods, then there are not enough augmented individuals in the overall dataset. In this case, the 97.5% quantile of  $\gamma_5$ , the recruitment probability in the final year of the study, was 0.14 and none of the other  $\gamma$ 's were close to 1 either. You can also look at the posterior distributions of  $N_t$  to make sure they are not truncated, Fig. 16.3 shows that the posterior distribution of  $N_5$  is not truncated. The posterior mean for survival,  $\phi$ , was 0.53. Although we did not do that in this example, it should be easy to see that we could allow survival to vary by time, as we did with recruitment. Our estimates of survival seem reasonable when compared with the ovenbird literature. Some studies have found annual male ovenbird survival to be around 0.62 (Porneluzi and Faaborg, 1999; Bayne and Hobson, 2002), whereas female ovenbird survival is much lower (0.21, Bayne and Hobson (2002)). With more individuals, we could run this model with survival estimated for each sex separately. However, researchers should be careful not to over-parameterize models based on the amount of data available. The results indicate that the posterior mean estimate of  $\phi$  was greater in the SCR model (0.53) than the non-spatial model (0.48) which suggests that the SCR model is starting to separate movement from survival in order to estimate the true rather than apparent survival.

## 16.3 CORMACK-JOLLY-SEBER MODELS

### 16.3.1 Traditional CJS models

Cormack-Jolly-Seber (CJS) models are used extensively to estimate survival probabilities. There are two common ways to fit these models, using either a multinomial likelihood approach (Lebreton et al., 1992) or a state-space formulation of the model (Gimenez et al., 2007; Royle, 2008). The multinomial likelihood approach is based on summarizing the data to counts of unique encounter histories, which have a multinomial distribution. The data are summarized over individuals and so it is not feasible to build general models that contain individual covariates. On the other hand, the state-space formulation of the model preserves individual identity and, therefore, parameters can be modeled at the individual level, and individual effects (covariates or heterogeneity) can be included. In the present context of spatial capture-recapture models, we naturally think about including individual locations, or activity centers, as individual covariates.

We can adopt a simple state-space parameterization of the basic single state, non-spatial CJS model in which the observation model is described conditional on the latent state variables  $z(i, t)$  – the “alive state” which indicates whether individual  $i$  is alive ( $z(i, t) = 1$ ) or not ( $z(i, t) = 0$ ) during each of  $t = 1, 2, \dots, T$  primary periods. Let  $y_{it}$  indicate the observed encounter data of individual  $i$  in primary period  $t$ . The model, specified conditional on  $z(i, t)$ , is:

$$y_{it}|z(i, t) \sim \text{Bernoulli}(p_t z(i, t)).$$

Analogous to the JS model, if individual  $i$  is alive at time  $t$  ( $z(i, t) = 1$ ), then the observations are Bernoulli with probability of detection  $p_t$ .

If the individual is not alive ( $z(i, t) = 0$ ), then the observations must be fixed zeros with probability 1. Contrary to the JS model, in the CJS model we condition on first capture, which means that  $z(i, t)$  will be 1 when  $t$  is the primary period individual  $i$  is first captured in. We denote this  $z(i, f_i)$ , where  $f_i$  indicates the primary occasion in which individual  $i$  is first captured, which can vary from  $1 \dots T$ . This ensures that each individual is alive upon entering the model, and is also the reason that recruitment is not estimated in the model.

The “alive state” at time  $t$  for each individual is a function of the state at the previous time step  $t - 1$ . Because we condition on the first capture, the initial state is set to one:

$$z(i, f_i) = 1$$

where  $f_i$  indicates the primary occasion in which individual  $i$  is captured and the model for the transition of individual states from  $t$  to  $t + 1$  for all  $t > f_i$  is

$$z(i, t) \sim \text{Bernoulli}(\phi z(i, t - 1)).$$

Because we start with  $z(i, f_i) = 1$ , the individual survives with probability  $\phi$  to time  $f_i + 1$  and so forth. Once an individual leaves the population (i.e.,  $z(i, t) = 0$ ), there is no mechanism for the individual to return. This means that under this specification individuals cannot temporarily emigrate. In the CJS model we are not estimating  $N_t$ , so we do not need to make use of data augmentation here in order to account for uncaptured individuals (remember: we explicitly condition on first capture). This version of the model

13390 is easy to construct in the **BUGS** (or **JAGS**) language which is shown in Panel 16.3.  
 13391 Variations on this basic model and associated code for fitting the model in **BUGS** are  
 13392 described in detail in Kéry and Schaub (2012, Chaps. 7-9).

---

```
model{
  phi ~ dunif(0,1) # Survival (constant over time)

  for(t in 1:T){
    p[t] ~ dunif(0, 1) # Detection (varies with time)
  }

  for(i in 1:M){
    z[i,first[i]] ~ dbern(1)
    for (t in (first[i]+1):T){
      tmp[i,t] <- z[i,t]*p[t]
      y[i,t] ~ dbern(tmp[i,t])
      phiUP[i,t] <- z[i,t-1]*phi
      z[i,t] ~ dbern(phiUP[i,t])
    }
  }
}
```

---

Panel 16.3: **JAGS** model specification for the non-spatial basic Cormack-Jolly-Seber (CJS) model. Note that the first alive state of each individual,  $z[i, \text{first}[i]]$ , is not stochastic. It is equal to 1 with probability 1.

13393 **Movement and survival of American shad in the Little River**

13394 As an example for the CJS model, we use data collected on American shad (*Alosa*  
 13395 *sapidissima*) in the Little River in North Carolina, U.S.A. (see photo in Fig. 16.4). The  
 13396 Little River is a tributary to the Neuse River and the confluence is near Goldsboro,  
 13397 North Carolina about 212 river kilometers from the Pamlico Sound. The motivation for  
 13398 this example stems from an interest in better understanding survival and movement of  
 13399 migratory fish. American shad are an anadromous fish that use rivers for spawning. The  
 13400 data were collected and analyzed as described in Raabe (2012). Using a resistance board  
 13401 weir near the river mouth, 315 fish were tagged with passive integrated transponders  
 13402 (PIT) in the spring of 2010. An array of 7 upstream PIT antennas passively recaptured  
 13403 individuals during upstream and downstream migrations. Each time a fish passed over  
 13404 the antenna, it was recorded and the data were summarized weekly for 12 weeks. The fish  
 13405 do not necessarily move past all antennae and may remain in the river between antennae  
 13406 for more than a week, thus they are not all detected at each time period. The antennae  
 13407 do not always operate perfectly either and fish that pass may not be recorded at some  
 13408 times.



**Figure 16.4.** American shad caught in North Carolina, U.S.A. Credit: Joshua Raabe, North Carolina State University

13409 To apply the basic CJS model, we create the encounter history for each individual for  
 13410 the 12 weeks and we also create a vector to indicate the period (week) of first capture.  
 13411 The code is not shown here but is available in the `scrbook` package within the function  
 13412 `shad.cjs()`. This function contains all of the code to fit the non-spatial, multi-state, and  
 13413 spatial CJS models to the American shad dataset.

13414 Table 16.2 shows the estimated detection probabilities for each of the 12 primary  
 13415 periods in the study. The posterior mean for detection probabilities ranges from 0.126 to  
 13416 0.880, which could potentially be due to variation in water flow, stream depth, storms,  
 13417 etc... The weekly survival probability,  $\phi$ , had a posterior mean estimate of 0.824. This  
 13418 estimate could be considered low for a weekly probability, but is likely due to the fact that  
 13419 the migration upstream can be quite energetically taxing and the fish are likely to only  
 13420 feed minimally in rivers (Leggett and Carscadden., 1978; Leonard and McCormick, 1999).  
 13421 Additionally, the CJS model is only estimating apparent survival and some fish may have  
 13422 left the stream temporarily or permanently heading back to the ocean or possibly to other  
 13423 tributaries that are not monitored. We demonstrate in Panel 16.3 how to allow  $p$  to vary  
 13424 by time, but we could also allow survival,  $\phi$  to vary by time by implementing it exactly  
 13425 as we do for  $p$ . As we move into the multi-state model, we can test for movement and  
 13426 survival by state, which allows more specific biological questions to be addressed.

### 13427 **16.3.2 Multi-state CJS models**

13428 The basic version of the CJS model only allows for estimation of survival and detection  
 13429 probabilities. However, researchers are often interested in addressing other ecological  
 13430 questions such as age-dependent survival rates, habitat based movements, etc. Multi-  
 13431 state models allow researchers to directly address such questions by incorporating more  
 13432 than one state that an individual may potentially be in (Arnason, 1972, 1973; Brownie  
 13433 et al., 1993). These possible states can be geographic location, age class, or reproductive  
 13434 status among many others. Instead of just having an encounter history for an individual,

**Table 16.2.** Results of the basic non-spatial CJS model for the American shad dataset.

	Mean	SD	2.5 %	50 %	97.5 %
p[1]	0.499	0.289	0.026	0.499	0.975
p[2]	0.627	0.058	0.511	0.628	0.738
p[3]	0.762	0.036	0.689	0.763	0.829
p[4]	0.880	0.025	0.828	0.882	0.925
p[5]	0.548	0.043	0.465	0.548	0.633
p[6]	0.259	0.038	0.190	0.258	0.337
p[7]	0.126	0.031	0.072	0.124	0.194
p[8]	0.236	0.045	0.155	0.234	0.332
p[9]	0.237	0.049	0.148	0.234	0.341
p[10]	0.589	0.072	0.447	0.590	0.728
p[11]	0.834	0.063	0.700	0.839	0.942
p[12]	0.468	0.072	0.330	0.466	0.614
$\phi$	0.824	0.011	0.802	0.825	0.846

we will also have auxiliary information on the state of that individual at capture (e.g., breeder or non-breeder). Since our interest is in movement of individuals, here we will consider states that represent spatial units or geographic locations. Generally speaking, we might think that the transition rates between locations could be due to habitat features (or quality) and we can use multi-state models to help us address such a question. In addressing movement through a multi-state modeling approach, the movement is often parameterized as random or Markovian between patches (Arnason, 1972, 1973; Schwarz et al., 1993).

In the simplest version of the multi-state model we have just two states. Thus, individuals can be marked and recaptured in one of two states (we'll call them A and B here). We will assume that the two "states" are different geographic sites. In the single-state model above, an individual  $i$  was either alive ( $z(i, t) = 1$ ) at time  $t$  or dead ( $z(i, t) = 0$ ). Now, we must consider that the individual could be alive in a given state or dead and that individuals can transition between states. An easy way to think about this is to look at the state transition matrix in Table 16.3. Here,  $\phi^A$  is the probability of surviving in State A from time  $t$  to  $t + 1$  and  $\phi^B$  is the analogous parameter for State B. The movement parameters are  $\psi^{AB}$  and  $\psi^{BA}$ , where  $\psi^{AB}$  is the probability that an individual, survives from  $t$  to  $t + 1$  and moves to State B just before  $t + 1$  and vice versa for  $\psi^{BA}$ . The movement could also be defined as occurring before the survival; i.e.,  $\psi^{AB}$  is the probability that an individual moves from State A to State B shortly after time  $t$  and then survives to time  $t + 1$  in State B.

**Table 16.3.** Transition matrix for a multi-state model with just two states.

	State A	State B	Dead
State A	$\phi^A(1 - \psi^{AB})$	$\phi^A\psi^{AB}$	$1 - \phi^A$
State B	$\phi^B\psi^{BA}$	$\phi^B(1 - \psi^{BA})$	$1 - \phi^B$
Dead	0	0	1

Because individuals are not necessarily observed in their given state, detection should

be estimated separately for each of the states. Hence, we also have  $p^A$  and  $p^B$ , the probability of detecting an individual in State A and State B respectively. Also, at this point, we assume that there is no error in observed State (i.e., if the animal is observed, then the State is recorded correctly).

In the next few paragraphs, we show how the formulation of the 2 state multi-state capture recapture presented above can be related directly to SCR models. To start, define  $\mathbf{s}$  as the index of which state an individual is actually in and  $u_{it}$  as the state in which individual  $i$  was observed during sample  $t$ . In this two state example,  $u_{it}$  can only take on values for being observed in A or B (i.e., 1 or 2). We can define a simplistic model as follows:

$$u_{it} \sim \text{dcat}(\psi)$$

where  $\psi$  is a constant vector.

We observe an individual with probability  $p_0$ , that is:

$$\Pr(y_{it} = 1 | u_{it}) = p_0$$

The state-transition probabilities are constant.

In an alternative formulation of this model, we can define  $\mathbf{s}$  as the index of which state an individual is in and then condition the observed locations,  $u_{it}$  as a function of the actual state,  $\mathbf{s}$ . This means that whether an individual moves or not, or where it moves to, is a function of where it is located. In this case, successive movement outcomes are *iid* and we can write the model according to:

$$u_{it} \sim \text{dcat}(\psi(\mathbf{s}_i))$$

Conditional on the state in which individual  $i$  is located, the probability of observing the individual is the same as above,  $p_0$ . However, in this formulation of the model, the state-transition probabilities are constant, conditional on  $\mathbf{s}$ . Other models for these transition probabilities are possible and we will discuss those later.

A slight modification of this model would define  $\mathbf{s}$  as a “home area” for each individual. Then the region the animal goes to is a function not of where it was last time, but which region is its home area. This model is only slightly different from the Markovian model and, as was shown in Chapt. 9 for closed populations models, is how we make the technical transition from multi-state models to SCR models. Essentially increasing to a large number of strata, this formulation of the multi-state model becomes an SCR model where the “area of activity”  $\mathbf{s}$  becomes the “activity center” for each individual. In this case, the vector  $\psi(\mathbf{s}_i)$  is a  $J \times 1$  vector, corresponding to the probability of observation in each trap, given the individual’s activity center. Therefore, SCR models are closely related to classical multi-state models where the state-variable is “space.”

To describe this model for **JAGS**, we use a slightly different formulation which combines  $u_{it}$  and  $y_{it}$  as defined above into one observation matrix such that  $y_{it} = 1, 2$ , or 3 where 3 indicates “not observed”. Additionally, we use  $z(i, t)$  to indicate the true state of individual  $i$  such that  $z(i, t) = 1, 2$ , or 3 where 1 indicates alive and in state 1, 2 indicates alive and in state 2, and 3 indicates “not alive”. Using this delineation, we just need to set up the transition matrix based on Table 16.3 and define each item within the model specification, shown in Panel 16.4. Note that this can become quite cumbersome when dealing with models that have many states.

---

```

model{

# r is an index for state (excluding the 'not alive' state)
for(r in 1:2){
    phi[r] ~ dunif(0,1)
    psi[r] ~ dunif(0,1)
    p[r] ~ dunif(0,1)
}

for (i in 1:M){
    z[i,first[i]] <- y[i, first[i]]
    for (t in (first[i]+1):T){
        z[i,t] ~ dcat(ps[z[i,t-1], i, ])
        y[i,t] ~ dcat(po[z[i,t], i, ])
    }
    ps[1, i, 1] <- phi[1] * (1-psi[1])
    ps[1, i, 2] <- phi[1] * psi[1]
    ps[1, i, 3] <- 1-phi[1]
    ps[2, i, 1] <- phi[2] * (1-psi[2])
    ps[2, i, 2] <- phi[2] * psi[2]
    ps[2, i, 3] <- 1-phi[2]
    ps[3, i, 1] <- 0
    ps[3, i, 2] <- 0
    ps[3, i, 3] <- 1

    po[1, i, 1] <- p[1]
    po[1, i, 2] <- 0
    po[1, i, 3] <- 1-p[1]
    po[2, i, 1] <- 0
    po[2, i, 2] <- p[2]
    po[2, i, 3] <- 1-p[2]
    po[3, i, 1] <- 0
    po[3, i, 2] <- 0
    po[3, i, 3] <- 1
}
}

```

---

Panel 16.4: **JAGS** model specification for a two-state version of the multi-state CJS model. Code modified from (Kéry and Schaub, 2012, Chapt. 9).

**Table 16.4.** Results of the multi-state CJS model for the migratory fish example.  $p^A$  is the detection probability in the first state (A), which in this case is the down stream area.  $\phi^A$  is the weekly survival probability in state A and  $\psi^{AB}$  is the probability that an individual, which survived from  $t$  to  $t + 1$  in Site A, moves to State B just before  $t + 1$ .

	Mean	SD	2.5 %	50 %	97.5 %
$p^A$	0.777	0.045	0.689	0.777	0.866
$p^B$	0.434	0.027	0.382	0.434	0.489
$\phi^A$	0.850	0.022	0.807	0.851	0.893
$\phi^B$	0.782	0.019	0.743	0.782	0.820
$\psi^{AB}$	0.421	0.034	0.356	0.421	0.489
$\psi^{BA}$	0.927	0.014	0.897	0.937	0.952

#### 13497    Movement and survival of American shad in the Little River

13498    Previously, we analyzed the American shad data using a basic (i.e., non-spatial) CJS  
 13499    model. However, the researchers were interested in movement of fish during migration and  
 13500    so we classified the stream into 2 states (regions) – “downstream” and “upstream”. Each  
 13501    antenna was assigned to a state based on the location, those below 20 river kilometers  
 13502    were considered in the downstream state. Each fish has an encounter history including  
 13503    whether or not the fish was detected during each week of the 12 week study, but also  
 13504    the “state” of capture (“downstream” or “upstream”). A vector to indicate the period  
 13505    of first capture was also created. Fish captured in more than one state during the week  
 13506    were assigned the state in which they were captured most during that week. And the  
 13507    model assumes that individuals observed in a state at consecutive primary periods did  
 13508    not move from that state within the primary period. The data manipulation and model  
 13509    specification for the multi-state CJS model is provided in the `scrbook` package under  
 13510    the function `shad.cjs()`.

13511    Survival between the two areas was quite different (see Table 16.4). This might suggest  
 13512    that fish moving further upstream are expending more energy and are more likely to die.  
 13513    While survival in the two states was different, it is intuitive that the average of the survival  
 13514    probabilities for A and B is essentially the same as that from the basic non-spatial CJS  
 13515    ( $\phi = 0.82$ , see Table 16.2). Also, it should be noted that  $\psi^{BA}$  was very high, indicating  
 13516    that fish in this study are returning downstream after spawning in the upstream area.  
 13517    These results highlight the utility in using a multi-state model to understand movement  
 13518    between states; here, we used spatial states, but age, class, breeding status, etc. are all  
 13519    possibilities. We did have to reduce the dataset however to fit this model and information  
 13520    on exact spatial location of detections was lost in creating just two states, downstream  
 13521    and upstream. Losing information is one potential effect of using a multi-state model;  
 13522    additionally when states are hidden or unknown (e.g., when animals are in a region not  
 13523    exposed to sampling or the state is misclassified), these models can be difficult to fit (see  
 13524    Conn and Cooch (2009) for an overview of these problems). Not losing information and  
 13525    unknown states are two issues that can be resolved using the fully spatial CJS model.  
 13526    Misclassification of state (or even individual) is a difficult problem to solve and current  
 13527    approaches (Link et al., 2010; McClintock et al., In press) are in development for SCR  
 13528    models.

**13529 16.3.3 Spatial CJS models**

13530 In Chapt. 9, we suggested that SCR models are essentially a type of multi-state model  
 13531 with spatially structured transition probabilities. As we noted, individuals can appear  
 13532 in  $> 1$  states simultaneously, which is not directly analogous to a standard multi-state  
 13533 model. However, building on the state-space and multi-state CJS models, we can explicitly  
 13534 incorporate individual movement as an individual covariate model (Royle, 2009a). To  
 13535 move from the basic and multi-state CJS models to the SCR version, we need only make a  
 13536 few changes to the model. We will not have discrete states and thus the biggest difference  
 13537 is that individuals do not “transition” between a finite set of states, but instead are allowed  
 13538 to move in continuous space.

13539 We may consider the same basic encounter models as described previously (i.e., Pois-  
 13540 son, Bernoulli, or multinomial). In particular, let  $y_{ijkt}$  indicate the observed encounter  
 13541 data of individual  $i$  in trap  $j$ , during interval (secondary period or sub-sample)  $k =$   
 13542  $1, 2, \dots, K$  and primary period  $t$ . We note that in some cases we may have intervals  
 13543 ( $K = 1$ ) which correspond to the design underlying a standard CJS or JS models whereas  
 13544 the case  $K > 1$  corresponds to the “robust design” (Pollock 1982). The Poisson observa-  
 13545 tion model, specified conditional on  $z(i, t)$ , is:

$$y_{ijkt} | z(i, t) \sim \text{Poisson}(\lambda_0 g_{ij} z(i, t))$$

13546 where  $\lambda_0$  is the baseline encounter rate and  $g_{ij}$  is the detection model as a function of  
 13547 distance. If the individual is not alive ( $z(i, t) = 0$ ), then the observations must be fixed  
 13548 zeros with probability 1. Remember that in the CJS formulation, we condition on first  
 13549 capture which means that  $z(i, t)$  will be 1 when  $t$  is the first primary period of capture.  
 13550 As before in the non-spatial CJS model, we can denote this as  $z(i, f_i)$  where  $f_i$  indicates  
 13551 the primary occasion in which individual  $i$  is first captured.

13552 Modeling time-effects either within or across primary periods is straightforward. For  
 13553 that, we define  $\lambda_0 \equiv \lambda_0(k, t)$  and then develop models for  $\lambda_0(k, t)$  as in our closed SCR  
 13554 models (we note that trap-specific effects could be modeled analogously).

13555 We follow the same model for survival as described in the non-spatial version of the  
 13556 CJS. The model is initialized by setting the alive state at first capture to one:

$$z(i, f_i) = 1$$

13557 and for the transition of an individual’s alive state from  $t$  to  $t + 1$ , for all  $t > f_i$ , we have

$$z(i, t) \sim \text{Bernoulli}(\phi z(i, t - 1)).$$

13558 An individual survives with probability  $\phi$  from one time step to the next. It is easy to see  
 13559 that we can let survival be time specific by allowing  $\phi$  to vary with each time step:

$$z(i, t) \sim \text{Bernoulli}(\phi_t z(i, t - 1)).$$

13560 In either case, once an individual leaves the population (i.e.,  $z(i, t) = 0$ ), there is no  
 13561 recruitment so individuals cannot return. Again, we are not estimating  $N_t$  in this model,  
 13562 hence we do not need any data augmentation. This conveniently makes the model run  
 13563 faster too!

**Table 16.5.** Results of the spatial Cormack-Jolly-Seber model fitted to the American shad data set.

	Mean	SD	2.5 %	50 %	97.5 %
lam0[1]	5.555	0.224	5.125	5.553	6.003
lam0[2]	4.442	0.155	4.143	4.437	4.752
lam0[3]	1.892	0.068	1.763	1.891	2.031
lam0[4]	1.126	0.055	1.021	1.125	1.238
lam0[5]	0.949	0.058	0.838	0.948	1.067
lam0[6]	0.359	0.040	0.284	0.357	0.443
lam0[7]	0.188	0.031	0.133	0.186	0.254
lam0[8]	0.309	0.044	0.230	0.307	0.402
lam0[9]	0.363	0.052	0.269	0.361	0.471
lam0[10]	0.627	0.072	0.493	0.625	0.777
lam0[11]	1.611	0.109	1.408	1.607	1.835
lam0[12]	0.939	0.139	0.697	0.929	1.241
$\phi$	0.784	0.012	0.760	0.785	0.807
$\sigma$	13.954	0.197	13.573	13.950	14.350

#### 13564    Movement and survival of American shad in the Little River

13565    Going back to our American shad example, we can consider that this is exactly a  
 13566    spatial capture recapture problem. In stream networks, the placement of PIT antennas  
 13567    along the stream mimics the type of spatial data collected in terrestrial passive detector  
 13568    arrays such as camera traps, hair snares, acoustic recording devices, etc. The difference is  
 13569    that for fish and aquatic species, the stream constrains the movement of individuals to a  
 13570    linear network. Using the data from the array of 7 PIT antennas and the number of times  
 13571    each fish passed over the antenna, we can apply the SCR CJS model to evaluate movement  
 13572    up and downstream of these fish. When we look at the individuals encountered at each  
 13573    antenna for each of the primary periods, the dimensions of the data are 315 individuals by  
 13574    7 antennas by 12 sample occasions. Individuals can encounter any antenna any number  
 13575    of times during the week, which means we just sum the encounters over the week and  
 13576    eliminate any need for explicit secondary occasions in the model. The result is a 3-D  
 13577    array instead of a 4-D array. Given the structure of the encounters, we use a Poisson  
 13578    encounter model in this example shown in 16.5. The code to carry out this model is  
 13579    provided in the `scrbook` package using the function `shad.cjs()`.

13580    The baseline encounter rate,  $\lambda_0$ , was allowed to vary by week and ranged from 0.188  
 13581    to 5.555. We use the Poisson encounter model in this spatial CJS example rendering  $\lambda_0$   
 13582    not directly comparable to  $p_0$  from the non-spatial and multi-state versions, which arises  
 13583    as the detection probability under the Binomial encounter model. The posterior mean for  
 13584     $\phi$  was 0.784 (see Table 16.5), again showing that the weekly survival probability is rather  
 13585    low, just as we saw in the two previous example analyses of these data. Here, we are  
 13586    modeling survival probability as constant, but there is reason to believe that it might vary  
 13587    by time (similar to detection) and we might consider this additional parameterization in a  
 13588    more complete analysis of the data set. The other parameter of interest is  $\sigma$ , the movement  
 13589    parameter, which had a posterior mean of 13.954. Stream locations are recorded in river  
 13590    kilometers (RKM), so  $\sigma$  is in units of kms. Our system here is linear, so we do not think  
 13591    of fish as having a home range radius. However,  $\sigma$  can still inform us about the linear

---

```

model {
# Priors
sigma ~ dunif(0,80)
sigma2 <- sigma*sigma
lam0 ~ dgamma(0.1, 0.1)
phi ~ dunif(0, 1)  # Survival (constant across time)
tauv~dunif(0, 30)
tau<-1/(tauv*tauv)

for (i in 1:M){
z[i,first[i]] <- 1
S[i,first[i]] ~ dunif(0,50)  #Fish enter the stream at 0, thus the
#first AC is set to the lower stream end

for(j in 1:nantenna) {
D2[i,j,first[i]] <- pow(S[i,first[i]]-antenna.loc[j], 2)
lam[i,j,first[i]]<- lam0*exp(-D2[i,j,first[i]]/(2*sigma2))
tmp[i,j,first[i]] <- lam[i,j,first[i]]
y[i,j,first[i]] ~ dpois(tmp[i,j,first[i]])
}

for (t in first[i]+1:T) {
S[i,t] ~ dunif(xl, xu)
for(j in 1:nantenna) {
D2[i,j,t] <- pow(S[i,t]-antenna.loc[j], 2)
lam[i,j,t] <- lam0 * exp(-D2[i,j,t]/(2*sigma2))
tmp[i,j,t] <- z[i,t]*lam[i,j,t]
y[i,j,t] ~ dpois(tmp[i,j,t])
}
phiUP[i,t] <- z[i,t-1]*phi
z[i,t] ~ dbern(phiUP[i,t])
}
}
}

```

---

Panel 16.5: **JAGS** model specification for the spatial Cormack-Jolly-Seber (CJS) model for the American shad dataset. Note that the first alive state of each individual,  $z[i, \text{first}[i]]$ , is not stochastic. It is equal to 1 with probability 1.

13592 distance fish are moving. One final note about this example, we have simplified the dataset  
 13593 for analysis here and some parameter estimates are different than found in Raabe (2012).

## 16.4 MODELING MOVEMENT AND DISPERSAL DYNAMICS

13594 To better understand the dynamics of a population, it is important to consider how the  
 13595 locations of activity centers evolve over time. It is known that home ranges and territories  
 13596 of animals can shift and other types of movement (migration, dispersal) can take place. In  
 13597 the framework of open population SCR models, these dynamics of individual locations can  
 13598 be reflected in appropriate models for the distribution of the activity centers. To begin,  
 13599 a plausible “null model” for the distribution of individual activity centers is to assume  
 13600 they are static over time and do not change across periods, i.e.,  $\mathbf{s}_i \sim \text{Uniform}(\mathcal{S})$ . This  
 13601 model may be appropriate for territorial species where the primary sampling periods are  
 13602 relatively close together in time or the overall time frame of the study is limited. It might  
 13603 seem more likely that activity centers change over time but are independent from year to  
 13604 year for a given individual such  $\mathbf{s}_i \sim \text{Uniform}(\mathcal{S})$ . This is also how the spatial version of  
 13605 the JS and CJS models were formulated above and this might a reasonable model when  
 13606 there are large time lags between surveys, or if the individuals redistribute themselves  
 13607 frequently in the study population.

13608 An intermediate option would be to assume that  $\mathbf{s}(i, t) \sim \text{Normal}(\mathbf{s}(i, t - 1), \tau^2 \mathbf{I})$  for  
 13609  $t > 1$  so that individual home range centers are perturbed randomly from their previous  
 13610 value. This is possibly the the most realistic model for many cases. For example, many  
 13611 migratory passerines, like the ovenbird, return to the same location, or nearly so each  
 13612 year.

13613 We could use also use models of the activity centers to look at patterns of animal dis-  
 13614 tribution with regard to habitat. For example, if the primary period is a season, it may be  
 13615 expected that individuals move as the available food sources change. Using telemetry data  
 13616 and/or capture recapture models a number of developments have been made to under-  
 13617 stand animal movement patterns relative to habitat or dynamic systems (e.g., Jonsen et al.  
 13618 (2005); Hooten and Wikle (2010)). Similarly, if we have an indicator of habitat that varies  
 13619 by season, then in SCR models we can model the location of activity centers as a function  
 13620 of the change in habitat. There are a number of options for modeling variation in activity  
 13621 centers or animal locations as a function of covariates such as habitat, season, or behavior.  
 13622 Other approaches to analyzing movement in a mark-recapture framework include but are  
 13623 not limited to diffusion and auto-regressive models (Ovaskainen, 2004; Ovaskainen et al.,  
 13624 2008)), agent-based (Grimm et al., 2005; Hooten et al., 2010) and dispersal kernels (Fu-  
 13625 jiwara et al., 2006). For example, we define  $\mathbf{u}_{ikt}$  as the individual's observed location  
 13626 at secondary period  $k$  in primary period  $t$ . Then  $\mathbf{u}_{ikt} \sim \text{Normal}(\mathbf{s}(i, t), \Sigma_t)$  where  $\Sigma_t$  is  
 13627 the variance-covariance matrix at time  $t$ . This is a model we have used quite frequently  
 13628 throughout the book, i.e., that individual observed locations are assumed to follow a bi-  
 13629 variate normal distribution about the activity center,  $\mathbf{s}$ . This is similar to the Guassian  
 13630 and Laplace dispersal kernels. We could further allow the observed locations to follow an  
 13631 auto-regressive model such that  $\mathbf{u}_{ikt} \sim \text{Normal}(\rho(\mathbf{u}_{i,k,t-1} - \mathbf{s}(i, t - 1)), \Sigma_t^*)$ . These are  
 13632 just a few simple examples; as more information becomes available and data are collected  
 13633 over longer time periods, we will be able to use more complex movement models in open  
 13634 SCR models.

13635 **Cautionary note:**

13636 Using such Markovian models for the change in location of activity centers across  
 13637 primary periods, activity centers are no longer bound by the limits of the state-space.  
 13638 Imagine an individual living at the very edge of  $S$  at time  $t$  – there is some probability that,  
 13639 under the perturbation model, the location of  $\mathbf{s}$  at  $t+1$  could be outside  $S$ . When activity  
 13640 centers are no longer bound to the state-space, the way we have determined density (i.e.,  
 13641  $D = N/||S||$ ) no longer directly applies. This is not an issue in the CJS models where only  
 13642 survival and detection are of interest. But in the JS model, if individuals can move outside  
 13643 the state-space and remain alive, then the density must be recalculated such that we only  
 13644 count those individuals with activity centers *within* the state-space. This was previously  
 13645 not an issue because the prior on the activity centers constrained all individuals to the  
 13646 state-space at all times.

13647 **16.4.1 Thoughts on movement of American shad**

13648 In our American shad example above, we had reason to believe that individual movement  
 13649 is directly related to stream flow. When the stream flow is low, we might expect that  
 13650 the fish move very little, and when the stream flow is high, they might move upstream to  
 13651 spawn. In this case, we could model the effect of stream flow in two ways. First, we might  
 13652 allow  $\sigma$  to be a function of flow and to vary for each primary occasion, according to:

$$\log(\sigma_t) = \mu_\sigma + \alpha_2 \text{Flow}_t$$

13653 But if we think that the change in activity centers between primary periods might be  
 13654 related to the general pattern of fish migrating upstream more during high flow or staying  
 13655 closer to the same location in low flow, then we could allow the correlation in activity  
 13656 centers to be a function of flow. In this case, for example, a low flow period might indicate  
 13657 that activity centers are more correlated to the previous time period because fish are not  
 13658 actively migrating during such a time. This means that we assume the activity centers  
 13659 are correlated so we have

$$\mathbf{s}(i, t) \sim \text{Normal}(\mathbf{s}(i, t-1), \tau^2 \mathbf{I})$$

13660 where

$$\log(\tau) = \mu_\tau + \alpha_2 \text{Flow}_t.$$

13661 These are just a few thoughts on simple ways to model movement as a function of habitat  
 13662 variables which we have only started exploring on these data. As we discussed in the  
 13663 previous section, there are many other movement models that could be used.

13664 **16.4.2 Modeling dispersal**

13665 Dispersal is a well studied area in population ecology and is often of heightened interest  
 13666 because it relates directly to population regulation, habitat quality, and linking of local  
 13667 populations. However, studying dispersal with capture-recapture data can be difficult  
 13668 for a few reasons. One common issue with using capture-recapture data for dispersal  
 13669 estimation is that short distances are sampled more frequently than long distances. This  
 13670 is particularly true if we consider that most trap arrays are not large relative the potential

13671 dispersal distances of animals. In some cases, such as with small mammals, we may be able  
 13672 to capture both short and long distance dispersals in one trap array; in other cases, we may  
 13673 have discrete study sites set up across a larger area which capture individuals within and  
 13674 between sites. Either way, data are likely to be sparse for long distance dispersal events  
 13675 and this is particularly true if there are different habitat types which are sampled with  
 13676 different levels of effort (Ovaskainen et al., 2008), thus causing more difficulty in fitting  
 13677 models to data where much information is missing. In addition to that, determining if an  
 13678 individual has left an area or died can be difficult if the sampling does not cover the area  
 13679 an individual has moved to or if the sampling method has failed (e.g., a band or tag falls  
 13680 off or a mark is lost).

13681 Irregardless of these common sampling limitations, let's look at an optimal the situation  
 13682 where we have the trap array large enough to observe some dispersal events (or possibly  
 13683 multiple trap arrays on the landscape where an individual is observed in different arrays). We sketch out a possible dispersal model but note that this is a simple example.  
 13684 In this case, each individual could have some probability of dispersing, say  $\eta$  where  
 13685  $pd_{i,t} \sim \text{Bernoulli}(\eta)$  indicates if an individual disperses at time  $t$  and then

$$13687 \quad s_{i,t+1,1} = s_{i,t,1} + pd_{i,t}(ds_{i,t}\cos(\theta_{i,t})) \\ s_{i,t+1,2} = s_{i,t,2} + pd_{i,t}(ds_{i,t}\sin(\theta_{i,t}))$$

13688 where  $ds_i$  is the dispersal distance for individual  $i$  and  $\theta$  is the dispersal direction (in  
 13689 radians). Thus when  $pd_i = 0$ , then the activity centers remain the same as the previous  
 13690 time step and if  $pd_i = 1$  then the individual disperses to a new activity center. For this  
 13691 specification, we have to provide a model for dispersal distance. One option is to let  $ds_{i,t} \sim$   
 13692 exponential( $L$ ) where  $L$  is the mean dispersal distance for individuals dispersing and let  
 13693  $\theta_{i,t} \sim \text{Uniform}(-\pi, \pi)$  where  $\pi$  is not a parameter in this case, but the mathematical  
 13694 constant(i.e.,  $\pi = 3.14159\dots$ ). If all individuals are expected to move some distance  
 13695 between periods, then the  $pd$  indicator could be removed. A number of distributions  
 13696 exists for fitting these parameters (e.g., the von Mises is commonly used for angles) and  
 13697 more complex models with components like weighted directional movement and various  
 13698 movement states could be fit (see Jonsen et al. (2005); Johnson et al. (2008a); McClintock  
 13699 et al. (2012))

## 16.5 SUMMARY AND OUTLOOK

13700 In this chapter we have described a framework for making inference not only about spatial  
 13701 and temporal variation in population density, but also demographic parameters including  
 13702 survival, recruitment, and movement. The ability to model population vital rates is es-  
 13703 sential for ecology, management, and conservation; and the models described here allow  
 13704 researchers to examine the spatial and temporal dynamics governing those population  
 13705 parameters. While we have covered a lot of ground in this chapter, but there are many  
 13706 variations of the basic JS and CJS models, such as dead recovery models or models that  
 13707 address transiency that we have not explicitly 'converted' to a spatial framework, and  
 13708 these areas provide a broad field of further model development.

13709 As open models are further developed, mechanisms for dealing directly with dispersal  
 13710 and transients will provide improved inference frameworks for understanding movement  
 13711 as well as the potential to estimate *true* survival instead of only *apparent survival*. This

13712 is a function of explicitly modeling movement, which means we can separate movement  
13713 from mortality, as we sketched out in the model above for dispersal, providing a huge  
13714 advantage over traditional models. Also, models of individual dispersal can be used to  
13715 examine dynamics of population dynamics relative to habitat, density-dependence, or  
13716 climatic events.

13717 Birth and death processes, as well as movement, all have the potential to be related  
13718 to the space usage of animals in the landscape. Understanding the impact of spatially  
13719 varying density on survival and recruitment will provide insights into the basic ecology  
13720 of species. With the advent of non-invasive techniques, like camera trapping and genetic  
13721 analysis of tissue, we can start to understand the population dynamics of species that are  
13722 rarely observed in the wild. As more and more data are collected, we can use the models  
13723 to explore the spatio-temporal patterns of survival, recruitment, density, and movement  
13724 of species, providing incredibly useful biological and ecological information as we face  
13725 broad changes in climate, land-use, habitat fragmentation, etc. Rathbun and Cressie  
13726 (1994) articulate a model for marked point processes where they separate out the spatial  
13727 birth, growth, and survival processes for longleaf pine trees. Because of the application,  
13728 these demographic parameters are slightly different than how they are often considered in  
13729 wildlife and ecology, but still, there are analogies. Allowing birth, growth, and survival  
13730 as well as density to arise from different spatially varying processes is the next stage in  
13731 development of the open SCR models.

13732

## Part IV

13733

---

13734

# Super-Advanced SCR Models



13735  
13736

---

# 17

13737  
13738

## DEVELOPING MARKOV CHAIN MONTE CARLO SAMPLERS

13739 In this chapter we will dive a little deeper into Markov chain Monte Carlo (MCMC)  
13740 sampling. We will construct custom MCMC samplers in **R**, starting with easy-to-code  
13741 GLMs and GLMMs and moving on to simple CR and SCR models. This material might  
13742 seem slightly out of place here, as it does not deal with specific aspects or modifications of  
13743 SCR models, but rather, with a particular way of implementing them (and other models,  
13744 too). Knowing how to build an MCMC sampler is not essential for any of the SCR models  
13745 we have covered so far, but we will need these skills to implement some models that  
13746 come up in the last few chapters of this book. The aim of this chapter is to provide you  
13747 with some working knowledge of building MCMC samplers. To this end, we will NOT  
13748 provide exhaustive background information on the theory and justification of MCMC  
13749 sampling – there are entire books dedicated to that subject and we refer you to Robert  
13750 and Casella (2004) and Robert and Casella (2010). Rather we aim to provide you with  
13751 enough background and technical know-how to start building your own MCMC samplers  
13752 for SCR models in **R**. You will find that quite a few topics that come up in this chapter  
13753 have already been covered in previous chapters, particularly the introduction into Bayesian  
13754 analysis in Chapt. 3. To keep you from having to leaf back and forth we will in some  
13755 places briefly review aspects of Bayesian analysis, but we try to focus on the more technical  
13756 issues of building MCMC samplers relevant to SCR models.

### 17.1 WHY BUILD YOUR OWN MCMC ALGORITHM?

13757 The standard programs we have used so far to do MCMC analyses are **WinBUGS** (Gilks  
13758 et al., 1994) and **JAGS** (Plummer, 2003). The wonderful thing about these **BUGS**  
13759 engines is that they automatically use appropriate and, most of the time, reasonably  
13760 efficient forms of MCMC sampling for the model specified by the user.

13761 The fact that we have such a Swiss Army knife type of MCMC machine begs the  
13762 question: Why would anyone want to build their own MCMC algorithm? For one, there

13763 are a limited number of distributions and functions implemented in **BUGS**. While **Open-**  
 13764 **BUGS** and **JAGS** provide more options, some more complex models may be impossible  
 13765 to build within these programs. A very simple example from spatial capture-recapture  
 13766 that can give you a headache in **WinBUGS** is when your state-space is an irregular-  
 13767 shaped polygon, rather than an ideal rectangle that can be characterized by four pairs  
 13768 of coordinates. It is easy to restrict activity centers to any arbitrary polygon in **R** using  
 13769 an ESRI shapefile (and we will show you an example in a little bit), but you cannot use  
 13770 a shapefile in a **BUGS** model. Similarly, models of space usage that take into account  
 13771 ecological distance (Chapt. 12) cannot be implemented in the **BUGS** engines.

13772 Sometimes, implementing an MCMC algorithm in **R** may be faster than in **Win-**  
 13773 **BUGS** - especially if you want to run simulation studies where you have hundreds or  
 13774 more simulated data sets, several years' worth of data or other large models, this can be  
 13775 a big advantage. Further, writing your own sampler gives you more control over which  
 13776 kind of updater is used (see following sections). Finally, building your own MCMC al-  
 13777 gorithm is a great exercise to understand how MCMC sampling works. So while using  
 13778 the **BUGS** language requires you to understand the structure of your model, building an  
 13779 MCMC algorithm requires you to think about the relationship between your data, priors  
 13780 and posteriors, and how these can be efficiently analyzed and characterized. However,  
 13781 if you don't think you will ever sit down and write your own MCMC sampler, consider  
 13782 skipping this chapter - apart from coding it will not cover anything SCR-related that is  
 13783 not covered by other, more model-oriented chapters as well.

## 17.2 MCMC AND POSTERIOR DISTRIBUTIONS

13784 MCMC is a class of simulation methods for drawing (correlated) random numbers from  
 13785 a target distribution, which in Bayesian inference is the posterior distribution. As a re-  
 13786 minder, the posterior distribution is a probability distribution for an unknown parameter,  
 13787 say  $\theta$ , given observed data and its prior probability distribution (the probability distribu-  
 13788 tion we assign to a parameter before we observe data). The great benefit of having the  
 13789 posterior distribution of  $\theta$  is that it can be used to make probability statements about  
 13790  $\theta$ , such as the probability that  $\theta$  is equal to some value, or the probability that  $\theta$  falls  
 13791 within some range of values. The posterior distribution summarizes all we know about a  
 13792 parameter and thus, is the central object of interest in Bayesian analysis. Unfortunately,  
 13793 in many if not most practical applications, it is nearly impossible to directly compute the  
 13794 posterior. Recall Bayes' theorem:

$$[\theta|y] = \frac{[y|\theta][\theta]}{[y]}, \quad (17.2.1)$$

13795 where  $\theta$  is the parameter of interest,  $y$  is the observed data,  $[\theta|y]$  is the posterior,  $[y|\theta]$  the  
 13796 likelihood of the data conditional on  $\theta$ ,  $[\theta]$  the prior probability of  $\theta$ , and, finally,  $[y]$  is the  
 13797 marginal probability of the data, defined as

$$[y] = \int [y|\theta][\theta]d\theta$$

13798 This marginal probability is a normalizing constant that ensures that the posterior  
 13799 integrates to 1. Often, the integral is difficult or impossible to evaluate, unless you are

13800 dealing with a really simple model. For example, consider a normal model, with a set of  
 13801  $n$  observations,  $y_i; i = 1, 2, \dots, n$ :

$$y_i \sim \text{Normal}(\mu, \sigma),$$

13802 where  $\sigma$  is known and our objective is to estimate  $\mu$ . To fully specify the model in a  
 13803 Bayesian framework, we first have to define a prior distribution for  $\mu$ . Recall from Chapt.  
 13804 3 that for certain data models, certain priors lead to conjugacy, i.e. if you choose a certain  
 13805 prior for your parameter, the posterior distribution will be of a known parametric form.  
 13806 More specifically, under conjugacy, the prior and posterior distributions are from the same  
 13807 parametric family. The conjugate prior for the mean of a normal model is also a normal  
 13808 distribution:

$$\mu \sim \text{Normal}(\mu_0, \sigma_0^2).$$

13809 If  $\mu_0$  and  $\sigma_0^2$  are fixed, the posterior for  $\mu$  has the following form (for some of the algebra  
 13810 behind this, see Chapt. 2 in Gelman et al. (2004)):

$$\mu|y \sim \text{Normal}(\mu_n, \sigma_n^2) \quad (17.2.2)$$

13811 where

$$\mu_n = \left( \frac{\sigma^2}{\sigma^2 + n\sigma_0^2} \right) \times \left( \mu_0 + \frac{n\sigma_0^2}{\sigma^2 + n\sigma_0^2} \right) \times \bar{y}$$

13812 and

$$\sigma_n^2 = \frac{\sigma^2 \sigma_0^2}{\sigma^2 + n\sigma_0^2}.$$

13813 We can directly obtain estimates of interest from this normal posterior distribution, such  
 13814 as its mean  $\hat{\mu}$  (which is equivalent to an estimate of  $\mu_n$ ) and variance; we do not need  
 13815 to apply MCMC, since we can recognize the posterior as a parametric distribution, in-  
 13816 cluding the normalizing constant  $[y]$ . But generally we will be interested in more complex  
 13817 models with several, say  $m$ , parameters. In this case, computing  $[y]$  from Eq. 17.2.1 re-  
 13818 quires  $m$ -dimensional integration, which can be difficult or impossible. Thus, the posterior  
 13819 distribution is generally only known up to a constant of proportionality:

$$[\theta|y] \propto [y|\theta][\theta]$$

13820 The power of MCMC is that it allows us to approximate the posterior using simulation  
 13821 without evaluating the high dimensional integrals, and to directly sample from the pos-  
 13822 terior, even when the posterior distribution is unknown! The price is that MCMC is  
 13823 computationally expensive. Although MCMC first appeared in the scientific literature in  
 13824 1949 (Metropolis and Ulam, 1949), widespread use did not occur until the 1980s when  
 13825 computational power and speed increased (Gelfand and Smith, 1990). It is safe to say that  
 13826 the advent of practical MCMC methods is the primary reason why Bayesian inference has  
 13827 become so popular during the past three decades.

13828 In a nutshell, MCMC lets us generate sequential draws of  $\theta$  (the parameter(s) of in-  
 13829 terest) from distributions approximating the unknown posterior over  $T$  iterations. The  
 13830 distribution of the draw at  $t$  depends on the value drawn at  $t-1$ ; hence, the draws from a  
 13831 Markov chain<sup>1</sup>. As  $T$  goes to infinity, the Markov chain converges to the desired distri-  
 13832 bution, in our case the posterior distribution for  $\theta|y$ . Thus, once the Markov chain has

<sup>1</sup>Remember that for  $T$  random samples  $\theta^{(1)}, \dots, \theta^{(T)}$  from a Markov chain the distribution of  $\theta^{(t)}$  depends only on the immediately preceding value,  $\theta^{(t-1)}$ .

reached its stationary distribution, the generated samples can be used to characterize the posterior distribution,  $[\theta|y]$ , and point estimates of  $\theta$ , its standard error and confidence bounds, can be obtained directly from this approximation of the posterior.

### 17.3 TYPES OF MCMC SAMPLING

There are several general MCMC algorithms in widespread use, the most popular being Gibbs sampling and Metropolis-Hastings sampling, both of which were briefly introduced in Chapt. 3. We will be dealing with these two classes in more detail and use them to construct MCMC algorithms for SCR models. Also, we will briefly review alternative techniques that are applicable in some situations.

#### 17.3.1 Gibbs sampling

Gibbs sampling was named after the physicist J.W. Gibbs by Geman and Geman (1984), who applied the algorithm to a Gibbs distribution<sup>2</sup>. The roots of Gibbs sampling can be traced back to work of Metropolis et al. (1953), and it is actually closely related to Metropolis sampling (see Chapt. 11.5 in Gelman et al. (2004), for the link between the two samplers). We will focus on the technical aspects of this algorithm, but if you find yourself hungry for more background, Casella and George (1992) provide a more in-depth introduction to the Gibbs sampler.

Let's go back to our simple example from above to understand the motivation and functioning of Gibbs sampling. Recall that for a normal model with known variance and a normal prior for  $\mu$ , the posterior distribution of  $\mu|y$  is also normal. Conversely, with a fixed (known)  $\mu$ , but unknown variance, the conjugate prior for  $\sigma^2$  is an inverse-gamma distribution with shape and scale parameters  $a$  and  $b$ :

$$\sigma^2 \sim \text{Inverse-Gamma}(a, b).$$

With fixed  $a$  and  $b$ , algebra reveals that the posterior  $[\sigma^2|\mu, y]$  is also an inverse-gamma distribution, namely:

$$\sigma^2|\mu, y \sim \text{Inverse-Gamma}(a_n, b_n), \quad (17.3.1)$$

where  $a_n = n/2 + a$  and  $b_n = (1/2) \sum_{i=1}^n (y_i - \mu)^2 + b$ . However, what if we know neither  $\mu$  nor  $\sigma^2$ , which is probably the more common case? The joint posterior distribution of  $\mu$  and  $\sigma^2$  now has the general structure

$$[\mu, \sigma^2|y] = \frac{[y|\mu, \sigma^2][\mu][\sigma^2]}{\int [y|\mu][\mu][\sigma^2]d\mu d\sigma^2}$$

or

$$[\mu, \sigma^2|y] \propto [y|\mu, \sigma^2][\mu][\sigma^2]$$

---

<sup>2</sup>a distribution from physics we are not going to worry about, since it has no immediate connection with Gibbs sampling other than giving its name

13860 This cannot easily be reduced to a distribution we recognize. However, we can con-  
 13861 dition  $\mu$  on  $\sigma^2$  (i.e., we treat  $\sigma^2$  as fixed) and remove all terms from the joint posterior  
 13862 distribution that do not involve  $\mu$  to construct the full conditional distribution,

$$[\mu|\sigma^2, y] \propto [y|\mu][\mu]$$

13863 The full conditional of  $\mu$  again takes the form of the normal distribution shown in Eq.  
 13864 17.2.2; similarly,  $[\sigma^2|\mu, y]$  takes the form of the inverse-gamma distribution shown in Eq.  
 13865 17.3.1, both distributions we can easily sample from. And this is precisely what we do  
 13866 when using Gibbs sampling: we break down high-dimensional problems into convenient  
 13867 one-dimensional problems by constructing the full conditional distributions for each model  
 13868 parameter separately; and we sample from these full conditionals, which, if we choose  
 13869 conjugate priors, are known parametric distributions. Let's put the concept of Gibbs  
 13870 sampling into the MCMC framework of generating successive samples, using our simple  
 13871 normal model with unknown  $\mu$  and  $\sigma^2$  and conjugate priors as an example. These are the  
 13872 steps you need in order to build a Gibbs sampler:

13873 **Step 0:** Begin with some initial values for  $\theta$ , say  $\theta^{(0)}$ . In our example,  $\theta = (\mu, \sigma)$ , so  
 13874 we have to specify initial values for  $\mu$  and  $\sigma$ , for example by drawing a random number  
 13875 from some uniform distribution, or by setting them close to what we think they might be.  
 13876 (Note: This step is required in any MCMC sampling; chains have to start from somewhere.  
 13877 We will get back to these technical details a little later.)

13878 **Step 1:** For iteration  $t$ , Draw  $\theta^{(t)}$  from the conditional distribution  $[\theta_1^{(t)}|\theta_2^{(t-1)}, \dots, \theta_d^{(t-1)}]$ .  
 13879 Here,  $\theta_1$  is  $\mu$ , which we draw from the normal distribution in Eq. 17.2.2 using  $\sigma^{(t-1)}$  as  
 13880 value for  $\sigma$ .

13881 **Step 2:** Draw  $\theta_2^{(t)}$  from the conditional distribution  $[\theta_2^{(t)}|\theta_1^{(t)}, \theta_3^{(t-1)}, \dots, \theta_d^{(t-1)}]$ . Here,  $\theta_2$   
 13882 is  $\sigma$ , which we draw from the inverse-gamma distribution of Eq. 17.3.1, using the newly  
 13883 generated  $\mu^{(t)}$  as value for  $\mu$ .

13884 **Step 3, ..., d:** Draw  $\theta_3^{(t)}, \theta_4^{(t)}, \dots, \theta_d^{(t)}$  from their conditional distribution  $[\theta_3^{(t)}|\theta_1^{(t)}, \theta_2^{(t)}, \theta_4^{(t-1)},$   
 13885  $\dots, \theta_d^{(t-1)}], \dots, [\theta_d^{(t)}|\theta_1^{(t)}, \dots, \theta_{d-1}^{(t)}]$ . In our example we have no additional parameters,  
 13886 so we only need step 0 through to 2.

13887 **Repeat Steps 1 to d for  $T =$**  a large number of samples.

13888 Note that the order in which we update the parameters within the Gibbs algorithm  
 13889 does not matter. In terms of **R** coding, this means we have to write Gibbs updaters for  
 13890  $\mu$  and  $\sigma^2$  and embed them into a loop over  $T$  iterations. The final code in the form of an  
 13891 **R** function is shown in Panel 17.1.

13892 This is it! You can go ahead and simulate some data,  $y \sim \text{Normal}(5, 0.5)$  and then  
 13893 use the function **NormGibbs()** in the **R** package **scrbook** to run your first Gibbs sampler  
 13894 (note that the **R** function **rnorm** requires you to supply the standard deviation  $\sigma$  and we  
 13895 have written **NormGibbs** so that it returns  $\sigma$  instead of  $\sigma^2$  so you can easily compare your  
 13896 input value and parameter estimate).

```
13897 > set.seed(13)
13898
```

---

```
Norm.Gibbs<-function(y=,mu_0=mu_0,sigma2_0=sigma2_0,a=a,b=b,niter=niter){

ybar<-mean(y)
n<-length(y)
mu<-1           #mean initial value
sigma2<-1        #sigma2 initial value
an<-n/2 + a      #shape parameter of InvGamma of sigma2
out<-matrix(nrow=niter, ncol=2)
colnames(out)<-c('mu', 'sig')

for (i in 1:niter) {

#update mu
mu_n<-((sigma2/(sigma2+n*sigma2_0))*mu_0
+ (n*sigma2_0/(sigma2 + n*sigma2_0))*ybar)
sigma2_n <- (sigma2*sigma2_0)/ (sigma2 + n*sigma2_0)
mu<-rnorm(1,mu_n, sqrt(sigma2_n))

#update sigma2
bn<- 0.5 * (sum((y-mu)^2)) + b
sigma2<-1/rgamma(1,shape=an, rate=bn)
out[i,]<-c(mu,sqrt(sigma2))
}
return(out)
}
```

---

Panel 17.1: R-code for a Gibbs sampler for a normal model with unknown  $\mu$  and  $\sigma$  and conjugate priors (normal and inverse-gamma, respectively) for both parameters.

```

13899 #true mean and sd are 5 and 0.5
13900 > y<-rnorm(1000, 5,0.5) #data
13901
13902 > mu_0<-0 #prior mean
13903 > sigma2_0<-100 #prior variance
13904
13905 #inverse-gamma hyperparameters
13906 > a<-0.1
13907 > b<-0.1
13908
13909 > mod=Norm.Gibbs(y, mu_0, sigma2_0, a,b,niter=10000)

```

13910 Your output, `mod`, will be a table with two columns, one per parameter, and  $T$  rows,  
 13911 one per iteration. For this 2-parameter example you can visualize the joint posterior by  
 13912 plotting samples of  $\mu$  against samples of  $\sigma$  (Fig. 17.1):

```
13913 > plot(out[,1], out[,2])
```

13914 The marginal distribution of each parameter is approximated by examining the samples  
 13915 of this particular parameter. You can visualize it by plotting a histogram of the samples  
 13916 (Fig. 17.2 upper left and right):

```

13917 > par(mfrow=c(1,2))
13918 > hist(out[,1]); hist(out[,2])

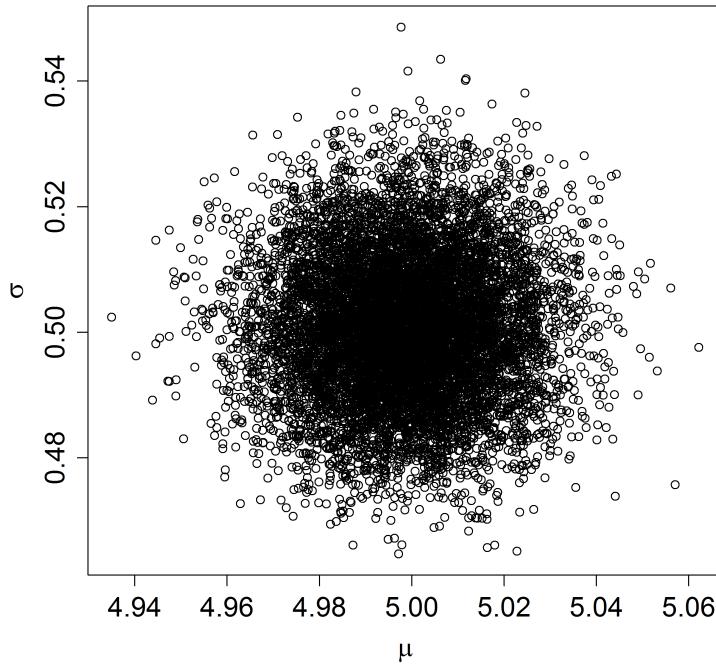
```

13919 Finally, recall an important characteristic of MCMC, namely, that the chain has to  
 13920 have converged (reached its stationary distribution) in order to regard samples as coming  
 13921 from the posterior distribution. In practice, that means you have to throw out some  
 13922 of the initial samples called the burn-in. We will talk about this in more detail when  
 13923 we talk about convergence diagnostics. For now, you can use the `plot(out[,1])` or  
 13924 `plot(out[,2])` command to make a time series plot of the samples of each parameter and  
 13925 visually assess how many of the initial samples you should discard. Fig. 17.2 bottom left  
 13926 and right shows plots for the samples of  $\mu$  and  $\sigma$  from our simulated data set; you see that  
 13927 in this simple example the Markov chain apparently reaches its stationary distribution  
 13928 very quickly – the chains look ‘grassy’ seemingly from the start. It is hard to discern a  
 13929 burn-in phase visually (but we will see examples further on where the burn-in is clearer)  
 13930 and you may just discard the first 500 draws to be sure you only use samples from the  
 13931 posterior distribution. The mean of the remaining samples are your estimates of  $\mu$  and  $\sigma$ :

```

13932 > summary(mod[501:10000,])
13933      mu          sig
13934 Min.   :4.935  Min.   :0.4652
13935 1st Qu.:4.988  1st Qu.:0.4930
13936 Median :4.998  Median :0.5006
13937 Mean   :4.998  Mean   :0.5008
13938 3rd Qu.:5.009  3rd Qu.:0.5084
13939 Max.   :5.062  Max.   :0.5486

```



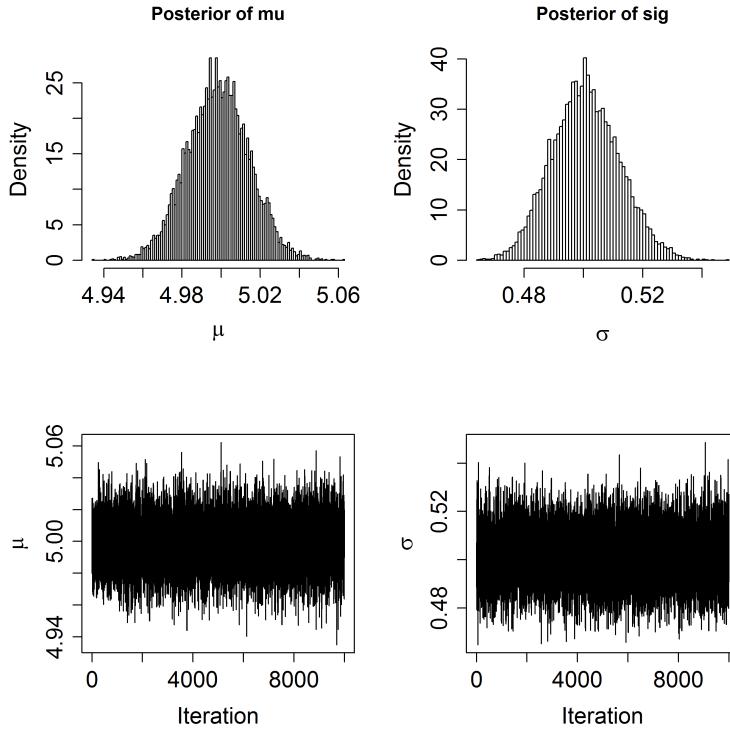
**Figure 17.1.** Joint posterior distribution of  $\mu$  and  $\sigma$  from a normal Model

### 13940 17.3.2 Metropolis-Hastings sampling

13941 Although it is applicable to a wide range of problems, the limitations of Gibbs sampling  
 13942 are obvious: what if we do not want to use conjugate priors or what if we cannot recognize  
 13943 the full conditional distribution as a parametric distribution, or simply do not want to  
 13944 worry about these issues? The most general solution is to use the Metropolis-Hastings  
 13945 (MH) algorithm, which also goes back to the work by Metropolis et al. (1953). You saw  
 13946 the basics of this algorithm in Chapt. 3. In a nutshell, because we do not recognize the  
 13947 posterior  $[\theta|y]$  as a parametric distribution, the MH algorithm generates samples from a  
 13948 known proposal distribution, say  $h(\theta)$ , that depends on the value of  $\theta$  at the previous time  
 13949 step,  $\theta^{(t-1)}$ . The candidate value  $\theta^*$  is accepted with probability

$$r = \min\left(1, \frac{[\theta^*|y]h(\theta^{(t-1)}|\theta^*)}{[\theta^{(t-1)}|y]h(\theta^*|\theta^{(t-1)})}\right)$$

13950 Proposal distributions must be chosen so that reversibility is ensured. That means,



**Figure 17.2.** Plots of the posterior distributions of  $\mu$  (upper left) and  $\sigma$  (upper right) from a normal model and time series plots of  $\mu$  (lower left) and  $\sigma$  (lower right).

it must be possibly to go from any one value to any other. But within that criterion the proposal distribution can be absolutely anything! You can generate candidate values from a Normal(0,1) distribution, from a Uniform(-3455,3455) distribution, or anything of proper support. Note, however, that good choices of  $h(\theta)$  are those that approximate the posterior distribution. Obviously if  $h(\theta) = [\theta|y]$  (i.e., the posterior) then you always accept the draw and it stands to reason that proposals that are more similar to  $[\theta|y]$  will lead to higher acceptance probabilities. Actually, when  $h(\theta) = [\theta|y]$  we can draw samples of  $\theta$  directly from  $h(\theta)$ , which brings us back to Gibbs sampling. Thus, Gibbs sampling is a special case of Metropolis-Hastings sampling.

The original Metropolis algorithm required  $h(\theta)$  to be symmetric so that

$$h(\theta^*|\theta^{(t-1)}) = h(\theta^{(t-1)}|\theta^*)$$

In that case these two terms just cancel out from the MH acceptance probability and  $r$  is then just the ratio of the target density evaluated at the candidate value to that evaluated

13963 at the current value. A later development of the algorithm by Hastings (1970) lifted this  
 13964 condition. Since using a symmetric proposal distribution makes life a little easier, we are  
 13965 going to focus on this specific case. A type of symmetric proposal useful in many situations  
 13966 is the so-called *random-walk* proposal distribution where candidate values are drawn from  
 13967 a normal distribution with mean equal to the current value and some standard deviation,  
 13968 say  $\delta$ , which is prescribed by the user (see below for further explanation).

13969 **Parameters with bounded support:** Many models contain parameters that have  
 13970 bounded support. E.g., variance parameters live on  $[0, \infty]$ , parameters that represent  
 13971 probabilities live on  $[0, 1]$ , etc.. For such cases, it is sometimes convenient to use a random  
 13972 walk proposal distribution that can generate any real number (e.g., a normal random walk  
 13973 proposal). Under these circumstances you should not constrain the proposal distribution  
 13974 itself, but you can just reject parameters that are outside of the parameter space (sec. 6.4.1  
 13975 in Robert and Casella, 2010). You will see plenty of examples of updating parameters with  
 13976 bounded support in this chapter.

13977 It is worth knowing that there are alternatives to the random walk MH algorithm.  
 13978 For example, in the independent MH, the proposal distribution  $h$  does not depend on  
 13979  $\theta^{(t-1)}$ , while the Langevin algorithm (Roberts and Rosenthal, 1998) aims at avoiding the  
 13980 random walk by favoring moves towards regions of higher posterior probability density.  
 13981 The interested reader should look up these algorithms in Robert and Casella (2004) or  
 13982 Robert and Casella (2010).

13983 Building a MH sampler can be broken down into several steps. We are going to  
 13984 demonstrate these steps using a different but still simple and common model: the logit-  
 13985 normal or logistic regression model. For simplicity, assume that

$$y|\theta \sim \text{Bernoulli} \left( \frac{\exp(\theta)}{1 + \exp(\theta)} \right)$$

13986 and

$$\theta \sim \text{Normal}(\mu, \sigma).$$

13987 The following steps are required to set up a random walk MH algorithm:

13988 **Step 0:** Choose initial values,  $\theta^{(0)}$ .

13989 **Step 1:** Generate a proposed value of  $\theta$  from  $h(\theta^*|\theta^{(t-1)})$ . We will use the random walk  
 13990 MH algorithm, so we draw  $\theta^*$  from  $\text{Normal}(\theta^{(t-1)}, \delta)$ , where  $\delta$  is the standard deviation  
 13991 of the normal proposal distribution, the tuning parameter that we have to set.

13992 **Step 2:** Calculate the ratio of posterior densities for the proposed and the original value  
 13993 for  $\theta$ :

$$r = \frac{[\theta^*|y]}{[\theta^{(t-1)}|y]}.$$

13994 In our example,

$$r = \frac{\text{Bernoulli}(y|\theta^*) \times \text{Normal}(\theta^*|\mu, \sigma)}{\text{Bernoulli}(y|\theta^{(t-1)}) \times \text{Normal}(\theta^{(t-1)}|\mu, \sigma)}$$

13995 **Step 3:** Set

$$\begin{aligned} \theta^t &= \theta^* \text{ with probability } \min(r, 1) \\ &= \theta^{(t-1)} \text{ otherwise} \end{aligned}$$

13996 We can do this last step by drawing a random number  $u$  from a Uniform(0, 1) and  
 13997 accept  $\theta^*$  if  $u < r$ . This is repeated for  $t = 1, 2, \dots, T$  a large number of samples. As for  
 13998 Gibbs sampling, the order in which we update parameters does not matter. The **R** code  
 13999 for this MH sampler is provided in Panel 17.2.

---

```
Logreg.MH<-function(y=y, mu0=mu0, sig0=sig0, delta=delta, niter=niter) {
  out<-c()
  theta<-runif(1, -3,3) #initial value
  for (iter in 1:niter){
    theta.cand<-rnorm(1, theta, delta)
    loglike<-sum(dbinom(y, 1, exp(theta)/(1+exp(theta)), log=TRUE))
    logprior <- dnorm(theta,mu0 ,sig0, log=TRUE)
    loglike.cand<-sum(dbinom(y, 1, exp(theta.cand)/(1+exp(theta.cand)),
      log=TRUE))
    logprior.cand <- dnorm(theta.cand, mu0, sig0, log=TRUE)
    if (runif(1)<exp((loglike.cand+logprior.cand)-(loglike+logprior))){
      theta<-theta.cand
    }
    out[iter]<-theta
  }
  return(out)
}
```

---

Panel 17.2: **R** code to run a Metropolis sampler on a simple logit-normal model.

14000 The reason why in the **R** code we sum the logs of the likelihood and the prior, rather  
 14001 than multiplying the original values, is simply computational. The product of small prob-  
 14002 abilities can be numbers very close to 0, which computers do not handle well. Thus we  
 14003 add the logarithms, sum, and exponentiate to achieve the desired result. Similarly, in  
 14004 case you have forgotten,  $x/y = \exp(\log(x) - \log(y))$ , with the latter being favored for  
 14005 computational reasons.

14006 Comparing MH sampling to Gibbs sampling, where all draws from the conditional  
 14007 distribution are used, in the MH algorithm we discard a portion of the candidate values,  
 14008 which inherently makes it less efficient than Gibbs sampling – the price you pay for its  
 14009 increased generality. In Step 1 of the MH sampler we had to choose a variance,  $\delta$ , for  
 14010 the normal proposal distribution. Choice of the parameters that define our candidate

distribution is also referred to as 'tuning', and it is important since adequate tuning will make your algorithm more efficient.  $\delta$  should be chosen (a) large enough so that each step of drawing a new proposal value for  $\theta$  can cover a reasonable distance in the parameter space, as otherwise, mixing of the Markov chain is inefficient and chains will tend to have strong autocorrelation; and (b) small enough so that proposal values are not rejected too often, as otherwise the random walk will 'get stuck' at specific values for too long. As a rule of thumb, your candidate value should be accepted in about 40% of all cases. Acceptance rates of 20 – 80% are probably ok, but anything below or above may well render your algorithm inefficient (this does not mean that it will give you wrong results, only that you will need more iterations to converge to the posterior distribution). In practice, tuning will require some 'trial-and-error', some common sense and, with enough experience, some intuition. Or, one can use an adaptive phase, where the tuning parameter is automatically adjusted until it reaches a user-defined acceptance rate, at which point the adaptive phase ends and the actual Markov chain begins. This is computationally a little more advanced. Link and Barker (2010) discuss this in more detail. It is important that the samples drawn during the adaptive phase are discarded.

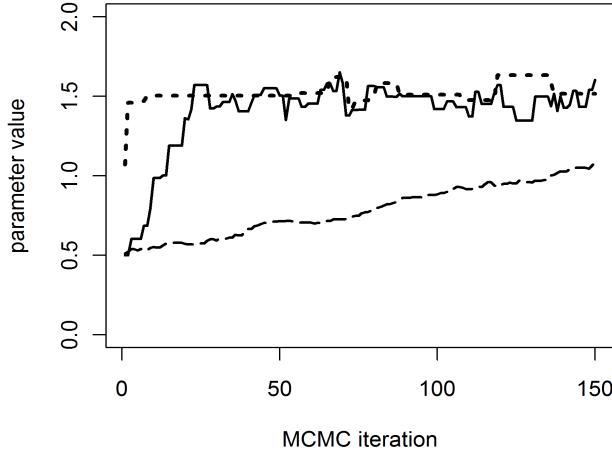
To illustrate the effects of tuning, we ran the Metropolis-Hastings algorithm in Panel 17.2 with  $\delta = 0.01$ ,  $\delta = 0.2$  and  $\delta = 1$ . The first 150 iterations for  $\theta$  are shown in Fig. 17.3. We see that for a very small  $\delta$  (the dashed line) the burn-in is extremely slow - after 150 iterations the chain isn't even half way there, while for the other two values of  $\delta$  (solid and dotted) the burn-in phase seems to be over after only about 10 iterations. While  $\delta = 0.2$  leads to reasonably good mixing, the chain clearly gets stuck on certain values with  $\delta = 1$ .

Other than graphically, you can easily check acceptance rates for the parameters you monitor (that are part of your output) using the `rejectionRate()` function of the package `coda` (we will talk more about this package a little later on). Do not let the term 'rejection rate' confuse you; it is simply  $1 - \text{acceptance rate}$ . There may be parameters – for example, individual values of a random effect or latent variables – that you do not want to save, though, and in our next example we will show you a way to monitor their acceptance rates with a few extra lines of code.

### 17.3.3 Metropolis-within-Gibbs

One weakness of the MH sampler is that formulating the joint posterior when evaluating whether to accept or reject the candidate values for  $\theta$  becomes increasingly complex or inefficient as the number of parameters in a model increases. As you already saw in Chapt. 3, in these cases you can simply combine MH sampling and Gibbs sampling. You can use the principles of Gibbs sampling to break down your high-dimensional parameter space into easy-to-handle one-dimensional conditional distributions and use MH sampling for these conditional distributions. Better yet, if you have some conjugacy in your model, you can use the more efficient Gibbs sampling for these parameters and one-dimensional MH for all the others. You have already seen the basics of how to build both types of algorithms, so we can jump straight into an example here and build a Metropolis-within-Gibbs algorithm.

**GLMMs: Poisson regression with a random effect** Let's assume a model that gets us closer to the problem we ultimately want to deal with - a GLMM. Here, we assume



**Figure 17.3.** Time series plots of  $\theta$  from a MH algorithm with tuning parameter  $\delta = 0.01$  (dashed line), 0.2 (solid line) and 1 (dotted line).

14055 we have Poisson counts,  $y_{ij}$ , from  $j = 1, 2, \dots, n$  plots in  $i$  different study sites, and we  
 14056 believe that the counts are influenced by some plot-specific covariate,  $\mathbf{x}$ , but that there is  
 14057 also a random site effect. So our model is:

$$\begin{aligned} 14058 \quad y_{ij} &\sim \text{Poisson}(\lambda_{ij}) \\ \lambda_{ij} &= \exp(\alpha_i + \beta x_{ij}) \end{aligned}$$

14059 Let's place normal priors on  $\alpha$  and  $\beta$ ,

$$\alpha_i \sim \text{Normal}(\mu_\alpha, \sigma_\alpha)$$

14060 and

$$\beta \sim \text{Normal}(\mu_\beta, \sigma_\beta)$$

14061 In this model, we do not specify  $\mu_\alpha$  and  $\sigma_\alpha$ , but instead, estimate them as well, so we  
 14062 have to specify hyperpriors for these parameters:

$$\begin{aligned} \mu_\alpha &\sim \text{Normal}(\mu_0, \sigma_0) \\ \sigma_\alpha^2 &\sim \text{Inverse-Gamma}(a_0, b_0) \end{aligned}$$

14063 Note that for simplicity we assume that  $\beta$  is constant across the  $i$  study sites, and for  
 14064 analysis we set  $\mu_\beta$  and  $\sigma_\beta$  (i.e., we don't estimate these parameters from the data). With

14065 the model completely specified, we can compile the full conditionals, breaking the multi-dimensional parameter space into one-dimensional components:  
 14066

$$\begin{aligned} [\alpha_1 | \alpha_2, \alpha_3, \dots, \alpha_i, \beta, \mathbf{y}_1] &\propto [\mathbf{y}_1 | \alpha_1, \beta][\alpha_1] \\ &\propto \text{Poisson}(\mathbf{y}_1 | \exp(\alpha_1 + \beta \mathbf{x}_1)) \times \text{Normal}(\alpha_1 | \mu_\alpha, \sigma_\alpha), \end{aligned}$$

14067 where  $\mathbf{y}_1 = (y_{11}, y_{12}, \dots, y_{1n})$  is the vector of observed counts for site  $i = 1$  and, in general,  
 14068  $\mathbf{y}_i$  is the vector of all counts for site  $i$ ; analogous,  $\mathbf{x}_i$  is the vector of all observations of  
 14069 the covariate for site  $i$ . The other full conditionals for each  $\alpha_i$  are constructed similarly:

$$\begin{aligned} [\alpha_2 | \alpha_1, \alpha_3, \dots, \alpha_i, \beta, \mathbf{y}_2] &\propto [\mathbf{y}_2 | \alpha_2, \beta][\alpha_2] \\ &\propto \text{Poisson}(\mathbf{y}_2 | \exp(\alpha_2 + \beta \mathbf{x}_2)) \times \text{Normal}(\alpha_2 | \mu_\alpha, \sigma_\alpha), \end{aligned}$$

14070 and so on for all elements of  $\alpha$ . The full-conditional for  $\beta$  is:

$$\begin{aligned} [\beta | \boldsymbol{\alpha}, \mathbf{y}] &\propto [\mathbf{y} | \boldsymbol{\alpha}, \beta][\beta] \\ &\propto \text{Poisson}(\mathbf{y} | \exp(\boldsymbol{\alpha} + \beta \mathbf{x})) \times \text{Normal}(\beta | \mu_\beta, \sigma_\beta). \end{aligned}$$

14071 Finally, we need to update the hyperparameters for the random effects vector  $\alpha$ :

$$\begin{aligned} [\mu_\alpha | \boldsymbol{\alpha}] &\propto [\boldsymbol{\alpha} | \mu_\alpha, \sigma_\alpha][\mu_\alpha] \\ [\sigma_\alpha | \boldsymbol{\alpha}] &\propto [\boldsymbol{\alpha} | \mu_\alpha, \sigma_\alpha][\sigma_\alpha] \end{aligned}$$

14072 Note that the likelihood contributions of the counts  $\mathbf{y}$  at each site, when conditioned  
 14073 on  $\boldsymbol{\alpha}$ , do not depend on the hyperparameters  $\mu_\alpha$  and  $\sigma_\alpha$ . As such, the full conditionals  
 14074 for these hyperparameters only depend on the collection of all  $\boldsymbol{\alpha}$ , not the data. Since we  
 14075 assumed  $\boldsymbol{\alpha}$  to come from a normal distribution, the choice of priors for  $\mu_\alpha$  (normal) and  
 14076  $\sigma_\alpha^2$  (inverse-gamma) leads to the same conjugacy we observed in our initial normal model,  
 14077 so that both hyperparameters can be updated using Gibbs sampling.  
 14078

14079 Now let's build the updating steps for these full conditionals. Again, for the MH steps  
 14080 that update  $\boldsymbol{\alpha}$  and  $\beta$  we use normal proposal distributions with standard deviations  $\delta_\alpha$   
 14081 and  $\delta_\beta$ .

14082 First, we set the initial values  $\boldsymbol{\alpha}^{(0)}$  and  $\beta^{(0)}$ . Then, starting with  $\alpha_1$ , we draw  $\alpha_1^{(1)}$   
 14083 from  $\text{Normal}(\alpha_1^{(0)}, \delta_\alpha)$ , calculate the conditional posterior density of  $\alpha_1^{(0)}$  and  $\alpha_1^{(1)}$  and  
 14084 compare their ratios,

$$r = \frac{\text{Poisson}(\mathbf{y}_1 | \exp(\alpha_1^{(1)} + \beta \mathbf{x}_1)) \times \text{Normal}(\alpha_1^{(1)} | \mu_\alpha, \sigma_\alpha)}{\text{Poisson}(\mathbf{y}_1 | \exp(\alpha_1^{(0)} + \beta \mathbf{x}_1)) \times \text{Normal}(\alpha_1^{(0)} | \mu_\alpha, \sigma_\alpha)}$$

14085 and accept  $\alpha_1^{(1)}$  with probability  $\min(r, 1)$ . We repeat this for all  $\boldsymbol{\alpha}$ .

14086 For  $\beta$ , we draw  $\beta^{(1)}$  from  $\text{Normal}(\beta^{(0)}, \delta_\beta)$ , compare the posterior densities of  $\beta^{(0)}$  and  
 14087  $\beta^{(1)}$ ,

$$r = \frac{\text{Poisson}(\mathbf{y} | \exp(\boldsymbol{\alpha} + \beta^{(1)} \mathbf{x})) \times \text{Normal}(\beta^{(1)} | \mu_\beta, \sigma_\beta)}{\text{Poisson}(\mathbf{y} | \exp(\boldsymbol{\alpha} + \beta^{(0)} \mathbf{x})) \times \text{Normal}(\beta^{(0)} | \mu_\beta, \sigma_\beta)},$$

14088 and accept  $\beta^{(1)}$  with probability  $\min(r, 1)$ .

14089 For  $\mu_\alpha$  and  $\sigma_\alpha^2$ , we sample directly from the full conditional distributions (Eq. 17.2.2  
 14090 and Eq. 17.3.1):

$$\mu_\alpha^{(1)} \sim \text{Normal}(\mu_n, \sigma_n^2)$$

14091 where

$$\mu_n = \frac{\sigma_\alpha^{2(0)}}{\sigma_\alpha^{2(0)} + n_\alpha \sigma_0^2} \times \mu_0 + \frac{n_\alpha \sigma_0^2}{\sigma_\alpha^{2(0)} + n_\alpha \sigma_0^2} \times \bar{\alpha}^{(1)}$$

14092 and

$$\sigma_n^2 = \frac{\sigma_\alpha^{2(0)} \sigma_0^2}{\sigma_\alpha^{2(0)} + n \sigma_0^2}$$

14093 Here,  $\bar{\alpha}$  is the current mean of the vector  $\alpha$ , which we updated before, and  $n_\alpha$  is the  
 14094 length of  $\alpha$ . For  $\sigma_\alpha^2$  we use

$$\sigma_\alpha^{2(1)} \sim \text{Inverse-Gamma}(a_n, b_n),$$

14095 where

$$a_n = n_a / 2 + a_0,$$

14096 and

$$b_n = 0.5 \sum_{i=1}^{n_\alpha} (\alpha_i^{(1)} - \mu_\alpha^{(1)})^2 + b_0.$$

14097 We repeat these steps over  $T$  iterations of the MCMC algorithm. Call the function  
 14098 **PoisGLMM()** in **scrbook** to check out what this algorithm looks like in **R**.

14099 In this example we may not want to save each individual  $\alpha_i$ , but are only interested in  
 14100 their mean and standard deviation. Since these two parameters will change as soon as the  
 14101 value for one element in  $\alpha$  changes, their acceptance rates will always be close to 1 and  
 14102 are not representative of how well your algorithm performs. To monitor the acceptance  
 14103 rates of parameters you do not want to save, you simply need to add a few lines of code  
 14104 into your updater to see how often the individual parameters are accepted. The code for  
 14105 updating  $\alpha$  from our Poisson GLMM below shows one way how to monitor acceptance of  
 14106 individual  $\alpha_i$ 's.

```
14107 #initiate counter for acceptance rate of alpha
14108 alphaUps<-0
14109
14110 #loop over sites, update intercepts alpha one at a time;
14111 #only data at site i contributes information
14112 #lev is the number of sites i
14113 for (i in 1:lev) {
14114   alpha.cand<-rnorm(1, alpha[i], delta_alpha)
14115   loglike<- sum(dpois (y[site==i], exp(alpha[i] + beta*x[site==i]),
14116     log=TRUE))
14117   logprior<- dnorm(alpha[i], mu_alpha,sig_alpha, log=TRUE)
14118   loglike.cand<- sum(dpois (y[site==i], exp(alpha.cand + beta *x[site==i]),
14119     log=TRUE))
14120   logprior.cand<- dnorm(alpha.cand, mu_alpha,sig_alpha, log=TRUE)
14121   if (runif(1)< exp((loglike.cand+logprior.cand) -(loglike+logprior))) {
```

---

```

14122 alpha[i]<-alpha.cand
14123 alphaUps<-alphaUps+1
14124 }
14125 }
14126
14127 #lets you check the acceptance rate of alpha at every 100th iteration
14128 if(iter %% 100 == 0) {
14129     cat("    Acceptance rates\n")
14130     cat("        alpha =", alphaUps/lev, "\n")
14131 }
```

#### 14132 17.3.4 Rejection sampling and slice sampling

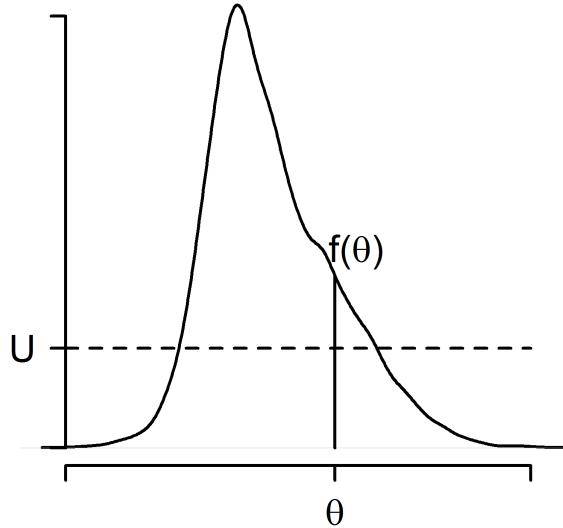
14133 While MH and Gibbs sampling are probably the most widely applied algorithms for posterior approximation, there are other options that work under certain circumstances and  
 14134 may be more efficient when applicable. **WinBUGS** applies these algorithms and we want  
 14135 you to be aware that there is more out there to approximate posterior distributions than  
 14136 Gibbs and MH. One alternative algorithm is rejection sampling. Rejection sampling is  
 14137 not an MCMC method, since each draw is independent of the others. The method can  
 14138 be used when the posterior  $[\theta|y]$  is not a known parametric distribution but can be ex-  
 14139 pressed in closed form. Then, we can use a so-called envelope function, say,  $g(\theta)$ , that  
 14140 we can easily sample from, with the restriction that  $[\theta|y] < M \times g(\theta)$ . We then sample a  
 14141 candidate value for  $\theta$  from  $g(\theta)$ , calculate  $r = [\theta|y]/M \times g(\theta)$  and keep the sample with  
 14142 the probability  $r$ .  $M$  is a constant that has to be picked so that  $r$  lies between 0 and 1, for  
 14143 example by evaluating both  $[\theta|y]$  and  $g(\theta)$  at  $n$  points and looking at their ratios. Rejec-  
 14144 tion sampling only works well if  $g(\theta)$  is similar to  $[\theta|y]$ , and packages like **WinBUGS** use  
 14145 adaptive rejection sampling (Gilks and Wild, 1992), where a complex algorithm is used to  
 14146 fit an adequate and efficient  $g(\theta)$  based on the first few draws. Though efficient in some  
 14147 situations, rejection sampling does not work well with high-dimensional problems, since  
 14148 it becomes increasingly hard to define a reasonable envelope function. For an example  
 14149 of rejection sampling in the context of SCR models, see Chapt. 11, where we use it to  
 14150 simulation inhomogeneous point processes.

14152 Another alternative is slice sampling (Neal, 2003). In slice sampling, we sample uni-  
 14153 formly from the area under the plot of  $[\theta|y]$ . Considering a single univariate  $\theta$ . Let's define  
 14154 an auxiliary variable,  $U \sim \text{Unif}(0, [\theta|y])$ . Then,  $\theta$  can be sampled from the vertical slice  
 14155 of  $[\theta|y]$  at  $U$  (Fig. 17.4):

$$\theta|U \sim \text{Unif}(B),$$

14156 where  $B = \{\theta : [\theta|y] \geq U\}$

14157 Slice sampling can be applied in many situations; however, implementing an efficient  
 14158 slice sampling procedure can be complicated. We refer the interested reader to Robert and  
 14159 Casella (2010, Chapt. 7) for a simple example. Both rejection sampling and slice sampling  
 14160 can be applied on one-dimensional conditional distributions within a Gibbs sampling setup.



**Figure 17.4.** Slice sampling. For  $U \sim \text{Unif}(0, [\theta|y])$ , we can sample  $\theta$  from the vertical slice of  $[\theta|y]$  at  $U$ ;  $\theta|U \sim \text{Unif}(B)$ , where  $B = \{\theta : [\theta|y] \geq U\}$ .

## 17.4 MCMC FOR CLOSED CAPTURE-RECAPTURE MODEL $M_H$

14161 By now you have seen MCMC samplers for some simple generalized (mixed) linear models.  
 14162 Now, to ease you into more complex models, we construct our own MCMC algorithm using  
 14163 a Metropolis-within-Gibbs sampler for the non-spatial model with individual heterogeneity  
 14164 in capture probability, model  $M_h$ , developed in Chapt. 4.

14165 To recapitulate: Under the non-spatial model, each of the  $n$  observed individuals is  
 14166 either detected (1) or not (0) during each of  $K$  sampling occasions. We estimate  $N$  using  
 14167 data augmentation and have a Bernoulli model for the data augmentation variables  $z_i$ .

$$z_i \sim \text{Bernoulli}(\psi)$$

14168 The binomial observation model is expressed conditional on the latent variables  $z_i$ .

$$y_i \sim \text{Binomial}(p_i \times z_i, K)$$

14169 Further, we prescribe a distribution for the capture probability  $p_i$ . Here we assume

$$\text{logit}(p_i) \sim \text{Normal}(\mu_p, \sigma_p^2)$$

14170 As usual, we have to go through two general steps before we write the MCMC algo-  
 14171 rithm:

- 14172 (1) Identify the model with all its components (including priors)  
 14173 (2) Recognize and express the full conditional distributions for all parameters

14174 Our model components are as follows:  $[y_i|p_i, z_i]$ ,  $[p_i|\mu_p, \sigma_p]$ , and  $[z_i|\psi]$  for each  $i =$   
 14175  $1, 2, \dots, M$  and then prior distributions  $[\mu_p]$ ,  $[\sigma_p]$  and  $[\psi]$ . The joint posterior distri-  
 14176 bution of all unknown quantities in the model is proportional to the joint distribution of  
 14177 all elements  $y_i, p_i, z_i$  and also the prior distributions of the prior parameters:

$$\left\{ \prod_{i=1}^M [y_i|p_i, z_i] [p_i|\mu_p, \sigma_p] [z_i|\psi] \right\} [\mu_p, \sigma_p, \psi]$$

14178 For prior distributions, we assume that  $\mu_p, \sigma_p, \psi$  are mutually independent and for  $\mu_p$   
 14179 and  $\sigma_p$  we use improper uniform priors, and  $\psi \sim \text{Uniform}(0, 1)$ . This is equivalent to  
 14180 Beta(1, 1), which will come in handy, as we will see in a moment. Note that the likelihood  
 14181 contribution for each individual, when conditioned on  $p_i$  and  $z_i$ , does not depend on  $\psi$ ,  
 14182  $\mu_p$ , or  $\sigma_p$ . As such, the full-conditional for the structural parameter  $\psi$  only depend on the  
 14183 collection of data augmentation variables  $z_i$ , and that for  $\mu_p$  and  $\sigma_p$  will only depend on  
 14184 the collection of latent variables  $p_i; i = 1, 2, \dots, M$  (this is equivalent to what we saw in the  
 14185 Poisson regression with random intercept  $\alpha$ , where hyperparameters for the distribution  
 14186 of  $\alpha$  did not depend on the observed data). The full conditionals for all the unknowns are  
 14187 as follows:

14188 (1) For  $p_i$ :

$$[p_i|y_i, \mu_p, \sigma_p, z_i] \propto [y_i|p_i][p_i|\mu_p, \sigma_p] \text{ if } z_i = 1 \\ [p_i|\mu_p, \sigma_p] \text{ if } z_i = 0$$

14189 (2) for  $z_i$ :

$$[z_i|y_i, p_i, \psi] \propto [y_i|z_i \times p_i] \text{Bernoulli}(z_i|\psi)$$

14190 (3) For  $\mu_p$ :

$$[\mu_p|p_i, \sigma_p] \sim \left\{ \prod_i [p_i|\mu_p, \sigma_p] \right\} \times \text{const}$$

14191 (4) For  $\sigma_p$ :

$$[\sigma_p|p_i, \mu_p] \sim \left\{ \prod_i [p_i|\mu_p, \sigma_p] \right\} \times \text{const}$$

14192 (5) For  $\psi$ :

$$[\psi|z_i] \propto \left\{ \prod_i [z_i|\psi] \right\} \text{Beta}(1, 1)$$

14193 Remember that Beta(1,1) is equivalent to Uniform(0,1). The beta distribution is the  
 14194 conjugate prior to the binomial and Bernoulli distributions and the general form of a full  
 14195 conditional of a beta-binomial model with  $x_i \sim \text{Bernoulli}(\theta)$  and  $\theta \sim \text{Beta}(a, b)$  is

$$[\theta|\mathbf{x}] \propto \text{Beta}(a + \sum_i x_i, b + n - \sum_i x_i).$$

14196 In our case that means

$$[\psi|z_i] \propto \text{Beta}(1 + \sum z_i, 1 + M - \sum z_i).$$

What we've done here is identify each of the full conditional distributions in sufficient detail to toss them into our Metropolis-Hastings algorithm. The constant terms in the full conditionals for  $\mu_p$  and  $\sigma_p$  reflect the improper prior we chose for both parameters. Because of the choice of an improper prior, prior probability densities for both parameters  $\propto 1$ , i.e. constant, and these constants cancel out of the MH acceptance ratio (see updating step below and following example). Below, you see the updating step for the detection parameter  $\mathbf{p}$ . Note that (1) we draw candidate values on the logit scale and (2) instead of looping through  $1 - M$  individuals to update all  $p_i$ , we update all elements of the vector of  $\mathbf{p}$  in parallel, for computational efficiency.

```
14197 14198 14199 14200 14201 14202 14203 14204 14205
14206 14207 14208 14209 14210 14211 14212 14213
14214 14215 14216
14217 14218 14219 14220
14221 14222 14223 14224 14225 14226 14227 14228 14229 14230
14231 14232 14233 14234 14235 14236 14237
```

---

```
### update the logit(p) parameters
lp.cand<- rnorm(M,lp,1) # 1 is a tuning parameter
p.cand<-plogis(lp.cand)
ll<-dbinom(ytot,K,z*p, log=T)
prior<-dnorm(lp,mu,sigma, log=T)
llcand<-dbinom(ytot,K,z*p.cand, log=T)
prior.cand<-dnorm(lp.cand,mu,sigma, log=T)

kp<- runif(M) < exp((llcand+prior.cand)-(ll+prior))
p[kp]<-p.cand[kp]
lp[kp]<-lp.cand[kp]
```

The parameters  $\mu_p$  and  $\sigma_p$  are also updated using MH steps (see the code for  $\mu_p$  below). In truth, we could also sample  $\mu_p$  and  $\sigma_p^2$  directly with certain choices of prior distributions. For example, if  $\mu_p \sim \text{Normal}(0, 1000)$  then the full conditional for  $\mu_p$  is also normal (see sec. 17.3.1), etc..

```
14221 p0.cand<- rnorm(1,p0,.05)
14222 if(p0.cand>0 & p0.cand<1){
14223 mu.cand<-log(p0.cand/(1-p0.cand))
14224 ll<-sum(dnorm(lp,mu,sigma,log=TRUE))
14225 llcand<-sum(dnorm(lp,mu.cand,sigma,log=TRUE))
14226 if(runif(1)<exp(llcand-ll)) {
14227 mu<-mu.cand
14228 p0<-p0.cand
14229 }
14230 }
```

For  $\psi$  we can easily sample directly from the beta distribution:

```
14232 psi<-rbeta(1, sum(z) + 1, M-sum(z) + 1)
```

To update the  $z_i$  we have opted for a MH updater (although they could be updated directly from their full-conditional). Since  $z_i$  can only take the values of 0 or 1, we generate candidate values using `z.cand<-ifelse(z==1,0,1)`. The updating step for  $z_i$  is detailed in the next example. You can check out the full code by invoking `modelMh()` from the **R** package `scrbook`.

---

## 17.5 MCMC ALGORITHM FOR MODEL SCR0

14238 Conceptually, but also in terms of MCMC coding, it is only a small step from the non-  
 14239 spatial model  $M_h$  to a fully spatial capture-recapture model. Next, we will walk you  
 14240 through the steps of building your own MCMC sampler for the basic SCR model (i.e.  
 14241 without any individual, site or time specific covariates) with both a Poisson and a binomial  
 14242 encounter process. As usual, we will have to go through two general steps before we write  
 14243 the MCMC algorithm:

- 14244 (1) Identify the model with all its components (including priors)  
 14245 (2) Recognize and express the full conditional distributions for all parameters

14246 It is worthwhile to go through all of step 1 for an SCR model, but you have probably  
 14247 seen enough of step 2 in our previous examples to get the essence of how to express a full  
 14248 conditional distribution. Therefore, we will exemplify step 2 for some parameters and tie  
 14249 these examples directly to the respective R code.

14250 **Step 1 – Identify your model** Recall the components of the basic SCR model with  
 14251 a Poisson encounter process from Chapt. 9: We assume that individuals  $i$ , or rather, their  
 14252 activity centers  $\mathbf{s}_i$ , are uniformly distributed across the state-space  $\mathcal{S}$ ,

$$\mathbf{s}_i \sim \text{Uniform}(\mathcal{S})$$

14253 and that the number of times individual  $i$  encounters trap  $j$ ,  $y_{ij}$ , is a Poisson variable  
 14254 with mean  $\lambda_{ij}$ ,

$$y_{ij} \sim \text{Poisson}(\lambda_{ij}).$$

14255 The link between individual location, movement and trap encounter rates is made by the  
 14256 assumption that  $\lambda_{ij}$ , is a decreasing function of the distance between  $\mathbf{s}_i$  and the location  
 14257 of  $j$ ,  $\mathbf{x}_j$ , say

$$d_{ij} = \|\mathbf{s}_i - \mathbf{x}_j\|,$$

14258 of the Gaussian (or half-normal) form

$$\lambda_{ij} = \lambda_0 \exp(-d_{ij}^2/2\sigma^2),$$

14259 where  $\lambda_0$  is the baseline trap encounter rate at  $d_{ij} = 0$  and  $\sigma$  is the scale parameter of the  
 14260 half-normal function.

14261 As in the non-spatial example for model  $M_h$ , we estimate  $N$ , here the number of  $\mathbf{s}_i$   
 14262 in  $\mathcal{S}$ , using data augmentation (sec. 4.2). We create  $M - n$  all-zero encounter histories  
 14263 and estimate  $N$  by summing over the auxiliary data augmentation variables,  $z_i$ , which we  
 14264 assume is a Bernoulli random variable,

$$z_i \sim \text{Bernoulli}(\psi).$$

14265 To link the two model components, we modify our trap encounter model to

$$\lambda_{ij} = \lambda_0 \times \exp(-d_{ij}^2/2\sigma^2) \times z_i.$$

14266 The model has the following structural parameters, for which we need to specify priors:

14267  $\psi$ : the Uniform(0, 1) is required as part of the data augmentation procedure and in general  
 14268 is a natural choice of an uninformative prior for a probability. It will also lead to  
 14269 conjugacy as we saw in the example of model  $M_h$ , so that we can update  $\psi$  directly  
 14270 from its full conditional distribution using Gibbs sampling.  
 14271  $s_i$ : since  $s_i$  is a pair of coordinates it is two-dimensional and we use a uniform prior  
 14272 limited by the extent of our state-space over both dimensions.  
 14273  $\sigma$ : we can conceive several priors for  $\sigma$  but let's assume an improper prior, one that is  
 14274 Uniform over  $(0, \infty)$ . As we already saw, this choice is convenient when updating the  
 14275 parameter, because the constant prior probability cancels out of the MH acceptance  
 14276 ratio.  
 14277  $\lambda_0$ : analogous, we will use a Uniform( $0, \infty$ ) improper prior for  $\lambda_0$ .

14278 **Step 2 – Construct the full conditionals:** Having completed step 1, let's look at  
 14279 the full conditional distributions for some of these parameters. We saw that with improper  
 14280 priors, full conditionals are proportional only to the likelihood of the observations; for  
 14281 example, consider  $\sigma$ :

$$[\sigma | s, \lambda_0, z, y] \propto \left\{ \prod_i [y_i | s_i, \lambda_0, z_i, \sigma] \right\}$$

14282 The R code to update  $\sigma$  is shown below. Notice that we automatically reject negative  
 14283 candidate values, since  $\sigma$  cannot be  $< 0$ .

```

14284 sig.cand <- rnorm(1, sigma, 0.1) #draw candidate value
14285 if(sig.cand>0){ #automatically reject sig.cand that are <0
14286   lam.cand <- lam0*exp(-(d*d)/(2*sig.cand*sig.cand))
14287   ll<- sum(dpois(y, lam*z, log=TRUE))
14288   llcand <- sum(dpois(y, lam.cand*z, log=TRUE))
14289   if(runif(1) < exp( llcand - ll) ){
14290     ll<-llcand
14291     lam<-lam.cand
14292     sigma<-sig.cand
14293   }
14294 }
```

14295 These steps are analogous for  $\lambda_0$  and  $s_i$  and we will use MH steps for all of these  
 14296 parameters. Similar to the random intercepts in our Poisson GLMM, we update each  $s_i$   
 14297 individually. Note that to be fully correct, the full conditional for  $s_i$  contains both the  
 14298 likelihood and prior component, since we did not specify an improper, but a proper uniform  
 14299 prior on  $s_i$ . However, with a uniform distribution the probability density of any value is  
 14300  $1/(\text{upper limit} - \text{lower limit}) = \text{constant}$ . Thus, the prior components are identical for  
 14301 both the current and the candidate value so that when you calculate the ratio of posterior  
 14302 densities,  $r$ , the identical prior component appears both in the numerator and denominator  
 14303 and cancel each other out.

14304 We still have to update  $z_i$ . The full conditional for  $z_i$  is

$$[z_i | y_i, \sigma, \lambda_0, s_i] \propto [y_i | z_i, \sigma, \lambda_0, s_i][z_i]$$

14305 and since  $z_i \sim \text{Bernoulli}(\psi)$ , the term has to be taken into account when updating  $z_i$ :

---

```

14306      zUps <- 0 #set counter to monitor acceptance rate
14307      for(i in 1:M) {
14308          #no need to update seen individuals, since their z =1
14309          if(seen[i])
14310              next
14311          zcand <- ifelse(z[i]==0, 1, 0)
14312          llz <- sum(dpois(y[i,],lam[i,]*z[i], log=TRUE))
14313          llcand <- sum(dpois(y[i,], lam[i,]*zcand, log=TRUE))

14314
14315          prior <- dbinom(z[i], 1, psi, log=TRUE)
14316          prior.cand <- dbinom(zcand, 1, psi, log=TRUE)
14317          if(runif(1) < exp((llcand+prior.cand)-(llz+prior))){
14318              z[i] <- zcand
14319              zUps <- zUps+1
14320          }
14321      }

```

14322 The parameter  $\psi$  is a hyperparameter of the model, with an uninformative prior distribution of Uniform(0, 1) or Beta(1, 1), so that

$$[\psi|\mathbf{z}] \propto \text{Beta}\left(1 + \sum_i z_i, 1 + M - \sum_i z_i\right).$$

14324 These are all the building blocks you need to write the MCMC algorithm for the spatial  
14325 null model with a Poisson encounter process. You can find the full **R** code by calling the  
14326 function (**SCR0pois**) in the **R** package **scrbook**.

### 14327 17.5.1 SCR model with binomial encounter process

14328 The equivalent SCR model with a binomial encounter process is very similar. Here, each  
14329 individual  $i$  can only be detected once at any given trap  $j$  during a sampling occasion  $k$ .  
14330 Thus

$$y_{ij} \sim \text{Binomial}(p_{ij}, K)$$

14331 Where  $p_{ij}$  is some function of distance between  $\mathbf{s}_i$  and trap location  $\mathbf{x}_j$ . Here we use:

$$p_{ij} = 1 - \exp(-\lambda_{ij})$$

14332 Recall from Chapt. 3 that this is the complementary log-log (cloglog) link function,  
14333 which constrains  $p_{ij}$  to fall between 0 and 1. For our MCMC algorithm that means that,  
14334 instead of using a Poisson likelihood,  $\text{Poisson}(y|\sigma, \lambda_0, \mathbf{s}, z)$ , we use a binomial likelihood,  
14335  $\text{Binomial}(y|\sigma, \lambda_0, \mathbf{s}, z; K)$ , in all the conditional distributions. An exemplary updating step  
14336 for  $\lambda_0$  under a binomial encounter model is shown below. The full MCMC code for the  
14337 binomial SCR with a cloglog link (**SCR0binom.cl**) can be found in the **R** package **scrbook**.

```

14338      lam0.cand <- rnorm(1, lam0, 0.1)
14339      #automatically reject lam0.cand that are <0
14340      if(lam0.cand >0){

```

```

14341      lam.cand <- lam0.cand*exp(-(d*d)/(2*sigma*sigma))
14342      p.cand <- 1-exp(-lam.cand)
14343      ll<- sum(dbinom(y, K, pmat *z, log=TRUE))
14344      llcand <- sum(dbinom(y, K, p.cand *z, log=TRUE))
14345      if(runif(1) < exp( llcand - ll) ){
14346          ll<-llcand
14347          pmat<-p.cand
14348          lam0<- lam0.cand
14349      }
14350  }

```

Another possibility is to model variation in the individual and site specific detection probability,  $p_{ij}$ , directly, without any transformation, such that

$$p_{ij} = p_0 \times \exp(-d_{ij}^2/(2\sigma^2))$$

and  $p_0 \in [0, 1]$ . This formulation is analogous to how detection probability is modeled in distance sampling under a half-normal detection function; however, in distance sampling  $p_0$  – detection of an individual on the transect line – is assumed to be 1 (Buckland et al., 2001). Under this formulation the updater for  $p_0$  becomes:

```

14357      p0.cand <- rnorm(1, p0, 0.1)
14358      if(p0.cand >0 & p0.cand < 1 ){
14359          #automatically rejects lam0.cand that are not {0,1}
14360          p.cand <- p0.cand*exp(-(d*d)/(2*sigma*sigma))
14361          ll<- sum(dbinom(y, K, pmat *z, log=TRUE))
14362          llcand <- sum(dbinom(y, K, p.cand *z, log=TRUE))
14363          if(runif(1) < exp( llcand - ll) ){
14364              ll<-llcand
14365              pmat<-p.cand
14366              p0<- p0.cand
14367          }
14368      }

```

## 17.6 LOOKING AT MODEL OUTPUT

Now that you have an MCMC algorithm to analyze spatial capture-recapture data with, let's run an actual analysis so we can look at the output. As an example, we will use the Fort Drum bear data set we first introduced in Chapt. 1 and already analyzed in several preceding chapters. You can load the Fort Drum data (`data(beardata)`), extract the trap locations (`trapmat`) and detection data (`bearArray`) and build the augmented  $M \times J$  array of individual encounter histories:

```

14375  > M=700
14376  > trapmat<-beardata$trapmat
14377  #summarizes captures across occasions
14378  > bearmat<-apply(beardata$bearArray, 1:2, sum)
14379  > Xaug<-matrix(0, nrow=M, ncol=dim(trapmat)[1])
14380  > Xaug[1:dim(bearmat)[1],]<-bearmat #create augmented data set

```

14381 In addition to these data, we need to specify the outermost coordinates of the state-  
 14382 space. Since bears are wide ranging animals we add a 20-km buffer to the maximum and  
 14383 minimum coordinates of the trap array:

```
14384 > xl<- min(trapmat[,1])- 20
14385 > yl<- min(trapmat[,2])- 20
14386 > xu<- max(trapmat[,1])+ 20
14387 > yu<- max(trapmat[,2])+ 20
```

14388 Finally, use the MCMC code for the binomial encounter model with the cloglog link  
 14389 (`SCR0binom.cl`) and run 5000 iterations. This should take approximately 25 minutes (in  
 14390 real life we would of course run the algorithm a lot longer but for demonstration purposes  
 14391 let's stick with a number of iterations that can be run in a manageable amount of time).

```
14392 > set.seed(13)
14393 > mod0<-SCR0binom.cl(y=Xaug, X=trapmat, M=M, xl=xl, xu=xu, yl=yl,
14394 + yu=yu, K=8, delta=c(0.1, 0.05, 2), niter=5000)
```

14395 Before, we used simple **R** commands to look at model results. However, there is a  
 14396 specific **R** package to summarize MCMC simulation output and perform some convergence  
 14397 diagnostics – package `coda` (Plummer et al., 2006). Download and install `coda`, then  
 14398 convert your model output to an `mcmc` object

```
14399 > chain<-mcmc(mod0)
```

14400 which can be used by `coda` to produce MCMC specific output.

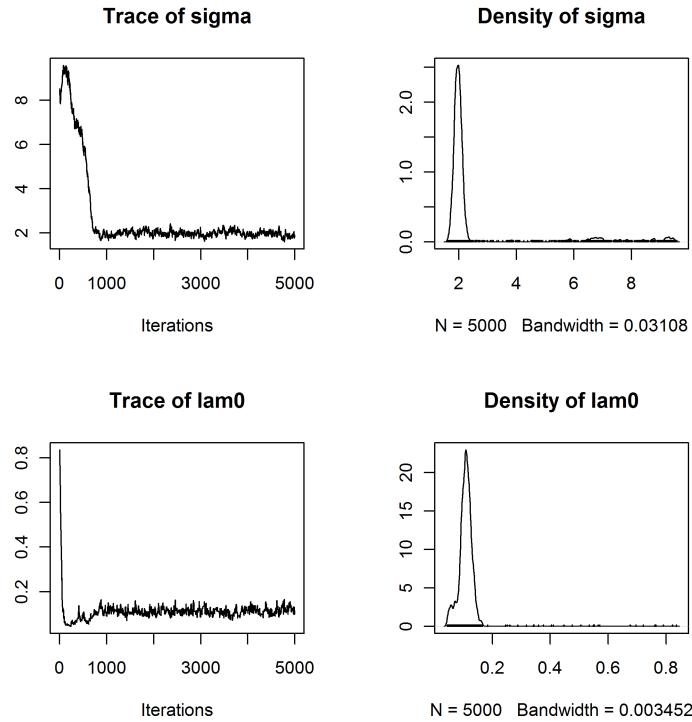
### 14401 17.6.1 Markov chain time series plots

14402 Start by looking at time series plots of your Markov chains using `plot(chain)`. This com-  
 14403 mand produces a time series plot and marginal posterior density plots for each monitored  
 14404 parameter, similar to what we did before using the `hist()` and `plot()` commands. Fig.  
 14405 17.5 shows an example of these plots for  $\sigma$  and  $\lambda_0$ . Time series plots will tell you several  
 14406 things: First, recall from sec. 17.3.2 that the way the chains move through the parameter  
 14407 space gives you an idea of whether your MH steps are well tuned. If chains were constant  
 14408 over many iterations you would need to decrease the tuning parameter of the (normal)  
 14409 proposal distribution. If a chain moves along some gradient to a stationary state very  
 14410 slowly, you may want to increase the tuning parameter so that the parameter space is  
 14411 explored more efficiently.

14412 Second, you will be able to see if your chains converged and how many initial sim-  
 14413 ulations you have to discard as burn-in. In the case of the chains shown in Fig. 17.5,  
 14414 we would probably consider the first 750 – 1000 iterations as burn-in, as afterwards the  
 14415 chains seem to be fairly stationary.

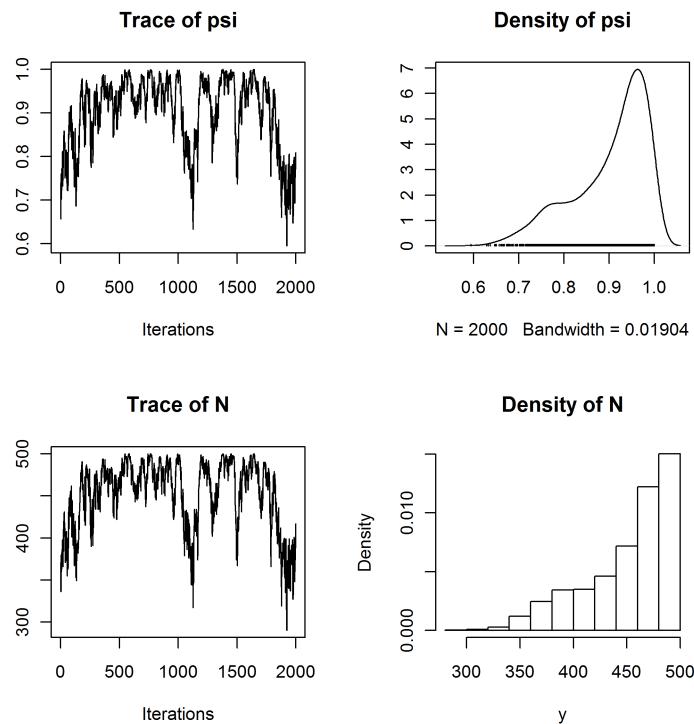
### 14416 17.6.2 Posterior density plots

14417 The `plot()` command also produces posterior density plots and it is worthwhile to look  
 14418 at those carefully. For parameters with priors that have bounds (e.g. uniform over some



**Figure 17.5.** Time series and posterior density plots for  $\sigma$  and  $\lambda_0$  for the Fort Drum black bear data.

interval), you will be able to see if your choice of the prior is truncating the posterior distribution. In the context of SCR models, this will mostly involve our choice of  $M$ , the size of the augmented data set. If the posterior of  $N$  has a lot of mass concentrated close to  $M$  (or equivalently the posterior of  $\psi$  has a lot of mass concentrated close to 1), as in the example in Fig. 17.6, we have to re-run the analysis with a larger  $M$ . A diffuse posterior plot suggests that the parameter may not be well-identified. There may not be enough information in your data to estimate model parameters and you may have to consider a simpler model. Finally, posterior density plots will show you if the posterior distribution is symmetrical or skewed – if the distribution has a heavy tail, using the mean as a point estimate of your parameter of interest may be biased and you may want to opt for the median or mode instead.



**Figure 17.6.** Time series and posterior density plots of  $\psi$  and  $N$  for the Fort Drum black bear data truncated by the upper limit of  $M$  (500).

### 14430 17.6.3 Serial autocorrelation and effective sample size

14431 Checking the degree of autocorrelation in your Markov chains and estimating the effective  
 14432 sample size your chain has generated should be part of evaluating your model output.  
 14433 If you use **WinBUGS** through the **R2WinBUGS** package, the **print()** command will au-  
 14434 tomatically return the effective sample size for all monitored parameters. In the **coda**  
 14435 package there are several functions you can use to do so. The function **effectiveSize()**  
 14436 will directly give you an estimate of the effective sample size for the parameters:

```
14437 > effectiveSize(window(chain, start=1001))
14438   sigma      lam0      psi      N
14439  93.89807 163.72311 51.96443 46.45394
```

14440 Alternatively, you can use the **autocorr.diag()** function, which will show you the  
 14441 degree of autocorrelation for different lag values (which you can specify within the function

14442 call, we use the defaults below):

```
14443 > autocorr.diag(window(chain, start=1001))
14444      sigma      lam0      psi      N
14445 Lag 0  1.0000000 1.0000000 1.0000000 1.0000000
14446 Lag 1  0.9316928 0.91464875 0.9745833 0.9663320
14447 Lag 5  0.7603332 0.67445407 0.8525272 0.8500215
14448 Lag 10 0.6065374 0.48724122 0.7514657 0.7530124
14449 Lag 50 0.1122331 0.06564406 0.3811939 0.3823236
```

14450 In the present case we see that autocorrelation is especially high for the parameter  $\psi$  and  
 14451 effective sample size for this parameter is only 52! This means we would have to run the  
 14452 model for much longer to obtain a reasonable effective sample size. Unfortunately, with  
 14453 many SCR data sets we observe high degrees of serial autocorrelation. For now, let's  
 14454 continue using this small number of samples to look at the output.

#### 14455 17.6.4 Summary results

14456 Now that we checked that our chains apparently have converged and pretending that  
 14457 we have generated enough samples from the posterior distribution, we can look at the  
 14458 actual parameter estimates. The **summary()** function will return two sets of results: the  
 14459 mean parameter estimates, with their standard deviation, the naïve standard error – i.e.  
 14460 your regular standard error calculated for  $T$  (= number of iterations) samples without  
 14461 accounting for serial autocorrelation – and the Time-series SE (in **WinBUGS** and earlier  
 14462 in this book referred to as MC error), which accounts for autocorrelation. Remember our  
 14463 rule of thumb that this error decreases with increasing chain length and should be 1% or  
 14464 less of the parameter estimate. In **WinBUGS** the MC error is only given in the log output  
 14465 within **BUGS** itself. You should adjust the **summary()** call by removing the burn-in from  
 14466 calculating parameter summary statistics. To do so, use the **window()** command, which  
 14467 lets you specify at which iteration to start 'counting'. In contrast to **WinBUGS**, which  
 14468 requires you to set the burn-in length before you run the model, this command gives us  
 14469 full flexibility to make decisions about the burn-in after we have seen the trajectories of  
 14470 our Markov chains. For our example, **summary(window(chain, start=1001))** returns the  
 14471 following output:

```
14472 Iterations = 1001:5000
14473 Thinning interval = 1
14474 Number of chains = 1
14475 Sample size per chain = 4000
14476
14477 1. Empirical mean and standard deviation for each variable,
14478 plus standard error of the mean:
14479
14480      Mean        SD  Naive SE Time-series SE
14481 sigma    1.9697  0.12534 0.0019818      0.012792
14482 lam0     0.1124  0.01521 0.0002405      0.001311
14483 psi      0.7295  0.11794 0.0018648      0.015278
```

---

```

14484 N      510.9190 81.99868 1.2965130      10.580567
14485
14486 2. Quantiles for each variable:
14487
14488      2.5%    25%    50%    75%   97.5%
14489 sigma   1.7288   1.8831   1.9666   2.0517   2.2240
14490 lam0    0.0863   0.1008   0.1112   0.1217   0.1449
14491 psi     0.5100   0.6423   0.7261   0.8170   0.9549
14492 N      359.0000 451.0000 508.0000 572.0000 668.0000

```

14493 Looking at the MC errors (column labeled `Time-series SE`), we see that in spite of the  
14494 high autocorrelation, the MC error for  $\sigma$  is below the 1% threshold, whereas for all other  
14495 parameters, MC errors are still above, another indication that for a thorough analysis we  
14496 should run a longer chain.

14497 Our algorithm gives us a posterior distribution of  $N$ , but we are usually interested  
14498 in the density,  $D$ . Density itself is not a parameter of our model, but we can derive a  
14499 posterior distribution for  $D$  by dividing each value of  $N$  ( $N$  at each iteration) by the area  
14500 of the state-space (here  $3032.719 \text{ km}^2$ ) and we can use summary statistics of the resulting  
14501 distribution to characterize  $D$ :

```

14502 > summary(window(chain[,4]/ 3032.719, start=1001))
14503
14504 Iterations = 1001:5000
14505 Thinning interval = 1
14506 Number of chains = 1
14507 Sample size per chain = 4000
14508
14509 1. Empirical mean and standard deviation for each variable,
14510 plus standard error of the mean:
14511
14512      Mean           SD        Naive SE Time-series SE
14513 0.1684690 0.0270380 0.0004275 0.0034888
14514
14515 2. Quantiles for each variable:
14516
14517      2.5%    25%    50%    75%   97.5%
14518 0.1184 0.1487 0.1675 0.1886 0.2203

```

14519 We see that our mean density of  $0.17/\text{km}^2$  is very similar to the estimate of  $0.18/\text{km}^2$   
14520 obtained under the non-spatial model  $M_0$  in Chapt. 4.

### 14521 17.6.5 Other useful commands

14522 While inspecting the time series plot gives you a first idea of how well you tuned your  
14523 MH algorithm, use `rejectionRate()` to obtain the rejection rates (1 – acceptance rates)  
14524 of the parameters that are written to your output:

```

14525 > rejectionRate(chain)
14526   sigma      lam0      psi      N
14527 0.42988598 0.78775755 0.00000000 0.03160632

```

14528 Recall (sec. 17.3.2) that rejection rates should lie between 0.2 and 0.8, so our tuning  
 14529 seems to have been appropriate here. Draws of the parameter  $\psi$  are never rejected since  
 14530 we update it with Gibbs sampling, where all candidate values are kept. And since  $N$  is  
 14531 the sum of all  $z_i$ , all it takes for  $N$  to change from one iteration to the next are small  
 14532 changes in the z-vector, so the rejection rate of  $N$  is always low. If you have run several  
 14533 parallel chains, you can combine them into a single mcmc object using the `mcmc.list()`  
 14534 command on the individual chains (note that each chain has to be converted to an mcmc  
 14535 object before combining them with `mcmc.list()`). You can then easily obtain the Gelman-  
 14536 Rubin diagnostic (Gelman et al., 2004), in **WinBUGS** called Rhat, using `gelman.diag()`,  
 14537 which will indicate if all chains have converged to the same stationary distribution. For  
 14538 details on these and other functions, see the `coda` manual, which can be found (together  
 14539 with the package) on the CRAN mirror.

## 17.7 MANIPULATING THE STATE-SPACE

14540 So far, we have constrained the location of the activity centers to fall within the outermost  
 14541 coordinates of our rectangular state-space by posing upper and lower bounds for  $x$  and  $y$ .  
 14542 But what if  $S$  has an irregular shape – maybe there is a large water body we would like  
 14543 to remove from  $S$ , because we know our terrestrial study species does not occur there. Or  
 14544 the study takes place in a clearly defined area such as an island.

14545 As mentioned before, this situation is difficult to handle in **BUGS** engines. In some  
 14546 simple cases we can adjust the state-space by setting one of the coordinates of  $s_i$  to be  
 14547 some function of the other and reject candidate  $s_i$  that do not fall within this modified  
 14548 state-space. In this manner, we can cut off corners of the rectangle to approximate the  
 14549 actual state-space<sup>3</sup>. To visualize this approach, plot the following rectangle, representing  
 14550 your state-space polygon, and line, representing, for example, the approximation of a shore  
 14551 line:

```

14552 > xlim<-c(-5,5)
14553 > ylim<-c(-7,7)
14554 > plot(xlim, ylim, type='n')
14555 > abline(a=4, b=0.4)

```

14556 The Y coordinates limiting your state-space to the habitat that is suitable to the species  
 14557 you study can now be expressed as a linear function of the X coordinates, in this case,  
 14558  $Y = 4 + 0.4 \times X$ . To include this new limit in a **BUGS** model, we need to change the  
 14559 following:

```

14560 #draw SX and SY as before
14561 SX[i]~dunif(xlim[1],xlim[2])
14562 SY[i]~dunif(ylim[1],ylim[2])

```

<sup>3</sup>This idea was pitched to us by Mike Meredith, Biodiversity Conservation Society Sarawak/WCS Malaysia

```

14563 #calculate upper limit for Y given X
14564 ymax[i]<-4+0.4*SX[i]
14565 # use step function to see if location [SX, SY]
14566 # is below the Y limit (Pin = 1) or not (Pin = 0)
14567 Pin[i] <- step(ymax[i] - SY[i])
14568 In[i] ~ dbern(Pin[i])

```

14569 The object `In` is a vector of  $M$  1's, passed as data to the model. If  $\text{Pin} = 0$ , the likelihood  
 14570 will be 0 and the candidate  $[\text{SX}, \text{SY}]$  pair will be rejected. If  $\text{Pin} = 1$ , this bit of the  
 14571 likelihood is equal to 1, and whether or not the the candidate pair of coordinates is  
 14572 accepted depends only on capture history of  $i$ . This approach can be very useful in some  
 14573 situations but is clearly restricted by the functional form of the relationship between  $\text{SX}$   
 14574 and  $\text{SY}$  that it requires.

14575 In **R**, we are much more flexible, as we can use the actual state-space polygon to  
 14576 constrain  $s_i$ . To illustrate that, let's look at a camera trapping study of raccoons (*Procyon*  
 14577 *lotor*) conducted on South Core Banks, a barrier island within Cape Lookout National  
 14578 Seashore, North Carolina (details of the study can be found in Sollmann et al. (2013) and  
 14579 in Chapt. 19 where we present the analysis of this data set with spatial mark-resight  
 14580 models). Since camera-traps were spread across the entire length of the island, we set  
 14581 the state-space to be delineated by the shore line of the island (Fig. 17.7), which clearly  
 14582 cannot easily be approximated as a rectangle. Instead, within **R** we can use an actual  
 14583 shapefile of the island.

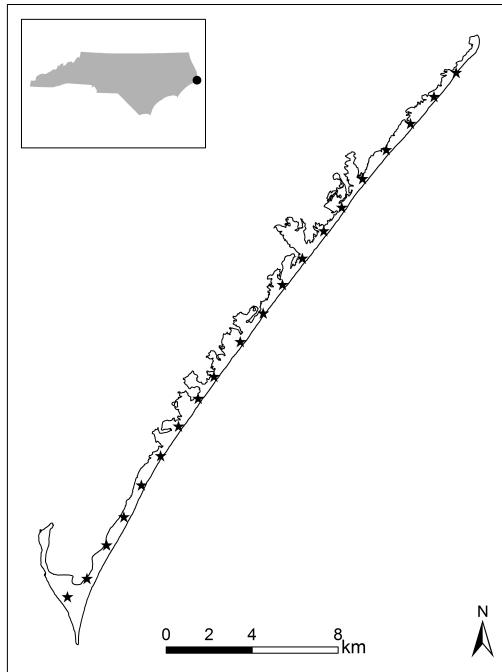
14584 In other circumstances you may still want to create the state-space as before, by adding  
 14585 some buffer to your trapping grid, but you may find that the resulting rectangle includes  
 14586 water bodies, paved parking lots or any other kind of habitat you know is never used by the  
 14587 species you study. In order to precisely describe the state-space, these features need to be  
 14588 removed. You can create a precise state-space polygon in **ArcGIS** and read it into **R**, or  
 14589 create the polygon directly within **R**, by intersecting two shapefiles – one of the rectangle  
 14590 defining the outer limits of your state-space state and one of the landscape feature you  
 14591 want to remove. While you will most likely have to obtain the shapefile describing the  
 14592 landscape of and around your trapping grid (coastlines, water bodies etc.) from some  
 14593 external source, the polygon shapefile buffering your outermost trapping grid coordinates  
 14594 can easily be written in **R**.

14595 If `xmin`, `xmax`, `ymin` and `ymax` mark the most extreme  $x$  and  $y$  coordinates of your  
 14596 trapping grid and `b` is the distance you want to buffer with, load the package **shapefiles**  
 14597 (Stabler, 2006) and issue the following **R** commands:

```

14598 > xl= xmin-b
14599 > xu= xmax+b
14600 > yl= ymin-b
14601 > yu= ymax+b
14602
14603         #create data frame with coordinate pairs
14604 > dd <- data.frame(Id=c(1,1,1,1,1),X=c(xl,xu,xu,xl,xl),
14605 +   Y=c(yl,yl,yu,yu,yl))
14606 > ddTable <- data.frame(Id=c(1),Name=c("Item1"))
14607             #convert to shapefile, type polygon

```



**Figure 17.7.** Camera traps (stars) set up on South Core Banks, a barrier island within Cape Lookout National Seashore, North Carolina (inset map) to estimate the raccoon population (see Chapt. 19 for details).

```

14608 > ddShapefile <- convert.to.shapefile(dd, ddTable, "Id", 5)
14609      # name and save to location of choice
14610 > write.shapefile(ddShapefile, 'c:/Test', arcgis=T)

```

14611 You can read shapefiles into **R** loading the package **maptools** (Lewin-Koh et al., 2011)  
 14612 and using the function **readShapeSpatial()**. Make sure you read in shapefiles in UTM  
 14613 format, so that units of the trap array, the movement parameter  $\sigma$  and the state-space  
 14614 are all identical. Intersection of polygons can be done in **R** also, using the package **rgeos**  
 14615 (Bivand and Rundel, 2011) and the function **gIntersect()**. The area of your (single)  
 14616 polygon can be extracted directly from the state-space object **SSp**:

```

14617 > area <- SSp@polygons[[1]]@Polygons[[1]]@area /1000000

```

14618 Note that dividing by 1000000 will return the area in  $\text{km}^2$  if your coordinates describing  
 14619 the polygon are in UTM. If your state-space consists of several disjunct polygons, you will  
 14620 have to sum the areas of all polygons to obtain the size of the state-space. To include  
 14621 this polygon into our MCMC sampler we need one last spatial **R** package, **sp** (Pebesma

and Bivand, 2011), which has a function, `over()`, which allows us to check if a pair of coordinates falls within a polygon or not.<sup>4</sup> All we have to do is embed this new check into the updating steps for the  $s_i$ :

```
14625 #draw candidate value
14626 Scand <- as.matrix(cbind(rnorm(M, S[,1], 2), rnorm(M, S[,2], 2)))
14627     #convert to spatial points on UTM (m) scale
14628 Scoord<-SpatialPoints(Scand*1000)
14629     # check if scand is within the polygon
14630 SinPoly<-over(Scoord,SSp)
14631
14632 for(i in 1:M) {
14633     #if scand falls within polygon, continue update
14634     if(is.na(SinPoly[i])==FALSE) {
14635         ... [rest of the updating step remains the same]
```

14636 Note that it is much more time-efficient to draw all  $M$  candidate values for  $s$  and check  
 14637 once if they fall within the state-space, rather than running the `over()` command for  
 14638 every individual pair of coordinates. To make sure that our initial values for  $s$  also fall  
 14639 within the polygon of  $\mathcal{S}$ , we use the function `runifpoint()` from the package `spatstat`  
 14640 (Baddeley and Turner, 2005), which generates random uniform points within a specified  
 14641 polygon. You'll find this modified MCMC algorithm (`SCR0poisSSp`) in the **R** package  
 14642 `scrbook`.

14643 Finally, observe that we are converting candidate coordinates of  $\mathcal{S}$  back to meters to  
 14644 match the UTM polygon. In all previous examples, for both the trap locations and the  
 14645 activity centers we have used UTM coordinates divided by 1000 to estimate  $\sigma$  on a km  
 14646 scale. This is adequate for wide ranging species like bears. In other cases you may center  
 14647 all coordinates on 0. No matter what kind of transformation you use on your coordinates,  
 14648 make sure to always convert candidate values for  $\mathcal{S}$  back to the original scale (UTM)  
 14649 before running the `over()` command.

## 17.8 INCREASING COMPUTATIONAL SPEED

14650 Using custom written MCMC algorithms in **R** is not only more flexible but can also be  
 14651 faster than using programs such as **JAGS** and especially **WinBUGS**. Also, **R** tends to  
 14652 use much less memory than **JAGS**, which can be crucial if you are running a large model  
 14653 but only have limited memory available. **WinBUGS** is limited in the amount of memory  
 14654 it can access and thus will likely not max out your memory, but as a trade-off, it will take a  
 14655 long time to run such models. In this chapter we have provided you with the guidelines to  
 14656 write your own MCMC sampler. But beyond the material that we have covered there are  
 14657 a number of ways you can make your sampler more efficient, through parallel computing  
 14658 or by accessing an alternative computer language such as **C++**. Exploring these options  
 14659 exhaustively is beyond the scope of this book; instead, in this section we will give you  
 14660 some pointers to get started with these more advanced computational issues.

---

<sup>4</sup>Remember from Chapt. (6.4.2) that the `over` function takes as its second argument (among others) an object of the class “`SpatialPolygons`” or “`SpatialPolygonsDataFrame`”. The former produces a vector while the latter produces a data frame (e.g., in the example above), which is important for how you index the output.

---

**14661 17.8.1 Parallel computing**

14662 If you are using a computer with several cores, you can make use of parallel computing to  
 14663 speed up overall computation. In parallel computing we execute commands simultaneously  
 14664 on different cores of the computer, instead of running them serially on one single core.  
 14665 For example, imagine you have 4 cores available and you want to implement a for-loop in  
 14666 **R**; instead of going through the loop iteration by iteration, you can prompt **R** to execute  
 14667 iterations 1 to 4 simultaneously on the 4 different cores. The core that finishes first will  
 14668 then continue with iteration 5, and so on. There are several packages in **R** that allow you  
 14669 to induce parallel computing, such as **snow** (Tierney et al., 2011) and **snowfall** (Knaus,  
 14670 2010), and the more current versions of **R** (from 2.14.0 upwards) come with a pre-installed  
 14671 set of functions grouped under the name **parallel**.

14672 The MCMC algorithms developed here and in other parts of this book come with plenty  
 14673 of opportunities to parallelize computation. In various instances within the algorithm, we  
 14674 have for-loops across our augmented data set of size  $M$ , or we may have for-loops across  
 14675 sampling occasions. We also have for-loops across iterations of the algorithm, but since  
 14676 one iteration of the Markov chain depends on the preceding iteration these should always  
 14677 be run serially, not in parallel. There is another dimension we can think of, and that is  
 14678 running multiple chains of an algorithm to assess convergence. This is a comparatively  
 14679 easy implementation of parallel computing and thus provides a good starting point to  
 14680 understand how it works in **R**.

14681 Let's go back to the Ft. Drum black bear data we analyzed above with the cloglog  
 14682 version of the binomial SCR model (sec. 17.6) and run 3 parallel chains using **snowfall**.  
 14683 All we need to do is wrap our function **SCR0binom.cl** within another function that can  
 14684 then be executed in parallel, returning a list with one output matrix for each chain (install  
 14685 **snowfall** before executing the code below; we assume the data objects are already in your  
 14686 workspace from the previous analysis):

```
14687 > library(snowfall)
14688 ## create wrapper function
14689 > wrapper<-function(a){
14690 + out<-SCR0binom.cl(y=Xaug, X=trapmat, M=M, xl=xl, xu=xu, yl=yl,
14691 + yu=yu, K=8, delta=c(0.1, 0.05, 2), niter=5000)
14692 + return(out)
14693 + }
```

14694 After creating the wrapper function we need to initialize the cluster of cores, defining  
 14695 that we want computation to be implemented in parallel and how many cores we want it  
 14696 to be run on. Here, we assume we have (at least) 3 cores, but if your computer only has 2,  
 14697 make sure to adjust the code accordingly (i.e., set **cpus=2**). In that case, 2 of the 3 chains  
 14698 will be run in parallel and whichever core finishes first will then pick up the third chain.  
 14699 Further, we have to export all **R** libraries and data to all the cores, and set up a random  
 14700 number generator, so that we do not get identical results from the different cores:

```
14701 > sfInit( parallel=TRUE, cpus=3 ) #initialize cluster
14702 > sfLibrary(scrbook) #export library scrbook
14703 > sfExportAll() #export all data in current workspace
14704 > sfClusterSetupRNG() #set up random number generator
```

---

```

14705 > outL=sfLapply(1:3,wrapper) # execute 'wrapper' 3 times

14706 The object outL is a list of length 3, with one out matrix from the function SCRObinom.cl
14707 for each chain. After computation is complete, terminate the cluster using the command
14708 sfStop(). Note that the intermediate output of current values and acceptance rates in the
14709 R console is suppressed when using parallel computing. We can now look at the output
14710 as described previously using the package coda, by first defining outL to be a list of mcmc
14711 objects.

14712 > library(coda)
14713 #turn output into MCMC list
14714 > res<-mcmc.list(as.mcmc(outL[[1]]),as.mcmc(outL[[2]]),as.mcmc(outL[[3]]))
14715 > summary(window(res, start=1001)) #remove first 1000 iterations as burn-in
14716
14717 [... some output removed ...]
14718
14719      Mean       SD  Naive SE Time-series SE
14720 sigma   1.9723  0.13093 0.0011952      0.0087055
14721 lam0    0.1115  0.01535 0.0001401      0.0009003
14722 psi     0.7130  0.10787 0.0009847      0.0077910
14723 N      499.6166 74.74934 0.6823650      5.4232653
14724
14725 2. Quantiles for each variable:
14726
14727      2.5%     25%     50%     75%   97.5%
14728 sigma   1.74339  1.8811  1.9637  2.0530  2.2618
14729 lam0    0.08443  0.1007  0.1105  0.1211  0.1438
14730 psi     0.52046  0.6350  0.7093  0.7814  0.9627
14731 N      366.00000 446.00000 497.00000 547.00000 674.0000

14732 Now that we have parallel chains we can also use the function gelman.diag to evaluate
14733 if chains have converged:
14734 > gelman.diag(window(res, start=1001)) #assess chain convergence
14735
14736 Potential scale reduction factors:
14737
14738      Point est. Upper C.I.
14739 sigma      1.01      1.04
14740 lam0       1.01      1.02
14741 psi        1.07      1.21
14742 N          1.07      1.21
14743
14744 Multivariate psrf
14745
14746 1.05

14747 We can see that estimates are similar to what we observed when running a single
14748 chain (see sec. 17.6) and that all 3 chains appear to have converged, based on their point

```

estimates of the  $\hat{R}$  statistic, but, as already noted before, for a real analysis we might want to run this model for quite a bit longer, to bring down the upper confidence interval limits on  $\hat{R}$  for  $\psi$  and  $N$ . If you have 3 cores then running these 3 parallel chains should not have taken longer than running a single chain. Yet if you look at the effective sample size now using `effectiveSize`, you can see that it has roughly tripled, as we would expect:

```
14749 > effectiveSize(window(res, start=1001))
14750
14751     sigma      lam0      psi        N
14752 272.6935 411.8384 167.4192 168.3355
```

### 17.8.2 Using C++

Parallel computing is a great tool to speed up computations, but its usefulness is limited by how many cores you have available. Even with a decent number of cores, large models may still take a long time to run. A major reason for this is that for-loops in **R** are time consuming, whereas they are handled much more time efficiently in other computer languages such as **C++**. As we saw above, MCMC algorithms consist of for-loops within for-loops, so that it stands to reason that implementing them in a language like **C++** should make those algorithms run much faster. Being avid **R** users, we cannot claim to be fluent in **C++** or to be aware of all the opportunities this language brings for faster computing. It is also beyond the scope of this book to go into the nuts and bolts of how **C++** works or provide a tutorial, and we refer you to the vast amounts of online and print material designed to give the interested user an introduction to **C++**. Just google “introduction **C++**” and you are sure to come across sites such as <http://www.cplusplus.com> that provide step by step instructions to get you started. Here, we only want to point out one approach to linking **R** with **C++**: the packages `inline` (Sklyar et al., 2010) and `RcppArmadillo` (Fran ois et al., 2011). These two packages provide a very convenient interface between the two languages, but there are other ways of calling **C++** functions from within **R**, such as the `.Call` command. If you are interested, we suggest you refer to the package manuals and vignettes, as well as the online document “Writing R extensions” (at <http://cran.r-project.org/doc/manuals/R-exts.html>) for a much more thorough treatment of this topic.

In order to use **C++** you need a compiler such as `g++` that (together with other compilers, for example for **C** and **FORTRAN**) comes with **Rtools**, which you can easily download from the web (at <http://cran.r-project.org/bin/windows/Rtools/>). All of these compilers are part of the GNU compiler collection (<http://gcc.gnu.org/>). Make sure the version of **Rtools** matches your version of **R** or you may run into compilation errors later on. To give you a taste of **C++** we will show you how to write a function that calculates the squared distances of individual activity centers to all traps, as is implemented in the `scrbook` package in the function `e2dist` (to be exact, `e2dist` calculates the distance, not the squared distance), and compare performance between **R** and **C++**. We will refer to these functions as “distance functions”. First, let us set up dummy data – a matrix holding the coordinates of the trap array, outer limits of the state-space and uniformly distributed activity centers for  $M = 700$  individuals:

```
14791 > gx<-seq(1,10,1)
```

```

14792 > gy<-seq(1,10,1)
14793 > X<-as.matrix(expand.grid(gx, gy))
14794 > M<-700
14795 > J<-dim(X)[1]
14796 > b<-3
14797 > xl<-min(gx)-b
14798 > xu<-max(gx)+b
14799 > yl<-min(gy)-b
14800 > yu<-max(gy)+b
14801 > S<-cbind(runif(M, xl, xu), runif(M, yl,yu))

```

14802 Next, we can write a “pedestrian” version of `e2dist` and check how long it takes to  
14803 calculate the squared distance matrix:

```

14804 > Dfun<-function(M, J, S, X){
14805 + D2<-matrix(0, nrow=M, ncol=J)
14806 + for (i in 1:M){
14807 + for(j in 1:J){
14808 + D2[i,j]<-(S[i,1]-X[j,1])^2 + (S[i,2]-X[j,2])^2
14809 + }
14810 + return(D2)
14811 +
14812
14813 > system.time(
14814 + (D2R<-Dfun(M, J, S, X)))
14815 +
14816
14817 user   system elapsed
14818     0.81    0.01    0.82

```

14819 The code to implement the same function in **C++** using the `inline` and `RcppArmadillo`  
14820 packages is shown in panel 17.3. These packages allow you to use a range of data formats  
14821 such as lists and matrices, and they take care of compiling the code in **C++** and loading  
14822 the resulting function into **R**. This is also referred to compiling **C++** code “on the fly”.  
14823 You will see that the way the code is set up is reasonably similar to **R**. One difference  
14824 that is worthy to point out is that in **C++** indices for vectors range from 0 to  $n - 1$ , NOT  
14825 from 1 to  $n$ , as in **R**. Note that with `inline` we only need to write the core of the code and  
14826 define the type of the variables we want to pass to the function, while the `cxxfunction`  
14827 call takes care of the rest. Once your function is compiled and loaded you should check  
14828 out the full **C++** code by calling `DfunArma@code`.

14829 Executing this code shows that it is faster than the **R** version of the distance function  
14830 or `e2dist`; in fact it is too fast for the time resolution of the `system.time()` function to  
14831 even give us a time estimate:

```

14832 > system.time(
14833 + (out<-DfunArma(M,J,S,X)))
14834
14835 user   system elapsed
14836     0       0       0

```

14837 While speed differences of less than 1 second may seem negligible, remember that  
14838 each command has to be executed at each iteration of the Markov chain. Especially with  
14839 time-consuming models such as those for open populations (Chapt. 16) or multi-session  
14840 models (Chapt. 14) we believe that **C++** holds large potential to make implementation  
14841 of such models more feasible.

## 17.9 SUMMARY AND OUTLOOK

14842 In a nutshell, programs like **JAGS** and **WinBUGS** do all the MCMC-related things that  
14843 we went through in this chapter (and quite a bit more). Looking through your model,  
14844 they determine which parameters they can use standard Gibbs sampling for (i.e. for  
14845 conjugate full conditional distributions). Then, they determine whether to use adaptive  
14846 rejection sampling, slice sampling or – in the ‘worst’ case – Metropolis-Hastings sampling  
14847 for the other full conditionals (how the sampler is chosen differs among softwares). For  
14848 MH sampling, they will automatically tune the updater so that it works efficiently.

14849 Although these programs are flexible and extremely useful to perform MCMC simulations,  
14850 it sometimes is more efficient to develop your own MCMC algorithm. Building an  
14851 MCMC code follows three basic steps: Identify your model including priors and express  
14852 full conditional distributions for each model parameter. If full conditionals are parametric  
14853 distributions, use Gibbs sampling to draw candidate parameter values from those dis-  
14854 tributions; otherwise use Metropolis-Hastings sampling to draw candidate values from  
14855 a proposal distribution and accept or reject them based on their posterior probability  
14856 densities.

14857 These custom-made MCMC algorithms give you more modeling flexibility than ex-  
14858 isting software packages, especially when it comes to handling the state-space: In **Win-**  
14859 **BUGS** and **JAGS** we define a continuous rectangular state-space using the corner coor-  
14860 dinates to constrain the uniform priors on the activity centers **s**. But what if a continuous  
14861 rectangle is an inadequate description of the state-space? In this chapter we saw that in  
14862 **R** it only takes a few lines of code to use any arbitrary polygon shapefile as the state-  
14863 space, which is especially useful when you are dealing with coastlines or large bodies of  
14864 water that need removing from the state-space. Another example is the SCR **R** package  
14865 **SPACECAP** (Gopalamswamy et al., 2012a) that was developed because implementation of an  
14866 SCR model with a discrete state-space was inefficient in **WinBUGS**.

14867 Another situations in which using a **BUGS** engine becomes increasingly complicated  
14868 or inefficient is when using point processes other than the homogeneous binomial point  
14869 process (“uniformity of density”) which underlies the basic SCR model (see sec. 5.10  
14870 in Chapt. 5). In Chapt. 11 you already saw an example of an inhomogeneous point  
14871 process model and we briefly introduce a different point processes, implemented using a  
14872 custom-made MCMC algorithm, in Chapt. 20. Finally, Chapt. 19 deals with partially  
14873 marked populations using hand-made MCMC algorithms to handle the (partially) latent  
14874 individual encounter histories. While some of these models can be written in the**BUGS**  
14875 language, they are painstakingly slow; others (for example the classes of models considered  
14876 in Chapt. 12) cannot be implemented in **WinBUGS/JAGS** at all and we have to either  
14877 use likelihood based inference or develop our own MCMC algorithms. In conclusion, while  
14878 you can certainly get by using **BUGS/JAGS** for standard SCR models, knowing how to  
14879 write your own MCMC sampler gives you more flexibility to tailor these models to your

14880 specific needs.

---

```
### calculate squared distances using RcppArmadillo
library(inline)
library(RcppArmadillo)

#write core of function code
code<-'
/*define input, assign correct class (matrix, vector etc)*/
arma::mat Sn=Rcpp::as<arma::mat>(S);
arma::mat Xn=Rcpp::as<arma::mat>(X);
int Ntot=Rcpp::as<int>(M);
int ntraps=Rcpp::as<int>(J);
/*create matrix to hold squared distances*/
arma::mat D2(Ntot, ntraps);

/*loop over M and J to calculate distances*/
for (int i=0; i<Ntot; i++){
  for(int j=0; j<ntraps; j++){
    D2(i,j)= pow(Sn(i,0)-Xn(j,0), 2) + pow(Sn(i,1)-Xn(j,1), 2);
  }
}
/*return D2 in R format*/
return Rcpp::wrap(D2);
'

# compile and load
DfunArma<-cxxfunction(signature(M="integer", J="integer", S="numeric",
X="numeric"), plugin="RcppArmadillo", body=code)
```

---

Panel 17.3: Code to compute squared distance between individual activity centers and traps in **C++** from within **R** using **inline** and **RcppArmadillo**



14881  
14882

14883

# 18

---

## UNMARKED POPULATIONS

14884 Traditional capture-recapture models share the fundamental assumption that each individual in a population can be uniquely identified when captured. Often, this can be  
14885 accomplished by marking individuals with color bands, ear tags, or some other artificial  
14886 mark that subsequently can be read in the field. For other species, such as tigers (*Panthera*  
14887 *tigris*) or marbled salamanders (*Ambystoma opacum*), individuals can be identified using  
14888 only their natural markings. However, many species do not possess adequate natural mark-  
14889 ings and are difficult to capture, making it impractical to use standard capture-recapture  
14890 techniques.

14892 Estimating density when individuals are unmarked can be accomplished using a variety  
14893 of alternatives to capture-recapture, such as distance sampling (Buckland et al., 2001) and  
14894 *N*-mixture models (Royle, 2004b). These methods, among others, can be very effective  
14895 when their assumptions are met, but in cases such as when it is not possible to obtain  
14896 accurate distance data, or when movement complicates the use of fixed-area plots, these  
14897 methods may not yield unbiased estimates of density (Chandler et al., 2011). Furthermore,  
14898 some species are so rare and cryptic that it is nearly impossible to collect enough data  
14899 using traditional survey methods.

14900 In this chapter, we investigate spatially explicit alternatives for estimating density  
14901 of unmarked populations, and we highlight the work of Chandler and Royle (2013) who  
14902 demonstrated that the “individual recognition” assumption of traditional capture-recapture  
14903 models is not a requirement of spatial capture-recapture models. They showed that, under  
14904 certain conditions, spatially correlated count data are sufficient for making inference about  
14905 animal distribution and density even when no individuals are marked. The Chandler and  
14906 Royle (2013) “spatial count model” (hereafter the SC model) requires neither distance data  
14907 nor fixed area plots. Instead, the observed data are trap- and occasion-specific counts,  
14908 which are modeled as a reduced-information summary of the *latent* encounter histories.  
14909 Because the model is formulated in terms of the data we wish we had, i.e. the typical  
14910 encounter history data observed in standard capture-recapture studies of marked animals,  
14911 the SC model is just a SCR model with a single extension to account for the fact that  
14912 the encounter history data are unobserved. However, this results in a drastically different  
14913 model than the models typically used for count data in ecology because the SC model

14914 is parameterized in terms of individuals, and specifically, their locations relative to the  
14915 sampling device.

14916 The ability to fit SCR models to data from unmarked populations has important  
14917 implications. For one, it means that SCR models can be applied to data collected using  
14918 methods like points counts in which observers record simple counts of animals at an array  
14919 of survey locations. The model can also be fitted to camera trapping data collected on  
14920 unmarked animals, representing one of the first formal method for estimating density  
14921 from such data (but see ?). So, is the SC model a free lunch? At face value, it sounds  
14922 as though it allows for estimation of all the quantities of interest in standard capture-  
14923 recapture studies, but with very little data. But of course the answer is no – lunch is  
14924 still not free because with this model come new assumptions, and as was demonstrated by  
14925 Chandler and Royle (2013), even with “perfect” data, parameter estimates will typically  
14926 not be very precise. This should not be surprising given that we are asking so much from  
14927 simple count data.

14928 The real value of the SC model is two-fold. First, it demonstrates an important  
14929 theoretical result, namely that spatial correlation in count data carries information about  
14930 density and distribution – and this stands in stark contrast to a prevailing view of spatial  
14931 correlation as a nuisance to be avoided or modeled out of unsightly residual plots. The  
14932 second reason why this model is important is that it provides the basis for numerous model  
14933 extensions that *can* result in precise estimates of density. We will discuss some of these  
14934 possibilities in this chapter, but perhaps the most useful extension – accomodating data  
14935 from marked and unmarked individuals – is treated separately in the next chapter. Here,  
14936 we focus on situations in which all individuals are unmarked, and we begin by presenting  
14937 the most basic formulation of the model. Then we proceed, by way of a few examples, to  
14938 consider extensions of the model in which ancillary information can be used to increase  
14939 precision.

## 18.1 EXISTING MODELS FOR INFERENCE ABOUT DENSITY IN UNMARKED POPULATIONS

14940 When capture-recapture methods are not a viable option, ecologists often collect simple  
14941 count data or even binary detection/non-detection data. These data are often treated  
14942 as an index of abundance or occurrence and are analyzed using generalized linear mod-  
14943 els such as Poisson regression or logistic regression, perhaps with random effects (Zuur  
14944 et al., 2009). However, index methods cannot be used to make unbiased inferences about  
14945 abundance or occurrence unless strong assumptions about constant detection probability  
14946 are valid (Williams et al., 2002; ?). In particular, index methods can be highly mis-  
14947 leading when covariates affect both the ecological process of interest and the observation  
14948 process. A classic example is given by Bibby and Buckland (1987) who found that song-  
14949 bird detection probability was negatively related to vegetation height, whereas density  
14950 was positively associated with vegetation height in restocked conifer plantations. This  
14951 intuitive phenomenon has been demonstrated repeatedly (e.g. Kéry, 2008; Sillett et al.,  
14952 2012) and has led to the development of a vast number of models to estimate population  
14953 size and occurrence probability when individuals are unmarked and detected imperfectly  
14954 (Buckland et al., 2001; Williams et al., 2002; MacKenzie et al., 2006; Royle and Dorazio,  
14955 2008). A review of these models is beyond the scope of this chapter, but we mention a

## EXISTING MODELS FOR INFERENCE ABOUT DENSITY IN UNMARKED POPULATIONS

14956 few deficiencies of existing methods that warrant the exploration of alternatives for robust  
14957 inference when standard capture-recapture methods do not apply.

14958 Distance sampling (Buckland et al., 2001; Buckland, 2004), which we briefly introduced  
14959 in Chapter 4, is perhaps the most widely used method for estimating population density  
14960 when individuals are unmarked and detection probability is less than one. This class of  
14961 methods is known to work impeccably when estimating the number of stakes in a field or  
14962 the number of duck nests in a wetland. Distance sampling can also work very well in more  
14963 interesting situations, and it is an extremely powerful method when the assumptions can  
14964 be met. However, the assumptions that distance data can be recorded without error and  
14965 that animals are distributed randomly with respect to the transect can be easily violated by  
14966 common processes such as animal movement and measurement error. Although numerous  
14967 methods have been proposed to relax some of these assumptions Royle et al. (2004);  
14968 Borchers et al. (1998); Johnson (2010); ?); Chandler et al. (2011), a more important issue  
14969 is that distance sampling is simply not practical in many settings. For example, many  
14970 species are so rare and elusive that they can only be reliably surveyed using “indirect”  
14971 methods such as camera traps or hair snares.

14972 In response to the increasing use of camera traps in studies of threatened species, and  
14973 the problems associated with commonly-used indices of abundance (?O’Brien, 2011; ?),  
14974 several density estimators have been developed for situations in which the species being  
14975 studied is unmarked ???. These estimators assume that (1) cameras are randomly placed  
14976 with respect to animal density (2) animals neither avoid nor are attracted to the cameras,  
14977 and (3) detection probability can be either modeled as a function of distance between the  
14978 animal and the camera or as a function of movement velocity. Although these methods  
14979 represent an important improvement over index-based methods, the assumptions may not  
14980 hold in many situations, especially when applied to data from standard designs in which  
14981 camera stations are either baited or placed along trails – issues that can be dealt with  
14982 directly using SCR models (see Chaps. 12 and 13). Nonetheless, empirical studies have  
14983 found that the assumptions do hold in some cases (?).

14984 Other common approaches to estimate density when individuals are unmarked include  
14985 double-observer sampling, removal sampling, and repeated counts, for which custom mod-  
14986 els have been developed (Nichols et al., 2000b; Farnsworth et al., 2002; Royle, 2004b,a;  
14987 Nichols et al., 2009; Fiske and Chandler, 2011). To obtain reliable density estimates using  
14988 these methods, the area surveyed must be well defined and closed with respect to move-  
14989 ment and demographic processes. Given a sufficiently short sampling interval, such as a  
14990 5-min point-count, the closure assumption may be reasonable. However, short sampling  
14991 intervals limit the number of detections, so observers generally visit each survey location  
14992 multiple times during a season. But then, animal movement may invalidate the closure  
14993 assumption, and a model of temporary emigration is required (Kendall et al., 1997; Chan-  
14994 dler et al., 2011). Furthermore, distance-related heterogeneity in detection probability can  
14995 introduce bias in these models, although this bias is negligible when the ratio of plot size  
14996 to the scale parameter of the detection function is low (Efford and Dawson, 2009).

14997 We mention these issues not to suggest that existing models do not have value – indeed  
14998 we believe that they can be used to obtain reliable density estimates in many situations –  
14999 rather, our aim is to highlight the need for alternative methods when the assumptions of  
15000 existing methods cannot be met and when spatially-explicit inference is the objective.

## 18.2 SPATIAL CORRELATION IN COUNT DATA

### 18.2.1 Spatial correlation as information

15002 All of the previous methods require some sort of auxiliary information to model both  
15003 abundance and detection. For instance, multiple observers or distance data or repeated  
15004 visits may be required to ensure that model parameters are identifiable (but see (Lele  
15005 et al., 2012; Sólymos et al., 2012)). The same is true for the SC model, but the auxiliary  
15006 information comes in the form of spatial correlation, which requires no extra effort to  
15007 collect (Chandler and Royle, 2013).

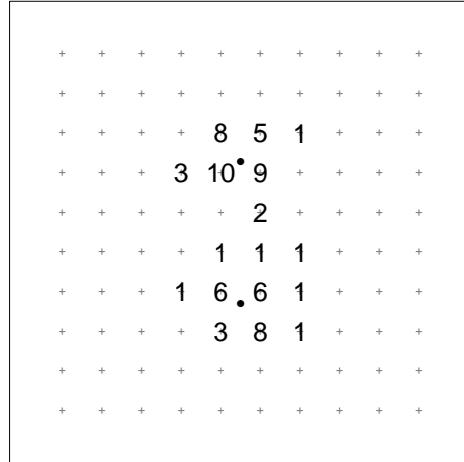
15008 It is natural to be suspicious of the claim that spatial correlation is a good thing. In  
15009 fact, elaborate methods have been devised to deal with spatial correlation as a nuisance  
1510 (Lichstein et al., 2002; Dormann et al., 2007), and ecologists have been admonished for  
1511 failing to obtain “real” replicates uncontaminated by spatial correlation (Hurlbert, 1984).  
1512 The following heuristic may be helpful for seeing the value of spatial correlation.

15013 Imagine a  $10 \times 10$  grid of camera traps and a single unmarked individual exposed to  
15014 “capture” whose home range center lies in the center of the trapping grid. If the individual  
15015 has a small home range size relative to the extent of the trapping grid, we can imagine what  
15016 the spatial correlation structure of the encounters might look like. If the animal’s home  
15017 range is symmetric around the activity center then the number of times the individual  
15018 is detected at each trap (the trap count) is a function of the distance between the home  
15019 range center and the trap; i.e., traps with the same distance from the activity center  
15020 will yield counts that are more highly correlated with one another than traps located at  
15021 different distances from the activity center. Thus, the correlation among the counts tells  
15022 us something about the location of the activity center. It is relatively intuitive that spatial  
15023 correlation carries information about distribution, but what about density?

15024 Imagine now that there are two activity centers located in the trapping grid. Using  
15025 trap counts alone, is it possible to determine the number and location of these activity  
15026 centers? The answer is yes, at least under certain circumstances. Figure 18.1 shows the  
15027 locations of the two hypothetical activity centers, and the total counts obtained at each  
15028 trap after 10 survey occasions. Assuming that animals have bivariate normal home ranges,  
15029 the fact that there are two areas in the map with high counts that dissipate with distance  
15030 suggests that the most likely number of individuals given these data is 2. Furthermore, the  
15031 degree to which the counts dissipate from the two areas of highest intensity is information  
15032 about the home range size parameter  $\sigma$ . These two pieces of information are enough to  
15033 estimate the number of individuals exposed to sampling – again, given that a bivariate  
15034 normal home range is a valid assumption. Of course, the data could just as well have  
15035 been generated by a single individual whose home range is distinctly bimodal, and thus *as*  
15036 *always* the assumptions of our model need to be carefully examined using our biological  
15037 knowledge of the system. If the assumptions do not hold, it is almost always possible to  
15038 relax them, for instance by allowing for non-stationary home ranges as we demonstrated  
15039 in Chapt. 12 and 13.

### 15040 18.2.2 Two types of spatial correlation

15041 The spatial correlation dealt with by the SC model is assumed to arise from animal  
15042 movement; however, this is just one type of spatial correlation that may exist in ecological



**Figure 18.1.** Simulated count data at each of 100 camera traps (crosses) after  $K = 10$  sampling occasions. The black dots are the locations of two animal activity centers. The spatial count model estimates both the location and number of activity centers exposed to sampling using such spatially-referenced count data.

15043 count data. Another common type of spatial correlation results from the spatial correlation  
 15044 of environmental covariates. Habitat variables, such as, for example, the percent cover of  
 15045 deciduous forest in North America, will often be patchy rather than randomly distributed,  
 15046 and this can result in spatial correlation in abundance, and hence in count data. Often,  
 15047 this type of spatial correlation can be dealt with by simply including the habitat covariate  
 15048 in the model. For example, a simple species distribution model with only a few habitat  
 15049 variables can result in a distribution map that reflects the spatial correlation in abundance  
 15050 (Sillett et al., 2012; Royle et al., 2012b). In such a case, there is no need to use spatially-  
 15051 explicit models (Besag and Kooperberg, 1995; Lichstein et al., 2002; Wikle, 2010). The  
 15052 reason is that the relevant assumption of non-spatial models (e.g. GLMs) is that no  
 15053 spatial correlation exists in the *residuals*, and often, any apparent spatial correlation can  
 15054 be accounted for using covariates. This may be obvious, but it is a point that seems to  
 15055 be frequently misunderstood.

15056 It does become important to account for environmentally-induced spatial correlation  
 15057 in situations where correlation exists in the residuals *after* accounting for covariate effects.  
 15058 This may be due to unobserved covariates, and Zuur et al. (2009) offer advice on how to  
 15059 check for and deal with such correlation in the context of GLMs and GLMMs. SCR models,  
 15060 including the SC model dealt with in this chapter, explicitly account for movement-induced  
 15061 spatial correlation, and they can also be used to account for environmentally-induced  
 15062 spatial correlation by adopting an inhomogeneous point process model for the activity  
 15063 centers. That is, the point process intensity can be modeled as a function of observed  
 15064 covariates, and theoretically, it should be possible to allow for spatially-correlated random

15065 effects to deal with unobserved covariates. See Chapt. 11 for details.

## 18.3 SPATIAL COUNT MODEL

### 15066 18.3.1 Data

15067 Whereas traditional SCR models require spatially-referenced individual encounter histo-  
 15068 ries, the SC model requires simple spatial count data. Let  $n_{jk}$  be the count data at  
 15069 sampling location  $j$  on occasion  $k$ . The  $J \times K$  matrix of counts will be denoted  $\mathbf{n}$ . A  
 15070 sampling location in this context could be any device capable of recording count data,  
 15071 such as a human observer or a camera trap, and one of the benefits of the SC model is  
 15072 that it can be applied to data collected using many different survey methods. For ease of  
 15073 presentation, we will refer to sampling devices as traps, but remember that a trap is just  
 15074 something capable of recording count data. As in all SCR models, we also require the  
 15075 coordinates of the  $J$  traps, and we denote the location of trap  $j$  by  $\mathbf{x}_j$ . In some instances,  
 15076 additional data might be available such as trap-specific covariates, state-space covariates,  
 15077 information on the identities of a subset of individuals, or perhaps even distance data.  
 15078 We consider some of these model extensions in Sec. 18.8, but for the time being we ignore  
 15079 these possibilities so that we can focus on the basic model.

### 15080 18.3.2 Model

15081 The state model is exactly the same as the one we have dealt with throughout this book.  
 15082 It is a point process describing the number and distribution of activity centers in the  
 15083 state-space  $\mathcal{S}$ . Although it might be possible to fit inhomogeneous point process models  
 15084 using the methods described in Chapt. 11, given the simplicity of the data, we concentrate  
 15085 on a homogeneous point process  $\{\mathbf{s}_i, \dots, \mathbf{s}_N\} \sim \text{Uniform}(\mathcal{S})$  where  $\mathbf{s}_i$  is the activity center  
 15086 of individual  $i$  in the population of size  $N$ . For the moment, we will assume that  $N$  is  
 15087 known.

15088 The observation model is the same as in other SCR models in the sense that it de-  
 15089 scribes the probability of encountering individual  $i$  at trap  $j$ , conditional on the location  
 15090 of the individual's activity center. The specific encounter process will depend on the sam-  
 15091 pling method, and here we consider the standard camera trapping situation in which an  
 15092 individual can be encountered at multiple traps during a single occasion, say one night  
 15093 during a camera-trapping study, and it can be detected multiple times at a single trap  
 15094 during an occasion. This is the Poisson encounter model (a.k.a. the count detector case)  
 15095 described in Chapt. 9. Our experience with alternative observation models such as the  
 15096 Bernoulli and multinomial models suggests that the parameters of the model may not be  
 15097 identifiable in these cases, at least when no additional information is available. This is a  
 15098 subject of ongoing research.

15099 As before, we define  $y_{ijk}$  as the encounter data for individual  $i$  at trap  $j$  on occasion  
 15100  $k$ , which we model as:

$$y_{ijk} \sim \text{Poisson}(\lambda_{ij}) \quad (18.3.1)$$

15101 where  $\lambda_{ij}$  is the encounter rate. A common encounter rate model is the Gaussian, or  
 15102 half-normal, model:

$$\lambda_{ij} = \lambda_0 \exp(-\|\mathbf{x}_j - \mathbf{s}_i\|/2\sigma^2)$$

15103 in which  $\lambda_0$  is the baseline encounter rate,  $\|\mathbf{x}_j - \mathbf{s}_i\|$  is the Euclidean distance between the  
 15104 trap and activity center, and  $\sigma$  is the scale parameter determining the degree to which  
 15105 encounter rate decreases with distance. In this context,  $\sigma$  also determines the amount of  
 15106 correlation among the counts because if  $\sigma$  is low relative to the trap spacing, then it is  
 15107 unlikely that an individual will be detected at multiple traps.

15108 When individuals cannot be uniquely identified, the encounter histories cannot be di-  
 15109 rectly observed, which seems like a massively insurmountable problem of epic proportions.  
 15110 The solution of Chandler and Royle (2013) is the same one we routinely apply when we  
 15111 cannot directly observe the process of interest – we regard the encounter histories as latent  
 15112 variables. This leaves the remaining task of specifying the relationship between the count  
 15113 data and the encounter histories, i.e. we need a model of  $[\mathbf{n}|\mathbf{y}]$  where  $\mathbf{y}$  represents the  
 15114 entire collection of encounter histories. In this case, there is only one possibility because,  
 15115 by definition, the count data are simply a reduced-information summary of the latent en-  
 15116 counter histories. That is, they are the sample- and trap-specific totals, aggregated over  
 15117 all individuals:

$$n_{jk} = \sum_{i=1}^N y_{ijk}. \quad (18.3.2)$$

15118 So, unlike most model-development problems faced in this book, we don't have to con-  
 15119 sider competing probability models for  $[\mathbf{n}|\mathbf{y}]$ , but instead, we recognize the fact that the  
 15120 relationship between the counts and the latent encounter histories is deterministic. This  
 15121 deterministic constraint poses some computational challenges, which we discuss below.  
 15122 But first we present some alternative formulations of the model.

15123 Recall from Chapt. 2 that the sum of two or more Poisson random variables is also  
 15124 a Poisson random variable. Specifically, if  $x_1 \sim \text{Poisson}(\lambda_1)$  and  $x_2 \sim \text{Poisson}(\lambda_2)$ , then  
 15125  $(x_1 + x_2) \sim \text{Poisson}(\lambda_1 + \lambda_2)$ . Thus, under this Poisson model for the latent encounter  
 15126 histories, the count data can be modeled as Poisson:

$$n_{jk} \sim \text{Poisson}(\Lambda_j) \quad (18.3.3)$$

15127 where

$$\Lambda_j = \lambda_0 \sum_i \exp(\|\mathbf{x}_j - \mathbf{s}_i\|/2\sigma^2),$$

15128 and because  $\Lambda_j$  does not depend on  $k$ , we can aggregate the replicated counts, defining  
 15129  $n_{j\cdot} = \sum_k n_{jk}$  and then

$$n_{j\cdot} \sim \text{Poisson}(K\Lambda_j).$$

15130 As such,  $K$  and  $\lambda_0$  serve equivalent roles as affecting baseline encounter rate. Formulating  
 15131 the model in terms of the aggregated count data demonstrates that the model can be  
 15132 applied to data from a single sampling occasion ( $J \equiv 1$ ) as has been noted elsewhere for  
 15133 standard SCR models (Efford et al., 2009b). In the context of studying marked popula-  
 15134 tions, the model parameters will only be identifiable in the  $J \equiv 1$  case if an animal can be  
 15135 captured at multiple traps during a single occasion. The SC model essentially requires the  
 15136 same thing, which is to say that it requires correlation in the count data resulting from  
 15137 an individual being captured in multiple, closely-spaced traps.

15138 This formulation of the model in terms of the aggregate count also simplifies computa-  
 15139 tions as the latent encounter histories do not need to be updated in the MCMC estimation

15140 scheme; however, retaining them in the formulation of the model is important if some in-  
 15141 dividuals are uniquely marked. This is because uniquely identifiable individuals produce  
 15142 observations of some of the  $y_{ijk}$  variables, which we elaborate on in the subsequent chapter.

## 18.4 HOW MUCH CORRELATION IS ENOUGH?

15143  $\sigma$  shouldn't be too small or too large relative to trap spacing or else the counts will be  
 15144 i.i.d. Poisson random variables. So how much correlation is enough? Phrased differently,  
 15145 what is the ideal ratio of  $\sigma$  to trap spacing to ensure correlation and minimize the variance  
 15146 of the posterior distributions. We see two options for answering this questions, both of  
 15147 which are topics in need of additional research. The first approach is to use the methods  
 15148 described in Chapt. 10, i.e. by either conducting simulation studies with various trap  
 15149 spacing to  $\sigma$  ratios, or to analytically minimize a variance criterion for a given set of  
 15150 sampling conditions and effort. The former approach was used by Chandler and Royle  
 15151 (2013) whose limited simulation study indicated that an ideal ratio is approximately 2.  
 15152 This agrees with findings from previous research on the optimal design of SCR studies  
 15153 (Chapt. 10), as it should.

15154 A second approach that may be of use if a data set has already been collected is to  
 15155 use standard techniques from spatial statistics to determine if adequate correlation exists  
 15156 in the counts. For example, one might compute Ripley's  $K$ -statistic or generate (semi-  
 15157 )variograms (Illian et al., 2008). We have not studied the utility of such approaches, but  
 15158 it seems worthy of investigation.

### 15159 18.4.1 On $N$ being unknown

Population size,  $N$ , is never known in practice, and thus we need a model for it. For homogeneous point process models,  $N$  is typically modeled as  $N \sim \text{Poisson}(\mu|\mathcal{S}|)$  or  $N \sim \text{Binomial}(M, \psi)$ , the latter of which is equivalent to a discrete uniform prior on  $N$  if  $\psi \sim \text{Uniform}(0, 1)$ . In Chapt. 11 and elsewhere, we demonstrated that the choice of prior has very little influence on parameter estimates, and so we favor the binomial prior because of its convenience when using MCMC, i.e. it allows us to fix the dimensions of the parameter space by setting  $M$  to some arbitrarily large integer. A binomial model is equivalent to a series of  $M$  independent Bernoulli trials, hence we can rewrite  $N \sim \text{Binomial}(M, \psi)$  as  $z_i \sim \text{Bernoulli}(\psi)$  where  $z_i$  is an auxiliary variable indicating if individual  $i$  is a member of the population, i.e.  $N = \sum_{i=1}^M z_i$ . Having expanded the model to include a prior on  $N$ , we can summarize the SC model, with a Gaussian observation model, as follows:

$$\begin{aligned} z_i &\sim \text{Bernoulli}(\psi) \\ y_{ijk} &\sim \text{Poisson}(\lambda_{ijk} z_i) \\ \lambda_{ijk} &= \lambda_0 \exp(-\|\mathbf{x}_j - \mathbf{s}_i\|^2)/(2\sigma^2) \\ n_{jk} &= \sum_{i=1}^M y_{ijk} \end{aligned}$$

15160 Bayesian analysis can proceed once suitable priors have been put on the hyperparameters  
 15161  $\psi$ ,  $\sigma$ , and  $\lambda_0$ . Chandler and Royle (2013) provided R code for fitting the model using

15162 MCMC, and they evaluated the model's performance with uniform priors on the three hyperparameters. They also discussed the possibilities and effects of including prior knowledge about  $\sigma$  into the model. In the next section, we explain how the model can be implemented using **JAGS**, but first we briefly contemplate the viability of classical analysis of this model.

15167 The obvious challenge faced when conducting a classical analysis of this model is that  
 15168 the number of latent variables is huge. In all SCR models, the activity centers are latent,  
 15169 but now, even the encounter histories are latent. Maximizing likelihoods with latent  
 15170 variables (random effects) involves integrating (or summing) over all possible values of  
 15171 the latent variables. For the activity centers, this is typically accomplished by integrating  
 15172 the conditional-on-s likelihood  $[y_i|s_i]$  over the two-dimensional state-space  $\mathcal{S}$  (Chapt. 6).  
 15173 However, with the SC model, we have to sum over all possible encounter histories meeting  
 15174 the constraint of Eq. 18.3.2. The number of possible encounter histories will, in general,  
 15175 be too high to make the likelihood tractable, and thus we do not think that maximum  
 15176 likelihood is a viable option for analyzing this model. However, one might be able to  
 15177 obtain approximate maximum likelihood estimates using simulation-based methods (Lele  
 15178 et al., 2010), which will typically be more computationally challenging than the Bayesian  
 15179 analysis.

## 18.5 SIMULATION EXAMPLE

15180 Simulating data under the SC model proceeds by first simulating standard SCR encounter  
 15181 history data and then collapsing it into count data. The following blocks of **R** code  
 15182 generate data from the model shown in Sec. 18.4.1, with parameters  $\sigma = 5$ ,  $\lambda_0 = 0.4$ , and  
 15183  $N = 50$ . The state-space is a  $[0, 100] \times [0, 100]$  square, and a grid of 100 traps is centered  
 15184 in the middle. The first block of code generates the trap coordinates  $X$  and the  $N = 50$   
 15185 activity centers:

```
15186 > tr <- seq(15, 85, length=10)
15187 > X <- cbind(rep(tr, each=length(tr)),
15188 +           rep(tr, times=length(tr)))      # 100 trap coords
15189 > set.seed(10)
15190 > xlim <- c(0, 100); ylim <- c(0, 100)      # S is [0,100]x[0,100] square
15191 > A <- (xlim[2]-xlim[1])*(ylim[2]-ylim[1])/1e4 # Area of S
15192 > mu <- 50                                     # Density (animals/unit area)
15193 > N <- rpois(1, mu*A)                         # Generate N=50 as Poisson deviate
15194 [1] 50
15195 > s <- cbind(runif(N, xlim[1], xlim[2]), runif(N, ylim[1], ylim[2]))
```

15196 We could have set  $N = 50$  directly, but instead we treated density as a fixed parameter  
 15197 ( $\mu = 50$ ) and generated  $N$  as a random variable – it just so happens that with the specified  
 15198 random seed,  $N$  equals 50.

15199 Now we can generate the encounter histories under the Poisson observation model.  
 15200 Let's suppose that sampling is conducted over  $K = 5$  nights.

```
15201 > sigma <- 5
15202 > lam0 <- 0.4
```

```

15203 > J <- nrow(X)
15204 > K <- 5
15205 > y <- array(NA, c(N, J, K))
15206 > for(j in 1:J) {
15207 +   dist <- sqrt((X[j,1]-s[,1])^2 + (X[j,2] - s[,2])^2)
15208 +   lambda <- lam0*exp(-dist^2/(2*sigma^2))
15209 +   for(k in 1:K) {
15210 +     y[,j,k] <- rpois(N, lambda)
15211 +   }
15212 + }
```

15213 The object `y` is the  $N \times J \times K$  array of encounter data, which cannot be directly observed if the animals are unmarked. Converting the encounter data to count data can be  
 15214 accomplished using a single `apply` command.  
 15215

```

15216 > n <- apply(y, c(2,3), sum)
15217 > dimnames(n) <- list(paste("trap", 1:J, sep=""),
15218 +                         paste("night", 1:K, sep=""))
15219 > n[1:4,]
15220   night1 night2 night3 night4 night5
15221 trap1    1     0     0     0     0
15222 trap2    1     2     2     0     1
15223 trap3    1     0     0     1     0
15224 trap4    0     0     0     0     0
```

15225 This displays the first 4 rows of `n`, the  $J \times K$  matrix of counts. It is worth contemplating  
 15226 how common such count data is in ecology and how many different mechanisms might  
 15227 generate it. Although the list of possibilities is immense, the SC model has advantages over  
 15228 some alternatives in that it includes an explicit model for the distribution of individuals in  
 15229 space *and* it includes a model describing how detections are generated given the distance  
 15230 between traps and individual activity centers. It also provides a foundation for extending  
 15231 the model in many ways as we discuss in Sec. 18.8 and in the next chapter.

15232 The question now is: Is it possible to estimate the parameters? In our simulated  
 15233 dataset we have  $J \times K = 500$  data points, but how many parameters do we need to  
 15234 estimate with this rather small set of data? A frequentist might say that there are only  
 15235 3 parameters:  $\lambda_0$ ,  $\sigma$ , and  $N$  (or density  $\mu$ ) because inference about the latent parameters  
 15236 is carried out using prediction methods after the 3 hyperparameters have been estimated.  
 15237 However, a Bayesian would probably say that each `s` and each element of the latent  
 15238 encounter array `y` is a parameter in need of a posterior. From this perspective there are far  
 15239 more parameters than data points, and thus it would appear as though the situation is  
 15240 dire. Whether or not the parameters are actually estimable is a rather difficult question  
 15241 to answer. One simplistic, but not definitive, approach for addressing the question is  
 15242 to conduct a simulation study and evaluate the frequentist performance of the model by  
 15243 asking how often the data-generating values are included in confidence/credible intervals,  
 15244 and how biased are point estimates. Chandler and Royle (2013) conducted such a sim-  
 15245 ulation study and found that, while the variance of the posterior distributions was high  
 15246 by most standards, the bias of the posterior mode of  $N$  was small and the coverage of  
 15247 the credible intervals was close to nominal. Moreover, they found no evidence that the

15248 posterior distributions were dominated by the priors, further supporting the conclusion  
 15249 that spatial correlation in the count data is sufficient for estimating density and encounter  
 15250 probability parameters. However, in such cases where identifiability has not formally been  
 15251 demonstrated, it may be wise to compare the results of models fit using both proper and  
 15252 improper priors, as we do below.

15253 At this point in time the SC model can only be fit using one of the **BUGS** engines,  
 15254 or using custom software like the **R** code accompanying Chandler and Royle (2013). Although **BUGS**  
 15255 might provide the most flexible option for fitting the model, it is not straight-forward because of the constraints in the model. In **WinBUGS**, the  $n_{jk} = \sum_i y_{ijk}$   
 15256 constraint can be enforced using the so-called “ones-trick”, but we prefer **JAGS** because it has a distribution called **dsum** that was designed for this type of situation in which  
 15257 the observed data are a sum of random variables. Panel 18.1 shows the **JAGS** code, but  
 15258 we abbreviated the arguments to **dsum** because in practice you need to provide all  $M$  of  
 15259 them. The code looks slightly unwieldy if  $M$  is large, but you can easily create it using the  
 15260 **paste** function in **R**. Here is an example, with an unrealistically small value of  $M = 10$ :

```
15263 > paste("y[", 1:10, ",j,k]", sep="", collapse=", ")
15264 [1] "y[1,j,k], y[2,j,k], y[3,j,k], y[4,j,k], y[5,j,k], y[6,j,k], "
15265 y[7,j,k], y[8,j,k], y[9,j,k], y[10,j,k]"
```

15266 The **JAGS** model in Panel 18.1 can be used to fit the version of the model in which  
 15267 the latent encounters are updated at each Monte Carlo iteration. One challenge faced  
 15268 when using this version of the model is that **JAGS** cannot auto-generate initial values  
 15269 that honor the constraints in the model, so it is necessary to provide them. The following  
 15270 code presents one fairly general way of creating acceptable starting values and formatting  
 15271 the data for analysis using the **rjags** package:

```
15272 > library(rjags)
15273 > dat1 <- list(n=n, X=X, J=J, K=K, M=200, xlim=xlim, ylim=ylim)
15274 > init1 <- function() {
15275 +   yi <- array(0, c(dat1$M, dat1$J, dat1$K))
15276 +   for(j in 1:dat1$J) {
15277 +     for(k in 1:dat1$K) {
15278 +       yi[sample(1:dat1$M, dat1$n[j,k]),j,k] <- 1
15279 +     }
15280 +   }
15281 +   list(sigma=runif(1, 1, 2), lam0=runif(1),
15282 +       y=yi, z=rep(1, dat1$M))
15283 + }
15284 > pars1 <- c("lam0", "sigma", "N", "mu")
```

15285 The code in Panel 18.1 is useful because it shows how closely this model is related to  
 15286 standard SCR models, and it provides the basis for including data on both marked and  
 15287 unmarked individuals, as will be discussed in the next chapter. However, this model runs  
 15288 very slowly, even when using a fast 64-bit machine with chains run in parallel. The code  
 15289 in Panel 18.2 runs much faster because it does not include the latent encounter histories.

15290 An even faster (but perhaps less efficient) alternative is to use the **scrUN** function in  
 15291 **scrbook**. The usage is as follows:

**Table 18.1.** Posterior summaries from the spatial count (“SC”) model applied to simulated data using **scrbook** and **JAGS**. 25000 samples were generated, but substantial Monte Carlo error is still evident.

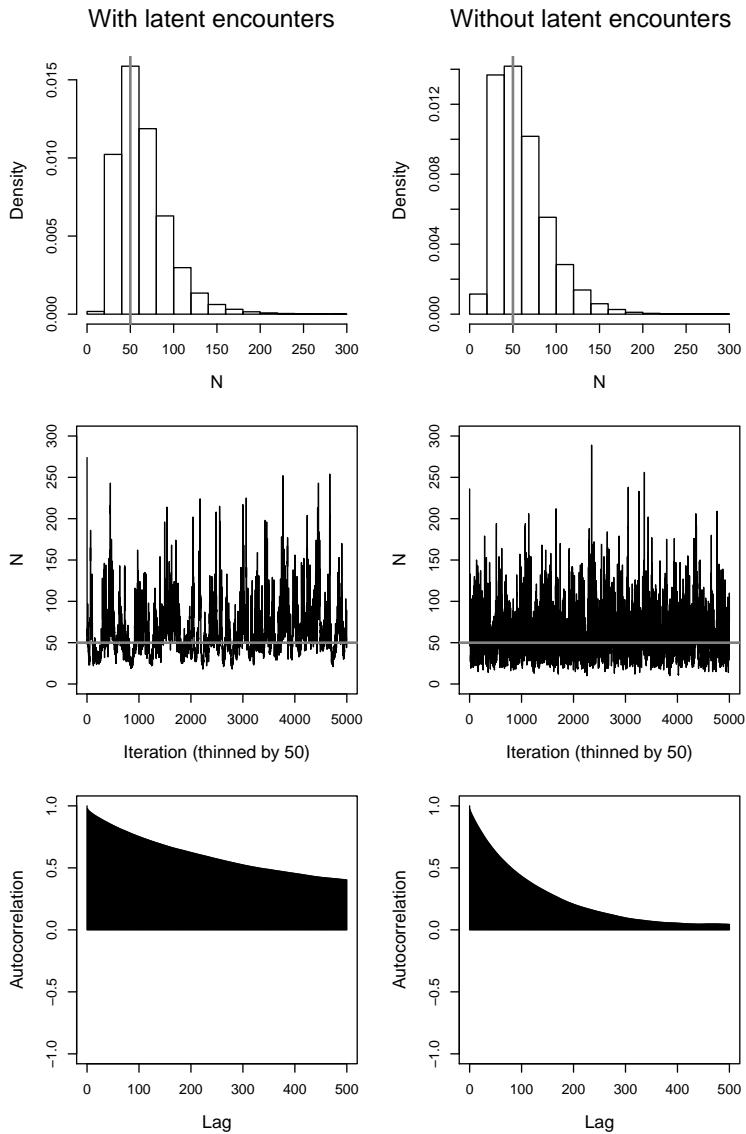
Parameter	Mean	SD	2.5%	50%	97.5%
scrUN(..., updateY=FALSE)					
$\sigma = 5$	4.718	0.922	3.239	4.615	6.833
$\lambda_0 = 0.4$	0.500	0.136	0.268	0.489	0.793
$N = 50$	60.653	31.067	21.000	54.000	137.000
scrUN(..., updateY=TRUE)					
$\sigma$	4.554	0.784	3.216	4.486	6.264
$\lambda_0$	0.489	0.131	0.262	0.479	0.775
$N$	64.772	30.162	26.000	59.000	140.000
<b>JAGS</b> (without latent encounter histories)					
$\sigma$	4.70	0.88	3.24	4.66	6.63
$\lambda_0$	0.52	0.14	0.27	0.52	0.80
$N$	58.55	30.30	20.00	52.00	135.00

```
15292 > out1 <- scrUN(n=n, X=X, M=300, niter=25000, xlims=xlim, ylims=ylim,
15293           inits=list(lam0=0.3, sigma=rnorm(1, 5, 0.1)), updateY=TRUE,
15294           tune=c(0.004, 0.09, 0.35))
```

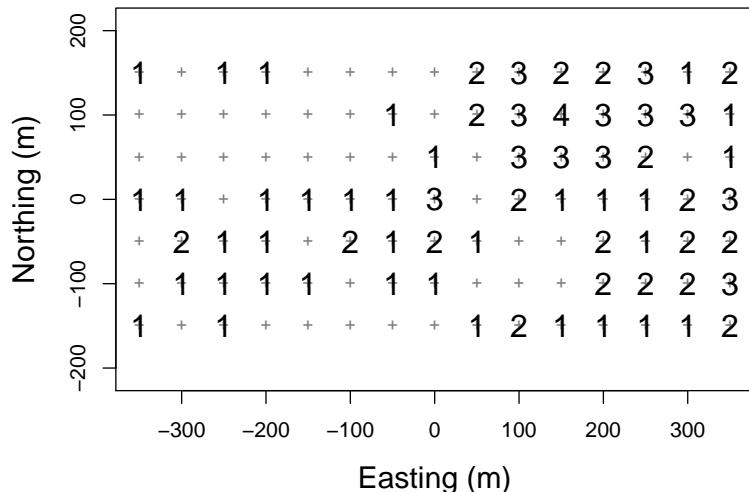
15295 where `n` is the matrix of counts, `X` is the trap coordinate matrix, `M` sets the size of the data-augmented latent data, `xlims` and `ylims` define the rectangular state-space, `inits` is a list of starting values, and `updateY` determines if the latent encounter histories are updated as part of the MCMC algorithm. In general, we recommend using the option `updateY=FALSE` because the Markov chains tend to mix better. Even so, it can be important to fiddle with the tuning parameters until the acceptance rates are between 40–60%. Otherwise, the Markov chains will exhibit extremely high autocorrelation. This is one reason to favor **JAGS** over our implementation in **scrbook** since **JAGS** finds suitable tuning parameters automatically during the adaptive phase (when using Metropolis updates).

15304 We fit the model to the simulated data using both formulations – with and without  
15305 the latent encounter histories – and the results are given in Table 18.1 and Fig. 18.2.  
15306 Table 18.1 shows summaries of 10000 posterior draws, and suggests that while the true  
15307 parameter values are easily covered by the 95% credible intervals, the intervals are rather  
15308 wide. In many cases, knowing that there are between 21 and 113 individuals in an area  
15309 will be considered relatively imprecise. This is not just a peculiarity of this particular data  
15310 set – in general, posterior precision will be low, as noted by Chandler and Royle (2013).  
15311 Furthermore, as indicated by Fig. 18.2, the autocorrelation of the samples is high, and  
15312 thus it may take many iterations to achieve convergence. Moreover, the algorithm that  
15313 includes the latent encounter histories seems to have a hard time exploring the region of  
15314 the posterior in which  $N$  is low. Given these technical difficulties, we recommend using  
15315 the **JAGS** implementation (based on Panel 18.2), and it is always a good idea to use  
15316 MCMC diagnostic tools such as those available in the `coda` package.

15317 The take-home message is that, even with simulated data, the precision of the posterior  
15318 distributions is low and mixing is poor. This should be expected given that we are asking so  
15319 much from so little data. In essence, we are trying to fit a point process model while being  
15320 twice removed from the actual point (activity center) locations. These difficulties may



**Figure 18.2.** MCMC results for the parameter  $N$  from the two algorithms (with and without the latent encounter histories). The first row contains the histograms of the posterior distributions, the second row contains the history plots, the third row shows the autocorrelation plots.



**Figure 18.3.** Spatially-correlated counts of northern parula. Gray crosses are the locations of the 105 point count stations. Superimposed are the number of detections after 3 survey occasions.

warrant the investigation of simpler models at the expense of the mechanistic description of the system. Another option is to figure out ways of improving model precision – options we discuss in Sec. 18.7. Before doing so, we re-analyze the Northern Parula (*Parula americana*) data described in Chandler and Royle (2013)

## 18.6 THE MARYLAND NORTHERN PARULA STUDY

The parula data are standard avian point count data, with one exception. Typically, points are spaced by  $> 200$  m when studying passerines in order to maintain statistical independence. In contrast, the parula data were collected at 105 points located on a 50-m grid, which virtually ensures spatial correlation since the parula song can be heard from distances  $> 50$  m. Each point was surveyed 3 times during June 2006, and Fig. 18.3 depicts the resulting spatially-correlated counts ( $n_j$ ). A total of 226 detections were made with a maximum count of 4 during a single survey. At 38 points, no birds were detected. All but one of the detections were of singing males, and this one observation was not included in the analysis.

We fit the model using **JAGS** and the code from Panel 18.2, which does not include the latent encounter histories. For comparative purposes, we used proper priors rather than the improper priors used by Chandler and Royle (2013), but all other aspects of the analysis were the same, including  $M = 300$  and a state-space created by buffering the grid of point count locations by 250 m. To reduce computation time, we used the **parallel** package and distributed 3 chains to 3 separate cores. The entire example can

15340 be reproduced using the code on the help page for `nopa` in our **R** package `scrbook`. The  
 15341 following illustrates the essential elements:

```

15342 > library(scrbook)
15343 > library(rjags)
15344 > dat2 <- list(n = nopa$n, X = nopa$X, M=300, J=nrow(nopa$n), K=ncol(nopa$n),
15345 +           xlim=c(-600, 600), ylim=c(-400, 400))
15346 > init2 <- function() {
15347 +   list(sigma=rnorm(1, 100), lam0=runif(1), z=rep(1, dat2$M))
15348 + }
15349 > cl2 <- makeCluster(3) # Open 3 parallel R instances
15350 > clusterExport(cl2, c("dat2", "init2", "pars1")) # send objects to 3 cores
15351 > system.time({
15352 +   out2 <- clusterEvalQ(cl2, { # executes the folowing command on each core
15353 +     library(rjags)
15354 +     jm <- jags.model("nopa2.jag", dat2, init2, n.chains=1, n.adapt=500)
15355 +     jc <- coda.samples(jm, pars1, n.iter=2500)
15356 +     return(as.mcmc(jc))
15357 +   })
15358 + })
15359 > mc2 <- mcmc.list(out2) # put the 3 chains together
15360 > plot(mc2)
15361 > summary(mc2)

15362      XXXX THE RESULTS OF THE TWO ANALYSES LOOK VERY SIMILAR AS
15363      EXPECTED XXXX
15364      Several aspects of this analysis could be improved via model extensions. In particular,
15365      we note that a more appropriate observation model would recognize the fact that detection
15366      in this case is the result of two processes. Specifically, an ideal encounter probability model
15367      would include a process describing the location of the bird (not just its home range center)
15368      as well as the probability of detecting it, given its location during the survey. Essentially,
15369      the model we would like to fit could be thought of as a latent distance sampling model
15370      allowing for movement. As it turns out, a very rudimentary form of distance data were
15371      collected – birds were determined to be either within 150 m or beyond 150 m from the
15372      observer. In Sec. 18.8, we propose a model to accommodate these auxilliary data.
```

## 18.7 IMPROVING PRECISION

15373 Chandler and Royle (2013) recommended two strategies for improving the precision of the  
 15374 posterior distributions obtained under the SC model: (1) mark a subset of individuals or  
 15375 (2) elicit informative priors from the published literature. The first option is the subject  
 15376 of the next chapter. The second option should be readily accomplished in many studies  
 15377 because extensive information on home range size has been compiled for many species in  
 15378 diverse habitats (*e.g.*, DeGraaf and Yamasaki, 2001). It is easy to embody this information  
 15379 as a prior distribution in Bayesian analyses (Chandler and Royle (2013), Chapt. 5).

15380 In some cases, it may not be possible to mark any individuals, and no prior information  
 15381 may exist about encounter parameters; however, it may be possible to collect axillary data,

such as the distance measurements recorded in the parula study. Other sources of auxiliary data could include removal counts or double observer counts, which are routinely collected in wildlife studies. Extending the model to accommodate such data is treated in the next section.

## 18.8 EXTENSIONS OF THE SPATIAL COUNT MODEL

If ancillary data such as distance measurements exist, why bother with the SC model at all? Isn't density estimable using the distance data alone? Yes, in fact it is, and in many situations a simple distance sampling model will be sufficient. However, unlike the situation we described earlier in this chapter where we viewed spatial correlation as a good thing, the model extension we describe now provides a means of dealing with spatial correlation when it is unwanted or perhaps unavoidable. In addition, extensions of this model We suspect that the SC model could can be used to make inferences about multiple processes in addition to spatial and temporal variation, such as home range size and movement.

As an example, consider again the northern parula data. As it turns out, observers recorded rudimentary distance sampling data by determining if each detected individual was within or beyond 100 m. Although not ideal, distance data binned into 2 intervals are sufficient for estimating the scale parameter of a distance sampling detection function, and thus we should be able to use that information to increase precision and develop a more realistic encounter model. Doing so requires that we consider not only the activity centers, but also the actual locations of individuals during each survey – much like in search-encounter models (Chapt. 15).

By including both activity centers ( $\mathbf{s}$ ) and actual locations ( $\mathbf{u}$ ) in the model, abundance in any region  $\mathcal{B}$  is given by

$$N(\mathcal{B}) = \sum_i I(\mathbf{u}_i \in \mathcal{B}). \quad (18.8.1)$$

Thus, in the context of distance sampling studies in which the distance data are recorded in discrete intervals, the region  $\mathcal{B}$  would be the area corresponding to a particular distance interval. The probability of detecting the individuals  $N(\mathcal{B})$  would be the average detection probability  $\bar{p}$ , which is computed by integrating a distance-based detection function over the distance interval.

In other contexts, such as when conducting removal surveys, the region  $\mathcal{B}$  could be a fixed-area plot, such as a stream segment. Again, Eq. 18.8.1 could be used to model local abundance ( $n(\mathcal{B})$ ), and detection probability within the region could be modeled conditional on  $n(\mathcal{B})$ . For instance, if a stream segment is uniformly surveyed using electrofishing equipment, then a standard non-spatial removal model could be used to estimate detection probability  $p$ , conditional on the spatially-explicit model of abundance. A reasonably

general description of this model is as follows:

$$\begin{aligned}\mathbf{s}_i &\sim \text{Uniform}(\mathcal{S}) \\ \mathbf{u}_{ik} &\sim \text{BVN}(\mathbf{s}_i, \tau) \\ N(\mathcal{B}_{jk}) &= \sum_{i=1}^M I(\mathbf{u}_{ik} \in \mathcal{B}_{jk}) \\ n_{jkl} &\sim \text{Binomial}(N(\mathcal{B}_{jkl}), p)\end{aligned}$$

15410 where  $\tau$  is the parameter of a bivariate normal distribution (with correlation  $\rho = 0$ )  
 15411 describing the locations of individuals on occasion  $k$ . The interpretation of the parameter  
 15412  $p$  will depend upon the survey protocol.

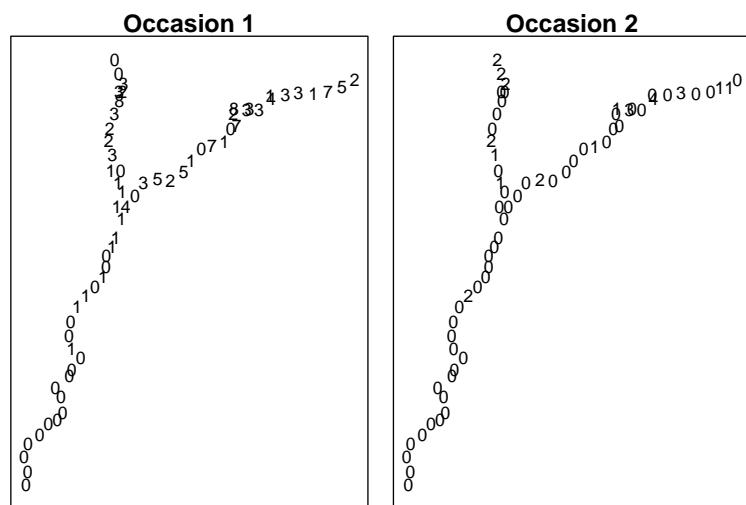
15413 When plots are far enough apart that individuals cannot move between them, the  
 15414 counts will be uncorrelated and the model can be approximated using a non-spatial  $N$ -  
 15415 mixture model allowing for temporary emigration (Chandler et al., 2011). In the next  
 15416 example, we consider data in which the plots are obviously not independent.

## 18.9 THE MARYLAND DUSKY SALAMANDER STUDY

15417 The independence assumption of the Chandler et al. (2011) model will not always hold. A  
 15418 prime example is in studies of aquatic species in stream networks. For example, consider  
 15419 the data depicted in Fig. 18.4. What is this spaghetti soup, you say? These are streams of  
 15420 numbers corresponding to counts of northern dusky salamanders (*Desmognathus fuscus*) in  
 15421 25-m stretches on a small stream in the Chesapeake and Ohio National Historic Park. The  
 15422 data were collected by E.H.C. Grant and colleagues with the objective of understanding  
 15423 the spatial and temporal dynamics of salamander populations in response to seasonal  
 15424 and annual variations in stream hydrology. In addition, movement processes, including  
 15425 dispersal are studied between years (see Grant et al. (2010) for more details).

15426 To sample the population, the stream networks are divided into 25-m stretches as  
 15427 illustrated in Fig. 18.4. In each stretch, “temporary” removal sampling is used, which  
 15428 involves capturing and removing salamanders on 3 consecutive passes. The salamanders  
 15429 are placed in a bucket of water for the brief 10-20 min duration of sampling, and then  
 15430 they are released at the location of capture. The entire process is repeated 3-4 times  
 15431 per season (May-Aug). In a subset of streams and years, individuals are marked, but in  
 15432 general it is too expensive to mark the entire population, and the data considered here  
 15433 consists entirely of unmarked individuals.

15434 The sampling protocol may be thought of as a “robust design” (Pollock, 1982), with  
 15435 “occasions” (typically 1 day) being the primary period, and secondary samples being the  
 15436 removal passes within the primary periods. An obvious feature of these data is that the  
 15437 neighboring counts are spatially correlated. In this case, we have reason to believe that  
 15438 this correlation is the result of habitat preferences, with individuals actively selecting  
 15439 habitat in the upper reaches of the streams. This could be modeled as a function of a  
 15440 covariate describing the distance from the mouth of the stream. Another obvious feature  
 15441 of this data is that the pattern of spatial correlation remains consistent between occasions,  
 15442 but the overall counts decline markedly over the course of the season. These phenomena  
 15443 can be explained by the fact that the salamanders have relatively small home ranges, and



**Figure 18.4.** Stream segment counts of northern dusky salamanders in the Chesapeake and Ohio National Historic Park, VA/MD. Each number is the count associated with a 25-m stretch in which 3 removal passes were made on 3 occasions each summer (only 2 occasions are shown here). Notice the consistency of the spatial correlation between occasions and the temporal decline in the counts.

**Table 18.2.** Posterior summarizes from removal model of salamander counts allowing for movement and decreasing population size over the course of a breeding season.

Parameter	Mean	SD	2.5%	50%	97.5%
$N_1$	178.393	16.346	151.000	177.000	214.000
$N_2$	62.322	6.884	51.000	62.000	77.000
$N_3$	21.202	3.695	15.000	21.000	29.000
$\phi$	0.348	0.038	0.275	0.348	0.425
$\tau$	27.427	3.200	21.293	27.173	33.706
$p$	0.396	0.053	0.294	0.394	0.502

15444 this results in the consistent pattern of correlation among occasions. Furthermore, as the  
 15445 season progresses, the streams dry out, and many individuals move underground.

15446 Given the importance of movement within home ranges, which determines the corre-  
 15447 lation among occasions, and movement underground, which results in a decreasing number  
 15448 of individuals being available for sampling, it would be helpful to have a model that de-  
 15449 scribes both processes and allows for evaluation of hypotheses regarding the effects of  
 15450 environmental variables. For example, one might ask how stream flow is related to the  
 15451 probability that an individual remains active. A model describing this process could be  
 15452 used to predict activity levels under future conditions. Although we do not investigate  
 15453 covariate effects in this section, we do present a general model allowing for movement  
 15454 among occasions, and for decreasing availability over the season.

15455 This expanded model is founded on the one described in the previous section, but it  
 15456 also includes a removal model for the observation process, and it includes a basic “open”  
 15457 population model to allow for a decline in abundance over time (Chapt. 16). Actually,  
 15458 the population is not thought to actually decline substantially during the season, but  
 15459 rather, the number of individuals *available* for detection declines because many individuals  
 15460 move underground as the streams dry. Each of these components is included in the  
 15461 **BUGS** description of the model presented in Panel 18.3.

15462 We fit this model to the data and obtained the posterior distributions summarized  
 15463 in Table 18.2. The results indicate that the population size available for detection did  
 15464 decrease rapidly during the season, the rate of which is determined by the  $\phi$  parameter.  
 15465 Modeling this parameter as a function of water flow or volume would allow one to predict  
 15466 salamander activity under future environmental conditions. Another result of the analysis  
 15467 is that the movement parameter,  $\tau$ , was relatively low, indicating that adult salamanders  
 15468 rarely move more than 100 m from their home range center during a season. This explains  
 15469 why the distribution of individuals within the stream remains relatively constant over time.  
 15470 Including this parameter in the model also provides a general mechanism for modeling  
 15471 temporal correlation in count data.

## 18.10 SUMMARY AND OUTLOOK

15472 Unlike traditional models of count data used in ecology, the SC model is parameterized  
 15473 in terms of *individuals* – individuals that just so happen not to be directly observed.  
 15474 The reason for accommodating this latent structure is that it provides a more mechanistic  
 15475 description of ecological systems. For example, the model allows us to attach a mechanism

15476 – movement – to the widely observed phenomenon of spatial correlation in count data.  
15477 In addition, by parameterizing the model in terms of individuals, it makes it possible to  
15478 incorporate data from both marked and unmarked individuals, as will be described in the  
15479 next chapter. This ability to combine different types of data should make it possible to  
15480 design effective monitoring programs when resources are too limited to conduct spatially  
15481 extensive capture-recapture studies, as has been done with non-spatial models (Conroy  
15482 et al., 2008).

15483 The SC model is a conceptually simple extension of standard SCR models, but in  
15484 terms of computational requirements and latent structure, it is perhaps at the extreme  
15485 end of what is possible to do with count data. As is always true, the harder we try to  
15486 mirror reality with our models, the harder it becomes to estimate the parameters of the  
15487 system. In this chapter, we tried to emphasize that as conceptually appealing as the SC  
15488 model may be, it is unlikely to produce satisfying results in the absence of additional  
15489 information. However, additional information such as home range size estimates will  
15490 often be available for many species, and if not, we have provided an alternative method of  
15491 accommodating additional data in the form of distance measurements or removal counts.  
15492 This can greatly increase precision in studies designed to make spatially-explicit inferences  
15493 about population processes.

```

model{
  sigma ~ dunif(0, 200) # Tailor this to your state-space
  lam0 ~ dunif(0, 5)    # consider dgamma() as an alternative
  psi ~ dbeta(1,1)
  for(i in 1:M) {
    z[i] ~ dbern(psi)
    s[i,1] ~ dunif(xlim[1], xlim[2])
    s[i,2] ~ dunif(ylim[1], ylim[2])
    for(j in 1:J) { # Number of traps
      distsq[i,j] <- (s[i,1] - X[j,1])^2 + (s[i,2] - X[j,2])^2
      lam[i,j] <- lam0 * exp(-distsq[i,j] / (2*sigma^2))
      for(k in 1:K) { # Number of occasions
        y[i,j,k] ~ dpois(lam[i,j]*z[i])
      }
    }
  }
  for(j in 1:J) {
    for(k in 1:K) {
      n[j,k] ~ dsum(y[1,j,k], y[2,j,k], ..., y[200,j,k]) # Code abbreviated!!
    }
  }
  N <- sum(z[]) # Realized population size
  A <- (xlim[2]-xlim[1])*(ylim[2]-ylim[1]) # Area of state-space
  D <- N / A # Realized density
  ED <- (M*psi)/A # Expected density
}

```

---

Panel 18.1: **JAGS** code defining the spatial count model. This version includes the latent encounter histories.

---

```

model{
  sigma ~ dunif(0, 200)
  lam0 ~ dunif(0, 5)
  psi ~ dbeta(1,1)
  for(i in 1:M) {
    z[i] ~ dbern(psi)
    s[i,1] ~ dunif(xlim[1], xlim[2])
    s[i,2] ~ dunif(ylim[1], ylim[2])
    for(j in 1:J) { # Number of traps
      distsq[i,j] <- (s[i,1] - X[j,1])^2 + (s[i,2] - X[j,2])^2
      lam[i,j] <- lam0 * exp(-distsq[i,j] / (2*sigma^2)) * z[i]
    }
  }
  for(j in 1:J) {
    bigLambda[j] <- sum(lam[,j])
    for(k in 1:K) {
      n[j,k] ~ dpois(bigLambda[j])
    }
  }
  N <- sum(z[])
  A <- (xlim[2]-xlim[1])*(ylim[2]-ylim[1]) * 10000 # Area of state-space (ha)
  D <- N / A      # Realized density
  ED <- (M*psi)/A # Expected density
}

```

---

Panel 18.2: **JAGS** code defining the spatial count model. This version does not include the latent encounter histories, and thus runs much faster than the code in Panel 18.1.

---

```

model {
  phi ~ dbeta(1,1)      # "availability" parameter
  tau ~ dunif(0, 1000)   # "movement parameter" of Gaussian kernel model
  p ~ dbeta(1,1)        # detection prob
  psi ~ dbeta(1,1)      # data augmentation parameter
  for(i in 1:M) {
    z[i,1] ~ dbern(psi)      # is the guy real?
    z[i,2] ~ dbern(z[i,1]*phi) # and still alive?
    z[i,3] ~ dbern(z[i,2]*phi) # still kicking
    s[i] ~ dcat(PrSeg[]) # location (stream segment) of activity center
    for(g in 1:G) {
      PrU[i,g] <- exp(-distmat[s[i],g]^2/(2*tau^2)) # Pr(u | s)
    }
    for(k in 1:K) {
      u[i,k] ~ dcat(PrU[i,]) # location of guy i at time k
      for(g in 1:G) {
        y[i,g,k] <- (u[i,k] == g)*z[i,k] # was guy at u==g?
      }
    }
  }
  for(j in 1:J) {
    for(k in 1:K) {
      N[j,k] <- sum(y[,seg[j],k]) # Number of individuals in seg j at time k
      # removal model:
      n[j,1,k] ~ dbin(p, N[j,k])
      N2[j,k] <- N[j,k] - n[j,1,k]
      n[j,2,k] ~ dbin(p, N2[j,k])
      N3[j,k] <- N2[j,k] - n[j,2,k]
      n[j,3,k] ~ dbin(p, N3[j,k])
    }
  }
  Ntot[1] <- sum(z[,1]) # Abundance, occasion 1
  Ntot[2] <- sum(z[,2]) # Abundance, occasion 2
  Ntot[3] <- sum(z[,3]) # Abundance, occasion 3
}

```

---

Panel 18.3: **BUGS** description of model for the data shown in Fig. 18.4. The model allows for spatially-explicit temporary emigration, and for a decrease in abundance as individuals move underground throughout the course of the season.



15494  
15495

15496

# 19

---

## SPATIAL MARK-RESIGHT MODELS

15497 So far, we have dealt with the situation where all detected individuals are identifiable  
15498 upon encounter because they carry some form of individual mark, and in Chapt. 18  
15499 we introduced and developed an SCR model for non-identifiable populations, a spatial  
15500 *non-capture-recapture* model, if you will. These two extremes are common in the study  
15501 of animal populations with non-invasive sampling methods. However, there is also an  
15502 intermediate situation where part of the population is tagged or otherwise marked and  
15503 can thus be identified upon recapture, while the unmarked portion remains unidentified.  
15504 In this situation so-called mark-resight models (Bartmann et al., 1987; Arnason et al.,  
15505 1991; Neal et al., 1993) can be used to estimate population size and density by combining  
15506 data from both the marked and unmarked individuals.

15507 Traditionally, capture-recapture studies involved physical capture and marking of in-  
15508 dividuals throughout the study. This methodology is still widely applied in the study  
15509 of species that are relatively easy to capture, such as small mammals, but can be very  
15510 costly, logically challenging and risky when dealing with larger species. In contrast, in  
15511 mark-resight studies a sample of individuals is captured and tagged (or otherwise marked)  
15512 during a single marking event. Marking is followed by resighting surveys, upon which both  
15513 the detection of marked and recognizable individuals and unmarked animals is recorded.  
15514 Resighting surveys are usually non-invasive (hence the name ‘resighting’), so that they  
15515 don’t involve handling of animals. As such, mark-resight models have a major advantage  
15516 over traditional capture-recapture models in that they only require individuals to be cap-  
15517 tured and handled once, during the initial marking. This reduces field costs and risks for  
15518 the animals (and potentially the researchers).

15519 Mark-resight models have a set of underlying assumptions, most of which are identi-  
15520 cal to those of capture-recapture models; e.g., demographic population closure (violation  
15521 of geographic population closure can be accommodated by some models) and no loss or  
15522 misidentification of marks (see also Chapt. 5). Just like standard capture-recapture mod-  
15523 els, there are means to incorporate heterogeneity in capture probability. An essential  
15524 assumption of mark-resight models is that the marked individuals are a representative  
15525 sample of the study population, so that inference about detection can be made for the

15526 whole population from the marked sample. However, in mark-resight means that the  
15527 process of actually marking individuals is an important consideration. This is usually  
15528 addressed by using a different method for marking than for resighting, and by marking a  
15529 random sample of the population.

15530 Owing to the advantages of mark-resight over capture-recapture, especially when deal-  
15531 ing with hard-to-trap species, mark-resight is a popular tool in wildlife population studies.  
15532 The method has been applied for decades to a suite of species and survey techniques,  
15533 ranging from banding and resighting Canada geese (Hestbeck and Malecki, 1989) to ear-  
15534 tagging and camera-trapping grizzly bears (Mace et al., 1994) to paintball marking and  
15535 areal resightings of large ungulates (Skalski et al., 2005).

15536 In this chapter we consider mark-resight within their spatial context and develop a  
15537 spatial mark-resight (SMR) model. To motivate this model development, imagine you  
15538 conduct a live-trapping study during which you capture and mark a number of animals  
15539 with individually recognizable tags. Subsequently, you go back out to the field and conduct  
15540 resighting surveys on an array of locations, and during these resighting surveys you see  
15541 some of your marked individuals as well as new, unmarked ones. Then, for the marked  
15542 animals you obtain the same form of spatially explicit individual encounter histories as  
15543 you would in a standard SCR study. In addition to that you obtain site (and occasion)  
15544 specific counts of individuals you did not tag. Thus, spatial mark-resight is an SCR  
15545 framework for populations where only a portion of the individuals can be identified and  
15546 the major difference between SCR and SMR is how we include those counts of unmarked  
15547 individuals in the model. In the following sections we first provide some background  
15548 information on mark-resight and the types of data such surveys can provide. We will  
15549 further explore the implications of the assumption of the marked individuals being a  
15550 random subset of the population, which, in the context of SMR models refers to both the  
15551 *demographic composition*, but also to the *spatial distribution* of the marked individuals  
15552 in  $\mathcal{S}$ . We then move on to the formal development of SMR models, which, as will be  
15553 shown, are hybrids of regular SCR models and the models presented in Chapt. 18 for  
15554 data where individuals cannot be uniquely identified. We explore models for both known  
15555 and unknown numbers of marked individuals, and for imperfect individual identification of  
15556 marks, and approaches to incorporate telemetry location data. In the spatial framework,  
15557 most of the information on model parameters comes from the marked individuals. But  
15558 in Sec. 19.5 we will see that, analogous to the models we developed previously in Chapt.  
15559 18, the spatial correlation in counts of unmarked individuals also contributes information  
15560 about detection and movement.

## 19.1 BACKGROUND

15561 Before we start exploring spatial mark-resight approaches in more detail, we need to  
15562 establish some terminology and gain a clear understanding of what types of mark-resight  
15563 data we can have, in order to appreciate and understand the different flavors of mark-  
15564 resight models.

### 15565 19.1.1 Resighting techniques

15566 As with capture-recapture surveys, there are numerous methods suitable to perform re-  
15567 sightings. Common methods are visits to a set of points for resightings by an observer,

15568 or camera-trapping; but resightings need not be restricted to a particular set of locations.  
15569 We can just as well envision a search-encounter kind of method, where a certain area is  
15570 searched, systematically or opportunistically, for marked animals (see Chapt. 15). In this  
15571 chapter we will only deal with fixed location resighting surveys, and we will refer to the set  
15572 of resighting locations as the resighting array. In some instances we will also be concerned  
15573 with where marked animals were captured, and we refer to these locations as the marking  
15574 locations.

15575 **19.1.2 Types of mark-resighting data**

15576 In general, we have (at least) two sets of data: encounter histories for marked, and thus,  
15577 identifiable individuals  $i$  at resighting location  $j$  and occasion  $k$ ,  $y_{ijk}$ , and counts of un-  
15578 marked records,  $n_{jk}$ , for each resighting location  $j$  and occasion  $k$ . Depending on the  
15579 sampling technique, we can conceive of three slightly different types of partial ID data.

15580 **(1) Known number of marked individuals:** If you implement a resighting survey  
15581 shortly after the marking session, you may be confident that none of the marked individuals  
15582 have died or lost their mark. Under these circumstances you know that the number of  
15583 marked individuals available for resighting,  $m$ , is equal to the number of individuals you  
15584 marked. Alternatively, the marking technique might involve radio-transmitters, allowing  
15585 you to confirm the presence or absence of marked individuals in the resighting survey area  
15586 using radio-telemetry (White and Shenk, 2001). In both cases, you know the number of  
15587 marked individuals in the surveyed population. In this situation, even though you may  
15588 fail to resight some of the marked individuals, you know how many there are, and so you  
15589 can simply assign all-zero encounter histories to the marked individuals not encountered  
15590 – in other words, contrary to regular capture-recapture models, in mark-resight models  
15591 with a known number of marked individuals, we can observe all-zero encounter histories.  
15592 Under these circumstances, estimating  $N$  reduces to estimating the number of unmarked  
15593 individuals,  $U$ .

15594 **(2) Unknown number of marked individuals:** If  $m$  is not known, for example because  
15595 we suspect that some of the marks may have been lost between tagging and conducting  
15596 the resighting surveys, we obtain a slightly different type of mark-resight data. Here, we  
15597 do not accurately know the number of marked individuals available for resighting. As a  
15598 consequence, individuals have to be resighted at least once for us to know they are still  
15599 marked and alive and thus available for resighting. So, contrary to the situation where  
15600 we know  $m$  and analogous to regular capture-recapture models, we cannot observe all-  
15601 zero encounter histories of marked individual. In this situation, estimating  $N$  involves  
15602 estimating both  $m$  and  $U$ .

15603 A special case of this kind of data can arise from camera trapping. Even when dealing  
15604 with a species that has no spots or stripes, some individuals in the study population can  
15605 have natural marks that make them identifiable on pictures, such as scars or a distinct  
15606 coloration. Again, in this scenario an individual has to be photographed at least once to  
15607 be known. Here, the fact that both the “marking” method and the subsequent resighting  
15608 method are the same (although marking in this case does not involve any actual physical  
15609 marking) can be cause for concern: our sample of “marked” individuals may not be a  
15610 random sample of the population but consist of individuals that for some reason are more

15611 likely to be photographed (e.g., individuals with activity center more interior to the trap  
 15612 array). In that case, a basic assumption of the mark-resight model is violated.

15613 **(3) Unknown marked status:** Finally, consider a scat or hair snare survey, where only  
 15614 a part of the sample is analyzed genetically (or DNA can only be extracted from a subset of  
 15615 samples due to sample quality). In this scenario,  $n_{jk}$  can contain both completely unknown  
 15616 individuals that are not represented at all in the complete set of encounter histories of  
 15617 marked animals,  $Y$ , but it can also contain samples from individuals that we previously  
 15618 identified. The difference is that in the first two scenarios, part of the population of  
 15619 individuals is identifiable, while in the second scenario, part of the sample of individuals  
 15620 is identifiable. This type of data violates one of the basic assumptions of mark-resight  
 15621 models, namely, that marked individuals are always correctly identified as such.

15622 To our knowledge there are currently no mark-resight models available that account for  
 15623 possible misidentification of the marking status of individuals (although some literature is  
 15624 available on misidentification of individuals in capture-recapture studies, e.g., Yoshizaki  
 15625 et al., 2009; Lukacs and Burnham, 2005; Link et al., 2010). In this chapter we will ignore  
 15626 this kind of data and focus instead on types (1) and (2).

15627 For both types of data a slightly different situation arises when we can only tell that  
 15628 an individual is marked, but not who it is. You may be able to see that an individual is  
 15629 marked but the identifying feature of the tag (a number or coloration) may have become  
 15630 unreadable, or may be hidden from view. In this case, in addition to the observed  $y_{ijk}$   
 15631 and  $n_{jk}$ , you also observe a number of sightings of marked but unidentified individuals,  
 15632 say  $r_{jk}$ .

### 15633 19.1.3 A short history of mark-resight models

15634 Initially, mark-resight methods focused on radio-tagged individuals to estimate popula-  
 15635 tion size (White and Shenk, 2001). Radio-collars provide a means of determining which  
 15636 of the animals are in the study area and available for sampling, thus determining the  
 15637 number of marked individuals in the population. Knowing this number was a prerequisite  
 15638 for most earlier mark-resight approaches (White, 1996). The oldest mark-resight model  
 15639 is the good old Lincoln-Petersen estimator, where individuals are marked and a single  
 15640 resight/recapture occasion is carried out (Krebs, 1999). We need not identify individuals,  
 15641 but only to tell apart marked from unmarked individuals. Let  $m$  be the number of marked  
 15642 individuals in the population,  $m_{(R)}$  the number of marked individuals seen on the resight-  
 15643 ing occasion, and  $n_{(R)}$  the total number of marked and unmarked individuals observed  
 15644 during resighting. Population size  $N$  is then estimated as

$$N = m \times n_{(R)} / m_{(R)}.$$

15645 A suite of more elaborate models using individual capture histories over several re-  
 15646 sighting occasions were developed in the 1980s and 90s and compiled into the program  
**15647 NOREMARK** (White, 1996). Apart from the basic model with known number of marked  
 15648 individuals and no individual variation in resighting probabilities (joint hypergeometric  
 15649 maximum likelihood estimator) (Bartmann et al., 1987; White and Garrot, 1990; Neal,  
 15650 1990; Neal et al., 1993), **15651 NOREMARK** contains models that account for lack of geo-  
 15652 graphic population closure (Neal et al., 1993), individual heterogeneity in resighting rates

and sampling with replacement (i.e. individuals can be seen more than once on any occasion, (Minta and Mangel, 1989; Bowden, 1993)). A first mark-resight model allowing for an unknown number of marked individuals was developed by Arnason et al. (1991).

While many of these models perform well under certain situations, they are somewhat limited in that they do not allow for combining data across several surveys (McClintock et al., 2006) and not all of them are likelihood-based or allow for different parameterizations (e.g., including a time effect on detection), so that selection of the most appropriate model cannot be based on standard approaches such as AIC, but is largely left up to educated guesswork (McClintock et al., 2006). Recently, more flexible and generalized likelihood-based mark-resight models have been developed. These models can account for individual heterogeneity in detection, unknown number of marked individuals and lack of geographical closure, as well as a less than 100% individual identification rate of marked individuals; they can be applied to sampling with and without replacement and can combine data across several primary sampling occasions in a robust design type of analysis (McClintock et al., 2009a,b). Since they are all likelihood-based, model selection among different parameterizations and model averaging based on AIC is an option. Most of these models have also been incorporated into the program **MARK** (McClintock and White, 2012).

For a detailed treatment of these different non-spatial mark-resight models, we refer you to the original papers cited in the preceding paragraph. In short, these models are based on the joint likelihood of two model components: one describing the resighting process of marked individuals and one describing the number of unmarked individuals observed. The resighting process of marked individuals can use either a Poisson or a Bernoulli observation model, depending on whether sampling is with or without replacement, and the resighting probabilities can have both fixed effects to model individual and environmental covariates, and a random-effect component to accommodate variation in detection due to individual heterogeneity. The process describing the number of unmarked individuals observed (or, under a Poisson observation model, the number of times unmarked individuals are observed),  $n_t$  ( $t$  here and in the following description denotes a primary sampling occasion, for example, a year or a season) which are approximated as a normal distribution (McClintock et al., 2006), or a normal distribution left-truncated at 0 (McClintock et al., 2009a):

$$n_t \sim \text{Normal}(\mathbb{E}(n_t), \text{Var}(n_t)).$$

For a single-season study, the  $t$  subscript does not need to be included. Although this is a simplification of the actual sampling process, McClintock et al. (2006) found this normal distribution to be a satisfactory approximation, which allows  $N$  to enter the model likelihood via  $\mathbb{E}(n_t)$  and  $\text{Var}(n_t)$ .

In the simplest model without any variation in detection, the expected number of resightings of unmarked individuals,  $\mathbb{E}(n_t)$ , can be written as the number of unmarked individuals times the expected number of detections of a single individual. This is the mean or expected value of the underlying observation model:

$$\mathbb{E}(n_t) = (N - m) * \theta \quad (19.1.1)$$

where  $\theta = K \times p$  for a Binomial observation model with  $K$  replicates and individual detection probability  $p$ , or  $\theta = \text{expected}/\text{average individual encounter rate } \lambda$  for a Poisson

15694 observation model. Similarly,  $\text{Var}(n_t)$  depends on the underlying observation model and  
15695 is based on the parameters that determine the individual detection probability/encounter  
15696 rate. Combining these two components,  $N$  is directly incorporated into the joint likelihood  
15697 of the model.

15698 While these mark-resight models are very flexible, they share the shortcomings of  
15699 traditional capture-recapture models when it comes to estimating population density (e.g.,  
15700 Chaps. 1 and 4). As long as resightings are collected across a number of locations,  
15701 however, they come with the same spatial information as (re)captures in a standard SCR  
15702 study. In the following sections we will consider mark-resight sampling in the framework  
15703 of spatial capture-recapture.

#### 15704 19.1.4 The random sample assumption

15705 In mark-resight studies it is a prerequisite that the marked portion of the studied popula-  
15706 tion is a random sample of the population, so that detection probability for the population  
15707 can be adequately estimated from the marked subset. If, for example, there is some latent  
15708 group structure in your population where one group has a higher detection probability  
15709 than the other, the marked part of the population should have the same composition with  
15710 regard to this group structure as the study population. Intuitively, people think of this  
15711 as a demographic problem. But if you think back to Chapt. 1 and one of the motivations  
15712 for the development of SCR models, this assumption also has spatial implications. In  
15713 a non-spatial mark-resight study, if all the individuals we mark live on the edge of the  
15714 resighting array, their exposure to resighting will be lower compared to the exposure of  
15715 unmarked individuals living in the center of the resighting array, thus artificially deflating  
15716 estimates of detection. So to obtain a truly random sample of the study population, the  
15717 *locations of the home ranges* of the marked individuals also have to be a random sample  
15718 of the home range locations of the entire population. In general, this will be difficult to  
15719 assess or even to incorporate into study design or analysis, unless the spatial context of  
15720 sampling is clearly defined.

15721 In the SMR framework, this issue manifests itself more explicitly, for two reasons:  
15722 (1) we define the spatial context of the population by setting a state-space; and (2) we  
15723 assume a certain distribution or point process for all individuals within that state-space,  
15724 in most cases a uniform distribution or homogeneous point process (but see Chapt. 11 for  
15725 models with inhomogeneous spatial point processes). So, for the marked individuals to be  
15726 a random subset of the population in  $\mathcal{S}$ , they have to be uniformly distributed throughout  
15727 the state-space. When we study a species where some individuals can be identified based  
15728 on natural marks, while others do not have unique marks (for example regular colored  
15729 versus melanistic leopards), and we can assume that the distribution of these two groups  
15730 of individuals across  $\mathcal{S}$  are identical, then the ‘random sample’ assumption should be met.  
15731 But what if we actively need to mark individuals in order to distinguish them? Then, if  
15732 we want to meet the random sample assumption, contrary to SCR models, the state-space  
15733 is no longer a quantity we can set after data collection for analysis purposes, but ideally,  
15734 we should make the definition of  $\mathcal{S}$  part of our study design.

15735 Here is another way to think about this: In SCR models, once the state-space is chosen  
15736 large enough, estimates of density are no longer sensitive to the size of  $\mathcal{S}$ , because  $N$  scales  
15737 with the area of  $\mathcal{S}$ . In spatial mark-resight, however, our population of individuals consists

of two groups, marked and unmarked. Consider the case where we have a known number of marks. Because we fix the size of the marked part of our population, total population size  $N$  no longer scales with the area of the state-space. While the number of unmarked individuals can go up as  $\mathcal{S}$  increases in size,  $m$  is fixed by design, and thus, as  $\mathcal{S}$  increases, overall density will decrease. Further, if our data contain all-zero encounter histories for some of the marked individuals, we have no immediate information about where in the state-space these never observed individuals live (apart from the vague information that they probably do not live in the middle of the trap array). As we increase  $\mathcal{S}$ , the area over which these marked unobserved individuals can live increases, too – and consequently, their density decreases. Even if we do not know  $m$ , we usually know an upper bound for it – the total number ever caught before resighting. This upper bound does not change, no matter the size of the state-space, so again, at some point, estimates of  $m$  will hit the upper limit and after that, density will decrease as we increase  $\mathcal{S}$ . There is also the opposite risk: if we choose  $\mathcal{S}$  too small so that it does not contain the activity centers of all of the marked individuals, but we assume, by fixing  $m$ , that they are all part of the population, we will overestimate density – just as we would if we chose  $\mathcal{S}$  too small in a regular SCR setting. But this problem can be avoided much more easily, by increasing  $\mathcal{S}$ .

If we wanted to make sure by design that marked individuals are a random sample from  $\mathcal{S}$ , then, in practical terms, we need to define the state-space, which includes the resighting array plus sufficient buffer to include all animals potentially exposed to this array, and uniformly mark individual throughout  $\mathcal{S}$ . Alternatively, we could imagine a marking scheme where marking happens uniformly across a larger area or landscape, for example, a large-scale bird banding program, and then resighting happens on a smaller spatial scale within this landscape, so that the state-space around the resighting array lies completely within the larger marking area. We can see some sampling situations in which either one of the two scenarios might be reasonable, or at least reasonably approximated. For example, later on in this chapter we present a study where raccoons were caught and marked throughout an island, the boundaries of which are a natural limit for the state-space of this particular system. For many studies, however, this might not be the case. Often, marking is the more difficult and logically challenging part of a mark-resight study – think about capturing large carnivores. Especially for rare and cryptic species, areas over which resighting is conducted might have to be large to accumulate sufficient data, and marking over an even larger area –  $\mathcal{S}$  – would be logically impossible.

So what happens if we capture and mark individuals in a subset of the state-space? Then, while we may well have an overall constant density across  $\mathcal{S}$ , we will have a higher density of marked than unmarked individuals in the vicinity of the marking locations – live traps, mist nets, whatever is used to catch animals – and the opposite situation is true as we get further away from the marking locations. You can think about it this way: Animals with activity centers close to marking locations have a higher probability of being caught and marked, so that the closer we are to the marking locations, the higher is the proportion of marked individuals relative to unmarked. We have to explicitly take this pattern into account in our model in order to obtain an overall homogeneous distribution of activity centers. As it turns out this is not a trivial problem, but we provide some ideas to approach this problem in later sections of this chapter. For now, we will go on developing SMR models assuming that the marked animals are, indeed, a random sample of the entire population in  $\mathcal{S}$ .

## 19.2 KNOWN NUMBER OF MARKED INDIVIDUALS

We begin the model development with the simplest situation. Here, a known number of individuals constituting a random, representative sample from the population are marked and a series of resight samples are conducted following marking. No marks (or marked animals) are lost between marking and resighting, all individuals are correctly identified as marked or unmarked, and marked individuals are 100% correctly identified to individual level.

Recall from Chapt. 18 that without any individual identity, the observed counts at trap  $j$  and occasion  $k$ ,  $n_{jk}$ , represent the sum of all latent individual detections at  $j$  and  $k$ ,  $\sum_{i=1}^N y_{ijk}$ , where  $y_{ijk}$  are the latent individual encounter histories. We can model these counts as

$$n_{jk} \sim \text{Poisson}(\Lambda_j)$$

where

$$\Lambda_j = \sum_{i=1}^M (\lambda_{ij}).$$

Under this formulation, in order to carry out MCMC, we do not need to update the individual  $y_{ijk}$  in our model, which is more efficient in terms of computing. However, we can also formulate the model as conditional on the latent  $y_{ijk}$ . This is useful because if we have  $m$  marked animals in our study population, than  $y_{ijk}$  for those  $m$  individuals are no longer latent, but fully observed and can easily be included in the analysis to provide information on detection parameters.

The formulation conditional on  $y_{ijk}$  basically brings us back to the original SCR model, where individual site and occasion specific counts,  $y_{ijk}$ , are modeled as

$$y_{ijk} \sim \text{Poisson}(\lambda_{ij})$$

and

$$\lambda_{ij} = \lambda_0 \exp(-d_{ij}^2 / (2\sigma^2)).$$

in the case of a Poisson encounter model. Unobserved  $y_{ijk}$  are treated as missing data and have to be updated as part of the MCMC procedure. We can do that by using their full conditional distribution, which is multinomial with sample size  $n_{jk}$ . Specifically, let  $\{y_{1jk}, \dots, y_{mjk}\}$  be the encounter history data for the  $m$  marked individuals, then the full-conditional distribution is

$$\{y_{1jk}, \dots, y_{mjk}\} | \cdot \sim \text{Multinomial}(n_{jk}, \{\lambda_{1jk}, \dots, \lambda_{mjk}\}).$$

Whereas unmarked individuals provide no direct information about individual detection probability (or rate) in non-spatial mark-resight models, in the spatial setting they inform  $\sigma$ , as described in Chapt. 18. Including known individuals into the analysis helps estimate model parameters more accurately and precisely. We will address the relationship between the number of marked individuals and accuracy of the estimated parameters in Sec. 19.5.

---

15814   **19.2.1 Implementing spatial mark-resight models**

15815   Implementing a spatial mark-resight model in **JAGS** is not trivial, since the program  
15816   does not accept partially observed multivariate nodes (in this case the partially observed  
15817   individual encounter histories which we model as coming from a multinomial distribution).  
15818   We can, however, work around that by separating the marked from the unmarked data.  
15819   The **JAGS** code for the model with a known number of marked individuals is shown  
15820   in Panel 19.1. You see that data augmentation is only applied to the unmarked part  
15821   of the population, and  $N$  is the sum of the estimated number of unmarked individuals  
15822   (`sum(z[])`) and the number of marked individuals, which is known. Also, to reduce run  
15823   time, we summed observations of marked individuals across occasions and account for that  
15824   by multiplying  $\lambda_{ij}$  with  $K$ . Although the two data sets are separated, both parts of the  
15825   population, marked and unmarked, have the same prior uniform distribution of activity  
15826   centers. A last noteworthy detail in this code is the `dsum()` distribution. This distribution  
15827   is specific to **JAGS** (i.e., you cannot run this model in **BUGS**), and allows you to model  
15828   data – here the counts of unmarked individuals,  $n_{jk}$  – as the sum of a number of latent  
15829   variables, which in this case are the latent encounter histories of unmarked individuals.  
15830   While it can be a pain writing out all the arguments of `dsum()`, it is this function that  
15831   allows us to implement SMR models in **JAGS**.

15832   Alternatively, we can use the technical skills presented in Chapt. 17 and derive our  
15833   own MCMC algorithm. To do so, we only have to make relatively simple modifications to  
15834   the MCMC code developed for regular SCR models in Chapt. 17. Essentially, since we  
15835   observe individual detections for the marked part of the population, we have to update  
15836   only the unobserved part of the full – augmented – set of encounter histories,  $\mathbf{Y}$ , and  
15837   modify the updating steps for  $z_i$  and  $\psi$ , the parameters introduced by data augmentation,  
15838   to reflect that these only apply to the unmarked part of the population, in other words, to  
15839   the  $M - m$  individuals in our data. You can find the full MCMC code in the accompanying  
15840   R package `scrbook` by invoking `scrPID`. The R code below shows how to simulate SMR  
15841   data using the `scrbook` function `sim.pID.data`, and running an SMR model on the data,  
15842   both in **JAGS** and using `scrPID`. The model file `mknown.jag` in the `jags.model` call should  
15843   contain the code from Panel 19.1.

```

15844 > set.seed(2013)
15845 > N=80 # pop. size
15846 > m<-45 # no. marked
15847 > sigma=0.5
15848 > lam0=0.5
15849 > K=5
15850 > #make resighting array
15851 > gx<-gy<-seq(0,6,1)
15852 > X<-as.matrix(expand.grid(gx, gy))
15853 > J=dim(X)[1]
15854 > #limits of S
15855 > xlims<-ylims<-c(-1.5, 7.5)
15856 > #simulate data
15857 > dat=sim.pID.data(N=N, K=K, sigma=sigma, lam0=lam0, knownID=m, X=X, xlims=xlims,
15858 ylims=ylims, obsmod='pois', nmarked='known')

```

```

model{

#priors
psi ~ dbeta(1,1)
lam0 ~ dunif(0, 5)
sigma ~ dunif(0, 5)

#marked part
for(i in 1:m) {
  sm[i,1] ~ dunif(xlim[1], xlim[2])
  sm[i,2] ~ dunif(ylim[1], ylim[2])
  for(j in 1:J) {
    distm[i,j] <- sqrt((sm[i,1]-X[j,1])^2 + (sm[i,2]-X[j,2])^2)
    lambdam[i,j] <- lam0*exp(-distm[i,j]^2/(2*sigma^2))
    y[i,j]~dpois(lambdam[i,j]*K)
  }
}

##unmarked part
for(i in 1:M) {
  z[i] ~ dbern(psi)
  s[i,1] ~ dunif(xlim[1], xlim[2])
  s[i,2] ~ dunif(ylim[1], ylim[2])
  for(j in 1:J) {
    dist[i,j] <- sqrt((s[i,1]-X[j,1])^2 + (s[i,2]-X[j,2])^2)
    lambda[i,j] <- lam0*exp(-dist[i,j]^2/(2*sigma^2))
    for(k in 1:K) {
      yu[i,j,k] ~ dpois(lambda[i,j]*z[i])
    }
  }
}

for(j in 1:J) {
  for(k in 1:K) {nU[j,k] ~ dsum(yu[1,j,k],yu[2,j,k],yu[3,j,k],
  [...code shortened...],
  yu[79,j,k],yu[80,j,k])
}
}

N <- sum(z[])+m
}

```

---

Panel 19.1: **JAGS** code for SMR model with known number of marked individuals. In this example,  $M$ , the size of the augmented unmarked data set, is 80. Note that the arguments  $yu[4,j,k]$  to  $yu[78,j,k]$  of the `dsum()` function are omitted from the code to conserve space.

```

15859
15860 > ### prep data for analysis in JAGS
15861 > n<-dat$n-apply(dat$Yknown,2:3,sum)
15862 > y<-apply(dat$Yknown,1:2,sum)
15863
15864 > M<-80 #augmentation only for unmarked
15865
15866 > #initial values for latent y
15867 > yin<-array(0, c(M,J,K))
15868 > for(j in 1:J){
15869   for(k in 1:K){
15870     yin[1:M,j,k]<-rmultinom(1, n[j,k], rep(1/M, M))
15871   }
15872
15873 > data<-list(y=y, nU=n, m=m, M=M, J=J, X=X, xlim=xlims, ylim=ylims, K=K)
15874 > inits<-function(){list(sigma=runif(1), lam0=runif(1),
15875 sm=cbind(runif(m, xlims[1], xlims[2]), runif(m, ylims[1], ylims[2])),
15876 s=cbind(runif(M, xlims[1], xlims[2]), runif(M, ylims[1], ylims[2])),
15877 z=rep(1, M),yu=yin)}
15878 > params<-c('lam0', 'sigma', 'N', 'psi')
15879
15880 > #analysis in JAGS
15881 > library(rjags)
15882 > mod<-jags.model('mknown.jag', data, inits, n.chains=1, n.adapt=800)
15883 > out<-coda.samples(mod,params, n.iter=5000)
15884
15885 > #analysis with scrbook MCMC code
15886 > library(scrbook)
15887 > library(coda)
15888 > inits2<-function(){list(psi=runif(1), sigma=0.5, lam0=0.5,
15889 S=cbind(runif(M+m, xlims[1],xlims[2]), runif(M+m, ylims[1],ylims[2]))}
15890 > out2<-scrPID(n=n, X=X, y=dat$Yknown, M=M+m, obsmod = "pois", niters=5800,
15891 xlims=xlims, ylims=ylims,inits=inits2(),delta=c(0.1,0.1,0.5))


```

15892 You can look at the two sets of output invoking `summary(out)` for the **JAGS** analysis  
 15893 and `summary(window(mcmc(out2),start=801))` for the custom MCMC algorithm, excluding  
 15894 the first 800 iterations (analogous to the adaptive phase in **JAGS**). We summarized  
 15895 the results in Table 19.2.1. The posterior mean is slightly higher than the data-generating  
 15896 value of  $N = 80$ , but it falls comfortably within the credible intervals. As expected, es-  
 15897 timates from both implementations are very similar; slight differences are probably the  
 15898 result of the relatively low number of iterations. You will find that sometimes, **JAGS**  
 15899 produces an error message, upon trying to compile the model, saying that some of the  
 15900 observed  $y$  are inconsistent with parent nodes at initialization. We have mentioned before  
 15901 that **JAGS** cannot always auto-generate acceptable initial values, and we believe this  
 15902 is what is happening here. If this error occurs, just repeat the `jags.model` command,  
 15903 usually, model compilation is successful on a second attempt (assuming, of course, that  
 15904 you followed the code above correctly). We further find that the custom MCMC alorithm

**Table 19.1.** Posterior summaries of the spatial mark-resight model with known number of marks, analyzed in JAGS and using scrPID.

Implementation	Parameter	Mean	SD	2.5%	50%	97.5%
<b>JAGS</b>	$N$	88.72	6.75	77	88	103
	$\lambda_0$	0.53	0.08	0.39	0.53	70
	$\sigma$	1.29	0.02	1.26	1.30	1.32
	$\psi$	0.47	0.03	0.45	0.47	0.53
<b>scrPID</b>	$N$	86.01	7.58	73	85	102
	$\lambda_0$	0.54	0.08	0.39	0.53	0.72
	$\sigma$	0.48	0.03	0.42	0.48	0.53
	$\psi$	0.51	0.11	0.32	0.51	0.73

15905 tends to be faster than **JAGS**, which is why the examples and simulation studies shown  
 15906 in the following sections were run solely in **R**.

### 19.3 UNKNOWN NUMBER OF MARKED INDIVIDUALS

15907 Now let us consider the case where we do not know the exact number of marked individuals  
 15908 available for resighting so that we have to capture an individual at least once to be sure that  
 15909 it is available. Unless we have a direct means of confirming the number of marked animals  
 15910 available for resighting, treating this number as unknown is probably more realistic in most  
 15911 circumstances. As a consequence of not knowing the exact number of marked individuals,  
 15912 we cannot observe all-zero encounter histories. When using maximum likelihood inference,  
 15913 this situation requires a model where detection rates of known individuals are modeled  
 15914 using a zero-truncated distribution (McClintock et al., 2009a). If we did not account for  
 15915 the fact that zeros are unobservable, our estimates of detection rates would be artificially  
 15916 inflated and estimates of population size would be negatively biased.

15917 Working with zero-truncated distributions in a spatial mark-resight setting is less  
 15918 straight-forward than for non-spatial mark-resight. A marked individual only has to show  
 15919 up once, anywhere on the resighting array, for us to know that it is there. When resightings  
 15920 are pooled across the entire sampling grid, then the total individual counts  $\sum_j y_{ijk}$  have  
 15921 to be  $> 0$  for all resighted individuals and a zero-truncated distribution can be used to  
 15922 model these counts. However, we are concerned with trap-specific encounters,  $y_{ijk}$ , which  
 15923 can easily be 0 for a resighted individual, as long as a single  $y_{ij}$  is  $> 0$ . Thus, the zero-  
 15924 truncation does not apply to the individual and trap specific counts we observe, but only  
 15925 to the sum of these counts over all traps.

15926 As an alternative to a zero-truncated distribution, in a Bayesian framework, we can  
 15927 make use of data augmentation to estimate the number of marked individuals (McClintock  
 15928 and Hoeting, 2010). In the SMR framework that means that we create two augmented  
 15929 data sets, one for the marked individuals and one for the unmarked, and estimate their  
 15930 number separately, having them share the parameters of the detection model. Sometimes  
 15931 we may know the maximum number that were ever marked before a resighting survey,  
 15932 in which case we can use that number as the data augmentation limit for the marked  
 15933 data set. Panel 19.2 shows the **JAGS** code for the SMR model with unknown number of  
 15934 marks, which is identical to the one in Panel 19.1, but for the augmentation of the marked

15935 data set. This introduces both a data augmentation parameter, `psim`, and an auxiliary  
 15936 “alive state” variable, `zm[i]`, into the description of the marked data model. Again, we  
 15937 provide an alternative, **R**-only MCMC algorithm within `scrbook - scrPID.um`.

15938 Note that we could look at the problem of not knowing the number of marked individuals  
 15939 in the study population as a manifestation of a lack of population closure. In  
 15940 other words, marked individuals may have emigrated, died or lost their marks in the time  
 15941 between marking and resighting. If we have information on the rates of these events, or  
 15942 a series of resighting surveys, we could develop an open population model for the marks  
 15943 in our population and estimate their number at a given resighting survey in this fashion.  
 15944 This kind of SMR model remains to be explored.

### 15945 19.3.1 Canada geese in North Carolina

15946 We applied the spatial mark-resight model with an unknown number of marks and a  
 15947 binomial encounter process to a dataset of Canada goose resightings (Rutledge, 2012)  
 15948 XXXget full citation with LizXXX. During the molt of 2008, 751 individual geese were  
 15949 captured and marked with neck and leg bands in Greensboro, North Carolina (Fig. 19.1).  
 15950 Geese were resighted at 87 locations on 81 resighting events over a period of 18 months.  
 15951 In addition to the banded geese, the number of unmarked geese was recorded during each  
 15952 resighting event. Here, we only looked at a subset of the data, from mid July to the  
 15953 end of October 2008, which corresponds to the first part of the post-molt season, before  
 15954 migratory Canada geese arrive in North Carolina. We treated this population as closed  
 15955 over this period. During this part of the study, 57 of the resighting sites were visited and  
 15956  $n = 654$  marked geese were resighted 3994 times at 40 different sites. In addition, 7944  
 15957 sightings of unmarked geese were recorded at 48 sites.

15958 In the model, we allowed  $\sigma$  to vary between males and females. We set the size of  
 15959 the augmented unmarked data set to 7000. We used the total number marked geese  
 15960 (751) as the upper limit for the augmented marked data set. We ran 50000 MCMC  
 15961 iterations and removed a burn-in of 5000 iterations. To describe the state-space, we  
 15962 buffered the resighting locations by 4.5 km. We assumed that marked geese were a random  
 15963 sample from the state-space, which seems reasonable because (a) marking took place across  
 15964 most of the extent of the resighting array; and (b) marking was done during the molting  
 15965 period, when geese are fairly immobile, and it seems reasonable to assume that, once the  
 15966 molt is complete, the marked geese redistributed themselves. We provide all the data  
 15967 (`data('geesedata')`) and functions (`geeseSMR`) for you to repeat this analysis but be  
 15968 aware that given the large data set it will take days to do so. The **R** code to set up the  
 15969 data and run 5000 iterations of the model for the geese data is given as an example on  
 15970 the help page for `geeseSMR`. The model results, including the derived parameter density  
 15971 ( $D$ ) in individuals per  $km^2$  are shown in Tab. 19.3.1.

15972 We see that credible intervals of estimates are pretty narrow, surely an effect of the  
 15973 large data set. Estimates of  $m$  indicate that most of the 751 geese originally banded are  
 15974 still alive and marked, which is not surprising, given that not much time passed between  
 15975 marking and this first resighting session. The parameter  $\phi$  in this model is the probability  
 15976 of being a male, a measure of the sex ratio of the population, which is slightly biased in  
 15977 favor of females.

```

model{

#priors
psim ~ dbeta(1,1)
psi ~ dbeta(1,1)
lam0 ~ dunif(0, 5)
sigma ~ dunif(0, 5)

#marked part
for(i in 1:max) {
  zm[i]~dbern(psim)
  sm[i,1] ~ dunif(xlim[1], xlim[2])
  sm[i,2] ~ dunif(ylim[1], ylim[2])
  for(j in 1:J) {
    distm[i,j] <- sqrt((sm[i,1]-X[j,1])^2 + (sm[i,2]-X[j,2])^2)
    lambdam[i,j] <- lam0*exp(-distm[i,j]^2/(2*sigma^2))*zm[i]
    y[i,j]~dpois(lambdam[i,j]*K*zm[i])
  }
}

##unmarked part
for(i in 1:M) {
  z[i] ~ dbern(psi)
  s[i,1] ~ dunif(xlim[1], xlim[2])
  s[i,2] ~ dunif(ylim[1], ylim[2])
  for(j in 1:J) {
    dist[i,j] <- sqrt((s[i,1]-X[j,1])^2 + (s[i,2]-X[j,2])^2)
    lambda[i,j] <- lam0*exp(-dist[i,j]^2/(2*sigma^2))
    for(k in 1:K) {
      yu[i,j,k] ~ dpois(lambda[i,j]*z[i])
    }
  }
}

for(j in 1:J) {
  for(k in 1:K) {nU[j,k] ~ dsum(yu[1,j,k],yu[2,j,k],yu[3,j,k],
  [...code shortened...],
  yu[79,j,k],yu[80,j,k])
}
}

Nu <- sum(z[])
Nm<-sum(zm[])
N<-Nu+Nm
}

```

Panel 19.2: JAGS code for SMR model with unknown number of marked individuals. In this example,  $M$ , the size of the augmented unmarked data set, is 80. Note that the arguments  $yu[4,j,k]$  to  $yu[78,j,k]$  of the `dsum()` function are omitted from the code for space reasons.

**Table 19.2.** Posterior summaries of the spatial mark-resight model for Canada geese in North Carolina.  $N$  is the total population size of marked and unmarked individuals;  $m$  is the number of marked individuals.

	Mean	SD	2.5%	50%	97.5%
$m$	739.77	3.24	733	740	746
$N$	5756.10	90.68	5577	5757	5932
$D$	13.76	0.19	13.38	13.76	14.14
$\lambda_0$	0.19	<0.01	0.18	0.19	0.19
$\sigma$ , females	1.29	0.02	1.26	1.30	1.32
$\sigma$ , males	1.06	0.02	1.02	1.06	1.11
$\psi$ , marked	0.99	<0.01	0.98	0.99	0.99
$\psi$ , unmarked	0.72	0.01	0.69	0.72	0.74
$\phi$	0.36	0.02	0.32	0.36	0.39

## 19.4 IMPERFECT IDENTIFICATION OF MARKED INDIVIDUALS

Often during resighting, it may be possible to see that an individual is marked but impossible to determine its individual identity. In such a situation in addition to the  $y_{ijk}$  and  $n_{jk}$ , we also have site and occasion specific counts of marked but unidentified individuals,  $r_{jk}$ . Here, the individual encounter histories of marked animals are incomplete, and if we used these incomplete data to inform the detection parameter of the model, we would run the risk of underestimating detection/trap encounter rate and overestimating abundance. Some non-spatial mark-resight models do not require that marked animals be identified individually, as long as the marking status can be observed unambiguously, but ignoring individual level information means that we cannot accommodate heterogeneity in detection (McClintock and White, 2012). In a spatial framework we could ignore marked and unmarked status completely and apply the model by Chandler and Royle (2013) discussed in Chapt. 18. But, that would mean losing important information on individual detection and movement. Therefore, being able to retain the individual identity of records that can be identified while at the same time accounting for imperfect identification of marked individuals is extremely useful.

McClintock et al. (2009a,b) suggest an intuitive means of correcting for this bias in a non-spatial model framework when dealing with a Poisson encounter model (or sampling with replacement). When marked but unknown resightings are part of the data, the expected number of records of unmarked individuals,  $n$ , changes from Eq. 19.1.3 to:

$$E(n) = (N - m)\lambda + \eta/m$$

where  $\lambda$  is the individual encounter rate estimated from the known resighted individuals and  $\eta$  is the number of records of marked but unidentified individuals. So, because the observed  $\lambda$  is known to be too low, the average number of unidentified pictures per known individual is added as a correction factor. This procedure assumes that the inability to identify a marked individual occurs at random throughout the population, which seems to be a reasonable assumption under most circumstances.

We can translate this same concept to the spatial mark-resight models. In the spatial framework we are interested in the individual and trap specific encounter rate,  $\lambda_{ij}$ . Further, we do not look at the sum of all records of unmarked individuals, but formulate the



**Figure 19.1.** Banded and unbanded Canada geese in a parking lot in Greensboro, North Carolina. (Photo credit: M.E. Rutledge, NCSU Canada goose project)

model conditional on the latent individual encounter histories. Thus, instead of using  $\eta/m$  as a correction factor, we need something that applies at the individual and trap level. If we take the sum of all correctly identified records of marked individuals,  $\sum y_c$  and divide it by the total number of records of marked individuals,  $\sum y_m$ , we get the average rate of correct individual identification for marked individuals, say,  $c$ :

$$c = \sum y_c / \sum y_m.$$

We can then apply  $c$  as a correction factor for  $\lambda_0$  for the marked individuals.

A more formal, model-based way to specify  $c$  is by assuming that

$$\sum y_c \sim \text{Binomial}(\sum y_m, c)$$

and estimating  $c$  as another model parameter, so that we account for the uncertainty about it. For the marked individuals we can then multiply  $\lambda_0$  by  $c$  to account for the fact that we observe incomplete individual encounter histories. Since we don't have this

16016 identification issue for unmarked individuals, their baseline trap encounter rate remains  
 16017 as before simply  $\lambda_0$  (or in other words,  $c$  for unmarked individuals equals 1).

16018 Incomplete individual identification of marked individuals is easily incorporated into  
 16019 our **JAGS** model, no matter whether  $m$  known or unknown, by adding the following two  
 16020 lines of code:

```
16021 c~dbeta(1,1) #prior for c
16022 npics[1]~dbin(c, npics[2]) #model for c
16023 and modifying the marked observation model description to
16024 y[i,j]~dpois(lambdam[i,j]*c*K)
```

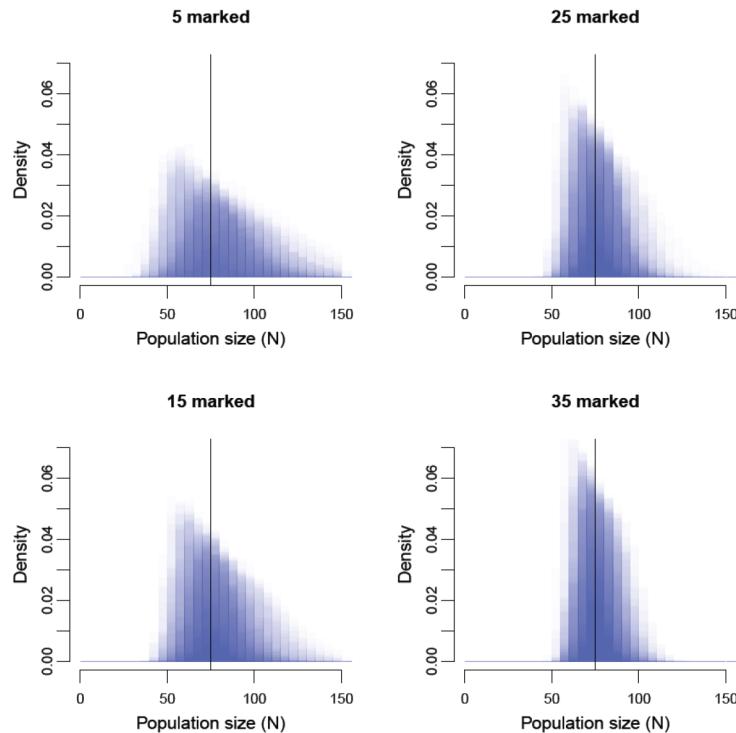
16025 Here, the data object **npics** is a vector with the number of correctly identified records  
 16026 of marked individuals and the total number of marked records. Accounting for imper-  
 16027 fect identification of marks is also included as an option in the **scrPID** and **scrPID.um**  
 16028 functions. Choosing an uninformative (and conjugate) beta(1,1) prior for  $c$ , within the  
 16029 **scrPID** algorithm we can update  $c$  directly from its full conditional distribution, which is  
 16030  $\text{beta}(1 + \sum y_c, 1 + (\sum y_m - \sum y_c))$ . We show an example of using  $c$  in an analysis in sec.  
 16031 19.6.

16032 Observe that now, in addition to assuming that failure to identify marked individuals  
 16033 occurs at random throughout the population, we also assume that it occurs at random  
 16034 throughout space, i.e. our success of identifying a marked individual does not depend on  
 16035 the trap we encounter it in. As long as individuals are identified based on the same type of  
 16036 tags the assumption that failure to identify marked individuals occurs at random through-  
 16037 out the population should be valid. The assumption that failure to identify marked indi-  
 16038 viduals occurs at random in space could be violated, for example when spatially varying  
 16039 habitat conditions influence the ability to recognize individual tags, or when an observer  
 16040 effect influences individual identification rates. While we haven't experimented with it,  
 16041 we believe that the approach described above could readily be extended to account for  
 16042 these differences. For example, identification rates could be calculated separately for dif-  
 16043 ferent observers, or be modeled as functions of habitat covariates. As an alternative to the  
 16044 approach we present here, model development could explore assigning records of marked  
 16045 but unidentified individuals to marked individuals in a fashion similar to how unmarked  
 16046 records are assigned to hypothetical individuals in this model, namely, based on the loca-  
 16047 tion of the record and the estimates of home range centers of marked individuals. While  
 16048 this is computationally more advanced it would make full use of the spatial information  
 16049 of the unmarked records.

## 19.5 HOW MUCH INFORMATION DO MARKED AND UNMARKED INDIVIDUALS CONTRIBUTE?

16050 It is intuitive that having marked individuals in the study population should lead to more  
 16051 accurate and precise parameter estimates than when no individuals are identifiable. To  
 16052 evaluate how strongly adding marked individuals to a population improves parameter  
 16053 estimates, Chandler and Royle (2013) performed a simulation study. They used a  $15 \times$   
 16054 15 resighting grid and simulated detection data of  $N = 75$  individuals in a  $20 \times 20$  units  
 16055 state-space over  $k = 5$  occasions with  $\sigma = 0.5$  and  $\lambda_0 = 0.5$ . They generated 100 datasets

16056 each for  $m = (0, 5, 15, 25, 35)$  where  $m$  is the known number of marked individuals  
 16057 randomly sampled from the population.



**Figure 19.2.** Overlaid posterior distributions of  $N$  from 100 simulations for four levels of marked individuals.

16058 Without any marked individuals in the population, the posterior distribution of  $N$   
 16059 turned out to be highly skewed, but the mode was still an approximately (frequentist)  
 16060 unbiased point estimator of  $N$ . As anticipated, posterior precision increased substantially  
 16061 with the proportion of marked individuals (Tab. 19.3 and Fig. 19.2). The posterior mode  
 16062 was approximately unbiased as a point estimator, and the relative root-mean squared error  
 16063 decreased from 0.246 when no individuals were marked to 0.085 when 35 individuals were  
 16064 marked (Tab. 19.3). Coverage was nominal for all values of  $m$  and posterior skew greatly  
 16065 diminished with increasing  $m$  (Tab. 19.3).

16066 As we saw in the previous chapter, the spatial correlation in unmarked counts can  
 16067 be sufficient to obtain estimates of movement and detection parameters. However, only  
 16068 marked and thus identifiable individuals provide us with direct information about these  
 16069 parameters and may well dominate estimates. To single out the contribution of marked  
 16070 and unmarked individuals to parameter estimates, we re-ran the same simulations but

**Table 19.3.** Posterior mean, mode, and associated relative RMSE for simulations in which  $m$  of  $N=75$  individuals were marked. One hundred simulations of each case were conducted.

	Parameter	Mean	rRMSE	Mode	rRMSE	BCI
m=0	$N$	85.866	0.259	77.720	0.242	0.950
	$\lambda_0$	0.506	0.180	0.488	0.182	0.960
	$\sigma$	0.495	0.115	0.486	0.113	0.960
m=5	$N$	80.898	0.184	76.360	0.182	0.970
	$\lambda_0$	0.510	0.178	0.494	0.180	0.950
	$\sigma$	0.496	0.089	0.488	0.086	0.970
m=15	$N$	79.028	0.148	76.250	0.147	0.950
	$\lambda_0$	0.508	0.163	0.494	0.164	0.950
	$\sigma$	0.496	0.073	0.492	0.071	0.970
m=25	$N$	77.765	0.114	75.810	0.113	0.950
	$\lambda_0$	0.511	0.153	0.498	0.157	0.950
	$\sigma$	0.496	0.067	0.493	0.065	0.940
m=35	$N$	76.446	0.085	74.900	0.085	1.000
	$\lambda_0$	0.513	0.142	0.501	0.144	0.950
	$\sigma$	0.497	0.056	0.493	0.057	0.940

let  $\sigma$  and  $\lambda_0$  be updated based solely on the data of marked individuals. Results are summarized in Tab. 19.4. We see that if we update  $\lambda_0$  and  $\sigma$  based on marked individuals only, estimates of these parameters are more biased and less precise. For estimates of  $N$ , especially for  $m=5$  and  $m=15$ , we observe a stronger positive bias, lower accuracy and considerably lower BCI coverage as compared to when both marked and unmarked individuals contribute to parameter estimates (Tab. 19.4). Thus, unmarked individuals do actually contribute noticeably to estimating model parameters.

**Table 19.4.** Posterior mean, mode, and associated relative RMSE for simulations in which  $m$  of  $N=75$  individuals were marked and unmarked individuals did not contribute to estimating  $\lambda_0$  and  $\sigma$ . One hundred simulations of each case were conducted.

	Parameter	Mean	RMSE	Mode	RMSE	BCI
m=5	$N$	88.621	0.369	83.139	0.421	0.810
	$\lambda_0$	1.255	1.247	0.606	1.148	0.950
	$\sigma$	0.472	0.252	0.426	0.333	0.910
m=15	$N$	81.031	0.192	78.361	0.175	0.820
	$\lambda_0$	0.535	0.281	0.476	0.284	0.970
	$\sigma$	0.503	0.109	0.490	0.107	0.940
m=25	$N$	78.206	0.129	76.594	0.123	0.920
	$\lambda_0$	0.531	0.204	0.496	0.202	0.960
	$\sigma$	0.497	0.081	0.489	0.084	0.950
m=35	$N$	76.833	0.099	75.422	0.096	0.940
	$\lambda_0$	0.528	0.192	0.505	0.186	0.940
	$\sigma$	0.499	0.069	0.493	0.070	0.960

## 19.6 INCORPORATING TELEMETRY DATA

As we expected, parameter estimates of spatial mark-resight models get better the more marked individuals we have in our study population. While this is great advice in theory, it may not be very helpful in practice, especially when dealing with animals that are hard or somewhat dangerous to capture, such as large carnivores. Oftentimes, studies involving the physical capture of such animals will employ telemetry tags in order to learn about the study species' spatial ecology and behavior. In the context of spatial mark-resight models, the actual locations collected by telemetry tags can provide detailed information on individual location and movement, and being able to incorporate this information directly into the SMR model should improve estimates of these parameters, especially when resighting information is sparse.

So how could we combine resighting data and telemetry data in a unified mark-resight model? Recall that the basic SCR model underlying all the SMR models we discuss here uses a Gaussian kernel to describe the trap encounter model. By using this function, we can relate the parameters  $\sigma$  and  $\mathbf{s}_i$  directly to those from a bivariate normal model of space usage, with mean =  $\mathbf{s}_i$ , and variance-covariance matrix  $\Sigma$ , where the variance in both dimensions is  $\sigma^2$  and the covariance is 0. Ordinarily, these parameters are estimated directly from the spatial distribution of individual captures/resightings. Telemetry data, however, provide more detailed information on individual location and movement, since the resolution and extent of the data are not limited by the trapping grid and potentially more locations can be accumulated through telemetry than resighting (depending on the monitoring frequency and resighting rates of individuals).

By assuming that the  $R_i$  locations of individual  $i$ ,  $\mathbf{l}_i$  (consisting of a pair of x and y coordinates,  $l_{ix}$  and  $l_{iy}$ ), are a bivariate normal random variable:

$$\mathbf{l}_i \sim \text{Normal}_2(\mathbf{s}_i, \Sigma)$$

we can estimate  $\sigma$  as well as  $\mathbf{s}_i$  for the collared individuals directly from telemetry locations, using their full conditional distributions:

$$[\sigma | \mathbf{l}, \mathbf{s}] \propto \left\{ \prod_{i=1}^m \prod_{r=1}^{R_i} \frac{1}{2\pi\sigma^2} \exp \left( -1/2 \left[ \frac{(l_{irx} - s_{ix})^2}{\sigma^2} + \frac{(l_{iry} - s_{iy})^2}{\sigma^2} \right] \right) \right\} * [\sigma]$$

and

$$[\mathbf{s}_i | \mathbf{l}, \sigma] \propto \left\{ \prod_{r=1}^{R_i} \frac{1}{2\pi\sigma^2} \exp \left( -1/2 \left[ \frac{(l_{irx} - s_{ix})^2}{\sigma^2} + \frac{(l_{iry} - s_{iy})^2}{\sigma^2} \right] \right) \right\} * [\mathbf{s}_i]$$

For the unmarked individuals  $\mathbf{s}_i$  are estimated as described before conditional on their latent encounter histories. Note that the bivariate normal model assumes that locations are independent of each other. If you have frequent telemetry fixes, for example from GPS collars that report animal locations every few hours or more, this assumption seems unrealistic and it might be advisable to thin your telemetry data (maybe to daily fixes) in order to approximate independence. Not all marked individuals need to be telemetry-tagged, but telemetry data should correspond to the period over which resighting surveys were conducted (as we discussed in Chapt. 5, both the  $\mathbf{s}_i$  and  $\sigma$  should only be interpreted against the specific sampling period).

Again, implementation of this model extension is straight-forward, both in **JAGS** and **R**. Take the SMR model description for the case where  $m$  is known (Panel 19.1). Then, all we have to do is add a description of the bivariate normal model for the telemetry locations, here `locs`, into the loop over the  $m$  marked individuals:

```

16117 [...] parts of model code omitted...
16118
16119 for(i in 1:m) {
16120
16121   sm[i,1] ~ dunif(xlim[1], xlim[2])
16122   sm[i,2] ~ dunif(ylim[1], ylim[2])
16123
16124   #telemetry model
16125   for (r in off1[i]:off2[i]){
16126     locs[r,1]~dnorm(sm[i,1], 1/(sigma^2))
16127     locs[r,2]~dnorm(sm[i,2], 1/(sigma^2))
16128   }
16129
16130   for(j in 1:J) {
16131     distm[i,j] <- sqrt((sm[i,1]-X[j,1])^2 + (sm[i,2]-X[j,2])^2)
16132     lambdam[i,j] <- lam0*exp(-distm[i,j]^2/(2*sigma^2))
16133     y[i,j]~dpois(lambdam[i,j]*K)
16134   }
16135 }
16136
16137 [...] parts of model code omitted...

```

The data object `locs` is a table with all  $\sum_i^m R_i$  telemetry locations. The two vectors `off1` and `off2` describe which subset of this matrix belongs to individual  $i$ . So if, say, the locations for individual 1 are contained in the first 10 rows of `locs`, `off1` and `off2` would be 1 and 10 for  $i = 1$ ; and if the locations of individual 2 are in the following 15 rows, `off1` and `off2` for  $i = 2$  would be 11 and 25, and so on. For the implementation of this SMR model with telemetry data in **R**, see the `scrPID.tel` function in `scrbook`. In a nutshell, in the MCMC algorithm we replaced the Metropolis-Hastings updating steps for  $\sigma$  and  $s_i$  of marked individuals, which were originally conditional on the resighting data, with updating steps conditional on the telemetry data. This is not quite what the above **JAGS** code does; rather **JAGS** will update these parameters conditional on both the telemetry *and* the resighting data. We could easily re-write `scrPID.tel` to do that, but believe that for most applications, the information on location and movement contained in the telemetry data will outweigh that in the resighting data, so that the resulting loss of information should be minimal.

### 19.6.1 Raccoons on the Outer Banks of North Carolina

Sollmann et al. (2013) applied a spatial mark-resight model with telemetry data to a camera-trap and radio-telemetry data set from the raccoon population on South Core Banks, a barrier island within Cape Lookout National Seashore, North Carolina. Between

16156 May and September 2007, 131 raccoons were marked with dog collars and large individ-  
16157 ually numbered cattle tags. Individuals were marked throughout the island, so that (a)  
16158 we do not have to deal with sensitivity to choice of the state-space, because it is clearly  
16159 defined by nature; and (b) it is reasonable to assume that marked raccoons are a random  
16160 sample of individuals from this state-space. Of the 131 tagged individuals, 44 were also  
16161 equipped with radio collars. Collared individuals were located using a VHF receiver and  
16162 antenna, and their locations were estimated approximately weekly. Twenty camera traps  
16163 were set up along the length of South Core Banks and camera trapping data collected be-  
16164 tween October 1 2007 to January 22 2008 constituted the resighting data in this analysis.  
16165 During this period 104 marked individuals, 38 radio-collared, were alive and available for  
16166 resighting with camera traps.



**Figure 19.3.** Camera trap picture of a raccoon marked with a cattle tag that cannot be read to determine individual identity. Taken on South Core Banks, North Carolina. (*Photo credit: Arielle Parsons*)

16167 The state-space  $\mathcal{S}$  was the entire area of South Core Banks island. A change in the  
16168 number of photocaptures over the course of the study suggested a variation of detection  
16169 rate with time. Since date recording in cameras malfunctioned, photographic records  
16170 could only be assigned to the time interval between subsequent trap checks, and these  
16171 intervals between checks are referred to as sampling occasions. These occasions ranged  
16172 from 2 to 43 days;  $\lambda_0$  was standardized to 7-day intervals and allowed to change with  
16173 sampling occasion. Since not all pictures of marked raccoons could be identified to the

**Table 19.5.** Summary statistics of posterior distributions from spatial mark-resight model for raccoon camera trapping and telemetry data. Baseline trap encounter rate  $\lambda_0$  was standardized to 7-day intervals;  $\lambda_0$  and the probability of identifying a picture of a marked individual,  $c$ , were allowed to vary among the 6 sampling occasions ( $t$ );  $\sigma$  is estimated from telemetry data of 38 radio-collared individuals.

	Mean (SE)	2.5%	50%	97.5%
$N$	186.71 (14.81)	162	185	220
$D$	8.29 (0.66)	7.19	8.22	9.77
$\lambda_0$ ( $t=1$ )	0.24 (0.05)	0.16	0.23	0.34
$\lambda_0$ ( $t=2$ )	0.40 (0.08)	0.26	0.39	0.57
$\lambda_0$ ( $t=3$ )	0.11 (0.03)	0.06	0.11	0.17
$\lambda_0$ ( $t=4$ )	0.30 (0.07)	0.17	0.29	0.46
$\lambda_0$ ( $t=5$ )	0.03 (0.01)	0.02	0.03	0.06
$\lambda_0$ ( $t=6$ )	0.03 (0.01)	0.02	0.03	0.05
$\sigma$	0.49 (0.01)	0.47	0.49	0.51
$c$ ( $t=1$ )	0.55 (0.09)	0.38	0.55	0.71
$c$ ( $t=2$ )	0.39 (0.11)	0.18	0.39	0.62
$c$ ( $t=3$ )	0.29 (0.11)	0.11	0.29	0.52
$c$ ( $t=4$ )	0.38 (0.16)	0.10	0.36	0.71
$c$ ( $t=5$ )	0.38 (0.16)	0.10	0.36	0.71
$c$ ( $t=6$ )	0.30 (0.14)	0.08	0.29	0.60

16174 individual level, the authors applied the correction factor  $c$  as described in sec. 19.4,  
 16175 estimated separately for each occasion.

16176 Camera-traps recorded 117 pictures of unmarked raccoons, 33 pictures of 18 marked  
 16177 and identifiable raccoons, and 49 records of marked but not individually identifiable indi-  
 16178 viduals (Fig. 19.3). An average of 16.32 telemetry locations (SD 4.91) were collected for  
 16179 each of the 38 collared individuals. Raccoon abundance on the island was estimated at  
 16180 186.71 (SE 14.81) individuals, which translated to a density of 8.29 (SE 0.66) individuals  
 16181 per  $km^2$ . Parameter estimates are listed in Tab. 19.5.

16182 In this study, although a large number of raccoons were tagged, photographic data of  
 16183 these tagged individuals were surprisingly sparse. Analysis of the photographic data set  
 16184 without the telemetry data did not render usable estimates as parallel Markov chains did  
 16185 not converge. One reason for the relatively sparse data was the camera trap study design:  
 16186 traps were spaced on average 1.77 km apart, which is about 3.5 times  $\sigma$ . Consequently,  
 16187 very few individual raccoons were photographed at more than one trap. Under these  
 16188 circumstances, the telemetry data provide the necessary spatial information to estimate  
 16189  $\sigma$  and the activity centers of individual animals and thus make other model parameter  
 16190 estimable. Similarly, in a camera-trapping study on Florida panthers (*Puma concolor*  
 16191 *coryi*), Sollmann et al. (in revision), including telemetry data from the 3 individuals  
 16192 that were collared and known to use the study area resulted in density estimates with  
 16193 considerably higher precision as compared to preliminary estimates *without* telemetry  
 16194 location data, reducing the width of the 95 % BCI by about 60 %. Such improvements  
 16195 in precision of estimates is especially important when we are interested in changes in the  
 16196 population over time.

## 19.7 MARKED ANIMALS ARE NOT A RANDOM SAMPLE FROM THE STATE-SPACE

As discussed in sec. 19.1.4, all the previously developed SMR models assume that marked individuals are a random sample, both spatially and demographically, from the population of the state-space. For many studies it may not be feasible to strive to meet or approximate this assumption and it is thus important to generalize SMR models to situations where marking does not take place throughout  $\mathcal{S}$ . If you think about it, even if you set up marking traps throughout the state-space, individuals at the border of  $\mathcal{S}$  would be exposed to fewer traps and be less likely to be caught, thus creating a gradient in the proportion of marked to unmarked animals (although if  $\mathcal{S}$  is large enough this is probably negligible). Here, we will develop one tentative approach to dealing with this situation, by having marked individuals be a random (uniformly distributed) sample from a smaller region within  $\mathcal{S}$ , say  $\mathcal{B}$ . The central point of this approach is that it establishes a spatial context for the marked part of the population. This spatial context is independent of  $\mathcal{S} - \mathcal{B}$  remains constant – and it provides a formal distribution of marked animals – uniform in  $\mathcal{B}$ , none outside of  $\mathcal{B}$  – so that unmarked animals can be distributed in a way that overall density is constant.

### 19.7.1 Marked animals are uniformly distributed in a smaller area

Imagine we perform an area search in a square,  $\mathcal{B}$ , for some species we want to study, maybe a reptile, and we mark all individuals we encounter. We conduct our sampling in a way that we can assume that the individuals we marked are uniformly distributed in  $\mathcal{B}$ . This also entails the assumption that  $\mathcal{B}$  can be clearly defined. We will come back to these assumptions in a minute. We then perform resighting surveys of some sort in an area that overlaps  $\mathcal{B}$ , so that, when we set a state-space around our resighting locations,  $\mathcal{B}$  is completely contained within  $\mathcal{S}$  (Figure 19.4), and we assume that the number of marked animals,  $m$ , is known. We further assume that individuals that were marked in  $\mathcal{B}$  continue to live within  $\mathcal{B}$  when resighting surveys are conducted, i.e. their activity centers do not shift during the complete mark-resight study. That means that we assume population closure across both the marking and the resighting part of the study.

Let the total population of  $\mathcal{B}$  be  $N_B$ . Under the conditions specified above, the number of marked animals  $m$  can be described as the outcome of a binomial random variable

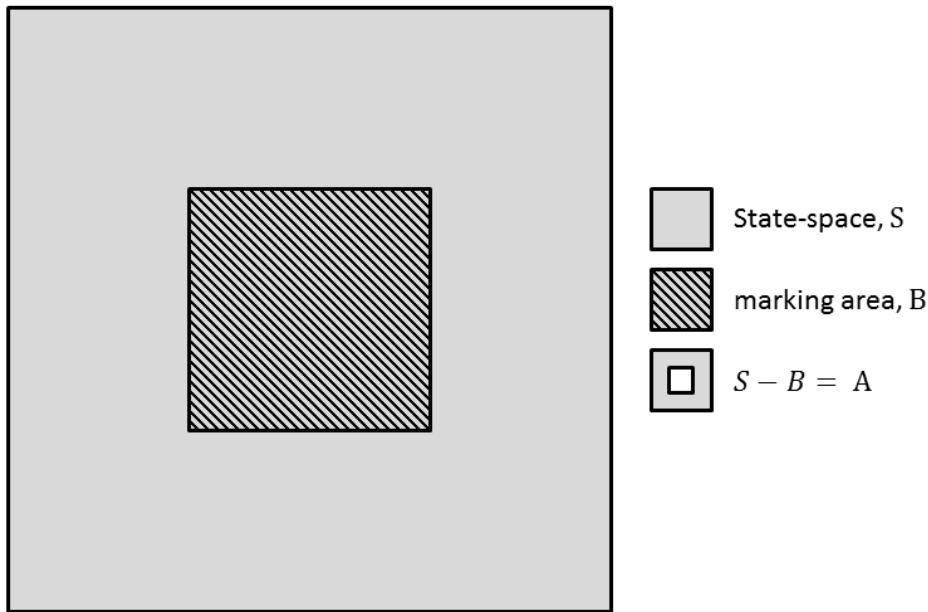
$$m \sim \text{Binomial}(\theta, N_B)$$

where  $\theta$  is the probability that an animal living in  $\mathcal{B}$  is marked. Remember that, by definition, all marked animals live inside  $\mathcal{B}$ . We now have to make sure that unmarked individuals get distributed across  $\mathcal{S}$  and  $\mathcal{B}$  so that *overall* density is constant. Let  $\mathcal{A}$  be  $\mathcal{S} - \mathcal{B}$ , i.e., the area of  $\mathcal{S}$  *not* covered by  $\mathcal{B}$  (Figure 19.4). Then, the proportion of  $N$  that should fall inside  $\mathcal{B}$ , say  $\pi_B$ , can be expressed as  $\mathcal{B}/(\mathcal{A} + \mathcal{B})$ . Consequently, the proportion of  $N$  in  $\mathcal{A}$ ,  $\pi_A$ , is  $\mathcal{A}/(\mathcal{A} + \mathcal{B})$ . Conditioning  $\pi_B$  on being an unmarked animal, we obtain

$$\pi_B | \text{unmarked} = \frac{(1 - \theta) * \pi_B}{U/N}$$

And we can use this conditional probability as prior probability for the activity centers of unmarked individuals to fall within  $\mathcal{B}$ . In other words, we now have two sets of priors

for activity centers. For marked animals,  $[s_i] \sim \text{Uniform}(\mathcal{B})$ . For unmarked animals, we introduce a binary variable, say  $b$ , and let  $b = 1$  mean that  $s_i$  lies within  $\mathcal{B}$ ; then  $[s_i] \sim \text{Bernoulli}(p_{iB} | \text{unmarked})$ . Because manipulating areas that are not simple rectangles (in this example,  $\mathcal{A}$ ) in **JAGS** is not straight forward, we wrote our own MCMC algorithm for this model, which can be found in the **scrbook** package by invoking **scrPIDBox**. A full example of how to simulate and analyze data under this model is given on the help page for **scrPIDBox**.



**Figure 19.4.** Relationship between marking area  $\mathcal{B}$  and state-space,  $\mathcal{S}$ .

The above model is an approach to specifying a spatial reference frame for marked individuals if these are not sampled uniformly from  $\mathcal{S}$ , and provides us with the ability to distribute unmarked individuals proportionally between the marking area and the rest of the state-space, so that an overall uniform density is obtained. Some of the assumptions of the model, however, are reminiscent of traditional capture-recapture and thus, suffer from the same shortcomings.  $\mathcal{B}$  needs to be clearly defined as the area the marked individuals live in, but how do we define it? Imagine again that  $\mathcal{B}$  is a quadrad search plot. Surely, we could capture an individual at the edge of the plot, whose activity center is located *off* that plot. Not accounting for this effect would overestimate density in  $\mathcal{B}$ . This is the equivalent of having to define an effective area sampled in traditional capture-recapture in order to estimate density. Further, we assume that  $\theta$ , the probability of an individual within the plot being marked is the same for all individuals in  $\mathcal{B}$ . But we discussed early

on in this book that this is unlikely to be true, because exposure to sampling depends on an individual's home range overlap with the sampled area. So individuals near the edge of  $\mathcal{B}$  are less likely to be marked than those in the center, assuming we dispense marking effort uniformly across  $\mathcal{B}$  (maybe we could counteract this effect to some extent by creating a decreasing gradient of sampling effort from the edge of the plot to the center).

### 19.7.2 Combining marking and resighting models

We can look at this approach from a slightly different angle. Effectively, what we did here is combine a non-spatial capture recapture model – more specifically, model  $M_0$  with equal capture probability – for the marking process with a spatial mark-resight model for the resighting process. In our simplified example, we only have a single marking occasion, but we can estimate  $\theta$ , the probability of being captured and marked, because we have enough information about how many unmarked individuals occur in  $\mathcal{B}$  coming from the spatial resighting model. While the underlying assumptions of the non-spatial marking model are questionable, we believe that the idea of combining marking and resighting into a unified model holds the key to developing a generalized spatial mark-resight model that does not rely on animals being marked throughout  $\mathcal{S}$ , at least when marking and resighting occur in a short enough time frame that the population can be assumed closed. As such, in spite of some shortcomings, the present approach is an important conceptual step forward. We can think of alternative distributions for marked individuals to move away from a completely non-spatial description of the marking process; for example, activity centers of marked individuals could follow a bivariate normal distribution around the centroid of the marking array or plot, so that the probability of being a marked individual is conditional on where you live and decreases with distance to the collection of marking locations. This avoids having to arbitrarily define an area,  $\mathcal{B}$ , but is still a simplification of the actual marking process. We have not yet implemented this approach. The essence of all of this is that including the marking process into an SMR model provides a spatial context for marked individuals, so that the resighting part of the model is no longer sensitive to the choice of the state-space. The next steps will be to develop a fully spatial model, where both the marking and the resighting process are spatially explicit, and to extend these models to the situation we can no longer use the information from the marking process directly to provide spatial context for marked individuals, because too much time passed between marking individuals and resighting them, so that the assumption that activity centers remain stationary is no longer reasonable.

## 19.8 SUMMARY AND OUTLOOK

In this chapter we combined SCR models and the spatial model for unmarked populations to derive a spatial mark-resight model, which accommodates that part of the population is individually identifiable, usually through artificial tags. Under the assumption that marked individuals are a random sample, both demographically and spatially, from the state-space, the basic model with known number of marked individuals and 100% individual identification of marked is easily modified for situations where the number of marked individuals is unknown, or where marked animals can sometimes not be identified to individual level. As expected, having marked individuals in the study population improved

accuracy and precision of parameter estimates when compared to fully unmarked populations, but we also saw that the spatial counts of unmarked individuals still contribute information to parameter estimates. Finally, we present an approach of how to incorporate telemetry location data into the spatial mark-resight model to inform estimates of  $\sigma$  and activity centers. Just as in SCR, the spatial mark-resight models can account for a variety of factors that may influence individual movement and detection, as well as survey-related parameters, and we saw one example for the Canada geese, where  $\sigma$  was sex-specific.

Many details of SMR models remain to be explored. We mentioned the assignment of marked but unidentified records to actual marked individuals based on their spatial location, which provides some (though imperfect) information of their identity (sec. 19.4). Similarly, records where the marked status cannot be determined could potentially be included in the model as some form of overall correction factor on detection. GPS telemetry devices and their ability to collect location data with much higher frequency offer the opportunity to assign records of collared animals to individuals based on how close to a given camera the collared individuals were, both in space and time. In this scenario, individual identity itself could be expressed probabilistically, leading to an SMR model accounting for potential misidentification. All these possible extensions can tailor SMR models to specific survey techniques.

A fundamental assumption of SMR models is that marked animals are a random sample from  $\mathcal{S}$ . This assumption entails that  $\mathcal{S}$  has to be part of our study design, which in practice is restrictive, as for most studies it might not be feasible to mark animals throughout an area that is larger than the resighting array. If marked animals are not a random sample from  $\mathcal{S}$ , we need to define a spatial context for marked animals within our model, and we showed one tentative approach to doing so by combining a non-spatial model describing the marking process with a spatial resighting model. We believe that combining marking and resighting models in a unified framework is the way forward to generalizing SMR models. Ideally, both components of the model should be spatially explicit to avoid issues associated with non-spatial capture-recapture modeling. Spatial mark-resight models are a fairly new development and work on how to relax the spatial component of the random sample assumption is ongoing. Within the limitations currently posed by this assumption, we believe that SMR modeling holds the potential to address a wide range of population estimation problems when dealing with animals that cannot be identified based on natural marks.



16327  
16328

# 20

16329  
16330

## 2012: A SPATIAL CAPTURE-RECAPTURE ODYSSEY

16331 Capture recapture methods have been a cornerstone of ecological modeling and analysis  
16332 for decades. Yet there are essentially no real capture-recapture data sets that come *without*  
16333 auxiliary spatial information about location of capture (but sometimes such information  
16334 is discarded). As such, classical capture-recapture models are usually not the right tool  
16335 for the job of analyzing real data sets, unless you happen to study fish living in a tank.  
16336 Instead, biologists need methods – spatial capture-recapture methods – that make use of  
16337 spatial information in their capture-recapture data sets.

16338 Spatial capture-recapture methods resolve the essential problem of capture-recapture,  
16339 that of estimating population size and density, but, by explicitly linking space occupied  
16340 by individuals with the spatial location or region of sampling, they do so in a manner  
16341 that provides a holistic framework for answering ecological questions related to the spatial  
16342 structure of populations – movement, space usage, spatial variation in density, landscape  
16343 connectivity, and other things. Thus, spatial capture-recapture methods enable ecologists  
16344 to integrate spatial ecology with their capture-recapture models based on individual level  
16345 encounter data.



**Figure 20.1.** A weasel taking bait on a hair snare. *A Fuller/NYSDEC hair snare study of fishers in southern NY, Photo credits:*

16346 Spurred on by the advent of new non-invasive technologies like DNA sampling, camera  
16347 trapping, acoustic sampling, and other methods, capture-recapture is more relevant and  
16348 widely used than ever before (see also Sec. 20.1). These new survey methods allow  
16349 researchers to use capture recapture for species that could not be studied efficiently even  
16350 a few years ago, especially those that are difficult to capture or handle including most cat  
16351 (Fig. ??) and bear species, mustelids such as weasels (Fig. ??), mink and fishers (Fig.  
16352 ??) and many other species.

16353 In this book we summarized, synthesized and extended recent developments of spatial  
16354 capture-recapture models. The “big idea”, if you could distill the whole thing into one  
16355 idea, is based on extending closed population models by augmenting them with a point  
16356 process model that describes the distribution of individuals (Efford, 2004) in space. As  
16357 a conceptual matter then, the underlying point locations are regarded as an individual  
16358 covariate in the encounter part of the capture-recapture model. In a sense, that's really  
16359 all there is to it. But the relevance is much bigger and more profound because, once we  
16360 have made space explicit in the model, we can think about building population models  
16361 that embody explicit spatial processes. We talked about some such ideas: landscape  
16362 connectivity, resource selection, modeling spatial variation in density. These are all by  
16363 themselves profound extensions of the basic capture-recapture method and they broaden



**Figure 20.2.** Fisher assaulting tree # 8-12. Credit: NYSDEC (New York State Department of Environmental Conservation), A Fuller/NYSDEC camera trap and hair snare study of fishers in southern NY

16364 and expand the relevance and utility of capture-recapture for studying animal populations.  
16365 Although we filled almost 600 book pages with SCR methods, there remains much to be  
16366 done in the continued development of SCR models. In the following sections, we highlight  
16367 the growth in the development and application of SCR models, and we highlight some  
16368 emerging topics that show promise or might be in need of further development. Some of  
16369 these would make ideal graduate projects or similar.

## 20.1 THE GROWTH OF SPATIAL CAPTURE-RECAPTURE

16370 On March 6, 2013, we did this Google Scholar search:

16371 “spatial capture recapture” OR “spatially explicit capture recapture”

16372 The results are described here and, we think, suggest a bright future for the development  
16373 and application of spatial capture-recapture models. Most (but not all) of these papers  
16374 are about the type of SCR models discussed in this book although a handful had to do  
16375 with other types of spatial analysis as related to capture-recapture models. The results  
16376 from this literature search are shown in tabular form in Tab. XXX.



**Figure 20.3.** Canada Lynx, ear-tagged and radio collared, producing high quality data in the name of science. *Credit: A Fuller, Cornell University*

16377 since 2002 274 cites  
16378 since 2003 274 cites 0 articles  
16379 since 2004 271 cites 3 articles published in 2003  
16380 since 2005 269 cites 2 articles published in 2004 Efford 2004  
16381 since 2006 264 cites 5 articles  
16382 since 2007 261 cites 3 articles  
16383 since 2008 253 cites 8 articles Borchers and Efford and Royle  
16384 since 2009 242 cites 11 articles  
16385 since 2010 222 cites 20 articles  
16386 since 2011 176 cites 46 articles  
16387 since 2012 111 cites 65 articles  
16388 since 2013 27 cites 84 articles published in 2012  
16389 27 so far since March 6

16390 We see rapid growth in the number of citations. Somewhere in 2003 the first relevant  
16391 article was published (this was Parminter et al. (2003)) and there was subsequent growth  
16392 fueled by publication of Efford (2004) and the release of the software DENSITY (Efford  
16393 et al., 2004). In 2012 there were 84 articles published and 27 through the first 9 weeks of  
16394 2013.

## 20.2 EMERGING TOPICS

16395 In this book, we provided an overview and synthesis of capture-recapture methods as  
16396 known to us around the end of 2012. There are many emerging topics which we have  
16397 not covered either because of lack of technical knowledge, lack of time for satisfactory  
16398 development, or lack of a good framework for implementation.

16399 A number of topics of importance, or that people are working on, or that might make  
16400 good PhD or Masters projects.

### 16401 20.2.1 Weasels

16402 We just want to say one word to you. Just one word. Weasels. Spatial capture-recapture  
16403 models can be used to study weasel populations.

### 16404 20.2.2 Inhomogeneous Point Processes

16405 In currently developing work, Reich et al. (2012) propose a model that accounts for spatial  
16406 variation in home range density and potential interactions between individuals' home  
16407 ranges. This model lets the activity centers follow an inhomogeneous Strauss process  
16408 (Strauss, 1975), which allows for spatial variation in the home range intensity, and includes  
16409 a parameter that determines the strength of repulsion between home ranges. The idea is  
16410 based on the notion that territorial species would have well defined (and defended) home  
16411 range and thus activity centers may be more regular on the landscape as individuals would  
16412 avoid one another and/or home ranges would be less likely to overlap. The development  
16413 of this work includes a simulation study, where it was found that properly accounting  
16414 for interactions between individuals can provide a substantial improvement in estimating  
16415 population size. And thus far, for simulated data generated with interaction, the usual  
16416 independence model has a significant bias for the population size, and generally has larger  
16417 uncertainty for the population size than the proposed Strauss process model.

16418 While the Strauss model is intuitive and shows great potential, it presents computa-  
16419 tional challenges. The first challenge is that the likelihood includes a high-dimensional  
16420 integral that has no closed form. To address this issue, ? develop an approximation to the  
16421 Strauss likelihood which allows for posterior sampling, extending related work for cate-  
16422 gorical Markov random fields (Green and Richardson, 2002; Smith and Smith, 2006). The  
16423 second challenge is that  $N$  is treated as an unknown parameter to be updated and hence  
16424  $N$  varies and so does the dimension of the likelihood, and thus the posterior. In this case,  
16425 the dimension-changing problem can be overcome by using an auxiliary variable scheme  
16426 in the Markov chain Monte Carlo algorithm. While the results from an initial analysis  
16427 of simulated data verifies that this computational approach leads to reliable inference,  
16428 there are still many areas to be explored in using the Strauss model and other models of  
16429 clustering.

### 16430 20.2.3 Combining data from different surveys

16431 In some instances, researchers apply different survey techniques to the population of inter-  
16432 est, because they yield complementary information. For example, camera trapping is the

16433 prime tool for estimating population size/density and other demographic parameters for  
16434 uniquely marked species; genetic surveys can yield additional information on the genetic  
16435 diversity and health of a population that we cannot study using camera traps. At the  
16436 same time, genetic surveys, when samples are analyzed to the individual level, also yield  
16437 spatial capture recapture data (see Chapt. 15). In this situation, we have two data sets at  
16438 hand that carry information on animal density, and we should be able to get more precise  
16439 estimates of density if we combine these two data sets into a single SCR model.

16440 Gopalaswamy et al. (2012a) developed two approaches to combining data from different  
16441 survey types. In the first case, both surveys are carried out at the same time, so that we  
16442 can assume that they both sample the same – closed – animal population, i.e., there are no  
16443 possible changes in population density between the two surveys. For camera trapping and  
16444 genetic surveys, we cannot match records of individuals between the two data sets. As a  
16445 consequence, in the combined model there are two sets of  $z_i$ , say,  $z_i^C$  for the camera trap  
16446 data and  $z_i^G$  for the genetic survey data. But defining a single state-space around both sets  
16447 of survey locations, we can define a single data augmentation parameter,  $\psi$ , that refers to  
16448 both these sets of  $mathbf{z}$ . **XXX ANDY; WE STILL HAVE TWO ESTIMATES  
16449 OF N AND THUS D; HOW DID YOU CHOOSE? OR DID YOU AVERAGE?  
1650 I DONT FIND ANY INFORMATION ON THAT IN ARJUNS MS XXXXX**

16451 Further, since the scale parameter of the trap encounter model (in Gopalaswamy et al.  
16452 (2012a) the Gaussian model),  $\sigma$ , is related to animal movement, we can expect this pa-  
16453 rameter to be the same for both surveys and share it across both data sets. Finally,  
16454 we need to define separate baseline detection parameters, say  $\lambda_0^C$  and  $\lambda_0^G$ , because the  
16455 processes governing detection by the survey methods should be different. Gopalaswamy  
16456 et al. (2012a) found that this combined model did indeed produce a more precise density  
16457 estimate, compared to single-data set models. We can, of course, imagine other pa-  
16458 rameterizations for this combined model – we could specify both  $\psi$  or  $\sigma$  as survey specific, if we  
16459 have reason to believe they changed between surveys, and we refer you to Gopalaswamy  
16460 et al. (2012a) for more details on these alternative parameterizations.

16461 A second approach of using information from one survey in the analysis of a second  
16462 survey (that maybe does not yield quite as much data as the primary survey) is by ana-  
16463 lyzing your primary data set alone, then taking the posterior distribution of a parameter  
16464 both surveys share and using it as an informative prior distribution in the analysis of  
16465 the second data set. Gopalaswamy et al. (2012a) refer to this as the stepwise approach,  
16466 and they implemented this approach by equating the mean and variance of the posterior  
16467 distribution of  $\psi$  and  $\sigma$  from the photographic survey to the mean and variance of a beta  
16468 and a gamma prior for these parameters, respectively, for the genetic survey. The authors  
16469 found that this approach produced almost identical density estimates compared to the  
16470 combined model approach described above.

16471 In summary, no matter which approach is chosen, combining data across surveys can  
16472 help researchers to obtain more precise population estimates, which is especially valuable  
16473 when dealing with rare and elusive species like big cats that almost always will produce  
16474 sparse individual data sets. Some thought has to go into the assumptions underlying  
16475 combining data – for example, if there is a chance that population size may have changed  
16476 between surveys, it might not be appropriate to have  $\psi$  be a parameter shared across  
16477 surveys; but the assumption that  $\sigma$  remains reasonably constant might still be valid. The  
16478 paper by Gopalaswamy et al. (2012a) considers the situation where we have two SCR data

sets, but we can imagine combining SCR data with other sources of information, such as telemetry data (see Chapt. 19 and Chapt. 13 for examples), and possibly opportunistic observations, although to our knowledge this latter issue has not been tackled in the context of SCR, yet.

#### 20.2.4 Imperfect identification of individuals

Imperfect identification of individuals can happen in a variety of ways. In genetic surveys there is usually some probability of mis-identification of individuals due to genotyping error (e.g. Lukacs and Burnham (2005)). In camera trap survey a different type of imperfect identification can occur when only the only one flank of an animal is recorded in a detection event and that imagine cannot be matched to any of the individuals identified by both flanks. If that case, we can match single-flank pictures with the same side flank pictures, but not with opposite side flank pictures and thus cannot construct definite encounter histories for these single-flank individuals (a right flank and a left flank picture could be the same individual, or could be from two distinct individuals. Finally, in Chapt. 19, in the context of mark-resight models, we discussed the case where individuals can either not definitely be identified as marked or not – a violation of a basic mark-resight assumption, and developed an approach to dealing with the situation where we can always tell if an animal is marked or not, but we are not always able to ascertain its individual identity.

In non-spatial capture recapture some efforts have been made to formally deal with misidentification. Stevick et al. (2001) address this problem by double-sampling to derive an error rate for genetic identification, and then including this error rate as a known constant into a Lincoln-Petersen estimator of abundance. Lukacs and Burnham (2005) develop an approach that includes an additional parameter in the model – the probability of a genotype being identified correctly, which is estimated as part of the model likelihood. Link et al. (2010) developed an approach towards solving the same problem implemented in a Bayesian framework that relaxes some of the assumptions of the initial approach. Yoshizaki et al. (2009) deal with misidentification from camera trap pictures due to evolving marks (i.e., natural marks that change over time, such as scars). This situation is different from the genotyping error one. Here, a change in marks creates a supposedly ‘new’ individual that can be recaptured several times, while the original individual is never captured again (its mark is no longer in the population). In contrast, in genotyping error it is assumed that misidentification creates a ‘new’ individual that is never observed again, because each error leads to a new unique genotype. Yoshizaki et al. (2009) approach this situation similarly, by including a parameter describing the probability of correctly identifying an individual upon recapture (the parameter can also be interpreted as the probability that a mark does not change between capture occasions). Because of the dependencies between true and false detection histories (when a ‘new’ individual is created, the ‘real’ one can no longer be recaptured), the standard multinomial approach to coming up with a model likelihood does not work and implementing the model in a maximum likelihood framework is difficult. The authors instead demonstrate an implementation of the model based on minimizing a function of the squared differences between the observed and expected frequencies of the observed capture histories.

To our knowledge no attempts have been made to deal with misidentification in an SCR framework. While all of the mis-ID cases described above require distinct approaches, we

believe that there is one unifying theme to all of them: the location of the un or potentially mis-identified records and the resulting probabilities of belonging to certain individuals conditional on their activity centers. For example, a right flank and a left flank camera trap picture that are taken at two neighboring camera traps are more likely to belong to the same individual than a right and a left flank picture taken at cameras at opposing ends of the trap array, especially if animal movement is smaller than the extent of the trap array; an unidentified record of a marked individual in a mark-resight survey is more likely to belong to a marked individual whose activity center is close by, than to an individual whose activity center is located far away; and so forth. Formally developing misidentification models should be a focus for future SCR model development.

#### 20.2.5 Gregarious species

One of the key assumptions of the SCR models that we described throughout this book is that the activity centers are independent of one another. There are good biological reasons why this shouldn't be the case, and one of those is when species associate with one another as a pair, or family group. Even species regarded as solitary often join family groups for some portion of their life cycle. We believe that general models for non-independence of activity centers can be developed (see Sec. 20.2.2 above).

There are two consequences of individuals that exist as a pair or group. For one, the detections are not independent. A trap that catches one of the individuals is likely to capture others in the group. The other consequence is that the activity centers  $s_i$  should appear clustered or, in fact, completely redundant in some cases. A possible way to account for this is to change our definition of  $s_i$  from the location of an individual's activity center, to the location of a group's activity center (Russell et al., 2012). Ideally, to accommodate unknown group size, the SCR model would be expanded to include a submodel for group size, so that formal estimation of both group density and group size would be possible.

#### 20.2.6 Design

model-based design is huge. Design of SCR studies is naturally amenable to standard considerations of model-based spatial design (Müller, 2007) which we introduced back in Chapt. 10. Clearly more work can be done on this problem and we think at some point not too far into the future, there will be general-purposes platforms for building capture-recapture sampling plans. A number of specific design problems require some investigation..... designing large landscape scale capture-recapture studies where uniform coverage with traps cannot be achieved, needs to be done. Design in the context of modeling spatial variation in density..... and the effect on design of having telemetry or other auxiliary data.

#### 20.2.7 Acoustic models

At the 2012 ISEC Borchers talked about a number of interesting developments and applications of acoustic models. Multiple observers "triangulating" on a source, and some other things.

---

**16563 20.2.8 Single Catch Traps**

16564 In Chapt. 9 we talked about multinomial models in which encounter of individuals is  
16565 independent for all individuals. This is the multi-catch type of device in which traps never  
16566 fill-up, but an individual can only be caught in one trap. We suggested (following that  
16567 of Efford et al. (2009a)) that the multi-catch, independent multinomial model, could be  
16568 used for “single catch” traps (traps that hold a single individual or “fill up”) and that  
16569 bias associated with mis-specifying the model would be low under certain conditions (i.e.,  
16570 when the proportion of occupied traps is low).

16571 As discussed in Chapt. 9, sec. , we recognize that the *time* of capture of an individual  
16572 in any trapping interval will affect the encounter probability of subsequently captured  
16573 individuals. Thus if the order of capture was known, then this information could be used  
16574 to write the likelihood. However, the order of capture is almost never known, but we think  
16575 that if you could consider all possible orderings, then you would have a close approximation  
16576 to the likelihood. All possible orderings of capture couple we quite computation intense  
16577 and so we are working on a solution that selects an arbitrary ordering the captures as a  
16578 practical approximation to the single-catch process.

16579 A formal model for this situation is needed.....

**16580 20.2.9 Model Fit and Selection**

16581 Evaluation of model adequacy or “fit” is an important part of any applied analysis. In  
16582 Chapt. 8, we offered up a number of ideas based on standard considerations and adapted  
16583 and applied them to SCR models. However, these ideas have not been widely applied, or  
16584 evaluated, and much work needs to be done. In particular, some basic analysis of their  
16585 power under meaningful alternatives would increase their relevance and possibly lead  
16586 to insights for devising better methods. This applies to both Bayesian and likelihood-  
16587 based methods, for which there are even fewer published applications of goodness-of-fit  
16588 assessment.

16589 Similarly, we discussed model selection strategies using more-or-less conventional ideas  
16590 based on AIC/DIC, and model indicator variables using the Kuo and Mallick (1998)  
16591 method. Calibration of these methods under alternatives is needed, along with some  
16592 analysis of sensitivity to density estimates to misspecification of certain model components.

**16593 20.2.10 Dynamics**

16594 Dynamic point processes: Movement, Dispersal, Migration, Transience.

**16595 20.2.11 Misc. Topics**

16596 Models for unmarked or partially marked individuals integrated with RSF data from  
16597 telemetry  
16598 Occupancy and counts data + SCR data (AOAS and Sollmann et al.)  
16599 Spatial genetics – can use SCR to study gene flow, related things....  
16600 SCR on dendritic networks (streams and trails).

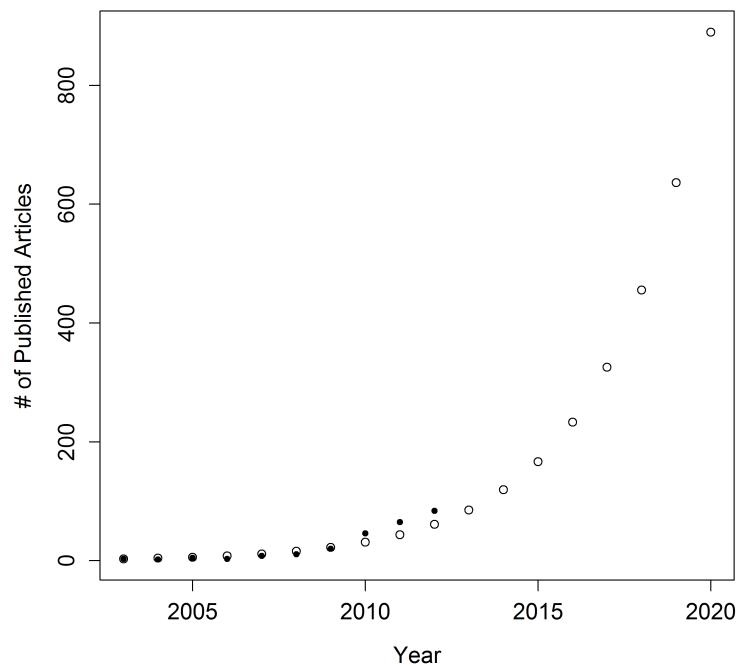
### 20.3 THE FUTURE OF SCR

16601 Everything in ecology is spatial, and now so too are capture-recapture models. We imagine  
16602 that as SCR models continue to be developed and extended, their use will continue to grow  
16603 as well. While there are not a huge number of applications of SCR models right now, we  
16604 showed that their use has expanded rapidly based on simple citation counts.

16605 Considering the number of citations as an ordinary population (without spatial con-  
16606 text!), we used these data to make a projection of the number of published articles that  
16607 involve SCR methods into the future. We fitted an exponential growth curve to these  
16608 data and estimate the annual rate of growth to be 33.4%, accounting for the partial year  
16609 of data observed in 2013. We used the model to project the number of publications that  
16610 involve SCR models into the future (Fig. 20.4). This includes the observed data (solid  
16611 circles) (omitting the partial year 2013) and projections up to 2020. We conclude that the  
16612 future of SCR is bright, with 800 or 900 publications predicted to occur in 2020. While  
16613 this is a long time in the future to be predicting based on an exponential growth model,  
16614 we can do some model checking along the way, as the prediction of 85 SCR publications in  
16615 2013 and 119 in 2014 can be assessed in short order, giving us some short-term feed-back  
16616 on the state of the system. Of course, this model is missing some important things. One  
16617 of those is carrying capacity. The model predicts > 800 publications in 2020, but there  
16618 may not be that much capacity to do capture-recapture studies!

16619 Historically the main use of capture-recapture was to obtain population size estimates,  
16620 but SCR models allow you to address basic and applied questions of population ecology  
16621 from individual encounter history data. Problems having to do with movement, space  
16622 usage, landscape connectivity, and how individuals organize themselves in space.

16623 We envision that SCR models will be useful in helping ecologists “do science” by devel-  
16624 oping explicit models of spatial processes, space usage, connectivity, etc., using ordinary,  
16625 cheap, and easy to obtain individual encounter history data. Much work needs to be  
16626 done to improve computational feasibility, to address many technical or methodological  
16627 holes in the literature (see previous sections), and to make these methods accessible to  
16628 practitioners.



**Figure 20.4.** exponential growth projection of population size of published articles that involve SCR models.



16629

## Part V

16630

---

16631

# Appendices



16632

## APPENDIX I - USEFUL SOFTWARE AND 16633 R PACKAGES

16634

---

16635 Throughout this book we have used a suite of software and R packages, all of which are  
16636 freely available online. To make life a little easier for you, here we provide you with a list  
16637 of all software and R packages, download links and some (hopefully) helpful tips regarding  
16638 their installation.

### 20.4 WINBUGS

16639 Although **WinBUGS** (Gilks et al., 1994) is becoming increasingly obsolete with the faster  
16640 and more flexible **OpenBUGS** and **JAGS**, there are still situations in which the pro-  
16641 gram comes in handy. The .exe file can be downloaded from <http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/contents.shtml>. On 32 bit machines you can just go ahead and  
16642 double-click on the .exe file and follow the installation instructions on the screen. On 64  
16643 bit machines, according to the BUGS project you should download a zip file (from the  
16644 same page) and unzip it into a folder of your choice. There are a couple of additional steps  
16645 to make BUGS run. First, you need to obtain a key (which is free and valid for life) here:  
16646 [http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/WinBUGS14\\_immortality\\_key.txt](http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/WinBUGS14_immortality_key.txt). The  
16647 key comes with instructions on how to activate it. Second, you need to update the ba-  
16648 sic **WinBUGS** version to the most current one (which is from August 2007) following  
16649 the instructions given here: [http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/WinBUGS14\\_cumulative\\_patch\\_No3\\_06\\_08\\_07\\_RELEASE.txt](http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/WinBUGS14_cumulative_patch_No3_06_08_07_RELEASE.txt). **WinBUGS** is ready to use after quit-  
16650 ting and re-opening it. Remember that **WinBUGS** only runs on Windows machines.  
16651 Also, there appears to be a problem installing the program in Vista, although we have no  
16652 personal experience with this.  
16653

#### 20.4.1 WinBUGS through R

16654 While you can run **WinBUGS** as a standalone application, we recommend you access  
16655 it from within **R** using the package **R2WinBUGS** (Sturtz et al., 2005), so you can conve-  
16656 niently process your output, make graphs etc. **R2WinBUGS** also allows you to run mod-  
16657 els in **OpenBUGS** (see below). You can install the package from within **R** directly  
16658 from a cran mirror. In addition to the usual package help document (<http://cran.r-project.org/web/packages/R2WinBUGS/R2WinBUGS.pdf>) you can also download a short  
16659 manual with some examples ([http://votevi.com/bayes\\_beach/R2WinBUGS.pdf](http://votevi.com/bayes_beach/R2WinBUGS.pdf)).

## 20.5 OPENBUGS

16663 **OpenBUGS** is the up-to-date version of **WinBUGS** and can be downloaded here: <http://www.openbugs.info/w/Downloads> (Windows, Mac and Linux versions are available).  
16664 The name **OpenBUGS** refers to the software being open source, so users do not need  
16665 to download a license key, like they have to for **WinBUGS** (although the license key  
16666 for **WinBUGS** is free and valid for life). For Windows, install by double-clicking on the  
16667 .exe file and following the instructions on the installer screen. Compared to **WinBUGS**,  
16668 **OpenBUGS** has more built-in functions. The method of how to determine the right  
16669 updater for each model parameter has changed and the user can manually control the  
16670 MCMC algorithm used to update model parameters. Several other changes have been  
16671 implemented in **OpenBUGS** and a detailed list of differences between the two **BUGS**  
16672 versions, can be found at <http://www.openbugs.info/w/OpenVsWin>. We have encountered  
16673 convergence problems with simple scr models in this program. There is an extensive  
16674 help archive for both **WinBUGS** and **OpenBUGS** and you can subscribe to a mailing  
16675 list, where people pose and answer questions of how to use these programs at <http://www.mrc-bsu.cam.ac.uk/bugs/overview/list.shtml>  
16676  
16677

16678 **20.5.1 OpenBUGS through R**

16679 Like **WinBUGS**, **OpenBUGS** can be used as a standalone application or through **R**.  
16680 There are several packages that allow **R** to interface with **OpenBUGS**, all of which can  
16681 be installed directly from a cran mirror:

16682 **R2WinBUGS**: One of the options in the `bugs()` call is `program`, which lets you specify either  
16683 **WinBUGS** or **OpenBUGS**. This is a convenient option because after having worked  
16684 through some of this book you will likely be familiar with the format of `bugs()` output  
16685 and other functions of the **R2WinBUGS** package.

16686 **R2openBUGS**: **R2openBUGS** (Sturtz et al., 2005) is very similar to, and actually based on,  
16687 **R2WinBUGS** and it is unclear to us what can be gained by using the former over the latter.  
16688 Arguments of the `bugs()` call differ slightly between the two packages and given that  
16689 **R2WinBUGS** allows for the use of both **OpenBUGS** and **WinBUGS** it is probably easiest  
16690 to stick with it.

16691 **BRugs**: **BRugs** (Thomas et al., 2006) can be installed from within **R** directly from a  
16692 cran mirror. In addition to the help document at [http://www.biostat.umn.edu/~brad/software/BRugs/BRugs\\_9\\_21\\_07.pdf](http://www.biostat.umn.edu/~brad/software/BRugs/BRugs_9_21_07.pdf) there is a **WinBUGS** style manual you can access  
16693 at <http://www.rni.helsinki.fi/openbugs/OpenBUGS/Docu/BRugs%20Manual.html>.  
16694 **BRugs** has the convenient feature that all pieces of a **BUGS** analysis can be run from within  
16695 **R**, including checking the model syntax, something that requires opening the **BUGS** GUI  
16696 with other packages.

## 20.6 JAGS

16698 **JAGS** (Just Another Gibbs Sampler) (Plummer, 2003) runs scr models considerably faster  
16699 than **WinBUGS**, does not have the convergence problem with simple scr models we have

16700 encountered in **OpenBUGS** but similar to the latter program, is flexible and constantly  
16701 updated. Writing a **JAGS** model is virtually identical to writing a **WinBUGS** model.  
16702 However, some functions may have slightly different names and you can look up available  
16703 functions and their use in the **JAGS** manual. One potential downside is that **JAGS** can  
16704 be very particular when it comes to initial values. These may have to be set as close to  
16705 truth as possible for the model to start. Although **JAGS** lets you run several parallel  
16706 Markov chains, this characteristic interferes with the idea of using overdispersed initial  
16707 values for the different chains. Also, we have found that when running models, sometimes  
16708 **JAGS** crashes for unclear reasons, taking **R** down with it. Oftentimes, in order to make  
16709 it run again you'll have to go through downloading and installing it again (remove the  
16710 non-functioning version first).

16711 **JAGS** has a variety of functions that are not available in **WinBUGS**. For example,  
16712 **JAGS** allows you to supply observed data for some deterministic functions of unobserved  
16713 variables. In **BUGS** we cannot supply data to logical nodes. Another useful feature is  
16714 that the adaptive phase of the model (the burn-in) is run separately from the sampling  
16715 from the stationary Markov chains. This allows you to easily add more iterations to the  
16716 adaptive phase if necessary without the need to start from 0. There are other, more  
16717 subtle differences and there is an entire manual section on differences between **JAGS** and  
16718 **OpenBUGS**.

16719 **JAGS** is available for download at <http://sourceforge.net/projects/mcmc-jags/files/>, together with the R package **rjags** (Plummer, 2011), which allows running **JAGS**  
16720 through **R**, user and installation manuals and examples. At this site **JAGS** is available for  
16721 Windows and Mac; Linux binaries are distributed separately and you can find links to various  
16722 sources here: <http://mcmc-jags.sourceforge.net/>. **JAGS** comes with a 32 bit and  
16723 a 64 bit version and can be installed by double-clicking on the .exe file and following the  
16724 instructions on the installer screen. For questions and problems concerning **JAGS** there is a  
16725 forum online at <http://sourceforge.net/projects/mcmc-jags/forums/forum/610037>.

### 16727 20.6.1 JAGS through R

16728 Unlike the two **BUGS** programs, **JAGS** does not have a GUI interface but a command  
16729 line interface that can be used to run the program as a standalone application. **JAGS**  
16730 will solely perform the MCMC simulation; analyzing and summarizing the output has to  
16731 be done outside of **JAGS**. To run **JAGS** through **R** you have two options.

16732 **rjags**: As mentioned above, **rjags** (Plummer, 2011) can be found together with **JAGS**  
16733 and was developed/is being maintained by the inventor of **JAGS**, which means it is  
16734 guaranteed to stay up to date when/as **JAGS** changes. The package can be installed  
16735 from a cran mirror and the help document can be accessed at <http://cran.r-project.org/web/packages/rjags/rjags.pdf>

16737 **R2jags**: Alternatively, the package **R2jags** (Su and Yajima, 2011) provides a means of  
16738 accessing **JAGS** through **R**. We prefer **rjags** for the reason named above, as well as because  
16739 it stores data in a more memory-efficient way and has better **plot()** and **summary()**  
16740 methods.

## 20.7 R

16741 At the time of the preparation of this list, **R** for Windows is at version 2.15.0, which can  
16742 be downloaded at [url`http://cran.r-project.org/bin/windows/base/`](http://cran.r-project.org/bin/windows/base/). This site also contains  
16743 helpful tips on how to install **R** in Windows Vista, how to update **R** packages etc. Instal-  
16744 lation of **R** in Windows is straightforward: download the .exe file, double-click on it and  
16745 follow the instructions of the Windows installer. The later versions of **R** come with versions  
16746 for both 64 bit and 32 bit machines. The **R** site (<http://mirrors.softliste.de/cran/>)  
16747 has an extensive FAQ section Hornik (2011), which includes instructions on how to install  
16748 R on Unix and Mac computers.

16749 **20.7.1 R packages**

16750 This section provides an alphabetical list of useful **R** packages. There is a large number  
16751 of **R** packages and by no means is this list intended to be complete in terms of what is  
16752 useful. Rather, we list packages that we are familiar with and that we employ at one point  
16753 or the other in this book. Unless explicitly stated otherwise, all packages can be installed  
16754 directly from within **R** trough a cran mirror.

16755 **adapt**: `adapt` (Genz et al., 2007) is a package for multidimensional numerical integration.  
16756 The package has been removed from the CRAN repository but can be obtained from  
16757 <http://cran.r-project.org/src/contrib/Archive/adapt/>.

16758 **coda**: `coda` (Plummer et al., 2006) lets you summarize and perform diagnostics on mcmc  
16759 output. For a list and description of functions, see the manual at <http://cran.r-project.org/web/packages/coda/coda.pdf>.

16761 **gdistance**: `gdistance` (van Etten, 2011) is a package for calculating distances and routes  
16762 on geographical grids and can be used to calculate least cost path surfaces. Manual at  
16763 <http://cran.r-project.org/web/packages/gdistance/gdistance.pdf>.

16764 **igraph**: `igraph` (Csardi and Nepusz, 2006) provides routines for graphs and network  
16765 analysis. Manual at <http://cran.r-project.org/web/packages/igraph/igraph.pdf>.

16766 **inline**: `inline` (Sklyar et al., 2010) allows the user to define R functions with in-lined **C**,  
16767 **C++** or **Fortran** code. Manual at <http://cran.r-project.org/web/packages/inline/inline.pdf>.

16769 **maps**: `maps` (Becker et al., 2012) is a library for the display of maps. Manual at <http://cran.r-project.org/web/packages/maps/index.html>.

16771 **maptools**: `maptools` (Bivand and Lewin-Koh, 2013) provides a set of tools for read-  
16772 ing and manipulating spatial data, especially ESRI shapefiles. Manual at <http://cran.r-project.org/web/packages/maptools/maptools.pdf>.

16774 **mvtnorm**: `mvtnorm` (Genz et al., 2013) computes multivariate normal and t probabilities,  
16775 quantiles, random deviates and densities. Manual at <http://cran.r-project.org/web/packages/mvtnorm/mvtnorm.pdf>.

16777 **parallel**: **parallel** contains a suite of functions for parallel computing on multiple computer cores and comes with **R** versions 2.14.0 or higher. More information about the package can be found at <http://stat.ethz.ch/R-manual/R-devel/library/parallel/doc/parallel.pdf>.

16781 **R2cuba**: **R2cuba** (Hahn et al., 2010) is another package for multidimensional integration. Manual at <http://cran.r-project.org/web/packages/R2Cuba/R2Cuba.pdf>.

16783 **raster**: **raster** (Hijmans and van Etten, 2012) provides functions for geographic analysis and modeling with raster data. Manual at <http://cran.r-project.org/web/packages/raster/raster.pdf>.

16786 **Rcpp**: **Rcpp** (Eddelbuettel and François, 2011) provides R functions as well as a C++ library which facilitate the integration of **R** and C++. Manual at <http://cran.r-project.org/web/packages/Rcpp/Rcpp.pdf>.

16789 **RcppArmadillo**: **RcppArmadillo** (François et al., 2011) is a templated C++ linear algebra library, integrating the **Armadillo** library and **R**. Manual at <http://cran.r-project.org/web/packages/RcppArmadillo/RcppArmadillo.pdf>.

16792 **reshape**: **reshape** (Wickham and Hadley, 2007) allows you to easily manipulate, summarize and reshape data. Manual at <http://cran.r-project.org/web/packages/reshape/reshape.pdf>.

16795 **rgeos**: **rgeos** (Bivand and Rundel, 2011) provides many useful functions for spatial operations such as intersecting or buffering spatial features. Manual at <http://cran.r-project.org/web/packages/rgeos/rgeos.pdf>.

16798 **SCRbayes**: (Russell et al., 2012). XXXXXXXX Manual at XXXXX.

16799 **secr**: **secr** (Efford et al., 2009a) is an allround package for fitting a wide array of SCR models in a frequentist framework. Manual at <http://cran.r-project.org/web/packages/secr/secr.pdf>.

16802 **shapefiles**: **shapefiles** (Stabler, 2006) allows you to read and write ESRI shapefiles (i.e. shapefiles you would use in **ArcGIS**). Manual at <http://cran.r-project.org/web/packages/shapefiles/shapefiles.pdf>.

16805 **snow**, **snowfall**: **snow** (Tierney et al., 2011) and **snowfall** (Knaus, 2010) provide functionality for parallel computing. The latter is a more user-friendly wrapper around the former. Manuals at <http://cran.r-project.org/web/packages/snowfall/snowfall.pdf> and <http://cran.r-project.org/web/packages/snow/snow.pdf>.

16809 **sp**: **sp** (Pebesma and Bivand, 2011) is a package for plotting, selecting, subsetting etc. spatial data. **sp** and **spatstat** (see below) are complementary in many ways and data formats can be easily converted between the two packages. Manual at <http://cran.r-project.org/web/packages/sp/sp.pdf>.

16813 **SPACECAP:** SPACECAP (Gopalaswamy et al., 2012a) provides a user friendly GUI interface  
16814 to fit SCR models with a Binomial observation model in a Bayesian framework. Manual  
16815 at <http://www.icesi.edu.co/CRAN/web/packages/SPACECAP/SPACECAP.pdf>.

16816 **spatstat:** spatstat (Baddeley and Turner, 2005) is an extensive package for analyzing  
16817 spatial data. We use it, for example, to generate random points within a state space  
16818 that cannot be described as a rectangle but consists of a (or several) arbitrary polygon(s).  
16819 Manual at <http://cran.r-project.org/web/packages/spatstat/spatstat.pdf>.

16820 **unmarked:** unmarked (Fiske and Chandler, 2011) fits hierarchical models of animal abun-  
16821 dance and occurrence to data collected using a range of predominantly direct observa-  
16822 tion based methods. Manual at <http://cran.r-project.org/web/packages/unmarked/unmarked.pdf>.

## BIBLIOGRAPHY

- 16826 Adriaensen, F., Chardon, J. P., De Blust, G., Swinnen, E., Villalba, S., Gulinkx, H.,  
 16827 and Matthysen, E. (2003), "The application of 'least-cost' modelling as a functional  
 16828 landscape model," *Landscape and urban planning*, 64, 233–247.
- 16829 Agresti, A. (2002), *Categorical data analysis*, vol. 359, John Wiley and Sons.
- 16830 Aitkin, M. (1991), "Posterior bayes factors," *Journal of the Royal Statistical Society. Series B (Methodological)*, 53, 111–142.
- 16831 Alho, J. M. (1990), "Logistic regression in capture-recapture models," *Biometrics*, 46,  
 16832 623–635.
- 16833 Alpízar-Jara, R. and Pollock, K. H. (1996), "A combination line transect and capture-  
 16834 recapture sampling model for multiple observers in aerial surveys," *Environmental and Ecological Statistics*, 3, 311–327.
- 16835 Amstrup, S. C., McDonald, T. L., and Manly, B. F. J. (2005), *Handbook of capture-  
 16836 recapture analysis*, Princeton Univ Pr.
- 16837 Anderson, D. R., Burnham, K. P., White, G. C., and Otis, D. L. (1983), "Density Estimation of Small-Mammal Populations Using a Trapping Web and Distance Sampling Methods," *Ecology*, 64, 674–680.
- 16838 Arnason, A., Schwarz, C., and Gerrard, J. (1991), "Estimating closed population size and number of marked animals from sighting data," *Journal of Wildlife Management*, 55, 716–730.
- 16839 Arnason, N. A. (1972), "Parameter estimates from mark-recapture experiments on two populations subject to migration and death," *Researches on Population Ecology*, 13, 97–113.
- 16840 — (1973), "The estimation of population size, migration rates and survival in a stratified population," *Researches on Population Ecology*, 15, 1–8.
- 16841 Baddeley, A. and Turner, R. (2005), "Spatstat: an R package for analyzing spatial point patterns," *Journal of Statistical Software*, 12, 1–42, ISSN 1548-7660.
- 16842 Bales, S., Hellgren, E., Leslie Jr, D., and Hemphill Jr, J. (2005), "Dynamics of a recolonizing population of black bears in the Ouachita Mountains of Oklahoma," *Wildlife Society Bulletin*, 33, 1342–1351.
- 16843 Balme, G. A., Slotow, R., and Hunter, L. T. B. (2010), "Edge effects and the impact of non-protected areas in carnivore conservation: leopards in the Phinda–Mkuze Complex, South Africa," *Animal Conservation*, 13, 315–323.
- 16844 Bartmann, R. M., White, G. C., Carpenter, L. H., and Garrott, R. A. (1987), "Aerial Mark-Recapture Estimates of Confined Mule Deer in Pinyon-Juniper Woodland," *The Journal of Wildlife Management*, 51, 41–46.
- 16845 Bayne, E. and Hobson, K. (2002), "Annual survival of adult American Redstarts and Ovenbirds in the southern boreal forest," *The Wilson Bulletin*, 114, 358–367.

- 16863 Becker, R. A., Wilks, A. R., Brownrigg, R., and Minka, T. P. (2012), *maps: Draw Geographical Maps*, r package version 2.3-0.
- 16864 Beier, P., Majka, D. R., and Spencer, W. D. (2008), "Forks in the road: choices in procedures for designing wildland linkages," *Conservation Biology*, 22, 836–851.
- 16865 Belant, J., Van Stappen, J., and Paetkau, D. (2005), "American black bear population size and genetic diversity at Apostle Islands National Lakeshore," *Ursus*, 16, 85–92.
- 16866 Berger, J., Liseo, B., and Wolpert, R. (1999), "Integrated likelihood methods for eliminating nuisance parameters," *Statistical Science*, 14, 1–28.
- 16867 Besag, J. and Kooperberg, C. (1995), "On conditional and intrinsic autoregressions," *Biometrika*, 82, 733–746.
- 16868 Bibby, C. J. and Buckland, S. T. (1987), "Bias of bird census results due to detectability varying with habitat," *Acta Ecologica*, 8, 103–112.
- 16869 Bibby, C. J., Burgess, N. D., and Hill, D. A. (1992), "Bird census techniques: Academic Press," London, UK, 257.
- 16870 Bivand, R. and Lewin-Koh, N. (2013), *maptools: Tools for reading and handling spatial objects*, r package version 0.8-23.
- 16871 Bivand, R. and Rundel, C. (2011), *rgeos: Interface to Geometry Engine - Open Source (GEOS)*, r package version 0.1-8.
- 16872 Blair, W. (1940), "A study of prairie deer-mouse populations in southern Michigan," *American Midland Naturalist*, 24, 273–305.
- 16873 Bolker, B. (2008), *Ecological models and data in R*, Princeton Univ Pr.
- 16874 Bondrup-Nielsen, S. (1983), "Density estimation as a function of live-trapping grid and home range size," *Canadian Journal of Zoology*, 61, 2361–2365.
- 16875 Borchers, D. L. (2012), "A non-technical overview of spatially explicit capture–recapture models," *Journal of Ornithology*, 152, 1–10.
- 16876 Borchers, D. L., Buckland, S. T., and Zucchini, W. (2002), *Estimating animal abundance: closed populations*, vol. 13, Springer Verlag.
- 16877 Borchers, D. L. and Efford, M. G. (2008), "Spatially explicit maximum likelihood methods for capture–recapture studies," *Biometrics*, 64, 377–385.
- 16878 Borchers, D. L., Zucchini, W., and Fewster, R. M. (1998), "Mark-recapture models for line transect surveys," *Biometrics*, 54, 1207–1220.
- 16879 Boulanger, J. and McLellan, B. (2001), "Closure violation in DNA-based mark-recapture estimation of grizzly bear populations," *Canadian Journal of Zoology*, 79, 642–651.
- 16880 Bowden, D. C. (1993), "A simple technique for estimating population size. Technical Report 93/12." Tech. rep., Department of Statistics, Colorado State University, Fort Collins, Colorado, USA.
- 16881 Box, G. E. P. and Draper, N. R. (1959), "A basis for the selection of a response surface design," *Journal of the American Statistical Association*, 54, 622–654.
- 16882 — (1987), *Empirical model-building with response surfaces*, Wiley, New York.
- 16883 Boyce, M. S. and McDonald, L. L. (1999), "Relating populations to habitats using resource selection functions," *Trends in Ecology & Evolution*, 14, 268–272.
- 16884 Brooks, S. P., Catchpole, E. A., and Morgan, B. J. T. (2000), "Bayesian Animal Survival Estimation," *Statistical Science*, 15, 357–376.
- 16885 Brownie, C., Hines, J. E., Nichols, J. D., and Pollock, K. H. and Hestbeck, J. B. (1993), "Capture-recapture studies for multiple strata including non-Markovian transitions." *Biometrics*, 49, 1173–1187.

- 16909 Buckland, S., Anderson, D., Burnham, K., Laake, J., Borcher, D., and L, T. (2001), *Introduction to distance sampling: estimating abundance of biological populations*, Oxford, UK: Oxford University Press.
- 16910
- 16911
- 16912 Buckland, S. T. (2004), *Advanced distance sampling*, Oxford University Press, USA.
- 16913 Burnham, K., Anderson, D., and Laake, J. (1980), "Estimation of density from line transect sampling of biological populations," *Wildlife monographs*, 3–202.
- 16914
- 16915 Burnham, K., Anderson, D., White, G., Brownie, C., and Pollock, K. (1987), "Design and analysis methods for fish survival experiments based on release-recapture," *American Fisheries Society Monograph 5. American Fisheries Society, Bethesda Maryland. 1987. 737.*
- 16916
- 16917
- 16918
- 16919 Burnham, K. P. (1997), "Distributional results for special cases of the Jolly-Seber model." *Communications in Statistics*, 26, 1395 – 1409.
- 16920
- 16921 Burnham, K. P. and Anderson, D. R. (2002), *Model selection and multimodel inference: a practical information-theoretic approach*, Springer Verlag.
- 16922
- 16923 Burnham, K. P. and Overton, W. S. (1978), "Estimation of the size of a closed population when capture probabilities vary among animals," *Biometrika*, 65, 625.
- 16924
- 16925 Burt, W. (1943), "Territoriality and home range concepts as applied to mammals," *Journal of mammalogy*, 24, 346–352.
- 16926
- 16927 Casella, G. and Berger, R. L. (2002), *Statistical inference*, Duxbury Press.
- 16928 Casella, G. and George, E. I. (1992), "Explaining the Gibbs sampler," *American Statistician*, 46, 167–174.
- 16929
- 16930 Caswell, H. (1989), *Matrix population models*, Sinauer assoc. Sunderland.
- 16931 Caswell, H. and Werner, P. A. (1978), "Transient behavior and life history analysis of teasel (*Dipsacus sylvestris* Huds.)" *Ecology*, 59, 53–66.
- 16932
- 16933 Chandler, R. B. and Royle, J. A. (2013), "Spatially-explicit models for inference about density in unmarked or partially marked populations," *Annals of Applied Statistics*, XXXX–XXXX.
- 16934
- 16935
- 16936 Chandler, R. B., Royle, J. A., and King, D. (2011), "Inference about density and temporary emigration in unmarked populations," *Ecology*, 92, 1429–1435.
- 16937
- 16938 Clavel, J., Robert, A., Devictor, V., and Julliard, R. (2008), "Abundance estimation with a transient model under the robust design," *Journal of Wildlife Management*, 72, 1203–1210.
- 16939
- 16940
- 16941 Clobert, J., Danchin, E., Dhondt, A., and Nichols, J. (2001), *Dispersal*, Oxford.
- 16942 Cochran, W. (2007), *Sampling techniques*, United States of America: John Wiley & Sons.
- 16943 Compton, B. W., McGarigal, K., Cushman, S. A., and Gamble, L. R. (2007), "A resistant-kernel model of connectivity for amphibians that breed in vernal pools," *Conservation Biology*, 21, 788–799.
- 16944
- 16945
- 16946 Conde, D., Colchero, F., Zarza, H., Christensen, N., Sexton, J., Manterola, C., Chávez, C., Rivera, A., Azuara, D., and Ceballos, G. (2010), "Sex matters: Modeling male and female habitat differences for jaguar conservation," *Biological Conservation*, 143, 1980–1988.
- 16947
- 16948
- 16949
- 16950 Conn, P. B. and Cooch, E. G. (2009), "Multistate capture - recapture analysis under imperfect state observation: an application to disease models," *Journal of Applied Ecology*, 46, 486–492.
- 16951
- 16952
- 16953 Conroy, M. J. and Carroll, J. P. (2009), *Quantitative Conservation of Vertebrates*, Wiley-Blackwell.
- 16954

- 16955 Conroy, M. J., Runge, J. P., Barker, R. J., Schofield, M. R., and Fonnesbeck, C. J. (2008),  
16956 “Efficient estimation of abundance for patchily distributed populations via two-phase,  
16957 adaptive sampling,” *Ecology*, 89, 3362–3370.
- 16958 Converse, S., White, G., and Block, W. (2006a), “Small mammal responses to thinning  
16959 and wildfire in ponderosa pine-dominated forests of the southwestern United States,”  
16960 *Journal of Wildlife Management*, 70, 1711–1722.
- 16961 Converse, S., White, G., Farris, K., and Zack, S. (2006b), “Small mammals and forest fuel  
16962 reduction: national-scale responses to fire and fire surrogates,” *Ecological Applications*,  
16963 16, 1717–1729.
- 16964 Converse, S. J. and Royle, J. A. (2012), “Dealing with incomplete and variable detectabil-  
16965 ity in multi-year, multi-site monitoring of ecological populations,” in *Design and Anal-*  
16966 *ysis of Long-term Ecological Monitoring Studies*, eds. Gitzen, R. R., Millspaugh, J. J.,  
16967 Cooper, A. B., and Licht, D. S., Cambridge University Press, pp. 426–442.
- 16968 Cooch, E. and White, G. (2006), “Program MARK: a gentle introduction,” *available  
16969 online with the MARK programme*, 7.
- 16970 Cormack, R. M. (1964), “Estimates of survival from the sighting of marked animals,”  
16971 *Biometrika*, 51, 429–438.
- 16972 Coull, B. A. and Agresti, A. (1999), “The Use of Mixed Logit Models to Reflect Hetero-  
16973 geneity in Capture-Recapture Studies,” *Biometrics*, 55, 294–301.
- 16974 Cox, D. (1955), “Some statistical methods connected with series of events,” *Journal of  
16975 the Royal Statistical Society. Series B (Methodological)*, 17, 129–164.
- 16976 Cressie, N. (1991), *Statistics for spatial data*, Wiley Series in Probability and Mathematical  
16977 Statistics.
- 16978 Csardi, G. and Nepusz, T. (2006), “The igraph software package for complex network  
16979 research,” *InterJournal, Complex Systems*, 1695.
- 16980 Cushman, S. A., Compton, B. W., and McGarigal, K. (2010), “Habitat fragmentation  
16981 effects depend on complex interactions between population size and dispersal ability:  
16982 Modeling influences of roads, agriculture and residential development across a range  
16983 of life-history characteristics,” in *Spatial complexity, informatics, and wildlife conser-  
16984 vation*, eds. Cushman, S. A. and Huettmann, F., New York: Springer, chap. 20, pp.  
16985 369–385.
- 16986 Cushman, S. A., McKelvey, K. S., Hayden, J., and Schwartz, M. K. (2006), “Gene flow in  
16987 complex landscapes: testing multiple hypotheses with causal modeling,” *The American  
16988 Naturalist*, 168, 486–499.
- 16989 Dawson, D. K. and Efford, M. G. (2009), “Bird population density estimated from acoustic  
16990 signals,” *Journal of Applied Ecology*, 46, 1201–1209.
- 16991 DeGraaf, R. M. and Yamasaki, M. (2001), *New England wildlife: habitat, natural history,  
16992 and distribution*, University Press of New England.
- 16993 DeSante, D. F., Burton, K. M., Saracco, J. F., and Walker, B. L. (1995), “Productiv-  
16994 ity indices and survival rate estimates from MAPS, a continent-wide programme of  
16995 constant-effort mist-netting in North America,” *Journal of Applied Statistics*, 22, 935–  
16996 948.
- 16997 Dice, L. R. (1938), “Some census methods for mammals,” *Journal of Wildlife Management*,  
16998 2, 119–130.
- 16999 — (1941), “Methods for estimating populations of mammals,” *Journal of Wildlife Man-  
17000 agement*, 5, 398–407.

- 17001 Diggle, P. J. (2003), *Statistical analysis of spatial point processes*, London: Arnold, 2nd  
17002 ed.
- 17003 Dijkstra, E. W. (1959), "A note on two problems in connexion with graphs," *Numerische  
17004 mathematik*, 1, 269–271.
- 17005 Dillon, A. and Kelly, M. (2007), "Ocelot *Leopardus pardalis* in Belize: the impact of trap  
17006 spacing and distance moved on density estimates," *Oryx*, 41, 469–477.
- 17007 Dixon, P. (2002), "Bootstrap resampling," *Encyclopedia of environmetrics*.
- 17008 Dorazio, R. M. (2007), "On the choice of statistical models for estimating occurrence and  
17009 extinction from animal surveys," *Ecology*, 88, 2773–2782.
- 17010 Dorazio, R. M. and Royle, J. A. (2003), "Mixture models for estimating the size of a closed  
17011 population when capture rates vary among individuals," *Biometrics*, 59, 351–364.
- 17012 Dormann, C. F., McPherson, J. M., Araújo, M., Bivand, R., Bolliger, J., Carl, G.,  
17013 G Davies, R., Hirzel, A., Jetz, W., Daniel Kissling, W., et al. (2007), "Methods to  
17014 account for spatial autocorrelation in the analysis of species distributional data: a re-  
17015 view," *Ecography*, 30, 609–628.
- 17016 Durban, J. and Elston, D. (2005), "Mark-recapture with occasion and individual effects:  
17017 abundance estimation through Bayesian model selection in a fixed dimensional parame-  
17018 ter space," *Journal of agricultural, biological, and environmental statistics*, 10, 291–305.
- 17019 Eddelbuettel, D. and François, R. (2011), "Rcpp: Seamless R and C++ Integration,"  
17020 *Journal of Statistical Software*, 40, 1–18.
- 17021 Efford, M. (2011a), *secr: Spatially explicit capture-recapture models*, R package version  
17022 2.3.1.
- 17023 Efford, M. G. (2004), "Density estimation in live-trapping studies," *Oikos*, 106, 598–610.  
17024 — (2011b), "Estimation of population density by spatially explicit capture-recapture anal-  
17025 ysis of data from area searches," *Ecology*, 92, 2202–2207.
- 17026 — (2012), "Spatially explicit capture-recapture for bear researchers and managers," Western  
17027 Black Bear Workshop, Coeur d'Alene, ID.
- 17028 Efford, M. G., Borchers, D. L., and Byrom, A. E. (2009a), "Density estimation by spatially  
17029 explicit capture-recapture: likelihood-based methods," *Modeling demographic processes  
17030 in marked populations*, 255–269.
- 17031 Efford, M. G. and Dawson, D. K. (2009), "Effect of distance-related heterogeneity on  
17032 population size estimates from point counts," *The Auk*, 126, 100–111.  
17033 — (2010), "SECR for acoustic data," .
- 17034 Efford, M. G., Dawson, D. K., and Borchers, D. L. (2009b), "Population density estimated  
17035 from locations of individuals on a passive detector array," *Ecology*, 90, 2676–2682.
- 17036 Efford, M. G., Dawson, D. K., and Robbins, C. S. (2004), "DENSITY: software for  
17037 analysing capture-recapture data from passive detector arrays," *Animal Biodiversity  
17038 and Conservation*, 27, 217–228.
- 17039 Efford, M. G. and Fewster, R. M. (2012), "Estimating population size by spatially explicit  
17040 capture-recapture," *Oikos*.
- 17041 Efford, M. G., Warburton, B., Coleman, M. C., and Barker, R. J. (2005), "A field test of  
17042 two methods for density estimation," *Wildlife Society Bulletin*, 33, 731–738.
- 17043 Epps, C. W., Palsbøll, P. J., Wehausen, J. D., Roderick, G. K., Ramey II, R. R., and  
17044 McCullough, D. R. (2005), "Highways block gene flow and cause a rapid decline in  
17045 genetic diversity of desert bighorn sheep," *Ecology Letters*, 8, 1029–1038.
- 17046 Epps, C. W., Wehausen, J. D., Bleich, V. C., Torres, S. G., and Brashares, J. S. (2007),

- 17047     “Optimizing dispersal and corridor models using landscape genetics,” *Journal of Applied  
17048     Ecology*, 44, 714–724.
- 17049     Farnsworth, G. L., Pollock, K. H., Nichols, J. D., Simons, T. R., E., H. J., and R.,  
17050     S. J. (2002), “A removal model for estimating detection probabilities from point-count  
17051     surveys,” *The Auk*, 119, 414–425.
- 17052     Fedorov, V. V. (1972), *Theory of Optimal Experiments*, Academic Press, New York.
- 17053     Fedorov, V. V. and Hackl, P. (1997), *Model-oriented design of experiments*, Springer.
- 17054     Fienberg, S. E., Johnson, M. S., and Junker, B. W. (1999), “Classical multilevel and  
17055     Bayesian approaches to population size estimation using multiple lists,” *Journal of the  
17056     Royal Statistical Society of London A*, 163, 383–405.
- 17057     Fiske, I. J. and Chandler, R. B. (2011), “unmarked: An R package for fitting hierarchical  
17058     models of wildlife occurrence and abundance,” *Journal Of Statistical Software*, 43, 1–23.
- 17059     Forester, J. D., Im, H. K., and Rathouz, P. J. (2009), “Accounting for animal movement  
17060     in estimation of resource selection functions: sampling and data analysis,” *Ecology*, 90,  
17061     3554–3565.
- 17062     Forester, J. D., Ives, A. R., Turner, M. G., Anderson, D. P., Fortin, D., Beyer, H. L., Smith,  
17063     D. W., and Boyce, M. S. (2007), “State-space models link elk movement patterns to  
17064     landscape characteristics in Yellowstone National Park,” *Ecological Monographs*, 77,  
17065     285–299.
- 17066     Fortin, D., Beyer, H. L., Boyce, M. S., Smith, D. W., Duchesne, T., and Mao, J. S. (2005),  
17067     “Wolves influence elk movements: behavior shapes a trophic cascade in Yellowstone  
17068     National Park,” *Ecology*, 86, 1320–1330.
- 17069     Foster, R. J. and Harmsen, B. J. (2012), “A critique of density estimation from camera-  
17070     trap data,” *The Journal of Wildlife Management*, 76, 224–236.
- 17071     Fowler, C. (1981), “Density dependence as related to life history strategy,” *Ecology*, 62,  
17072     602–610.
- 17073     François, R., Eddelbuettel, D., and Bates, D. (2011), *RcppArmadillo: Rcpp integration  
17074     for Armadillo templated linear algebra library*, r package version 0.2.25.
- 17075     Frary, V., Duchamp, J., Maehr, D., and Larkin, J. (2011), “Density and distribution of  
17076     a colonizing front of the American black bear *Ursus americanus*,” *Wildlife Biology*, 17,  
17077     404–416.
- 17078     Fujiwara, M., Anderson, K., Neubert, M., and Caswell, H. (2006), “On the estimation of  
17079     dispersal kernels from individual mark-recapture data.” *Environmental and Ecological  
17080     Statistics*, 13, 183–197.
- 17081     García-Alaníz, N., Naranjo, E. J., and Mallory, F. F. (2010), “Hair-snares: A non-invasive  
17082     method for monitoring felid populations in the Selva Lacandona, Mexico,” *Tropical  
17083     Conservation Science*, 3, 403–411.
- 17084     Gardner, B., Reppucci, J., Lucherini, M., and Royle, J. (2010a), “Spatially explicit in-  
17085     ference for open populations: estimating demographic parameters from camera-trap  
17086     studies,” *Ecology*, 91, 3376–3383.
- 17087     Gardner, B., Royle, J. A., and Wegan, M. T. (2009), “Hierarchical models for estimating  
17088     density from DNA mark-recapture studies,” *Ecology*, 90, 1106–1115.
- 17089     Gardner, B., Royle, J. A., Wegan, M. T., Rainbolt, R. E., and Curtis, P. D. (2010b),  
17090     “Estimating black bear density using DNA data from hair snares,” *The Journal of  
17091     Wildlife Management*, 74, 318–325.
- 17092     Garshelis, D. L. and Hristienko, H. (2006), “State and provincial estimates of American

- 17093 black bear numbers versus assessments of population trend," *Ursus*, 17, 1–7.
- 17094 Gelfand, A. and Smith, A. (1990), "Sampling-based approaches to calculating marginal  
17095 densities," *Journal of the American statistical association*, 85, 398–409.
- 17096 Gelman, A. (2006), "Prior distributions for variance parameters in hierarchical models,"  
17097 *Bayesian analysis*, 1, 515–533.
- 17098 Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004), *Bayesian data analysis*,  
17099 second edition., Boca Raton, Florida, USA: CRC/Chapman & Hall.
- 17100 Gelman, A., Meng, X. L., and Stern, H. (1996), "Posterior predictive assessment of model  
17101 fitness via realized discrepancies," *Statistica Sinica*, 6, 733–759.
- 17102 Geman, S. and Geman, D. (1984), "Stochastic relaxation, Gibbs distributions, and the  
17103 Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine  
17104 Intelligence*, PAMI-6, 721–741.
- 17105 Genz, A., Bretz, F., Miwa, T., Mi, X., Leisch, F., Scheipl, F., and Hothorn, T. (2013),  
17106 *mvtnorm: Multivariate Normal and t Distributions*, r package version 0.9-9994.
- 17107 Genz, A. S., Meyer, M. R., Lumley, T., and Maechler, M. (2007), "The adapt Package. R  
17108 package version 1.0-4." .
- 17109 Gerlach, G. and Musolf, K. (2000), "Fragmentation of landscape as a cause for genetic  
17110 subdivision in bank voles," *Conservation Biology*, 14, 1066–1074.
- 17111 Gilks, W. and Wild, P. (1992), "Adaptive rejection sampling for Gibbs sampling," *Applied  
17112 Statistics*, 41, 337–348.
- 17113 Gilks, W. R., Thomas, A., and Spiegelhalter, D. J. (1994), "A Language and Program for  
17114 Complex Bayesian Modelling," *Journal of the Royal Statistical Society. Series D (The  
17115 Statistician)*, 43, 169–177, ArticleType: primary\_article / Issue Title: Special Issue:  
17116 Conference on Practical Bayesian Statistics, 1992 (3) / Full publication date: 1994 /  
17117 Copyright 1994 Royal Statistical Society.
- 17118 Gilroy, J., Virzi, T., Boulton, R. L., and Lockwood, J. (2012), "A new approach to  
17119 the "apparent survival" problem: estimating true survival rates from mark-recapture  
17120 studies," *Ecology*, 93, 1509–1516.
- 17121 Gimenez, O., Rossi, V., Choquet, R., Dehais, C., Doris, B., Varella, H., Vila, J. P., and  
17122 Pradel, R. (2007), "State-space modelling of data on marked individuals," *Ecological  
17123 Modelling*, 206, 431–438.
- 17124 Gopalaswamy, A. M. (2012), "Capture-recapture models, spatially explicit," in *Encyclopedia of Environmentrics Second Edition*, eds. El-Shaarawi, A. H. and Piegorsch, W.,  
17125 John Wiley and Sons Ltd, Chichester, UK.
- 17126 Gopalaswamy, A. M., Royle, A. J., Hines, J., Singh, P., Jathanna, D., Kumar, N. S., and  
17127 Karanth, K. U. (2012a), "Program SPACECAP: software for estimating animal density  
17128 using spatially explicit capture-recapture models," *Methods in Ecology and Evolution*,  
17129 online early, r package version 1.0.4.
- 17130 Gopalaswamy, A. M., Royle, J. A., Delampady, M., Nichols, J. D., Karanth, K. U., and  
17131 Macdonald, D. W. (2012b), "Density estimation in tiger populations: combining information  
17132 for strong inference," *Ecology*, 93, 1741–1751.
- 17133 Grant, E. H. C., Nichols, J. D., Lowe, W. H., and Fagan, W. F. (2010), "Use of multiple  
17134 dispersal pathways facilitates amphibian persistence in stream networks," *Proceedings  
17135 of the National Academy of Sciences*, 107, 6936–6940.
- 17136 Green, P. and Richardson, S. (2002), "Hidden Markov Models and Disease Mapping,"  
17137 *Journal of the American Statistical Association*, 97, 1055–1070.

- 17139 Greig-Smith, P. (1964), *Quantitative plant ecology*, Butterworths (Washington).
- 17140 Grimm, V., Revilla, E., Berger, U., Jeltsch, F., Mooij, W., Railsback, S., Thulke, H.-  
17141 H., Weiner, J., Wiegand, T., and DeAngelis, D. (2005), "Pattern-oriented modeling of  
17142 agent-based complex systems: Lessons from ecology." *Science*, 310, 987–991.
- 17143 Hahn, T., Bouvier, A., and Kiêu, K. (2010), *R2Cuba: Multidimensional Numerical Inte-*  
17144 *gration*, R package version 1.0-6.
- 17145 Hall, R. J., Henry, P. F. P., and Bunck, C. M. (1999), "Fifty-year trends in a box turtle  
17146 population in Maryland," *Biological Conservation*, 88, 165–172.
- 17147 Hanks, E. M. and Hooten, M. B. (in press), "Circuit Theory and Model-Based Inference  
17148 for Landscape Connectivity," *Journal of the American Statistical Association*.
- 17149 Hanski, I. A. (1999), *Metapopulation Ecology*, Oxford Univiversity Press.
- 17150 Hardin, R. H. and Sloane, N. J. A. (1993), "A New Approach to the Construction of  
17151 Optimal Designs," *Journal of Statistical Planning and Inference*, 37, 339–369.
- 17152 Hastings, W. (1970), "Monte Carlo sampling methods using Markov chains and their  
17153 applications," *Biometrika*, 57, 97–109.
- 17154 Hawkins, C. E. and Racey, P. A. (2005), "Low population density of a tropical forest  
17155 carnivore, Cryptoprocta ferox: implications for protected area management," *Oryx*, 39,  
17156 35–43.
- 17157 Hayes, R. J. and Buckland, S. T. (1983), "Radial distance models for the line transect  
17158 method," *Biometrics*, 39, 29–42.
- 17159 Hayne, D. (1950), "Apparent home range of Microtus in relation to distance between  
17160 traps," *Journal of mammalogy*, 31, 26–39.
- 17161 Hayne, D. W. (1949), "An Examination of the Strip Census Method for Estimating Animal  
17162 Populations," *The Journal of Wildlife Management*, 13, 145–157.
- 17163 Hedley, S. L., Buckland, S. T., and Borchers, D. L. (1999), "Spatial modelling from line  
17164 transect data," *Journal of Cetacean Research and Management*, 1, 255–264.
- 17165 Hendriks, I., Tenan, S., Tavecchia, G., Marbá, N., Jordá, G., Deudero, S., Álvarez, E., and  
17166 Duarte, C. M. (2013), "Boat anchoring impacts coastal populations of the pen shell,  
17167 the largest bivalve in the Mediterranean," *Biological Conservation*, 160, 105–113.
- 17168 Hestbeck, J. B. and Malecki, R. A. (1989), "Mark-resight estimate of Canada Goose  
17169 midwinter numbers," *Journal of Wildlife Management*, 53, 749–752.
- 17170 Hijmans, R. J. and van Etten, J. (2012), *raster: Geographic analysis and modeling with  
17171 raster data*, R package version 1.9-67.
- 17172 Hines, J. E., Kendall, W. L., and Nichols, J. D. (2003), "On the use of the robust design  
17173 with transient capture-recapture models," *The Auk*, 120, 1151–1158.
- 17174 Hobbs, N. T. (2011), "An Ecological Modeler's Primer on JAGS," .
- 17175 Holdenried, R. (1940), "A population study of the long-eared chipmunk (*Eutamias quadri-*  
17176 *maculatus*) in the central Sierra Nevada," *Journal of Mammalogy*, 21, 405–411.
- 17177 Hooten, M., Johnson, D., Hanks, E., and Lowry, J. (2010), "Agent-based inference for  
17178 animal movement and selection," *Journal of agricultural, biological, and environmental  
17179 statistics*, 15, 523–538.
- 17180 Hooten, M. and Wikle, C. (2010), "Statistical agent-based models for discrete spatio-  
17181 temporal systems." *Journal of the American Statistical Association*, 105, 236–248.
- 17182 Hornik, K. (2011), "The R FAQ," ISBN 3-900051-08-9.
- 17183 Huggins, R. M. (1989), "On the statistical analysis of capture experiments," *Biometrika*,  
17184 76, 133.

- 17185 Hurlbert, S. H. (1984), "Pseudoreplication and the design of ecological field experiments,"  
17186     *Ecological monographs*, 54, 187–211.
- 17187 Illian, J., Penttinen, A., Stoyan, H., and Stoyan, D. (2008), *Statistical analysis and mod-*  
17188     *elling of spatial point patterns*, Wiley.
- 17189 Ivan, J. (2012), "Density, demography, and seasonal movements of snowshoe hares in  
17190     central Colorado," Ph.D. thesis, Colorado State University.
- 17191 Ivan, J., White, G., and Shenk, T. (2013a), "Using auxiliary telemetry information to  
17192     estimate animal density from capture-recapture data," *Ecology*.
- 17193 — (2013b), "Using simulation to compare methods for estimating density from capture-  
17194     recapture data," *Ecology*.
- 17195 Jackson, R., Roe, J., Wangchuk, R., and Hunter, D. (2006), "Estimating Snow Leop-  
17196     ard Population Abundance Using Photography and Capture-Recapture Techniques,"  
17197     *Wildlife Society Bulletin*, 34, 772–781.
- 17198 Jennelle, C. S., Runge, M. C., and MacKenzie, D. I. (2002), "The use of photographic rates  
17199     to estimate densities of tigers and other cryptic mammals: A comment on misleading  
17200     conclusions," *Animal Conservation*, 5, 119–120.
- 17201 Jett, D. A. and Nichols, J. D. (1987), "A Field Comparison of Nested Grid and Trapping  
17202     Web Density Estimators," *Journal of Mammalogy*, 68, 888–892.
- 17203 Jhala, Y. V., Qureshi, Q., Gopal, R., and R., S. P. (2011), "Status of tigers, co-predators  
17204     and prey in India." Tech. rep., National Tiger Conservation Authority, Govt. of India,  
17205     New Delhi and Wildlife Institute of India, Dehradun.
- 17206 Johnson (2010), "A Model-Based Approach for Making Ecological Inference from Distance  
17207     Sampling Data," *Biometrics*, 66, 310–318.
- 17208 Johnson, D. (1980), "The comparison of usage and availability measurements for evaluat-  
17209     ing resource preference," *Ecology*, 61, 65–71.
- 17210 Johnson, D. H. (1999), "The insignificance of statistical significance testing," *The journal  
17211     of wildlife management*, 63, 763–772.
- 17212 Johnson, D. S., London, J. M., Lea, M. A., and Durban, J. (2008a), "Continuous-time  
17213     correlated random walk model for animal telemetry data," *Ecology*, 89, 1208–1215.
- 17214 Johnson, D. S., Thomas, D. L., Ver Hoef, J. M., and Christ, A. (2008b), "A general  
17215     framework for the analysis of animal resource selection from telemetry data," *Biometrics*,  
17216     64, 968–976.
- 17217 Jolly, G. M. (1965), "Explicit estimates from capture-recapture data with both death and  
17218     dilution-stochastic model," *Biometrika*, 52, 225–247.
- 17219 Jonsen, I. D., Flemming, J. M., and Myers, R. A. (2005), "Robust state-space modeling  
17220     of animal movement data," *Ecology*, 86, 2874–2880.
- 17221 Kadane, J. B. and Lazar, N. A. (2004), "Methods and criteria for model selection," *Journal  
17222     of the American Statistical Association*, 99, 279–290.
- 17223 Karanth, K. U. (1995), "Estimating tiger *Panthera tigris* populations from camera-trap  
17224     data using capture–recapture models," *Biological Conservation*, 71, 333–338.
- 17225 Karanth, K. U. and Nichols, J. D. (1998), "Estimation of Tiger Densities in India Using  
17226     Photographic Captures and Recaptures," *Ecology*, 79, 2852–2862.
- 17227 — (2000), "Ecological status and conservation of tigers in India," *WCS, US Fish and  
17228     Wildlife Service. Centre for Wildlife Studies. Bangalore, India*, 123.
- 17229 — (2002), *Monitoring tigers and their prey: a manual for researchers, managers and  
17230     conservationists in Tropical Asia*, Bangalore, India: Centre for Wildlife Studies.

- 17231 Kass, R. and Wasserman, L. (1996), "The selection of prior distributions by formal rules,"  
17232 *Journal of the American Statistical Association*, 91, 1343–1370.
- 17233 Kays, R. W., Slauson, K. M., Long, R. A., MacKay, P., Zielinski, W. J., and Ray, J. C.  
17234 (2008), "Remote cameras." *Noninvasive survey methods for carnivores*, 110–140.
- 17235 Kelly, M., Noss, A., Di Bitetti, M., Maffei, L., Arispe, R., Paviolo, A., De Angelo, C., and  
17236 Di Blanco, Y. (2008), "Estimating puma densities from camera trapping across three  
17237 study sites: Bolivia, Argentina, and Belize," *Journal of Mammalogy*, 89, 408–418.
- 17238 Kendall, K. C., Stetz, J. B., Boulanger, J., Macleod, A. C., Paetkau, D., and White, G. C.  
17239 (2009), "Demography and genetic structure of a recovering grizzly bear population,"  
17240 *The Journal of Wildlife Management*, 73, 3–16.
- 17241 Kendall, W. L. (1999), "Robustness of Closed Capture-Recapture Methods to Violations  
17242 of the Closure Assumption," *Ecology*, 80, 2517–2525.
- 17243 Kendall, W. L., Nichols, J. D., and Hines, J. E. (1997), "Estimating Temporary Emigration  
17244 Using Capture-Recapture Data with Pollock's Robust Design," *Ecology*, 78, 563–578.
- 17245 Kéry, M. (2008), "Estimating abundance from bird counts: binomial mixture models  
17246 uncover complex covariate relationships," *The Auk*, 125, 336–345.
- 17247 — (2010), *Introduction to WinBUGS for Ecologists: Bayesian Approach to Regression,  
17248 ANOVA, Mixed Models and Related Analyses*, Academic Press.
- 17249 Kéry, M. (2011), "Towards the modelling of true species distributions," *Journal of Bio-  
17250 geography*, 38, 617–618.
- 17251 Kéry, M., Gardner, B., and Monnerat, C. (2010), "Predicting species distributions from  
17252 checklist data using site-occupancy models," *Journal of Biogeography*, 37, 1851–1862.
- 17253 Kéry, M., Gardner, B., Stoeckle, T., Weber, D., and Royle, J. A. (2011), "Use of Spatial  
17254 Capture-Recapture Modeling and DNA Data to Estimate Densities of Elusive Animals,"  
17255 *Conservation Biology*, 25, 356–364.
- 17256 Kéry, M., Royle, J. A., and Schmid, H. (2005), "Modeling avian abundance from replicated  
17257 counts using binomial mixture models," *Ecological Applications*, 15, 1450–1461.
- 17258 Kéry, M. and Schaub, M. (2012), *Bayesian Population Analysis Using WinBugs*, Academic  
17259 Press.
- 17260 Kiefer, J. (1959), "Optimal experimental designs (with discussion)," *Journal of the Royal  
17261 Statistical Society, Series B*, 272–319.
- 17262 King, R. and Brooks, S. (2001), "On the Bayesian analysis of population size," *Biometrika*,  
17263 88, 317–336.
- 17264 King, R., Brooks, S., and Coulson, T. (2008), "Analyzing Complex Capture–Recapture  
17265 Data in the Presence of Individual and Temporal Covariates and Model Uncertainty,"  
17266 *Biometrics*, 64, 1187–1195.
- 17267 Knape, Jonas & de Valpine, P. (2012), "Are patterns of density dependence in the Global  
17268 Population Dynamics Database driven by uncertainty about population abundance?"  
17269 *Ecology Letters*, 15, 17–23.
- 17270 Knaus, J. (2010), *snowfall: Easier cluster computing (based on snow)*., r package version  
17271 1.84.
- 17272 Koehler, G. M. and Pierce, D. J. (2003), "Black Bear Home-Range Sizes in Washington:  
17273 Climatic, Vegetative, and Social Influences," *Journal of Mammalogy*, 84, 81–91, Article-  
17274 Type: research-article / Full publication date: Feb., 2003 / Copyright 2003 American  
17275 Society of Mammalogists.
- 17276 Kohn, M., York, E., Kamradt, D., Haught, G., Sauvajot, R., and Wayne, R. (1999),

- 17277      "Estimating population size by genotyping faeces," *Proceedings of the Royal Society of*  
17278      *London. Series B: Biological Sciences*, 266, 657–663.
- 17279      Krebs, C. J. (1999), *Ecological methodology*, Menlo Park, CA: Benjamin/Cummings.
- 17280      Kucera and Barrett (2011), "missing," *missing*, *missing*.
- 17281      Kuo, L. and Mallick, B. (1998), "Variable selection for regression models," *Sankhy: The*  
17282      *Indian Journal of Statistics, Series B*, 60, 65–81.
- 17283      Laird, N. M. and Ware, J. H. (1982), "Random-effects models for longitudinal data,"  
17284      *Biometrics*, 38, 963–974.
- 17285      Langtimm, C. A., Dorazio, R. M., Stith, B. M., and Doyle, T. J. (2011), "New aerial  
17286      survey and hierarchical model to estimate manatee abundance," *The Journal of Wildlife*  
17287      *Management*, 75, 399–412.
- 17288      Le Cam, L. (1990), "Maximum likelihood: an introduction," *International Statistical Re-*  
17289      *view/Revue Internationale de Statistique*, 58, 153–171.
- 17290      Lebreton, J. D., Burnham, K. P., Clobert, J., and Anderson, D. R. (1992), "Modeling  
17291      Survival and Testing Biological Hypotheses Using Marked Animals: A Unified Approach  
17292      with Case Studies," *Ecological Monographs*, 62, 67–118.
- 17293      Leggett, W. C. and Carscadden, J. E. (1978), "Latitudinal variation in reproductive  
17294      characteristics of American shad (*Alosa sapidissima*): evidence for population specific  
17295      life history strategies in fish." *Journal of the Fisheries Research Board of Canada*, 35,  
17296      1469–1477.
- 17297      Lele, S. R. and Keim, J. L. (2006), "Weighted distributions and estimation of resource  
17298      selection probability functions," *Ecology*, 87, 3021–3028.
- 17299      Lele, S. R., Moreno, M., and Bayne, E. (2012), "Dealing with detection error in site  
17300      occupancy surveys: What can we do with a single survey?" *Journal of Plant Ecology*.
- 17301      Lele, S. R., Nadeem, K., and Schmuland, B. (2010), "Estimability and likelihood infer-  
17302      ence for generalized linear mixed models using data cloning," *Journal of the American*  
17303      *Statistical Association*, 105, 1617–1625.
- 17304      Leonard, J. B. K. and McCormick, S. D. (1999), "Effects of migration distance on whole-  
17305      body and tissue-specific energy use in American shad (*Alosa sapidissima*)," *Canadian*  
17306      *Journal of Fisheries and Aquatic Sciences*, 56, 1159–1171.
- 17307      Lewin-Koh, N. J., Bivand, R., contributions by Edzer J. Pebesma, Archer, E., Baddeley,  
17308      A., Bibiko, H.-J., Dray, S., Forrest, D., Friendly, M., Giraudoux, P., Golicher, D., Rubio,  
17309      V. G., Hausmann, P., Hufthammer, K. O., Jagger, T., Luque, S. P., MacQueen, D.,  
17310      Niccolai, A., Short, T., Stabler, B., and Turner, R. (2011), *maptools: Tools for reading*  
17311      *and handling spatial objects*, r package version 0.8-10.
- 17312      Lichstein, J. W., Simons, T. R., Shriner, S. A., and Franzreb, K. E. (2002), "Spatial  
17313      autocorrelation and autoregressive models in ecology," *Ecological Monographs*, 72, 445–  
17314      463.
- 17315      Link, W. (2003), "Nonidentifiability of Population Size from Capture-Recapture Data  
17316      with Heterogeneous Detection Probabilities," *Biometrics*, 59, 1123–1130.
- 17317      Link, W. A. and Barker, R. J. (1994), "Density Estimation Using the Trapping Web  
17318      Design: A Geometric Analysis," *Biometrics*, 50, 733–745.
- 17319      — (2005), "Modeling Association among Demographic Parameters in Analysis of Open  
17320      Population Capture-Recapture Data," *Biometrics*, 61, 46–54.
- 17321      — (2006), "Model weights and the foundations of multimodel inference," *Ecology*, 87,  
17322      2626–2635.

- 17323 — (2010), *Bayesian Inference: With Ecological Applications*, London, UK: Academic  
17324 Press.
- 17325 Link, W. A. and Eaton, M. J. (2011), “On thinning of chains in MCMC,” *Methods in*  
17326 *Ecology and Evolution*, 3, 112–115.
- 17327 Link, W. A., Yoshizaki, J., Bailey, L. L., and Pollock, K. H. (2010), “Uncovering a latent  
17328 multinomial: analysis of markrecapture data with misidentification,” *Biometrics*, 66,  
17329 178–185.
- 17330 Liu and Wu (1999), “Parameter expansion for data augmentation,” *J Am Stat Assoc*, 94,  
17331 1264–1274.
- 17332 Lukacs, P. M. and Burnham, K. P. (2005), “Estimating population size from DNA-based  
17333 closed capture-recapture data incorporating genotyping error,” *Journal of Wildlife*  
17334 *Management*, 69, 396–403.
- 17335 Lunn, D., Spiegelhalter, D., Thomas, A., and Best, N. (2009), “The BUGS project: Evo-  
17336 lution, critique, and future directions,” *Statistics in Medicine*, 28, 3049–3067.
- 17337 Lunn, D. J., Thomas, A., Best, N., and Spiegelhalter, D. (2000), “WinBUGS-a Bayesian  
17338 modelling framework: concepts, structure, and extensibility,” *Statistics and Computing*,  
17339 10, 325–337.
- 17340 Mace, R., Minta, S., Manley, T., and Aune, K. (1994), “Estimating grizzly bear population  
17341 size using camera sightings,” *Wildlife Society Bulletin*, 22, 74–83.
- 17342 MacEachern, S. N. and Berliner, L. M. (1994), “Subsampling the Gibbs sampler,” *Amer-*  
17343 *ican Statistician*, 48, 188–190.
- 17344 MacKay, P., Smith, D. A., Long, R. A., Parker, M., Long, R. A., MacKay, P., Zielinski,  
17345 W. J., and Ray, J. C. (2008), “Scat detection dogs,” *Noninvasive survey methods for*  
17346 *carnivores*, 183–222.
- 17347 MacKenzie, D. I., Nichols, J. D., Lachman, G. B., Droege, S., Royle, J. A., and Langtimm,  
17348 C. A. (2002), “Estimating site occupancy rates when detection probabilities are less than  
17349 one,” *Ecology*, 83, 2248–2255.
- 17350 MacKenzie, D. I., Nichols, J. D., Royle, J. A., Pollock, K. H., Bailey, L. L., and Hines,  
17351 J. E. (2006), *Occupancy estimation and modeling: inferring patterns and dynamics of*  
17352 *species occurrence*, Academic Press.
- 17353 Maffei, L. and Noss, A. J. (2008), “How small is too small? Camera trap survey areas and  
17354 density estimates for ocelots in the Bolivian Chaco,” *Biotropica*, 40, 71–75.
- 17355 Magoun, A. J., Long, C. D., Schwartz, M. K., Pilgrim, K. L., Lowell, R. E., and Valken-  
17356 burg, P. (2011), “Integrating motion-detection cameras and hair snags for wolverine  
17357 identification,” *The Journal of Wildlife Management*, 75, 731–739.
- 17358 Manel, S., Schwartz, M. K., Luikart, G., and Taberlet, P. (2003), “Landscape genetics:  
17359 combining landscape ecology and population genetics,” *Trends in Ecology & Evolution*,  
17360 18, 189–197.
- 17361 Manly, B., McDonald, L., Thomas, D., McDonald, T., and Erickson, W. (2002), *Resource*  
17362 *selection by animals: statistical design and analysis for field studies*, Springer, 2nd ed.
- 17363 Marques, T., Buckland, S., Borchers, D., Tosh, D., and McDonald, R. (2010), “Point  
17364 transect sampling along linear features,” *Biometrics*, 66, 1247–1255.
- 17365 Marques, T., Thomas, L., Ward, J., DiMarzio, N., and Tyack, P. (2009), “Estimating  
17366 cetacean population density using fixed passive acoustic sensors: An example with  
17367 Blainvilles beaked whales,” *The Journal of the Acoustical Society of America*, 125,  
17368 1982.

- 17369 Marques, T. A., Thomas, L., and Royle, J. A. (2011), "A hierarchical model for spatial  
17370 capture-recapture data: Comment," *Ecology*, 92, 526–528.
- 17371 Matthysen, E. (2005), "Density-dependent dispersal in birds and mammals," *Ecography*,  
17372 28, 403–416.
- 17373 McCarthy, M. A. (2007), *Bayesian Methods for Ecology*, Cambridge: Cambridge University  
17374 Press.
- 17375 McClintock, B. and Hoeting, J. (2010), "Bayesian analysis of abundance for binomial  
17376 sighting data with unknown number of marked individuals," *Environmental and Ecological  
17377 Statistics*, 17, 317–332.
- 17378 McClintock, B., King, Thomas, Matthiopoulos, McConnell, and Morales (2012), "A general  
17379 discrete-time modeling framework for animal movement using multi-state random  
17380 walks," *Ecological Monographs*, 82, 335–349.
- 17381 McClintock, B. and White, G. (2012), "From NOREMARK to MARK: software for esti-  
17382 mating demographic parameters using mark-resight methodology," *Journal of Ornithol-  
17383 ogy*, 152, 641–650.
- 17384 McClintock, B., White, G., Antolin, M., and Tripp, D. (2009a), "Estimating abundance  
17385 using mark-resight when sampling is with replacement or the number of marked indi-  
17386 viduals is unknown," *Biometrics*, 65, 237–246.
- 17387 McClintock, B., White, G., and Burnham, K. (2006), "A robust design mark-resight abun-  
17388 dance estimator allowing heterogeneity in resighting probabilities," *Journal of agricul-  
17389 tural, biological, and environmental statistics*, 11, 231–248.
- 17390 McClintock, B. T., Conn, P., Alonso, R., and Crooks, K. R. (In press), "Integrated  
17391 modeling of bilateral photo-identification data in mark-recapture analyses." *Ecology*,  
17392 <http://dx.doi.org/10.1890/12-1613.1>.
- 17393 McClintock, B. T., White, G. C., Burnham, K. P., and Pryde, M. A. (2009b), "A general-  
17394 ized mixed effects model of abundance for mark-resight data when sampling is without  
17395 replacement," in *Modeling demographic processes in marked populations*, ed. Thomson,  
17396 D., New York: Springer, pp. 271–289.
- 17397 McCullagh, P. and Nelder, J. A. (1989), *Generalized linear models*, Chapman & Hall/CRC.
- 17398 McRae, B. H. and Beier, P. (2007), "Circuit theory predicts gene flow in plant and animal  
17399 populations," *Proceedings of the National Academy of Sciences*, 104, 19885–19890.
- 17400 McRae, B. H., Dickson, B. G., Keitt, T. H., and Shah, V. B. (2008), "Using circuit theory  
17401 to model connectivity in ecology, evolution, and conservation," *Ecology*, 89, 2712–2724.
- 17402 Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., Teller, E., et al. (1953), "Equa-  
17403 tion of state calculations by fast computing machines," *The journal of chemical physics*,  
17404 21, 1087–1092.
- 17405 Metropolis, N. and Ulam, S. (1949), "The Monte Carlo method," *Journal of the American  
17406 Statistical Association*, 44, 335–341.
- 17407 Meyer, R. K. and Nachtsheim, C. J. (1995), "The coordinate-exchange algorithm for  
17408 construction exact optimal experimental designs," *Technometrics*, 37(1), 60–69.
- 17409 Millar, R. B. (2009), "Comparison of hierarchical Bayesian models for overdispersed count  
17410 data using DIC and Bayes' Factors," *Biometrics*, 65, 962–969.
- 17411 Mills, L. S., Citta, J. J., Lair, K. P., Schwartz, M. K., and Tallmon, D. A. (2000), "Es-  
17412 timating animal abundance using noninvasive DNA sampling: promise and pitfalls,"  
17413 *Ecological Applications*, 10, 283–294.
- 17414 Minta, S. and Mangel, M. (1989), "A Simple Population Estimate Based on Simulation

- 17415 for Capture-Recapture and Capture-Resight Data," *Ecology*, 70, 1738–1751.  
17416 Mitchell, T. J. (1974), "An algorithm for the construction of' D-optimal" experimental  
17417 designs," *Techometrics*, 203–210.  
17418 Mohr, C. (1947), "Table of equivalent populations of North American small mammals,"  
17419 *American midland naturalist*, 37, 223–249.  
17420 Molinari-Jobin, A., Kéry, M., Marboutin, E., Marucco, F., Zimmermann, F., Molinari, P.,  
17421 Frick, H., Wölfl, S., Bled, F., Breitenmoser-Würsten, C., Fuxjäger, C., Huber, T., I.,  
17422 K., Kos, I., Manfred Wölfl, M., and Breitenmoser, U. (2013), "Mapping range dynamics  
17423 from opportunistic data: spatio-temporal distribution modeling of lynx *Lynx lynx* L.  
17424 in the Alps," *Biological Conservation*, xx, xxxx–xxxx.  
17425 Mollet, P., Kéry, M., Gardner, B., Pasinelli, G., and A, R. J. (In review), "Population size  
17426 estimation for capercaille (*Tetrao urogallus* L.) using DNA-based individual recognition  
17427 and spatial capture-recapture models," missing, missing, missing.  
17428 Morrison, M. L., Strickland, M. D., Block, W. M., Collier, B. A., and Peterson, M. J.  
17429 (2008), *Wildlife Study Design*, Springer.  
17430 Müller, W. G. (2007), *Collecting spatial data: Optimum design of experiments for random*  
17431 *fields*, Springer.  
17432 Murdoch, W. W. (1994), "Population redulation in theory and practice," *Ecology*, 75,  
17433 271–287.  
17434 Neal, A., White, G., Gill, R., Reed, D., and Olterman, J. (1993), "Evaluation of mark-  
17435 resight model assumptions for estimating mountain sheep numbers," *Journal of Wildlife*  
17436 *Management*, 57, 436–450.  
17437 Neal, A. K. (1990), "Evaluation of mark-resight population estimates using simulations  
17438 and field data from mountain sheep." M.S. thesis, Colorado State University, Fort  
17439 Collins, Colorado, USA.  
17440 Neal, R. (2003), "Slice sampling," *Annals of Statistics*, 31, 705–741.  
17441 Nelder, J. A. and Wedderburn, R. W. M. (1972), "Generalized linear models," *Journal of*  
17442 *the Royal Statistical Society. Series A (General)*, 135, 370–384.  
17443 Nichols, J. and Pollock, K. (1990), "Estimation of recruitment from immigration versus  
17444 in situ reproduction using Pollock's robust design," *Ecology*, 21–26.  
17445 Nichols, J., Thomas, L., and Conn, P. (2009), "Inferences about landbird abundance from  
17446 count data: recent advances and future directions," *Modeling demographic processes in*  
17447 *marked populations*, 201–235.  
17448 Nichols, J. D., Hines, J. E., Lebreton, J.-D., and Pradel, R. (2000a), "Estimation of con-  
17449 tributions to population growth: a reverse-time capture-recapture approach," *Ecology*,  
17450 81 (2), 3362–3376.  
17451 Nichols, J. D., Hines, J. E., Pollock, K. H., Hinz, R. L., and Link, W. A. (1994), "Es-  
17452 timating Breeding Proportions and Testing Hypotheses about Costs of Reproduction  
17453 with Capture-Recapture Data," *Ecology*, 75, 2052–2065.  
17454 Nichols, J. D., Hines, J. E., Sauer, J. R., Fallon, F. W., Fallon, J. E., and Heglund,  
17455 P. J. (2000b), "A double-observer approach for estimating detection probability and  
17456 abundance from point counts," *The Auk*, 117, 393–408.  
17457 Nichols, J. D. and Karanth, K. U. (2002), "Statistical concepts: assessing spatial distri-  
17458 butions," in *Monitoring tigers and their prey: a manual for researchers, managers and*  
17459 *conservationists in Tropical Asia*, eds. Karanth, K. U. and Nichols, J. D., Bangalore,  
17460 India: Centre for Wildlife Studies, pp. 29–38.

- 17461 Niemi, A. and Fernández, C. (2010), "Bayesian Spatial Point Process Modeling of Line  
17462 Transect Data," *Journal of Agricultural, Biological, and Environmental Statistics*, 15,  
17463 327–345.
- 17464 Norris, J. L. and Pollock, K. H. (1996), "Nonparametric MLE under two closed capture-  
17465 recapture models with heterogeneity," *Biometrics*, 52, 639–649.
- 17466 Nowak, R. M. (1999), *Walker's Mammals of the World, 6th Edition, Volume I*, Baltimore:  
17467 John's Hopkins University Press, first printing ed.
- 17468 Nychka, D., Yang, Q., and Royle, J. A. (1997), "Constructing spatial designs for moni-  
17469 toring air pollution using regression subset selection," *Statistics for the Environment 3:*  
17470 *Pollution Assessment and Control*, 131–154.
- 17471 O'Brien, T. (2011), "Abundance, density and relative abundance: A conceptual frame-  
17472 work," in *Camera traps in animal ecology: methods and analyses*, eds. O'Connel, A.  
17473 F. J., Nichols, J. D., and Karanth, U., Tokyo, Japan: Springer Verlag, pp. 71–96.
- 17474 O'Connell, A. F., Nichols, J. D., and Karanth, U. K. (2010), *Camera traps in animal*  
17475 *ecology: Methods and analyses*, Springer.
- 17476 O'Hara, R. and Sillanpää, M. (2009), "A review of Bayesian variable selection methods:  
17477 what, how and which," *Bayesian Analysis*, 4, 85–118.
- 17478 Otis, D. L., Burnham, K. P., White, G. C., and Anderson, D. R. (1978), "Statistical  
17479 inference from capture data on closed animal populations," *Wildlife monographs*, 3–  
17480 135.
- 17481 Ovaskainen, O. (2004), "Habitat-specific movement parameters estimated using mark-  
17482 recapture data and a diffusion model," *Ecology*, 85, 242–257.
- 17483 Ovaskainen, O., Rekola, H., Meyke, E., and Arjas, E. (2008), "Bayesian methods for  
17484 analyzing movements in heterogeneous landscapes from mark-recapture data," *Ecology*,  
17485 89, 542–554.
- 17486 Parmenter, R. R. and MacMahon, J. A. (1989), "Animal Density Estimation Using a  
17487 Trapping Web Design: Field Validation Experiments," *Ecology*, 70, 169–179.
- 17488 Parmenter, R. R., Yates, T. L., Anderson, D. R., Burnham, K. P., Dunnum, J. L., Franklin,  
17489 A. B., Friggins, M. T., Lubow, B. C., Miller, M., Olson, G. S., and Others (2003),  
17490 "Small-mammal density estimation: A field comparison of grid-based vs. web-based  
17491 density estimators," *Ecological Monographs*, 73, 1–26.
- 17492 Patterson, T., Thomas, L., Wilcox, C., Ovaskainen, O., and Matthiopoulos, J. (2008),  
17493 "State-space models of individual animal movement," *Trends in ecology & evolution*,  
17494 23, 87–94.
- 17495 Paviolo, A., De Angelo, C., Di Blanco, Y., and Di Bitetti, M. (2008), "Jaguar *Panthera*  
17496 *onca* population decline in the Upper Paraná Atlantic Forest of Argentina and Brazil,"  
17497 *Oryx*, 42, 554.
- 17498 Paviolo, A., Di Blanco, Y., De Angelo, C., and Di Bitetti, M. (2009), "Protection af-  
17499 fects the abundance and activity patterns of pumas in the Atlantic Forest," *Journal of*  
17500 *Mammalogy*, 90, 926–934.
- 17501 Pebesma, E. and Bivand, R. (2011), *Package 'sp'*, r package version 0.9-91.
- 17502 Pledger, S. (2004), "Unified maximum likelihood estimates for closed capture–recapture  
17503 models using mixtures," *Biometrics*, 56, 434–442.
- 17504 Plummer, M. (2003), "JAGS: A program for analysis of Bayesian graphical models us-  
17505 ing Gibbs sampling," in *Proceedings of the 3rd International Workshop on Distributed*  
17506 *Statistical Computing (DSC 2003)*. March, pp. 20–22.

- 17507 — (2009), “rjags: Bayesian graphical models using mcmc. R package version 1.0 3-12,” .  
17508 — (2011), “rjags: Bayesian graphical models using mcmc. R package version 3-5,” .  
17509 Plummer, M., Best, N., Cowles, K., and Vines, K. (2006), “CODA: Convergence Diagnosis  
17510 and Output Analysis for MCMC,” *R News*, 6, 7–11.  
17511 Pollock, K. H. (1982), “A Capture-Recapture Design Robust to Unequal Probability of  
17512 Capture.” *Journal of Wildlife Management*, 46, 752–757.  
17513 Pollock, K. H., Nichols, J. D., Brownie, C., and Hines, J. E. (1990), “Statistical inference  
17514 for capture-recapture experiments,” *Wildlife monographs*, 3–97.  
17515 Porneluzzi, P. A. and Faaborg, J. (1999), “Season long fecundity, survival, and viability  
17516 of Ovenbirds in fragmented and unfrangmented landscapes,” *Conservation Biology*, 13,  
17517 1151–1161.  
17518 Pradel, R. (1996), “Utilization of Capture-Mark-Recapture for the Study of Recruitment  
17519 and Population Growth Rate,” *Biometrics*, 52, 703–709.  
17520 Pradel, R., Hines, J. E., Lebreton, J. D., and Nichols, J. D. (1997), “Capture-Recapture  
17521 Survival Models Taking Account of Transients,” *Biometrics*, 53, 60–72.  
17522 Raabe, J. (2012), “Factors Influencing Distribution and Survival of Migratory Fishes Fol-  
17523 lowing Multiple Low-Head Dam Removals on a North Carolina River. PhD disserta-  
17524 tion.” Ph.D. thesis, North Carolina State University, Raleigh, NC.  
17525 Rathbun, S. (1996), “Estimation of Poisson intensity using partially observed concomitant  
17526 variables,” *Biometrics*, 52, 226–242.  
17527 Rathbun, S. and Cressie, N. (1994), “A space-time survival point process for a longleaf  
17528 pine forest in southern Georgia,” *Journal of the American Statistical Association*, 89,  
17529 1164–1174.  
17530 Rathbun, S., Shiffman, S., and Gwaltney, C. (2007), “Modelling the effects of partially  
17531 observed covariates on Poisson process intensity,” *Biometrika*, 94, 153–165.  
17532 Reich, B. J., Gardner, B., and Wilting, A. (2012), “A spatial capture-recapture model for  
17533 territorial species,” *Biometrics in review*, xx, xx–xxx.  
17534 Robert, C. P. and Casella, G. (2004), *Monte Carlo statistical methods*, New York, USA:  
17535 Springer.  
17536 — (2010), *Introducing Monte Carlo Methods with R*, New York, USA: Springer.  
17537 Roberts, G. O. and Rosenthal, J. S. (1998), “Optimal scaling of discrete approximations  
17538 to Langevin diffusions,” *Journal of the Royal Statistical Society: Series B (Statistical  
17539 Methodology)*, 60, 255–268.  
17540 Rowcliffe, J., Carbone, C., Jansen, P. A., Kays, R., and Kranstauber, B. (2011), “Quanti-  
17541 fying the sensitivity of camera traps: an adapted distance sampling approach,” *Methods  
17542 in Ecology and Evolution*, 2, 464–476.  
17543 Rowcliffe, J. M., Field, J., Turvey, S. T., and Carbone, C. (2008), “Estimating animal  
17544 density using camera traps without the need for individual recognition,” *Journal of  
17545 Applied Ecology*, 45, 1228–1236.  
17546 Royle, J., Kéry, M., and Guélat, J. (2011a), “Spatial capture-recapture models for search-  
17547 encounter data,” *Methods in Ecology and Evolution*, 2, 602–611.  
17548 Royle, J. A. (2004a), “Generalized estimators of avian abundance from count survey data,”  
17549 *Animal Biodiversity and Conservation*, 27, 375–386.  
17550 — (2004b), “N-Mixture Models for Estimating Population Size from Spatially Replicated  
17551 Counts,” *Biometrics*, 60, 108–115.  
17552 — (2006), “Site occupancy models with heterogeneous detection probabilities,” *Biomet-*

- 17553        *rics*, 62, 97–102.  
17554        — (2008), “Modeling individual effects in the Cormack–Jolly–Seber model: a state–space  
17555        formulation,” *Biometrics*, 64, 364–370.  
17556        — (2009a), “Analysis of capture–recapture models with individual covariates using data  
17557        augmentation,” *Biometrics*, 65, 267–274.  
17558        — (2009b), “Analysis of capture–recapture models with individual covariates using data  
17559        augmentation,” *Biometrics*, 65, 267–274.  
17560        Royle, J. A. and Chandler, R. B. (2012), “Integrating Resource Selection Information with  
17561        Spatial Capture–Recapture,” *arXiv preprint arXiv:1207.3288*.  
17562        Royle, J. A., Chandler, R. B., Gazenski, K. D., and Graves, T. A. (2013), “Spatial capture–  
17563        recapture for jointly estimating population density and landscape connectivity,” *Ecology*, to appear.  
17564  
17565        Royle, J. A., Chandler, R. B., Sun, C., and Fuller, A. (2012a), “Integrating Resource  
17566        Selection Information with Spatial Capture–Recapture,” *needs updated MEE*.  
17567        Royle, J. A., Chandler, R. B., Yackulic, C., and Nichols, J. D. (2012b), “Likelihood  
17568        analysis of species occurrence probability from presence-only data for modelling species  
17569        distributions,” *Methods in Ecology and Evolution*, 3, 545–554.  
17570        Royle, J. A. and Converse, S. J. (in review), “Hierarchical spatial capture–recapture mod-  
17571        els: Modeling population density from replicated capture–recapture experiments,” *Ecol-  
17572        ogy*.  
17573        Royle, J. A., Converse, S. J., and Link, W. A. (2012c), “Data Augmentation for Hierar-  
17574        chical Capture–recapture Models,” *arXiv preprint arXiv:1211.5706*.  
17575        Royle, J. A., Dawson, D. K., and Bates, S. (2004), “Modeling abundance effects in distance  
17576        sampling,” *Ecology*, 85, 1591–1597.  
17577        Royle, J. A. and Dorazio, R. M. (2006), “Hierarchical models of animal abundance and  
17578        occurrence,” *Journal of Agricultural, Biological, and Environmental Statistics*, 11, 249–  
17579        263.  
17580        — (2008), *Hierarchical modeling and inference in ecology: the analysis of data from pop-  
17581        ulations, metapopulations and communities*, Academic Press.  
17582        — (2012), “Parameter-expanded data augmentation for Bayesian analysis of capture–  
17583        recapture models,” *Journal of Ornithology*, 152, S521–S537.  
17584        Royle, J. A., Dorazio, R. M., and Link, W. A. (2007), “Analysis of multinomial models with  
17585        unknown index using data augmentation,” *Journal of Computational and Graphical  
17586        Statistics*, 16, 67–85.  
17587        Royle, J. A. and Dubovský, J. A. (2001), “Modeling spatial variation in waterfowl band-  
17588        recovery data,” *The Journal of wildlife management*, 65, 726–737.  
17589        Royle, J. A. and Gardner, B. (2011), “Hierarchical models for estimating density from trap-  
17590        ping arrays,” in *Camera traps in animal ecology: methods and analyses*, eds. O’Connel,  
17591        A. F. J., Nichols, J. D., and Karanth, U., Tokyo, Japan: Springer Verlag, pp. 163–190.  
17592        Royle, J. A., Karanth, K. U., Gopalaswamy, A. M., and Kumar, N. S. (2009a), “Bayesian  
17593        inference in camera trapping studies for a class of spatial capture–recapture models,”  
17594        *Ecology*, 90, 3233–3244.  
17595        Royle, J. A. and Kéry, M. (2007), “A Bayesian state-space formulation of dynamic occu-  
17596        pancy models.” *Ecology*, 88, 1813–1823.  
17597        Royle, J. A. and Link, W. A. (2006), “Generalized site occupancy models allowing for  
17598        false positive and false negative errors,” *Ecology*, 87, 835–841.

- 17599 Royle, J. A., Magoun, A. J., Gardner, B., Valkenburg, P., and Lowell, R. E. (2011b),  
17600 "Density estimation in a wolverine population using spatial capture–recapture models,"  
17601 *The Journal of Wildlife Management*, 75, 604–611.
- 17602 Royle, J. A. and Nichols, J. D. (2003), "Estimating abundance from repeated presence–  
17603 absence data or point counts," *Ecology*, 84, 777–790.
- 17604 Royle, J. A., Nichols, J. D., Karanth, K. U., and Gopalaswamy, A. M. (2009b), "A  
17605 hierarchical model for estimating density in camera-trap studies," *Journal of Applied  
17606 Ecology*, 46, 118–127.
- 17607 Royle, J. A. and Nychka, D. (1998), "An algorithm for the construction of spatial coverage  
17608 designs with implementation in SPLUS," *Computers and Geosciences*, 24, 479–488.
- 17609 Royle, J. A. and Young, K. V. (2008), "A Hierarchical Model For Spatial Capture–  
17610 Recapture Data," *Ecology*, 89, 2281–2289.
- 17611 Russell, R. E., Royle, J. A., Desimone, R., Schwartz, M. K., Edwards, V. L., Pilgrim,  
17612 K. P., and McKelvey, K. S. (2012), "Estimating abundance of mountain lions from  
17613 unstructured spatial samples," *Journal of Wildlife Management*.
- 17614 Rutledge, M. (2012), "XXX to be determined XXX," Ph.D. thesis, North Carolina State  
17615 University.
- 17616 Sacks, J., Welch, W. J., Mitchell, T. P., and Wynn, H. (1989), "Design and analysis of  
17617 computer experiments," *Statistical Science*, 4, 409–435.
- 17618 Sæther, B.-E. and Bakke, Ø. (2000), "Avian life history variation and contribution of  
17619 demographic traits to the population growth rate," *Ecology*, 81, 642653.
- 17620 Saether, B. E., Engen, S., and Matthysen, E. (2002), "Demographic characteristics and  
17621 population dynamical patterns of solitary birds," *Science*, 295, 2070–2073.
- 17622 Saïd, S. and Servany, S. (2005), "The influence of landscape structure on female roe deer  
17623 home-range size," *Landscape ecology*, 20, 1003–1012.
- 17624 Salom-Pérez, R., Carrillo, E., Sáenz, J., and Mora, J. (2007), "Critical condition of the  
17625 jaguar *Panthera onca* population in Corcovado National Park, Costa Rica," *Oryx*, 41,  
17626 51.
- 17627 Sanathanan, L. (1972), "Estimating the size of a multinomial population," *The Annals of  
17628 Mathematical Statistics*, 43, 142–152.
- 17629 Sauer, J. R. and Link, W. A. (2002), "Hierarchical modeling of population stability and  
17630 species group attributes from survey data," *Ecology*, 83, 1743–1751.
- 17631 Schofield, M. and Barker, R. (2008), "A unified capture-recapture framework," *Journal of  
17632 agricultural, biological, and environmental statistics*, 13, 458–477.
- 17633 Schwartz, M. K., Copeland, J. P., Anderson, N. J., Squires, J. R., Inman, R. M., McKelvey,  
17634 K. S., Pilgrim, K. L., Waits, L. P., and Cushman, S. A. (2009), "Wolverine gene flow  
17635 across a narrow climatic niche," *Ecology*, 90, 3222–3232.
- 17636 Schwartz, M. K. and Monfort, S. L. (2008), *Genetic and endocrine tools for carnivore  
17637 surveys*, Island Press Washington, DC, USA.
- 17638 Schwarz, C. J. and Arnason, A. N. (1996), "A General Methodology for the Analysis of  
17639 Capture-Recapture Experiments in Open Populations," *Biometrics*, 52, 860–873.
- 17640 — (2005), "Jolly-Seber Models in MARK," in *Program MARK: A Gentle Introduction*,  
17641 5th ed., eds. Cooch, E. G. and White, G.
- 17642 Schwarz, C. J., Bailey, R. E., Irvine, J. R., and Dalziel, F. C. (1993), "Estimating salmon  
17643 spawning escapement using capture-recapture methods," *Canadian Journal of Fisheries  
17644 and Aquatic Sciences*, 50, 1181–1191.

- 17645 Seber, G. A. F. (1965), "A Note on the Multiple-Recapture Census," *Biometrika*, 52,  
17646 249–259.  
17647 — (1982), *The estimation of animal abundance and related parameters*, Macmillan Pub-  
17648 lishing Co.  
17649 Sepúlveda, M. A., Bartheld, J. L., Monsalve, R., Gómez, V., and Medina-Vogel, G. (2007),  
17650 "Habitat use and spatial behaviour of the endangered Southern river otter (*Lontra*  
17651 *provocax*) in riparian habitats of Chile: conservation implications," *Biological Conserva-*  
17652 *tion*, 140, 329–338.  
17653 Shirk, A. J., Wallin, D. O., Cushman, S. A., Rice, C. G., and Warheit, K. I. (2010),  
17654 "Inferring landscape effects on gene flow: a new model selection framework," *Molecular*  
17655 *ecology*, 19, 3603–3619.  
17656 Sillett, S., Chandler, R. B., Royle, J. A., Kéry, M., and Morrison, S. (2012), "Hierarchical  
17657 distance sampling models to estimate population size and habitat-specific abundance  
17658 of an island endemic," *Ecological Applications*.  
17659 Sillett, T., Rodenhouse, N., and Holmes, R. (2004), "Experimentally reducing neighbor  
17660 density affects reproduction and behavior of a migratory songbird," *Ecology*, 85, 2467–  
17661 2477.  
17662 Skalski, J. R., Millspaugh, J. J., and Spencer, R. D. (2005), "Population estimation and  
17663 biases in paintball, mark-resight surveys of elk," *Journal of Wildlife Management*, 69,  
17664 1043–1052.  
17665 Skaug, H. J. and Schweder, T. (1999), "Hazard models for line transect surveys with  
17666 independent observers," *Biometrics*, 55, 29–36.  
17667 Sklyar, O., Murdoch, D., Smith, M., Eddelbuettel, D., and François, R. (2010), *inline:*  
17668 *Inline C, C++, Fortran function calls from R*, r package version 0.3.8.  
17669 Smith, D. and Smith, M. S. (2006), "Estimation of binary Markov Random Fields using  
17670 Markov chain Monte Carlo," *Journal of Computational and Graphical Statistics*, 15,  
17671 1–21.  
17672 Smith, M. H., Blessing, R., Chelton, J. G., Gentry, J. B., Golley, F. B., and McGinnis,  
17673 J. T. (1971), "Determining density for small mammal populations using a grid and  
17674 assessment lines," *Acta Theriologica*, 16, 105–125.  
17675 Soisalo, M. K. and Cavalcanti, S. M. C. (2006), "Close-up Space in "Radio-telemetry","  
17676 *Biological Conservation*, 129, 487–496.  
17677 Sollmann, R., Furtado, M. M., Gardner, B., Hofer, H., Jacomo, A. T. A., Trres, N. M., and  
17678 Silveira, L. (2011), "Improving density estimates for elusive carnivores: Accounting for  
17679 sex-specific detection and movements using spatial capture-recapture models for jaguars  
17680 in central Brazil," *Biological Conservation*, 144, 1017–1024.  
17681 Sollmann, R., Gardner, B., and Belant, J. L. (2012), "How Does Spatial Study Design  
17682 Influence Density Estimates from Spatial Capture-Recapture Models?" *PloS one*, 7,  
17683 e34575.  
17684 Sollmann, R., Gardner, B., Chandler, R. B., Shindle, D., Onorato, D. P., Royle, J. A., and  
17685 O'Connell, A. F. (in revision), "Using multiple data sources provides density estimates  
17686 for endangered Florida panther," *Journal of Applied Ecology*.  
17687 Sollmann, R., Gardner, B., Parsons, A., Stocking, J., McClintock, B., Simons, T., Pollock,  
17688 K., and O'Connell, A. (2013a), "A spatial mark-resight model augmented with telemetry  
17689 data," *Ecology*.  
17690 Sollmann, R., Mohamed, A., Samejima, H., and Wilting, A. (2013b), "Risky business or

- 17691 simple solution—Relative abundance indices from camera-trapping,” *Biological Conservation*, 159, 405–412.  
17692  
17693 Sólymos, P., Lele, S., and Bayne, E. (2012), “Conditional likelihood approach for analyzing  
17694 single visit abundance survey data in the presence of zero inflation and detection error,”  
17695 *Environmetrics*, 23, 197–205.  
17696 Spiegelhalter, D., Best, N., Carlin, B., and Van Der Linde, A. (2002), “Bayesian mea-  
17697 sures of model complexity and fit,” *Journal of the Royal Statistical Society: Series B  
(Statistical Methodology)*, 64, 583–639.  
17698 Stabler, B. (2006), *shapefiles: Read and Write ESRI Shapefiles*, r package version 0.6.  
17699 Stanley, T. and Burnham, K. (1999), “A closure test for time-specific capture-recapture  
17700 data,” *Environmental and Ecological Statistics*, 6, 197–209.  
17701 Stanley, T. and Richards, J. (2013), “Software review: a program for testing capture-  
17702 recapture data for closure,” *Wildlife Society Bulletin*, 33, 782–785.  
17703 Stevens Jr, D. and Olsen, A. (2004), “Spatially balanced sampling of natural resources,”  
17704 *Journal of the American Statistical Association*, 99, 262–278.  
17705 Stevick, P. T., Palsbøll, P. J., Smith, T. D., Bravington, M. V., and Hammond, P. S.  
17706 (2001), “Errors in identification using natural markings: rates, sources, and effects on  
17707 capturerecapture estimates of abundance,” *Canadian Journal of Fisheries and Aquatic  
17708 Sciences*, 58, 1861–1870.  
17709 Stoyan, D. and Penttinen, A. (2000), “Recent Applications of Point Process Methods in  
17710 Forestry Statistics,” *Statistical Science*, 15, 61–78.  
17711 Strauss, D. (1975), “A model for clustering.” *Biometrika*, 63, 467 – 475.  
17712 Sturtz, S., Ligges, U., and Gelman, A. (2005), “R2WinBUGS: A Package for Running  
17713 WinBUGS from R,” *Journal of Statistical Software*, 12, 1–16.  
17714 Su, Y.-S. and Yajima, M. (2011), *R2jags: A Package for Running jags from R*, r package  
17715 version 0.02-17.  
17716 Sun, C. C. (in prep), “Population estimation, genetic diversity, and structure of black  
17717 bears in southwestern New York,” Master’s thesis, Cornell University.  
17718 Taberlet, P. and Bouvet, J. (1992), “Bear conservation genetics,” *Nature*, 358, 197–197.  
17719 Takemura, A. (1999), “Some superpopulation models for estimating the number of pop-  
17720 ulation uniques,” in *Proceedings of the International Conference on Statistical Data  
17721 Protection SDP*, Citeseer, vol. 98, pp. 45–58.  
17722 Tanner, M. A. and Wong, W. H. (1987), “The calculation of posterior distributions by  
17723 data augmentation,” *J Am Stat Assoc*, 82, 528–540.  
17724 Thomas, A., O’Hara, B., Ligges, U., and Sturtz, S. (2006), “Making BUGS Open,” *R  
17725 News*, 6, 12–17.  
17726 Thompson, C., Royle, J. A., and Garner, J. (2012), “A framework for inference about  
17727 carnivore density from unstructured spatial sampling of scat using detector dogs,” *The  
17728 Journal of Wildlife Management*, 76, 863–871.  
17729 Thompson, S. K. (2002), *Sampling*, New York: Wiley.  
17730 Tierney, L., Rossini, A. J., Li, N., and Sevcikova, H. (2011), *snow: Simple Network of  
17731 Workstations*, r package version 0.3-7.  
17732 Tilman, D. and Kareiva, P. (1997), *Spatial ecology: the role of space in population dynamics  
17733 and interspecific interactions*, vol. 30, Princeton University Press.  
17734 Tischendorf, L. and Fahrig, L. (2000), “On the usage and measurement of landscape  
17735 connectivity,” *Oikos*, 90, 7–19.  
17736

- 17737 Tischendorf, L., Grez, A., Zaviezo, T., and Fahrig, L. (2005), "Mechanisms affecting  
17738 population density in fragmented habitat," *Ecology and Society*, 10, 7.
- 17739 Tobler, M., Carrillo-Percastegui, S., Leite Pitman, R., Mares, R., and Powell, G. (2008),  
17740 "An evaluation of camera traps for inventorying large-and medium-sized terrestrial  
17741 rainforest mammals," *Animal Conservation*, 11, 169–178.
- 17742 Tobler, M. W., Hibert, F., Debeir, L., and Hansen, C. (2012), "Density and sustainable  
17743 harvest estimates for the lowland tapir in the Amazon of French Guiana using a spatial  
17744 capture-recapture model," *none yet*, none, none.
- 17745 Tracy, J. A. (2006), "Individual-based movement modeling as a tool for conserving con-  
17746 nectivity," in *Connectivity conservation*, eds. Crooks, K. and Sanjayan, M., Cambridge  
17747 University Press, pp. 343–368.
- 17748 Trolle, M. and Kéry, M. (2003), "Estimation of ocelot density in the Pantanal using  
17749 capture-recapture analysis of camera-trapping data," *Journal of Mammalogy*, 84, 607–  
17750 614.
- 17751 — (2005), "Camera-trap study of ocelot and other secretive mammals in the northern  
17752 Pantanal," *Mammalia*, 69, 409–416.
- 17753 Tufto, J., Andersen, R., and Linnell, J. (1996), "Habitat use and ecological correlates  
17754 of home range size in a small cervid: the roe deer," *Journal of Animal Ecology*, 65,  
17755 715–724.
- 17756 Tyre, A. J., Tenhumberg, B., Field, S. A., Niejalke, D., Parris, K., and Possingham,  
17757 H. P. (2003), "Improving precision and reducing bias in biological surveys: estimating  
17758 false-negative error rates," *Ecological Applications*, 13, 1790–1801.
- 17759 Valiere, N. and Taberlet, P. (2000), "Urine collected in the field as a source of DNA for  
17760 species and individual identification," *Molecular Ecology*, 9, 2150–2152.
- 17761 van Etten, J. (2011), *Package gdistance*, R package version 1.1-2.
- 17762 Venables, W. and Ripley, B. (2002), *Modern applied statistics with S*, Springer verlag.
- 17763 Venables, W., Smith, D., and Team, R. D. C. (2012), "An introduction to R," .
- 17764 Ver Hoef, J. and Boveng, P. (2007), "Quasi-poisson vs. negative binomial regression: How  
17765 should we model overdispersed count data?" *Ecology*, 88, 2766–2772.
- 17766 Ver Hoef, J. M. (2012), "Who Invented the Delta Method?" *The American Statistician*,  
17767 66, 124–127.
- 17768 Wallace, R. B., Gomez, H., Ayala, G., and Espinoza, F. (2003), "Camera trapping for  
17769 jaguar (*Panthera onca*) in the Tuichi Valley, Bolivia," *Journal of Neotropical Mammal-  
17770 ogy*, 10, 133–139.
- 17771 Wegan, M., Curtis, P., Rainbolt, R., and Gardner, B. (2012), "Temporal sampling frame  
17772 selection in DNA-based capture mark-recapture investigations." *Ursus*, 23, 42 – 51.
- 17773 Wegan, M. T. (2008), "Aversive conditioning, population estimation, and habitat prefer-  
17774 ence of black bears (*Ursus americanus*) on Fort Drum Military Installation in northern  
17775 New York," Ph.D. thesis, Cornell University, Jan.
- 17776 Wegge, P., Pokheral, C. P., and Jnawali, S. R. (2004), "Effects of trapping effort and trap  
17777 shyness on estimates of tiger abundance from camera trap studies," *Animal Conserva-  
17778 tion*, 7, 251–256.
- 17779 White, G. (1996), "NOREMARK: population estimation from mark-resighting surveys,"  
17780 *Wildlife Society Bulletin*, 24, 50–52.
- 17781 White, G. and Bennetts, R. (1996), "Analysis of frequency count data using the negative  
17782 binomial distribution," *Ecology*, 77, 2549–2557.

- 17783 White, G. and Shenk, M. (2001), "Popluation estimation with radio-marked iniividuals." in  
17784 *Radio tracking adn animal populations*, eds. Millspaugh, J. and Marzluff, J., San Diego,  
17785 USA: Academic Press, pp. 329–350.
- 17786 White, G. C., Anderson, D. R., Burnham, K. P., and Otis, D. (1982), *Capture-recapture*  
17787 *and removal methods for sampling closed populations*, Los Alamos: Los Alamos National  
17788 Laboratory.
- 17789 White, G. C. and Garrot, R. (1990), *Analysis of wildlife radiolocation data*, New York,  
17790 USA: Academic Press.
- 17791 White, G. C. and Shenk, T. M. (2000), *Population estimation with radio-marked animals*,  
17792 San Diego, California: Academic Press.
- 17793 Whitman, J., Ballard, W., and Gardner, C. (1986), "Home range and habitat use by  
17794 wolverines in southcentral Alaska," *The Journal of wildlife management*, 460–463.
- 17795 Wickham and Hadley (2007), "Reshaping data with the reshape package," *Journal of*  
17796 *Statistical Software*, 21.
- 17797 Wikle, C. K. (2010), "Hierarchical Modeling with spatial data," in *Handbook of Spatial*  
17798 *Statistics*, eds. Gelfand, A., Diggle, P., Fuentes, M., and Guttorp, P., Chapman and  
17799 Hall, pp. 89–106.
- 17800 Williams, B. K., Nichols, J. D., and Conroy, M. J. (2002), *Analysis and management of*  
17801 *animal populations: modeling, estimation, and decision making*, Academic Pr.
- 17802 Wilson, K. R. and Anderson, D. R. (1985a), "Evaluation of a Density Estimator Based  
17803 on a Trapping Web and Distance Sampling Theory," *Ecology*, 66, 1185–1194.
- 17804 — (1985b), "Evaluation of Two Density Estimators of Small Mammal Population Size,"  
17805 *Journal of Mammalogy*, 66, 13–21.
- 17806 With, K. and Crist, T. (1995), "Critical thresholds in species' responses to landscape  
17807 structure," *Ecology*, 76, 2446–2459.
- 17808 Woods, J. G., Paetkau, D., Lewis, D., McLellan, B. N., Proctor, M., and Strobeck, C.  
17809 (1999), "Genetic tagging of free-ranging black and brown bears," *Wildlife Society Bul-*  
17810 *letin*, 27, 616–627.
- 17811 Wright, J., Barker, R., Schofield, M., Frantz, A., Byrom, A., and Gleeson, D. (2009), "In-  
17812 corporating Genotype Uncertainty into Mark–Recapture-Type Models For Estimating  
17813 Abundance Using DNA Samples," *Biometrics*, 65, 833–840.
- 17814 Wynn, H. P. (1970), "The sequential generation of D-optimum experimental designs,"  
17815 *Annals of Mathematical Statistics*, 41, 1655–1664.
- 17816 Yang, H. C. and Chao, A. (2005), "Modeling Animals' Behavioral Response by Markov  
17817 Chain Models for Capture–Recapture Experiments," *Biometrics*, 61, 1010–1017.
- 17818 Yoshizaki, J., Pollock, K. H., Brownie, C., and Webster, R. A. (2009), "Modeling misiden-  
17819 tification errors in capture-recapture studies using photographic identification of evolv-  
17820 ing marks," *Ecology*, 90, 3–9.
- 17821 Zeller, K., McGarigal, K., and Whiteley, A. (2012), "Estimating landscape resistance to  
17822 movement: a review," *Landscape Ecology*, 27, 777–797.
- 17823 Zuur, A. F., Ieno, E. N., Walker, N. J., Saveliev, A. A., and Smith, G. M. (2009), *Mixed*  
17824 *effects models and extensions in ecology with R*, Springer Verlag.
- 17825 Zylstra, E., Steidl, R., and Swann, D. (2010), "Evaluating survey methods for monitoring  
17826 a rare vertebrate, the Sonoran desert tortoise," *The Journal of Wildlife Management*,  
17827 74, 1311–1318.