

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>Preface</b>	<b>13</b>
13	0.1 Computing . . . . .	13
14	0.2 R package scrbook . . . . .	13
15	0.3 Organization of This Book . . . . .	14
16	0.3.1 Who should read this book . . . . .	15
17	<b>Acknowledgments</b>	<b>16</b>
18	<b>I Background and Concepts</b>	<b>1</b>
19	<b>1 Introduction</b>	<b>3</b>
20	1.1 The Study of Populations by Capture-Recapture . . . . .	4
21	1.2 Scope of this Book . . . . .	5
22	1.3 Lions and Tigers and Bears, oh my: Genesis of Spatial capture-recapture data . . . . .	6
23	1.3.1 Camera trapping . . . . .	7
24	1.3.2 DNA Sampling . . . . .	7
25	1.3.3 Acoustic surveys . . . . .	8
26	1.3.4 Search-Encounter Methods . . . . .	8
27	1.4 Historical Context: A Brief Synopsis . . . . .	9
28	1.4.1 Buffering . . . . .	9
29	1.4.2 Temporary Emigration . . . . .	10
30	1.5 Capture-Recapture for Modeling Encounter Probability . . . . .	11
31	1.5.1 An Example: Fort Drum Bear Study . . . . .	12
32	1.5.2 Inadequacy of Capture-Recapture . . . . .	15
33	1.6 Extension of Closed Population Models . . . . .	15
34	1.6.1 Efford's Formulation . . . . .	16
35	1.6.2 Abundance as the Aggregation of a Point Process . . . . .	17
36	1.6.3 The Activity Center Concept . . . . .	17
37	1.6.4 The state-space . . . . .	18
38	1.6.5 Abundance and Density . . . . .	18
39	1.7 A Survey of Spatial Capture-Recapture . . . . .	19
40	1.8 Elements of SCR Models . . . . .	19

42	1.8.1	Biological focus . . . . .	19
43	1.8.2	Sampling focus . . . . .	20
44	1.9	Summary and Outlook . . . . .	20
45	<b>2</b>	<b>Hierarchical Modeling and SCR</b>	<b>23</b>
46	2.1	Random Variables and Probability Distributions . . . . .	24
47	2.1.1	Stochasticity in ecology . . . . .	24
48	2.1.2	Properties of Probability Distributions . . . . .	27
49	2.2	Common Probability Distributions . . . . .	29
50	2.2.1	The Binomial Distribution . . . . .	29
51	2.2.2	The Bernoulli Distribution . . . . .	30
52	2.2.3	The Multinomial and Categorical Distributions . . . . .	32
53	2.2.4	The Poisson Distribution . . . . .	33
54	2.2.5	The Uniform Distribution . . . . .	34
55	2.2.6	Other Distributions . . . . .	35
56	2.3	Statistical Inference and Parameter Estimation . . . . .	36
57	2.4	Joint, Marginal, and Conditional Distributions . . . . .	39
58	2.5	Hierarchical Models and Inference . . . . .	42
59	2.5.1	Spatial Capture-Recapture models as hierarchical models . .	44
60	2.6	Characterization of SCR models . . . . .	44
61	2.6.1	Variations on the SCR theme . . . . .	48
62	2.7	Summary and Outlook . . . . .	48
63	<b>3</b>	<b>GLMs and Bayesian Analysis</b>	<b>51</b>
64	3.1	GLMs and GLMMs . . . . .	52
65	3.2	Bayesian Analysis . . . . .	54
66	3.2.1	Bayes' Rule . . . . .	54
67	3.2.2	Principles of Bayesian Inference . . . . .	56
68	3.2.3	Prior distributions . . . . .	57
69	3.2.4	Posterior Inference . . . . .	58
70	3.2.5	Small sample inference . . . . .	59
71	3.3	Characterizing posterior distributions by MCMC simulation . .	60
72	3.3.1	What Goes on Under the MCMC Hood . . . . .	61
73	3.3.2	Rules for constructing full conditional distributions . . . . .	63
74	3.3.3	Metropolis-Hastings algorithm . . . . .	63
75	3.4	Bayesian Analysis using the BUGS language . . . . .	64
76	3.4.1	Linear Regression in WinBUGS . . . . .	65
77	3.5	Practical Bayesian Analysis and MCMC . . . . .	67
78	3.5.1	Choice of prior distributions . . . . .	68
79	3.5.2	Convergence and so-forth . . . . .	68
80	3.5.3	Bayesian confidence intervals . . . . .	73
81	3.5.4	Estimating functions of parameters . . . . .	73
82	3.6	Poisson GLMs . . . . .	74

83	3.6.1	Example: Breeding Bird Survey Data . . . . .	74
84	3.6.2	Doing it in WinBUGS . . . . .	77
85	3.6.3	Constructing your own MCMC algorithm . . . . .	78
86	3.7	Poisson GLM with Random Effects . . . . .	81
87	3.8	Binomial GLMs . . . . .	83
88	3.8.1	Binomial regression . . . . .	84
89	3.8.2	Example: Waterfowl Banding Data . . . . .	84
90	3.9	Model Checking and Selection . . . . .	86
91	3.9.1	Goodness-of-fit . . . . .	87
92	3.9.2	Model Selection . . . . .	88
93	3.10	Summary and Outlook . . . . .	90
94	<b>4</b>	<b>Closed Population Models</b>	<b>91</b>
95	4.1	The Simplest Closed Population Model: Model $M_0$ . . . . .	92
96	4.1.1	The Core Capture-Recapture Assumptions . . . . .	94
97	4.1.2	Conditional likelihood . . . . .	95
98	4.2	Data Augmentation . . . . .	95
99	4.2.1	DA links occupancy models and closed population models . .	96
100	4.2.2	Model $M_0$ in BUGS . . . . .	98
101	4.2.3	Formal development of data augmentation (DA) . . . . .	99
102	4.2.4	Remarks on Data Augmentation . . . . .	101
103	4.2.5	Example: Black Bear Study on Fort Drum . . . . .	102
104	4.3	Temporally varying and behavioral effects . . . . .	106
105	4.4	Models with individual heterogeneity . . . . .	106
106	4.4.1	Analysis of Model $M_h$ . . . . .	108
107	4.4.2	Analysis of the Fort Drum data with model $M_h$ . . . . .	109
108	4.4.3	Comparison with MLE . . . . .	111
109	4.5	Individual Covariate Models: Toward Spatial Capture-Recapture .	112
110	4.5.1	Example: Location of capture as a covariate. . . . .	114
111	4.5.2	Fort Drum Bear Study . . . . .	115
112	4.5.3	Extension of the Model . . . . .	117
113	4.5.4	Invariance of density to $B$ . . . . .	120
114	4.5.5	Toward Fully Spatial Capture-recapture Models . . . . .	120
115	4.6	Distance Sampling: A Primitive SCR Model . . . . .	120
116	4.6.1	Example: Sonoran Desert Tortoise Study . . . . .	123
117	4.7	Summary and Outlook . . . . .	125
118	<b>II</b>	<b>Extension to SCR Models</b>	<b>127</b>
119	<b>5</b>	<b>Fully Spatial Capture-Recapture Models</b>	<b>129</b>
120	5.1	Sampling Design and Data Structure . . . . .	130
121	5.2	The binomial encounter model . . . . .	131

122	5.2.1	Definition of home range center . . . . .	132
123	5.2.2	Distance as a latent variable . . . . .	133
124	5.2.3	The Core SCR Assumptions . . . . .	134
125	5.3	The Implied Model of Space Usage . . . . .	135
126	5.3.1	Bivariate normal case . . . . .	136
127	5.3.2	Empirical analysis . . . . .	137
128	5.3.3	Relevance of Understanding Space Usage . . . . .	139
129	5.4	The Binomial Point Process Model . . . . .	139
130	5.4.1	The state-space of the point process . . . . .	140
131	5.4.2	Connection to Model $M_h$ and Distance Sampling . . . . .	142
132	5.5	Simulating SCR Data . . . . .	144
133	5.5.1	Formatting and manipulating real data sets . . . . .	145
134	5.6	Fitting Model SCR0 in BUGS . . . . .	146
135	5.7	Unknown N . . . . .	148
136	5.7.1	Analysis using data augmentation in WinBUGS . . . . .	149
137	5.7.2	Implied Home Range Area . . . . .	152
138	5.7.3	Use of other <b>BUGS</b> engines: <b>JAGS</b> . . . . .	152
139	5.8	Wolverine Camera Trapping Study . . . . .	153
140	5.8.1	Practical Data Organization . . . . .	153
141	5.8.2	Fitting the model in WinBUGS . . . . .	156
142	5.8.3	Summary of the Wolverine Analysis . . . . .	158
143	5.8.4	Wolverine Space Usage . . . . .	159
144	5.9	Using a Discrete Habitat Mask . . . . .	160
145	5.9.1	Evaluation of Coarseness of Discrete Approximation . . . . .	160
146	5.9.2	Analysis of the wolverine camera trapping data . . . . .	161
147	5.10	Summarizing Density and Activity Center Locations . . . . .	163
148	5.10.1	Realized vs. Expected Density . . . . .	164
149	5.10.2	Constructing Density Maps . . . . .	165
150	5.10.3	Example: Wolverine density map . . . . .	167
151	5.10.4	Predicting where an individual lives . . . . .	168
152	5.11	Effective Sample Area . . . . .	168
153	5.12	Summary and Outlook . . . . .	171
154	<b>6</b>	<b>Likelihood Analysis of Spatial Capture-Recapture Models</b>	<b>175</b>
155	6.1	MLE with known N . . . . .	176
156	6.1.1	Implementation (simulated data) . . . . .	177
157	6.2	MLE when N is Unknown . . . . .	181
158	6.2.1	Integrated Likelihood under data augmentation . . . . .	184
159	6.2.2	Extensions . . . . .	184
160	6.3	Classical model selection and assessment . . . . .	185
161	6.4	Likelihood analysis of the wolverine camera trapping data . . . . .	186
162	6.4.1	Using a habitat mask (Restricted state-space) . . . . .	187
163	6.5	DENSITY and the R package <b>secr</b> . . . . .	190

---

164	6.5.1	Encounter device types and detection models . . . . .	191
165	6.5.2	Analysis using the <b>secr</b> package . . . . .	192
166	6.5.3	Likelihood Analysis in the <b>secr</b> Package . . . . .	195
167	6.5.4	Multi-Session Models in <b>secr</b> . . . . .	197
168	6.5.5	Some additional capabilities of <b>secr</b> . . . . .	197
169	6.6	Summary and Outlook . . . . .	199
170	<b>7</b>	<b>Model Selection and Assessment</b>	<b>201</b>
171	7.1	Strategies for Model Selection . . . . .	202
172	7.1.1	Scope of the model selection problem . . . . .	202
173	7.1.2	Model selection by AIC . . . . .	202
174	7.1.3	AIC analysis of the wolverine data . . . . .	203
175	7.2	Bayesian model selection . . . . .	207
176	7.2.1	Deviance Information Criterion (DIC) . . . . .	207
177	7.2.2	DIC analysis of the wolverine data . . . . .	208
178	7.2.3	Bayesian model averaging with indicator variables . . . . .	210
179	7.2.4	Choosing among detection functions . . . . .	213
180	7.3	Evaluating Goodness-of-Fit . . . . .	215
181	7.4	The Two Components of Model Fit . . . . .	217
182	7.4.1	Testing Uniformity or Spatial Randomness . . . . .	217
183	7.4.2	Assessing Fit of the Observation Model . . . . .	220
184	7.4.3	Does the SCR model fit the wolverine data? . . . . .	221
185	7.5	Summary and Outlook . . . . .	224
186	<b>8</b>	<b>Modeling Encounter Probability</b>	<b>227</b>
187	8.1	Introduction . . . . .	227
188	8.2	Detection Functions . . . . .	228
189	8.3	Bayesian Analysis with <b>bear.JAGS</b> . . . . .	230
190	8.4	Bayesian Analysis of detection functions . . . . .	230
191	8.5	Modeling Covariate Effects . . . . .	232
192	8.5.1	Date and Time . . . . .	233
193	8.5.2	Trap-specific covariates . . . . .	235
194	8.5.3	Behavior or Trap Response by Individual . . . . .	236
195	8.5.4	Individual Covariates . . . . .	237
196	8.6	Individual heterogeneity. . . . .	240
197	8.6.1	Models of Heterogeneity . . . . .	241
198	8.7	Likelihood analysis in <b>secr</b> . . . . .	243
199	8.7.1	Notes for fitting standard models . . . . .	243
200	8.7.2	Sex Effects . . . . .	245
201	8.7.3	Individual heterogeneity . . . . .	245
202	8.7.4	Model selection in <b>secr</b> with AIC . . . . .	245
203	8.8	Summary and Outlook . . . . .	246

204	<b>9 Alternative Models for the Encounter Process</b>	<b>247</b>
205	9.1 Poisson Observation Model . . . . .	248
206	9.1.1 Poisson model of space usage . . . . .	249
207	9.1.2 Poisson relationship to the Bernoulli model . . . . .	250
208	9.1.3 A cautionary note on modeling encounter frequencies . . . . .	252
209	9.1.4 Analysis of the Poisson SCR model in BUGS . . . . .	252
210	9.1.5 Simulating Data and Fitting the Model . . . . .	253
211	9.1.6 Analysis of the Wolverine Study Data . . . . .	255
212	9.1.7 Count detector models in the <b>secr</b> package . . . . .	256
213	9.2 Independent Multinomial Observations . . . . .	256
214	9.2.1 Multinomial Resource Selection Models . . . . .	258
215	9.2.2 Simulating data and analysis using JAGS . . . . .	259
216	9.2.3 Multinomial Relationship to Poisson . . . . .	262
217	9.3 Avian Mist-Netting Example . . . . .	263
218	9.3.1 Multiple Sample Sessions . . . . .	264
219	9.3.2 Analysis in JAGS . . . . .	265
220	9.3.3 Analysis in <b>secr</b> . . . . .	267
221	9.4 Single-catch traps . . . . .	269
222	9.4.1 Inference for single-catch systems . . . . .	270
223	9.4.2 Analysis of Efford's possum trapping data . . . . .	270
224	9.5 Acoustic sampling . . . . .	273
225	9.5.1 Implementation in <b>secr</b> . . . . .	275
226	9.5.2 Implementation in <b>BUGS</b> . . . . .	276
227	9.5.3 Other types of acoustic data . . . . .	276
228	9.6 Summary and Outlook . . . . .	277
229	<b>10 Sampling Design</b>	<b>279</b>
230	10.1 General Considerations . . . . .	280
231	10.1.1 Model-based not design-based . . . . .	280
232	10.1.2 Sampling space or sampling individuals? . . . . .	281
233	10.1.3 Scope of inference vs. state-space . . . . .	281
234	10.2 Study design for (spatial) capture-recapture . . . . .	282
235	10.3 Trap spacing and array size relative to animal movement . . . . .	284
236	10.3.1 Example: Black bears from Pictured Rocks National Lakeshore: . . . . .	285
237	10.3.2 Final musings: SCR models, trap spacing and array size . . . . .	287
238	10.4 Spacing of traps with telemetered individuals . . . . .	287
239	10.5 Sampling over large scales . . . . .	288
240	10.6 Model-based Spatial Design . . . . .	288
241	10.6.1 Formalization of the Design Problem for SCR Studies . . . . .	289
242	10.6.2 An Optimal Design Criterion for SCR . . . . .	290
243	10.6.3 Optimization of the criterion . . . . .	292
244	10.6.4 Illustration . . . . .	293

---

246	10.7 Covariate models . . . . .	293
247	10.8 Summary and Outlook . . . . .	294
248	<b>III Spatial Processes in SCR</b>	<b>295</b>
249	<b>11 Integrating Resource Selection with Spatial Capture-Recapture Models</b>	<b>297</b>
250	11.1 A Simple Model of Space Usage . . . . .	298
251	11.1.1 Poisson use model . . . . .	301
252	11.1.2 Thinning . . . . .	302
253	11.1.3 Capture-recapture Data . . . . .	303
254	11.2 The Joint RSF/SCR Likelihood . . . . .	303
255	11.3 Application: New York Black Bear Study . . . . .	305
256	11.4 Simulation Study . . . . .	307
257	11.5 Summary and Outlook . . . . .	310
258		
259	<b>12 Modeling Landscape Connectivity</b>	<b>315</b>
260	12.1 Shortcomings of Euclidean Distance Models . . . . .	316
261	12.2 Least-Cost Path Distance . . . . .	317
262	12.2.1 Example of Computing Cost-weighted distance . . . . .	318
263	12.3 Simulating SCR Data using Ecological Distance . . . . .	321
264	12.4 Likelihood Analysis of Ecological Distance Models . . . . .	324
265	12.4.1 Example of SCR with Least-Cost Path . . . . .	325
266	12.5 Bayesian Analysis . . . . .	325
267	12.6 Simulation Evaluation of the MLE . . . . .	325
268	12.6.1 Simulation Results . . . . .	326
269	12.7 Distance In an Irregular Patch . . . . .	327
270	12.7.1 Basic Geographic Analysis in R . . . . .	329
271	12.8 Summary and Outlook . . . . .	333
272		
273	<b>13 Modeling Spatial Variation in Density</b>	<b>335</b>
274	13.1 Homogeneous point process revisited . . . . .	336
275	13.2 Inhomogeneous point processes . . . . .	339
276	13.3 Observed Point Processes . . . . .	341
277	13.4 Fitting inhomogeneous point process SCR models . . . . .	344
278	13.4.1 Continuous space . . . . .	344
279	13.4.2 Discrete space . . . . .	347
280	13.5 Ecological Distance and Density Covariates . . . . .	348
281	13.6 The Jaguar Data . . . . .	351
281	13.7 Summary . . . . .	354

---

282	<b>IV Advanced SCR Models</b>	<b>357</b>
283	<b>14 Developing Markov Chain Monte Carlo Samplers</b>	<b>359</b>
284	14.0.1 Why build your own MCMC algorithm? . . . . .	359
285	14.1 MCMC and posterior distributions . . . . .	360
286	14.2 Types of MCMC sampling . . . . .	362
287	14.2.1 Gibbs sampling . . . . .	362
288	14.2.2 Metropolis-Hastings sampling . . . . .	364
289	14.2.3 Metropolis-within-Gibbs . . . . .	369
290	14.2.4 Rejection sampling and slice sampling . . . . .	372
291	14.3 MCMC for closed capture-recapture Model Mh . . . . .	372
292	14.4 MCMC algorithm for model SCR0 . . . . .	375
293	14.4.1 SCR model with binomial encounter process . . . . .	378
294	14.4.2 Looking at model output . . . . .	379
295	14.4.3 Posterior density plots . . . . .	380
296	14.4.4 Serial autocorrelation and effective sample size . . . . .	381
297	14.4.5 Summary results . . . . .	383
298	14.4.6 Other useful commands . . . . .	384
299	14.5 Manipulating the state-space . . . . .	385
300	14.6 Increasing computational speed . . . . .	388
301	14.6.1 Parallel computing . . . . .	388
302	14.6.2 Using C++ . . . . .	391
303	14.7 Summary and Outlook . . . . .	393
304	<b>15 Open Population Models</b>	<b>397</b>
305	15.1 Introduction . . . . .	397
306	15.1.1 Overview of Population Dynamics . . . . .	398
307	15.1.2 Animal movement related to population demography . . . . .	399
308	15.1.3 Basic assumptions of JS and CJS models . . . . .	399
309	15.2 Traditional Jolly-Seber Models . . . . .	400
310	15.2.1 Data Augmentation for the Jolly-Seber Model . . . . .	402
311	15.2.2 Mist-netting example . . . . .	404
312	15.2.3 Shortcomings of the traditional JS models . . . . .	405
313	15.3 Spatial Jolly-Seber Models . . . . .	406
314	15.3.1 Mist-netting example . . . . .	406
315	15.4 Traditional CJS models . . . . .	411
316	15.4.1 Migratory fish example . . . . .	411
317	15.5 Multi-state CJS models . . . . .	413
318	15.5.1 Migratory fish example . . . . .	415
319	15.6 Spatial CJS models . . . . .	417
320	15.6.1 Migratory fish example . . . . .	418
321	15.7 Moving Activity Centers . . . . .	420
322	15.7.1 Migratory Fish Example Notes . . . . .	421

---

323	15.8 Summary and Outlook . . . . .	422
324	<b>16 Stratified Populations: Multi-session and Multi-site Data</b>	<b>423</b>
325	16.1 Data Structure . . . . .	424
326	16.2 Multinomial Abundance Models . . . . .	425
327	16.2.1 Observation Models . . . . .	426
328	16.2.2 Simulating group structured capture-recapture data . . . . .	428
329	16.2.3 Fitting in BUGS . . . . .	428
330	16.2.4 Approach B modeling $\psi$ . . . . .	428
331	16.3 Spatial Capture-Recapture . . . . .	428
332	16.4 Application . . . . .	429
333	16.4.1 Results . . . . .	431
334	16.5 Topics in Multi-Session models . . . . .	434
335	16.5.1 Temporal models . . . . .	434
336	16.5.2 Dependence – is it a problem? . . . . .	434
337	16.6 Multi-session models in <b>secr</b> . . . . .	434
338	16.6.1 Ovenbird data in WinBUGS? . . . . .	436
339	16.6.2 Converse data in <b>secr</b> ? . . . . .	436
340	16.7 Summary and Outlook . . . . .	436
341	<b>17 Models for Search-Encounter Data</b>	<b>439</b>
342	17.1 Search-Encounter sampling designs . . . . .	440
343	17.2 A Model for Search-Encounter Data . . . . .	442
344	17.2.1 Ecological process model . . . . .	444
345	17.2.2 Other stuff . . . . .	445
346	17.3 Examples . . . . .	445
347	17.3.1 Hard plot boundaries . . . . .	446
348	17.3.2 Analysis of other protocols . . . . .	447
349	17.4 Design 3: Ad hoc implementation of Design 1. . . . .	447
350	17.5 Capricaille crap . . . . .	447
351	17.6 Design 4 – no location info . . . . .	447
352	17.7 Summary and Outlook . . . . .	448
353	<b>18 Spatial Capture-Recapture for Unmarked Populations</b>	<b>449</b>
354	18.1 Existing Models for Inference About Density in Unmarked Populations	450
355	18.2 Spatial Correlation as Information . . . . .	451
356	18.3 Data . . . . .	452
357	18.4 Model . . . . .	452
358	18.5 Northern Parula Example . . . . .	454
359	18.6 Improving Precision with Prior Information . . . . .	455
360	18.7 Design issues . . . . .	458
361	18.7.1 How Much Correlation Is Enough? . . . . .	458
362	18.7.2 Linear Designs . . . . .	458

363	18.7.3 Quadrat counts . . . . .	458
364	18.8 Alternative Observation Models . . . . .	458
365	18.8.1 Spatial point process models . . . . .	458
366	18.9 Conclusion . . . . .	459
367	<b>19 Spatial Mark-Resight Models for partially identifiable populations</b>	<b>461</b>
368	19.1 Background . . . . .	462
369	19.1.1 Types of partial ID data . . . . .	462
370	19.1.2 A short history of mark-resight models . . . . .	464
371	19.2 Known number of marked individuals . . . . .	465
372	19.2.1 MCMC for a spatial mark-resight model . . . . .	466
373	19.2.2 Binomial encounter model . . . . .	468
374	19.3 Unknown number of marked individuals . . . . .	469
375	19.4 Individual identification rate < 100 % . . . . .	472
376	19.5 How much information do marked and unmarked individuals contribute? . . . . .	474
377	19.6 Incorporating telemetry data . . . . .	476
378	19.7 Summary and Outlook . . . . .	480
381	<b>20 2012: A Spatial Capture-Recapture Odyssey</b>	<b>483</b>
382	20.1 10 thesis or dissertation topics . . . . .	484
383	20.2 Three dimesional space . . . . .	484
384	20.3 Gregarious species . . . . .	484
385	<b>V Appendices</b>	<b>485</b>
386	20.4 WinBUGS . . . . .	487
387	20.4.1 WinBUGS through R . . . . .	487
388	20.5 OpenBUGS . . . . .	488
389	20.5.1 OpenBUGS through R . . . . .	488
390	20.6 JAGS . . . . .	488
391	20.6.1 JAGS through R . . . . .	489
392	20.7 R . . . . .	489
393	20.7.1 R packages . . . . .	490
394	<b>Bibliography</b>	<b>493</b>

## Preface

395

396

397     Stuff on hierarchical models and how great those are.  
398     Bayesian and likelihood analysis..... We don't care, and neither should you.

### 0.1 COMPUTING

399     Emphasis on doing things in the R programming language - A link to CRAN or  
400     whatever. Reference some other books about using R and so forth. Which packages  
401     do we use? (see Appendix).

402     We do Bayesian analysis almost exclusively in the BUGS language, using Win-  
403     BUGS and JAGS. Mostly we are transitioning to use of JAGS but we still like  
404     WinBUGS a lot. Old habits die hard..... WinBUGS is not in active development  
405     anymore, but JAGS is. So is OpenBUGS but we didn't want to use all of them  
406     (what's the point?). We love the BUGS language because, as Marc Kery said, it  
407     “frees the modeler in you”. If you can express your model algebraically, in the  
408     BUGS language, then JAGS or WinBUGS or OpenBUGS will do the MCMC for  
409     you. That's pretty handy.

410     We do a limited amount of developing our own custom MCMC algorithms (see  
411     chapt. XXXX) which we think is really handy for certain problems. In fact, there  
412     are problems that WinBUGS or JAGS can't do, and so we have had to develop  
413     our own custom algorithms (e.g., Sollmann et al. 2012; Chandler and Royle 2013).  
414     This is really handy because you can then exploit large linux or windows clusters  
415     to distribute your computing efficiently. There are R packages for that (snowfall  
416     XXXX and others).

417     We do a fair amount of likelihood analysis in this book. Mostly we rely on the R  
418     package **secr** (Efford et al., 2009a) but we also show how to compute the likelihood  
419     for certain specific models in a few chapters.

### 0.2 R PACKAGE SCRBOOK

420     We have an R package that comes with the book. Almost all analyses are in there  
421     – R/BUGS scripts, etc.... The purpose of the package is not meant to be general-  
422     purpose and flexible software for doing SCR models but, rather, a set of examples  
423     and templates to see how specific things are done. Because we use so many different  
424     software packages and computing platforms, we think its impossible to put all of  
425     what is covered in this book into a single integrated package.

426     A commented Gibbs sampler written in **R** is available in the accompanying  
427     R package **scrbook** (see `?scrIPP`). This function is not meant to be an all purpose  
428     tool for fitting SCR models using MCMC—instead, it is presented so that interested  
429     readers can better understand the computational aspects of the problem and can  
430     modify it for their purposes.

### 0.3 ORGANIZATION OF THIS BOOK

431 Comment on the 4 parts of the book.....

432 In the following chapters we develop a comprehensive synthesis and extension of  
433 spatial capture-recapture models. Roughly the first third of the book is introductory  
434 material – In Chapt. 3 we provide the basic analysis tools to understand and analyze  
435 SCR models - namely generalized linear models (GLMs) with random effects, and  
436 their analysis in **R** and **WinBUGS**. Because SCR models represent extensions  
437 of basic closed population models, we cover ordinary closed population models in  
438 Chapt. 4 wherein, along with Chaps. 5 and 9<sup>1</sup>, provides the basic introduction to  
439 capture-recapture models and their spatial extension. We will see that SCR models  
440 are a conceptual and technical intermediates between the class of models referred  
441 to as model  $M_h$ , and so-called individual covariate models. We develop technical  
442 tools for likelihood (Chapt. 6) and Bayesian analysis (Chapt. 14). The middle  
443 part of the book expands set of models that we can deal with to include alternative  
444 observation models related to the type of encounter device (Chapt. 9), models for  
445 encounter probability (Chapt. 8), [should include search-encounter models right  
446 after Poisson-mn type models?] and provides basic tools for model fit and selection  
447 (Chapt. 7). [should include the design chapter right here]. Finally in the last  
448 third of the book we address more advanced stuff including modeling space usage  
449 in the encounter process (Chapt. 12), modeling state-space covariates, covariates  
450 that affect density, (Chapt. 13), open population models (Chapt. 15), models  
451 that include unmarked individuals either entirely (Chapt. 18) or partially marked  
452 samples (Chapt. 19).

453 In Chapter XXXX We cover a mish-mash of ideas: using telemetry data, multi-  
454 ple encounter methods, alternative point-process models, and other topics that are  
455 useful but that are not fully developed or that we don't have room for in this book.

456 **From Ch1b:**

457 In our experience, students in ecology and even many established scientists  
458 simply cannot separate what they need to do from how to do it. They cannot dis-  
459 tinguish clearly (either conceptually or actually) the difference between the model  
460 for their data, and the actual procedure of how to estimate parameters of that  
461 model, or make predictions - ie., how to do the calculations. Sometimes this issue  
462 raises itself in an email from some hapless grad student wondering “what is the  
463 right statistical test for this type of data?” In a sense it is this view that drives our  
464 approach to developing elements of this book.

465 In contemporary statistical ecology, models and methods are sometimes ob-  
466 scured by named procedures often that are completely uninformative, the technical  
467 details of which hide in obscurity in some black boxes such as MARK, PRESENCE,  
468 DISTANCE, etc., known only by the few specialist experts in the field. While it is  
469 sometimes convenient to refer to a type or class of models by a name (logistic regres-  
470 sion or even “model  $M_h$ ”) in order to emphasize a broad concept or methodological

---

<sup>1</sup>might ought to put Modeling Encounter Probability as chapter 5 instead

area, this is only useful if the fundamental statistical and mathematical structure underlying that name is clear. As such, we try to focus on model development and keep the model development distinct from how to combine our data with the model to produce estimates and so forth. We talk a lot about hypothetical data we wish we could observe - complete data sets - data sets as if  $N$  were known, etc.. We talk about the model in precise terms and then break down various ways for analyzing the model either using likelihood methods or Bayesian methods or some black-box that does one or the other.

To fit models, we rely heavily on the various implementations of the **BUGS** language including **WinBUGS** (Lunn et al., 2000), **JAGS** (Plummer, 2003) and **OpenBUGS** (Thomas et al., 2006). We really like the **BUGS** language, not merely as a computational device for fitting models but because it emphasizes understanding of what the model is and fosters understanding how to build models - as Kery XYZ XYZ says “it frees the modeler in you.” (direct citation for this would be nice). However, in addition to using the **BUGS** language and its various implementations, we also develop our own **R** code both for doing MCMC and maximum likelihood, for which we also use the R package **secr** (Efford, 2011). In addition, we have created an **R** package to go with this book, **scrbook**, which contains the data sets, **R** and **BUGS** scripts, and **R** code for doing summary analyses, and some likelihood and MCMC functions written solely in **R**.

### 0.3.1 Who should read this book

This book is not a book about Bayesian analysis, not a book about hierarchical models, not a book about capture-recapture, and not about programming in R. In a sense though, our book integrates elements of all of these things into what we hope is a coherent package for analyzing data from this enormous class of data collection methods that produce spatially-explicit capture-recapture data. As such, we expect that people have a basic understanding of statistical models and classical inference (What is frequentist inference? what is a likelihood? Generalized linear model? Generalized linear mixed model?), **R** programming, Bayesian analysis (what is a prior distribution and a posterior distribution?), and maybe even a little bit of Bayesian computation (MCMC and perhaps the BUGS language). The ideal candidate for reading this book has basic knowledge of these topics. However, we do provide introductory chapters on the necessary components which we hope can serve as a brief and cursory tutorial for those who might have only limited technical knowledge, e.g., many carnivore biologists who implement field sampling programs but do not have extensive experience analyzing data.



509

## Part I

510

---

511

# Background and Concepts



---

**1****INTRODUCTION**

515 Knowledge of the spatial structure of populations is central to applied and theoretical population ecology, landscape ecology, conservation biology and many other  
516 ecological disciplines. For examples, understanding distribution or spatial variation  
517 in density are important in the management of populations; movement, dispersal,  
518 space usage, are important to understanding landscape connectivity, and density de-  
519 pendence, and spatial interactions among individuals contribute to these population  
520 processes. At the same time, the inherent spatial aspect of *sampling* populations  
521 strongly affects apparent biases in how we observe population structure.

523 Books have been written on spatial processes in animal populations (Tilman and  
524 Kareiva, 1997; Hanski, 1999; Clobert et al., 2001) and similarly on how to sample  
525 these populations using capture-recapture methods (Seber, 1982; Williams et al.,  
526 2002). However, and despite the central roll of space and spatial processes to both  
527 understanding population dynamics and how we observe or sample populations,  
528 a coherent framework that integrates spatial population processes with the spatial  
529 nature of how we observe populations has not been fully realized either conceptually  
530 or methodologically.

531 Capture-recapture methods represent perhaps the most common technique for  
532 studying animal populations, and it is growing in popularity due to recent tech-  
533 nological innovations that expand the utility of such methods to many taxa that  
534 until recently could not be studied by capture-recapture. A major deficiency with  
535 classical capture-recapture methods is that they do not admit the spatial structure  
536 of either ecological processes that give rise to encounter history data, nor the spa-  
537 tial aspect of sampling populations. While many technical limitations of this lack  
538 of spatial explicitness have been recognized for decades (Hayne, 1950), it has only  
539 been very recently (Efford, 2004) that spatially explicit capture-recapture methods  
540 – those which accomodate space – have been developed.

These spatial capture-recapture (SCR) methods resolve a host of technical problems that arise in applying capture-recapture methods to animal populations. However, SCR models are not merely an extension of technique but rather they represent an extention in a much more profound way in that they make ecological processes explicit in the model – processes of density, spatial organization of individuals, movement and space-usage of individuals. The practical importance of SCR models is that they allow ecological scientists to study elements of ecological theory using observational data that exhibits various biases relating to the observation mechanisms employed. In the context of capture-recapture, we observe individual encounter history data from which we can use SCR models to infer where individual live, how they organize themselves in space and move around in space and how they interact with other individuals. Thus, SCR models enable ecologists to explicitly integrate biological context and theory with encounter history data. We therefore believe that SCR models will enable ecologists to ask questions of space usage, environmental or landscape effects, social behavior and other important elements of ecological theory.

## 1.1 THE STUDY OF POPULATIONS BY CAPTURE-RECAPTURE

Information about abundance or density of populations, and their vital rates, is fundamental to applied ecology and conservation biology. To that end, a huge variety of statistical methods have been devised, and among these, the most well-developed are collectively known as capture-recapture (or capture-mark-recapture) methods. For example, the volumes by Seber (1982), Borchers et al. (2002), Williams et al. (2002), and Amstrup et al. (2005) are largely synthetic treatments of such methods, and contributions on modeling and estimation using capture-recapture are plentiful in the peer-reviewed ecology literature. Capture-recapture techniques make use of individual *encounter history* data, by which we mean sequences (usually) of 0's and 1's denoting if an individual was encountered during sampling over a certain time period. For example, the encounter history "010" indicates that this individual was encountered only during the second of three trapping occasions. As we will see, these data contain information about encounter probability, abundance, and other parameters of interest in the study of population dynamics.

Capture-recapture methods have been important in studies of animal populations for many decades, and their importance is growing dramatically in response to technological advances that improve our ability and efficiency to obtain encounter history data. While such information was obtainable historically only using physical traps, which capture and retain animals until visited by a biologist who removes the individual, marks it, or otherwise molests it in some scientific fashion, new methods do not require physical capture or handling of individuals. A large number of passive detection devices produce individual encounter history data including camera traps (Karanth and Nichols, 1998; O'Connell et al., 2010), acoustic recording devices (Dawson and Efford, 2009), and methods that obtain DNA samples such

581 as hair snares for bears (Gardner et al., 2010b), scent posts for many carnivores  
582 (Kéry et al., 2010), and related methods which allow DNA to be extracted from  
583 scat, urine or animal tissue in order to identify individuals. This book is concerned  
584 with how such data can be used to carry out inference about animal abundance  
585 or density, and other demographic parameters such as survival, recruitment, and  
586 movement using new classes of capture-recapture models which utilize auxiliary  
587 spatial information related to the encounter process. We refer to such methods as  
588 spatial capture-recapture (SCR) models<sup>1</sup>.

589 As the name implies, the primary feature of SCR models that distinguishes  
590 them from traditional CR methods is that they make use of the spatial informa-  
591 tion inherent to capture-recapture studies. That is, when encounter histories are  
592 associated with auxiliary information on the location of capture, which we refer  
593 to as *spatial encounter histories*, then such information is informative about spa-  
594 tial processes such as spatial variation in density, movement, resource selection,  
595 and space usage. As we will see, this allows us to overcome critical deficiencies of  
596 non-spatial methods, and also integrate into capture-recapture models explicit eco-  
597 logical hypotheses related to spatial processes such as movement, space usage, and  
598 the spatial organization of individuals in a population. This greatly expands the  
599 practical utility and scientific relevance of capture-recapture methods and studies  
600 based on encounter history data.

## 1.2 SCOPE OF THIS BOOK

601 In this book, we try to achieve a broad methodological scope from basic closed  
602 population models for inference about population density, movement, space usage  
603 and resource selection, on up to open population models for inference about vital  
604 rates such as survival and recruitment. Much of the material is a synthesis of recent  
605 research but we also expand SCR models in a number of useful directions, including  
606 to accommodate unmarked individuals (Chapt. 18), use of telemetry information  
607 (Chapt. 11), and developing explicit models of landscape connectivity based on eco-  
608 logical or least-cost distance (Chapt. 12), and many other new topics that have only  
609 recently appeared in the literature. Our intent is to provide a comprehensive re-  
610 source for ecologists interested in understanding and applying SCR models to solve  
611 common problems faced in the study of populations. To do so, we make use of hier-  
612 archical models (Royle and Dorazio, 2008), which allow extraordinary flexibility in  
613 accommodating many types of capture-recapture data. We present many example  
614 analyses, of real and simulated data using likelihood-based and Bayesian methods—  
615 examples that readers can replicate using the code presented in the text and the  
616 resources made available on-line and in our accompanying **R** package **scrbook**.

617 Although we aim to reach a broad audience, at times we go into details that may  
618 only be of interest to advanced practitioners who need to extend capture-recapture

---

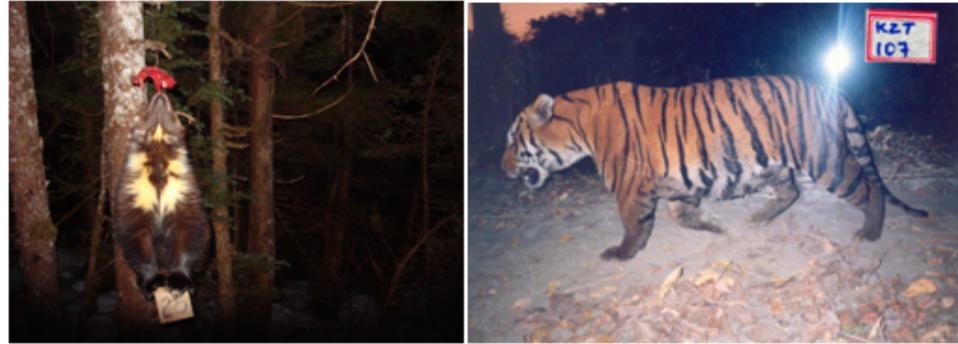
<sup>1</sup>In the literature the term spatially explicit capture-recapture (SECR) is also used

619 models to unique situations. We hope that these advanced topics will not discourage  
620 those new to these methods, but instead our intent is to allow readers to advance  
621 their own understanding and become less reliant on restrictive tools and software.  
622 A number of conceptual and methodological themes unify the main topical coverage  
623 of this book, and those are:

- 624 (1) Hierarchical modeling. We develop hierarchical models consisting of explicit  
625 models for both the observation process and the underlying “ecological process”  
626 which describes the organization of individuals in space.
- 627 (2) Spatial processes in capture-recapture. We emphasize the linkage of capture-  
628 recapture data to underlying ecological processes including density or distribution  
629 of individuals, space usage, resource selection and movement.
- 630 (3) Formal inference using both classical (frequentist, likelihood-based) and Bayesian  
631 methods. We often emphasize Bayesian analysis because this allows us to focus  
632 the technical formulation of models, and spatial capture-recapture is mainly con-  
633 cerned with modeling random effects and estimating functions of random effects.  
634 However, we also explore likelihood methods using existing software such as the  
635 **R** package **secr** (Efford, 2011), as well as development of custom solutions along  
636 the way.
- 637 (4) In developing Bayesian analyses of SCR models, we emphasize the use of the  
638 **BUGS** language for describing models. The **BUGS** language emphasizes the  
639 syntactic description of the essential assumptions of models in a special kind of  
640 pseudo-code language, which is used in software (**WinBUGS**, **JAGS**, **Open-**  
641 **BUGS**) to devise Markov chain Monte Carlo (MCMC) algorithms for Bayesian  
642 analysis of models. The **BUGS** language focuses your thinking on model devel-  
643 opment and lets you develop an understanding of models at the level of their basic  
644 assumptions and structure. Despite our focus on describing models in **BUGS**, we  
645 also show readers how to devise their own MCMC algorithms for Bayesian anal-  
646 ysis of SCR models, which can be convenient (even necessary) in some practical  
647 situations.

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

648 A diverse number of methods and devices exist for producing individual encounter  
649 history data with auxiliary spatial information about individual locations. Histori-  
650 cally, physical “traps” have been widely used to sample animal populations. These  
651 include live traps, leg-hold traps, mist nets, pitfall traps and many other types of  
652 devices. Although these are still widely used, recent technological advances for ob-  
653 taining encounter history data non-invasively have made it possible to study many  
654 species that were difficult if not impossible to study effectively just a few years ago.  
655 These methods have revolutionized the study of animal populations by capture-  
656 recapture methods, and will lead to their increasing relevance in the future. We

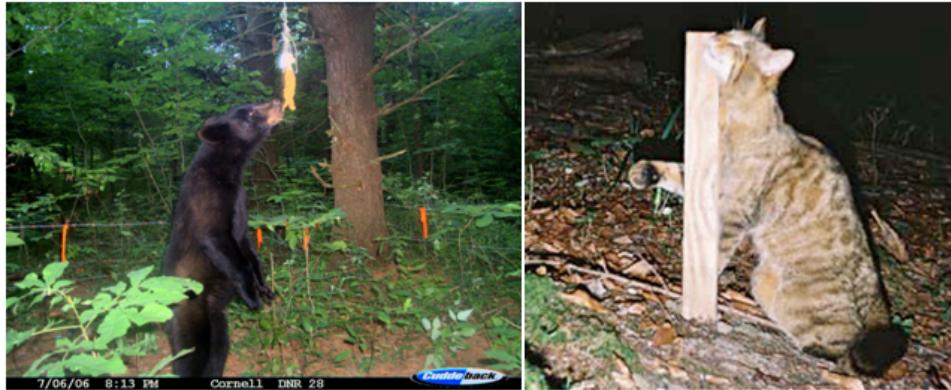


**Figure 1.1.** Left: Wolverine being encounter by a camera trap (*Photo credit: Audrey Magoun*). Right: Tiger encountered by camera trap (*Photo credit: Ullas Karanth*).

briefly review some of these here, which we consider more explicitly in later chapters of this book.

### 1.3.1 Camera trapping

Considerable recent work has gone into the development of camera-trapping methodologies. For a historical overview of this method see Kays et al. (2008) and Kucera and Barrett (2011). Several recent synthetic works have been published including Nichols and Karanth (2002), and an edited volume by O'Connell et al. (2010) devoted solely to camera trapping concepts and methods. As a method for estimating abundance, some of the earliest work that relates to the use of camera trapping data in capture-recapture models originates from Karanth and colleagues (Karanth, 1995; Karanth and Nichols, 1998, 2000). In camera trapping studies, cameras are situated along trails or at baited stations and individual animals are photographed and subsequently identified either manually by a person sitting behind a computer, or sometimes now using computational methods. Camera trapping methods are widely used for species that have unique stripe or spotting patterns such as tigers (Karanth, 1995; Karanth and Nichols, 1998), ocelots (Trolle and Kéry, 2003, 2005), leopards (Balme et al., 2010), and many other cat species. Camera traps are also used for other species such as wolverines (*Gulo gulo* Magoun et al., 2011; Royle et al., 2011b), and even species that are less easy to identify uniquely such as mountain lions (Sollmann et al., in revision) and coyotes (e.g. Kelly et al. (2008)). We note that even for species that are not readily identified by pelage patterns, it might be efficient to use camera traps in conjunction with spatial capture-recapture models to estimate density (see Chapt. 18).



**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* )

680 **1.3.2 DNA Sampling**

681 DNA obtained from hair, blood or scat is now routinely used to obtain individual  
682 identity and encounter history information about individuals (Taberlet and Bouvet,  
683 1992; Kohn et al., 1999; Woods et al., 1999; Mills et al., 2000; Schwartz and Monfort,  
684 2008). A common method is based on the use of “hair snares” (Fig. 1.2) which are  
685 widely used to study bear populations (Woods et al., 1999; Gardner et al., 2010b;  
686 Garshelis and Hristienko, 2006; Kendall et al., 2009). A sample of hair is obtained as  
687 individuals pass under or around barbed-wire (or other physical mechanism) to take  
688 bait. Hair snares and scent sticks have also been used to sample felid populations  
689 (García-Alaníz et al., 2010; Kéry et al., 2010) and other species. Research has  
690 even shown that DNA information can be extracted from urine deposited in the  
691 wild (e.g., in snow; see Valiere and Taberlet (2000)) and as a result this may prove  
692 another future data collection technique where SCR models are useful.

693 **1.3.3 Acoustic surveys**

694 Many studies of birds (Dawson and Efford, 2009), bats, and whales (Marques et al.,  
695 2009) now collect data using devices that record vocalizations. When vocalizations  
696 can be identified by individual from multiple recording devices, spatial encounter  
697 histories are produced that are amenable to the application of SCR models (Dawson  
698 and Efford, 2009; Efford et al., 2009b). Recently, these ideas have been applied to  
699 data on direction or distance to vocalizations by multiple simultaneous observers  
700 (Borchers et al., xxxx).



**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.*

701    **1.3.4 Search-Encounter Methods**

702    There are other methods which don't fall into a nice clean taxonomy of "devices".  
703    Spatial encounter histories are commonly obtained by conducting manual searches  
704    of geographic sample units such as quadrats, transects or road or trail networks.  
705    For example, DNA-based encounter histories can be obtained from scat samples  
706    located along roads or trails or by specially trained dogs (MacKay et al., 2008)  
707    searching space (Fig. 1.3). This method has been used in studies of martens, fishers  
708    (Thompson et al., 2012), lynx, coyotes, birds Kéry et al. (2010), and many other  
709    species. Similar data structure arises from the use of standard territory or spot  
710    mapping of birds Bibby et al. (1992) or area sampling in which space is searched by  
711    observers to physically capture individuals. This is common in surveys that involve  
712    reptiles and amphibians, e.g., we might walk transects picking up box turtles (Hall  
713    et al., 1999), or desert tortoises (Zylstra et al., 2010), or search space for lizards  
714    (Royle and Young, 2008).

715       These methods don't seem like normal capture-recapture in the sense that the  
716    encounter of individuals is not associated with specific trap location, but SCR  
717    models are equally relevant for analysis of such data as we discussed in Chapt. 17.

## 1.4 HISTORICAL CONTEXT: A BRIEF SYNOPSIS

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

723 **1.4.1 Buffering**

724 The standard approach to estimating density even now is to estimate  $N$  using  
 725 conventional closed population models (Otis et al., 1978) and then try to associate  
 726 with this estimate some specific sampled area, say  $A$ , the area which is contributing  
 727 individuals to the population for which  $N$  is being estimated. The strategy is to  
 728 define  $A$  by placing a buffer of say  $W$  around the trap array or some polygon which  
 729 encloses the trap array. The historical context is succinctly stated by (O'Brien,  
 730 2011) from which we draw this description:

731 “At its most simplistic,  $A$  may be described by a concave polygon defined by connect-  
 732 ing the outermost trap locations ( $A_{tp}$ ; Mohr (1947)). This assumes that animals do  
 733 not move from outside the bounded area to inside the area or vice versa. Unless the  
 734 study is conducted on a small island or a physical barrier is erected in the study area  
 735 to limit movement of animals, this assumption is unlikely to be true. More often, a  
 736 boundary area of width  $W$  ( $A_w$ ) is added to the area defined by the polygon  $A_{tp}$  to  
 737 reflect the area beyond the limit of the traps that potentially is contributing animals  
 738 to the abundance estimate (Otis et al., 1978). The sampled area, also known as the  
 739 effective area, is then  $A(W) = A_{tp} + A_w$ . Calculation of the buffer strip width ( $W$ )  
 740 is critical to the estimation of density and is problematic because there is no agreed  
 741 upon method of estimating  $W$ . Solutions to this problem all involve ad hoc methods  
 742 that date back to early attempts to estimate abundance and home ranges based on  
 743 trapping grids (see Hayne, 1949). Dice (1938) first drew attention to this problem  
 744 in small mammal studies and recommended using one-half the diameter of an av-  
 745 erage home range. Other solutions have included use of inter-trap distances (Blair,  
 746 1940; Burt, 1943), mean movements among traps, maximum movements among traps  
 747 (Holdenried, 1940; Hayne, 1949), nested grids (Otis et al., 1978), and assessment lines  
 748 (Smith et al., 1971).”

749 The idea of using 1/2 mean maximum distance moved (“MMDM” Wilson and  
 750 Anderson, 1985b) to create a buffer strip seems to be the standard approach even to-  
 751 day, presumably justified by Dice’s suggestion to use 1/2 the home range diameter,  
 752 with the mean over individuals of the maximum distance moved being an estimator  
 753 of home range diameter. Alternatively, some studies have used the full MMDM  
 754 (e.g. Parmenter et al. (2003)), because the trap array might not provide a full cov-  
 755 erage of the home range (home ranges near the edge should be truncated) and so  
 756 1/2 MMDM should be biased smaller than the home range radius. And, sometimes  
 757 home range size is estimated by telemetry (Karanth, 1995; Bales et al., 2005). Use  
 758 of MMDM summaries to estimate home range radius is usually combined with an  
 759 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 has increased with the advent of new technologies and 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) 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 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 closed population models and fully spatial capture-recapture models, which we address directly in Chapt. 4.

While these procedures are all heuristically appealing, they are also essentially ad hoc in the sense that the underlying model remains unspecified or at least imprecisely characterized and so there is little or no basis for modifying, extending or generalizing the methods. These methods are distinctly *not* model-based procedures even though they might well be heuristically appealing under specific move-

801   ment models. Despite this, there seems to be an enormous amount of literature  
802   developing, evaluating and “validating” these literally dozens of heuristic ideas that  
803   solve specific problems, as well as various related tweaks and tunings of them and  
804   really it hasn’t led to any substantive breakthroughs that are sufficiently general or  
805   theoretically rigorous.

## 1.5 CAPTURE-RECAPTURE FOR MODELING ENCOUNTER PROBABILITY

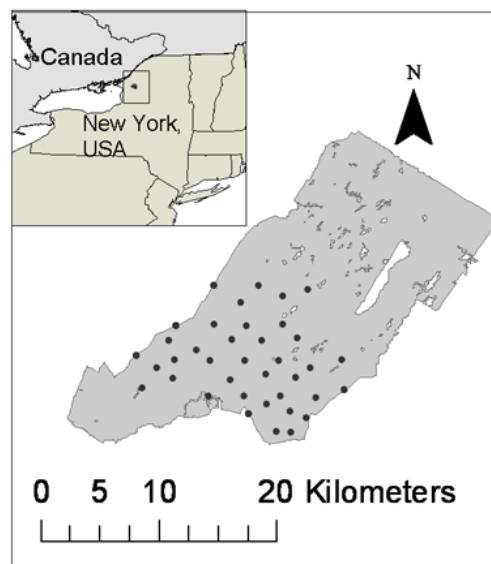
806   We briefly introduced and reviewed a number of classical techniques for applying  
807   non-spatial capture-recapture models to studies of animal populations. These tech-  
808   niques, such as buffering, are based on many heuristically appealing ideas. But  
809   these are just heuristics and do not resolve the essential, basic problem with con-  
810   ventional (“non-spatial”) capture-recapture models which is that there is no linkage  
811   *in the model* between the quantity being informed by the data (i.e.,  $N$ ) and any  
812   stated or prescribed “area”,  $A$ .

### 813   1.5.1 An Example: Fort Drum Bear Study

814   Here we confront some of the issues that motivate the need for spatial capture-  
815   recapture models by considering analysis of data from a study design to estimate  
816   black bear abundance on the Fort Drum Military Installation in upstate New York  
817   (see Chapt. 4 for more details). The specific data used here are encounter histories  
818   on 47 individuals obtained from an array of 38 baited “hair snares” during June  
819   and July 2006. The study area and locations of the 38 hair snares are shown in  
820   Fig. 1.4. Barbed wire traps (see Fig. 1.2) were baited and checked for hair samples  
821   each week for eight weeks. Analysis of these data appears in Gardner et al. (2010b)  
822   and we use the data in a number of analyses in later chapters.

823   We regarded this data set as a standard capture-recapture data set - an en-  
824   counter history matrix with 47 rows and 8 columns with entries  $y_{ik}$ , where  $y_{ik} = 1$   
825   if individual  $i$  was captured in sample  $k$  and  $y_{ik} = 0$  otherwise. There is a standard  
826   closed population model, colloquially referred to as “model  $M_0$ ” (see Chapt. 4),  
827   which assumes that encounter probability  $p$  is constant for all individuals and sam-  
828   ple periods. We fitted model  $M_0$  to the Fort Drum data using traditional likelihood  
829   methods, yielding the maximum likelihood estimate (MLE) of  $\hat{N} = 49.19$  with an  
830   asymptotic standard error (SE) of 1.9.

831   The key issue in using closed population models with such data is how on earth  
832   do we interpret this estimate of  $\hat{N} = 49.19$  bears? Does it represent the entire  
833   population of Fort Drum? Certainly not – the trapping array covers less than half  
834   of Fort Drum! (Fig. 1.4). So to get at the total bear population size of Fort  
835   Drum, we’d have to convert our  $\hat{N}$  to an estimate of density and extrapolate. To  
836   get at density, then, should we assert that  $N$  applies to the southern half of Fort Drum  
837   below some arbitrary line? Surely bears move on and off of Fort Drum



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

838 without regard to hypothetical boundaries. Without additional information there  
 839 is simply no way of converting this estimate of  $N$  to density, and hence it is really  
 840 not meaningful biologically. To resolve this problem, we will adopt the customary  
 841 approach of converting  $N$  to  $D$  by buffering the convex hull around the trap array.  
 842 The convex hull has area  $157.135 \text{ km}^2$ . We follow Bales et al. (2005) in buffering  
 843 the convex hull of the trap array by the radius of the mean female home range size.

844 The mean female home range radius was estimated (Wegan, 2008) for our study  
 845 region to be  $2.19 \text{ km}$ , and the area of the convex hull buffered by  $2.19 \text{ km}$  is  
 846  $277.01 \text{ km}^2$ . (**R** commands to compute the convex hull, buffer it, and compute the  
 847 area are given in the **R** package **scrbook** which accompanies the book). Hence,  
 848 the estimated density here is approximately  $0.178 \text{ bears/km}^2$  using the estimated  
 849 population size obtained by model  $M_0$ . We could assert that the problem has been  
 850 solved, go home, and have a beer. But then, on the other hand, maybe we should  
 851 question the use of the estimated home range radius – after all, this is only the  
 852 female home range radius and the home ranges change for many reasons. Instead,  
 853 we may decide to rely on a buffer width based on one-half MMDM estimated from  
 854 the actual hair snare data as is more customary (Dice, 1938). In that case the  
 855 buffer width is  $1.19 \text{ km}$ , and the resulting estimated density is increased to  $0.225$   
 856 bears/ $\text{km}^2$  about 27 % larger. But wait – some studies actually found the full  
 857 MMDM (Parmenter et al., 2003) to be a more appropriate measure of movement  
 858 (e.g Soisalo and Cavalcanti (2006)). So maybe we should use the full MMDM which  
 859 is  $2.37 \text{ km}$ , pretty close to the telemetry-based estimate and therefore providing a  
 860 similar estimate of density ( $0.171 \text{ bears/km}^2$ ). So in trying to decide how to buffer  
 861 our trap array we have already generated 3 density estimates. The crux of the  
 862 matter is obvious: Although it is intuitive that  $N$  should scale with area – the  
 863 number of bears should go up as area increases and go down as area decreases – in  
 864 this ad hoc approach of accounting for animal movement  $N$  remains the same, no  
 865 matter what area we assert was sampled. The number of bears and the area they  
 866 live in are not formally tied together within the model, because estimating  $N$  and  
 867 estimating the area  $N$  relates to are two completely independent analytical steps  
 868 which are unrelated to one another by a formal model.

869 Unfortunately, our problems don't end here. In thinking about the use of model  
 870  $M_0$ , we might naturally question some of the basic assumptions that go into that  
 871 model. The obvious one to question is that which declares that  $p$  is constant.  
 872 One obvious source of variation in  $p$  is variation *among individuals*. We expect  
 873 that individuals may have more or less exposure to trapping due to their location  
 874 relative to traps, and so we try to model this “heterogeneous” encounter probability  
 875 phenomenon. To illustrate this here are the number of traps that each individual  
 876 was captured in:

```
877 #traps: 1 2 3 4 5 6 9
878 #bears: 23 13 6 2 1 1 1
```

879 suggesting quite a range in traps exposed to by different bears. This has led many

880 to consider capture-recapture models that allow for individual heterogeneity in  $p$ .  
 881 Such models have the colloquial name of “model  $M_h$ ” (Otis et al., 1978). We fitted  
 882 this model (see Chapt. 4 for details) to the Fort Drum data using each of the 3 buffer  
 883 widths previously described (telemetry, 1/2 MMDM and MMDM), producing the  
 884 estimates reported in Table 1.1. While we can tell by the models’ AIC that  $M_h$  is  
 885 clearly favored by more than 30 units, we might still not be entirely happy with our  
 886 results. Clearly there is information in our data that could tell us something about  
 887 the exposure of individual bears to the trap array – where they were captured, and  
 888 how many times – but since space has no representation in our model, we can’t  
 889 make use of this information. Model  $M_h$  thus merely accounts for what we observe  
 890 in our data (some bears were more frequently captured than others) rather than  
 891 explicitly accounting for the processes that generated the data.

892 XXXXX SCRIPT NEEDS put IN PACKAGE XXXXXXXX

893 So what are we left with? Our density estimates span a range from 0.17 to 0.43  
 894 bears/km<sup>2</sup> depending on which estimator of  $N$  we use and what buffer strip we  
 895 apply. Should we feel strongly about one or the other? Which buffer should we  
 896 prefer? AIC favors model  $M_h$ , but did it adequately account for the differences in  
 897 exposure of individuals to the trap array? Are we happy with a purely phenomeno-  
 898 logical model for heterogeneity? It assumes that all individuals are iid draws from  
 899 some distribution but does not account for the explicit mechanism of induced het-  
 900 erogeneity. And, further, we have information about that (trap of capture) which  
 901 model  $M_h$  ignores. And if we choose one type of buffer, how do we compare our  
 902 density estimates to those from other studies that may opt for a different kind of  
 903 buffer? The fact that  $N$  does not scale with  $A$ , as part of the model, renders this  
 904 choice arbitrary. The buffer isn’t part of the model.

**Table 1.1.** Table on estimates of density ( $D$ , bears/km<sup>2</sup>) 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	$\bar{D}$	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

### 905 1.5.2 Inadequacy of Capture-Recapture

906 \*\*\*Models are not integrated with any ecological theory.\*\*\*\*

907 The parameter  $N$  in an ordinary capture-recapture model is functionally unre-  
 908 lated 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.

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.

More specifically, 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. For capture-recapture models to provide a coherent framework for inference about population density,  $N$  has to scale, as part of the model, with  $A$  so that the model imposes biological context on  $A$  (i.e., as the area over which the  $N$  individuals reside). SCR models achieve this. (2) Ordinary CR models *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-level covariates which exist in a large proportion of real studies. Again, SCR models formally accommodate heterogeneity due to trap variation; (4) Ordinary CR models do not accommodate formal consideration of any spatial process that gives rise to the observed data.

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

## 1.6 EXTENSION OF CLOSED POPULATION MODELS

The deficiency with classical closed population models is that they have no spatial context.  $N$  is just an integer parameter that applies equally well to some population that exists in a computer, estimating the number of unique words in a book, or a bucket full of goldfish. The question of *where* the  $N$  items belong is central both to interpretation of data and estimates from all capture-recapture studies and, in fact, to the construction of spatial capture-recapture models considered in this book. Surely it must matter whether the  $N$  items exist as words in a book, or goldfish in a bowl, or birds in a forest patch! That classical closed population models have no spatial context leads to a number of conceptual and methodological problems or limitations as we have encountered previously. More important, ecologists seldom

949 care only about  $N$  – space is often central to objectives of many population studies  
950 – movement, space usage, resource selection, how individuals are distributed in  
951 space and in response to explicit factors related to landuse or habitat. because  
952 space is central to so many real problems, this is probably the # 1 reason that  
953 most ecologists don't bother with capture-recapture. They haven't seen that as  
954 being able to solve their problems.

955 Thus, the essential problem is that classical closed population models are too  
956 simple - they ignore the spatial attribution of traps and encounter events, movement  
957 and variability in exposure of individuals to trap proximity, and, because ordinary  
958 closed population models possess no notion of “area”, they do not yield estimates  
959 of *density*. These problems can be addressed formally by the development of more  
960 general models.

### 961 **1.6.1 Efford's Formulation**

962 The solution to the various issues that arise in the application of ordinary capture-  
963 recapture models is to extend the closed population model so that  $N$  becomes  
964 spatially explicit. Efford (2004) was the first to formalize an explicit model for  
965 spatial capture-recapture problems in the context of trapping arrays. He adopted  
966 a Poisson point process model to describe the distribution of individuals and then  
967 what is essentially a distance sampling formulation of the observation model which  
968 describes the probability of detection as a function of individual location, regarded  
969 as a latent variable governed by the point process model. While earlier (and con-  
970 temporary) methods of estimating density from trap arrays have been ad hoc in  
971 the sense of lacking a formal description of the spatial model, Efford achieved a  
972 formalization of the model, describing explicit mechanisms governing the spatial  
973 distribution of individuals and how they are encountered by traps, but adopted  
974 a more or less ad hoc framework for inference under that spatial model using a  
975 simulation based method known as inverse prediction (Gopalaswamy, 2012).

976 Recently, there has been a flurry of effort devoted to formalizing inference un-  
977 der this model-based framework for the analysis of spatial capture-recapture data  
978 (Royle and Gardner, 2011; Borchers, 2012; Gopalaswamy, 2012). There are two  
979 distinct lines of work which adopt the model-based formulation in terms of the  
980 underlying point process but differ primarily by the manner in which inference is  
981 achieved. One approach (Borchers and Efford, 2008) is a classical inference ap-  
982 proach based on likelihood (see Chapt. 6), and the other (Royle and Young, 2008)  
983 adopts a Bayesian framework for inference (Chaps. 5 and 14).

### 984 **1.6.2 Abundance as the Aggregation of a Point Process**

985 Spatial point process models represent a major methodological theme in spatial  
986 statistics (Cressie, 1991) and they are widely applied as models for many ecological  
987 phenomena (Stoyan and Penttinen, 2000; Illian et al., 2008). Point process models

988 apply to situations in which the random variable in question represents the locations  
 989 of events or objects: trees in a forest, weeds in a field, bird nests, etc. As such,  
 990 it seems natural to describe the organization of individuals in space using point  
 991 process models. SCR models represent the extension of ordinary capture-recapture  
 992 models by augmenting the model with a point process model to describe individual  
 993 locations.

994 Specifically, let  $\mathbf{s}_i; i = 1, 2, \dots, N$  be the locations of all individuals in the popula-  
 995 tion. One of the key features of SCR models is that the point locations are latent,  
 996 or unobserved, and we only obtain imperfect information about the point locations  
 997 by observing individuals at trap or observation locations. Thus, the realized loca-  
 998 tions of individuals represent a type of “thinned” point process, where the thinning  
 999 mechanism is not random but, rather, biased by the observation mechanism. It is  
 1000 also natural to think about the observed point process as some kind of a compound  
 1001 or aggregate point process with a set of “parent” nodes being the locations of in-  
 1002 dividual home ranges or their centroids, and the observed locations as “offspring”  
 1003 - i.e., a Poisson cluster process (PCP). In that context, density estimation in SCR  
 1004 models is analogous to estimating the number of parents of a Poisson cluster process  
 1005 (Chandler and Royle, In press).

1006 Most of the recent developments in modeling and inference from spatial en-  
 1007 counter history data, including most methods discussed in this book, are predi-  
 1008 cated on the view that individuals are organized in space according to a relatively  
 1009 simple point process model. More specifically, we assume that the collection of in-  
 1010 dividual activity centers are independent and identically distributed (abbreviated  
 1011 “iid”) random variables distributed uniformly over some region. This is consistent  
 1012 with the assumption that the activity centers represent the realization of a Poisson  
 1013 point process or, if the total number of activity centers is fixed, then this is usually  
 1014 referred to as a binomial point process.

### 1015 1.6.3 The Activity Center Concept

1016 In the context of SCR models, and because most animals we study by capture-  
 1017 recapture are not sessile, there is not a unique and precise mathematical definition  
 1018 of the point locations  $\mathbf{s}$ . Rather, we imagine these to be the centroid of individual’s  
 1019 home ranges, or the centroid of an individual’s activities during the time of sam-  
 1020 pling. In general, this point is unknown for any individual but if we could track an  
 1021 individual over time and take many observations then we could perhaps get a good  
 1022 idea of where that point is. We’ll think of the collection of these points as defining  
 1023 the spatial distribution of individuals in the population.

1024 We use the terms home range or activity center interchangeably. The term  
 1025 “home range center” suggests that models are only relevant to animals that exhibit  
 1026 behavior of establishing home ranges or territories, or central place foragers, and  
 1027 since not all species do that, perhaps the construction of SCR models based on this  
 1028 idea is flawed. However, the notion of a home range center is just a conceptual

device and we don't view this concept as being strictly consistent with classical notions of animal territories. Rather our view is that a home range or territory is inherently dynamic, temporally, and thus it is a transient quantity - where the animal lived during the period of study, a concept that is completely analogous to the more conventional notion of utilization distributions. Therefore, whether or not individuals of a species establish home ranges is irrelevant because, once a precise time period is defined, this defines a distinct region of space that an individual must have occupied.

#### 1.6.4 The state-space

Once we introduce the collection of activity centers,  $\mathbf{s}_i; i = 1, 2, \dots, N$ , then the question "what are the possible values of  $\mathbf{s}$ ?" needs to be addressed because the individual  $\mathbf{s}_i$  are *unknown*. As a technical matter, we will regard them as random effects and in order to apply standard methods of statistical inference we need to provide a distribution for these random effects. In the context of the point process model, the possible values of the point locations referred to as the "state-space" of the point process and this is some region or set of points which we will denote by  $\mathcal{S}$ . This is analogous to what is sometimes called the *observation window* for  $\mathbf{s}$  in the point process literature. The region  $\mathcal{S}$  serves as a prior distribution for  $\mathbf{s}_i$  (or, equivalently, the random effects distribution). In animal studies, as a description of where individuals that could be captured are located, it includes our study area, and should accommodate all individuals that could have been captured in the study area. In the practical application of SCR models, in most cases estimates of density will be relatively insensitive to choice of state-space which we discuss further in Chapt. 5 and elsewhere.

#### 1.6.5 Abundance and Density

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

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

A deficiency with some classical methods of "adjustment" is they attempted to prescribe something like a state-space - a "sampled area" - except absent any precise linkage of individuals with the state-space. SCR models formalize the linkage between individuals and space and, in doing so, provide an explicit definition of  $N$  associated with a well-defined spatial region, and hence density. That is, the provide a model in which  $N$  scales, as part of the model, with the size of the prescribed state-space. In a sense, the whole idea of SCR models is that by defining this point

1066 process and its state-space  $\mathcal{S}$ , this gives context and meaning to  $N$  which can be  
1067 estimated directly for that specific state-space. Thus, it is fixing  $\mathcal{S}$  that resolves  
1068 the problem of “unknown area” that we have previously discussed.

## 1.7 A SURVEY OF SPATIAL CAPTURE-RECAPTURE

1069 combine with next and cover basic sampling issues and models?

## 1.8 ELEMENTS OF SCR MODELS

1070 Formulation of capture-recapture models conditional on the latent point process  
1071 is the critical and unifying element of *all* SCR models. However, there are many  
1072 more aspects relevant to the formulation of SCR models for specific situations. We  
1073 address various of these things in later chapters.

### 1074 1.8.1 Biological focus

1075 SCR models differ in how the underlying process model is formulated, or its com-  
1076 plexity. Most of the development and application of SCR models has focused on  
1077 their use to estimate density and touting the fact that they resolve certain spe-  
1078 cific technical problems related to the use of ordinary capture-recapture models (as  
1079 noted in sec. xXXXXX above). This is achieved with a simple process model being  
1080 a basic point process of independently distributed points. At the same time, there  
1081 are models of CR data that focus exclusively on *movement* modeling, or models  
1082 with explicit dynamics (Ovaskainen, 2004; Ovaskainen et al., 2008). Conceptually,  
1083 these are akin to spatial versions of so-called Cormack-Jolly-Seber (CJS) models  
1084 in the traditional capture-recapture literature, except they involve explicit mathe-  
1085 matical models of movement based on diffusion or Brownian motion. Finally, there  
1086 are now a very small number of papers that focus on *both* movement and density  
1087 simultaneously (Royle and Young, 2008; Royle et al., 2011a; Royle and Chandler,  
1088 2012) or population dynamics and density (Gardner et al., 2010b).

1089 A key thing is that these models, whether focused just on density, or just on  
1090 movement, or both, are similar models in terms of the underlying concepts, the  
1091 latent structure, and the observation model. They are rather just different in what  
1092 the ecological focus is.

1093 It is great to focus on elaborate models of movement.... but a strict focus on  
1094 developing elaborate movement models will be limited by two practical consid-  
1095 erations: (1) most capture-recapture data e.g., by camera trapping or whatever,  
1096 produces only a few observations of each individual (between 1-5 would be typi-  
1097 cal). So there is not too much information about complex movement models. (2)  
1098 Typically people have an interest in density of individuals and therefore you need  
1099 models that can be extrapolated from the sample to the unobserved part of the  
1100 population. My sense in looking at some of the movement modeling papers is that

they are focused on "what is this individual doing in relation to the space it has available" and there is no formal attempt to extrapolate a sample to the population. That said, there are clearly some cases where more elaborate movement models should come into play. If one has some telemetry data in addition to SCR then there is additional info on fine-scale movements that should be useful.

### 1.8.2 Sampling focus

A second way to characterize SCR models is based on how the encounter observations arise.

Broadly speaking we differentiate between two situations: Sampling based on fixed arrays or sampling based on "search encounter" methods. The former includes things like camera traps, hair snares, mist nets and conventional traps. Fixed arrays limit the observation location to pre-defined points, where traps are located. Using such methods the model is a little simpler because the "movement process" of individuals is confounded with the "observation process". The 2nd type of model – search encounter models – typically will allow locations in continuous space, possibly only restricted by polygon boundaries (Royle and Young, 2008). Search-encounter data usually allow for the separate modeling and estimation of movement model parameters from encounter model parameters but not always, depending on whether replication of the sampling is done.

## 1.9 SUMMARY AND OUTLOOK

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

But SCR models are much more than an extension of technique that resolves certain technical problems of ordinary capture-recapture models. Rather, for the first time, they provide a coherent, flexible framework for making ecological processes explicit in models of individual encounter history data. Spatial capture-recapture models show great promise in their ability to integrate explicit ecological theories directly into the models so that we can directly test hypotheses about either space usage (e.g., Chapt. 12) or movement (Chapt. 17) or the distribution of individuals in space (Chapt. 13). We imagine that in the near future SCR models will include point process models that allow for interactions among individuals such as inhibition or clustering (Reich et al., 2012). Thus, SCR models are capture-recapture

models that enable ecologists to explicitly integrate biological context and theory with encounter history data, which is something that has always been the focus of “open population” models but never, until very recently, has been considered formally in closed population models. We therefore believe that SCR models will enable ecologists to test theories of space usage and environmental effects, social behavior and other important theories.

**key graph:** SCR = holistic framework for studing animal populations – individual movement, space usage, population dynamics, and density. Historically these things are all studied differently using ostensibly unrelated study designs and statistical procedures. RSF models for resource selection or space usage, state-space models for movement, density using closed capture-recapture, and dynamics from a hodge-podge of open extensions of those. SCR brings all of these problems together into a single unified framework for modeling and inference.

In the following chapters we develop a comprehensive synthesis and extension of spatial capture-recapture models. Roughly the first third of the book is introductory material – In Chapt. 3 we provide the basic analysis tools to understand and analyze SCR models - namely generalized linear models (GLMs) with random effects, and their analysis in **R** and **WinBUGS**. This is important material because we find that SCR models are essentially variations of generalized linear mixed models (GLMMs). This in a sense makes them consistent with many important methodologies used in ecology (e.g., see Zuur et al. (2009); Kéry et al. (2010)), and because of the connection with standard modeling concepts, we believe that the material presented in this book can be understood and used by most ecologists with some modeling experience.

Because SCR models represent extensions of basic closed population models, we cover ordinary closed population models in Chapt. 4 wherein, along with Chaps. 5 and 9 provides the basic introduction to capture-recapture models and their spatial extension. We delve more deepling into the details of both likelihood (Chapt. 6) and Bayesian analysis (Chapt. 14) of SCR models. In the last third of the book, we address more advanced stuff including modeling space usage in the encounter process (Chapt. 12), modeling state-space covariates, covariates that affect density, (Chapt. 13), open population models (Chapt. 15), models that include unmarked individuals either entirely (Chapt. 18) or partially marked samples (Chapt. 19). This last third of the book is largely based on research that has only very recently been published in the primary literature but we feel is important to provide a full picture of the importance of SCR models.

1176  
1177

## 2

1178

---

# HIERARCHICAL MODELING AND SCR

1179 In the previous chapter we described the basics of capture-recapture methods and  
1180 the advantages that spatial models have over traditional non-spatial models. In  
1181 doing so, we avoided statistical terminology. Although it is critical to understand  
1182 the non-technical motivation for this broad class of models, it is impossible to  
1183 fully appreciate them, and apply them to real data, without a solid grasp of the  
1184 fundamentals of statistical inference.

1185 In this chapter, we present a brief overview of the key statistical concepts that  
1186 are referenced throughout the remainder of this book. Emphasis is placed on the  
1187 definition of a random variable, the common probability distributions used to model  
1188 random variables, and how hierarchical models can be used to describe conditionally  
1189 related random variables. For some readers, this material will be familiar, perhaps  
1190 even elementary, and thus you may want to skip to the next chapter. However, our  
1191 experience is that many basic statistics courses taken by ecologists do not emphasize  
1192 the important subjects covered in this chapter. Instead, there seems to be much  
1193 attention paid to pedantics such as computing the number of degrees of freedom in  
1194 various  $F$ -tests which, although useful in some contexts, do not provide the basis  
1195 for drawing conclusions from data and evaluating scientific hypotheses.

1196 The material in the beginning of this chapter is explained in numerous other  
1197 texts. Technical treatments that emphasize ecological problems are given by Williams  
1198 et al. (2002), Royle and Dorazio (2008) and Link and Barker (2010), to name just  
1199 a few. A very accessible introduction to some of the topics covered in this chapter  
1200 is presented in Chapt. 3 of MacKenzie et al. (2006). With all these sources, one  
1201 might wonder why we bother rehashing these concepts here. Our motivation is  
1202 two-fold: first, we wish to develop this material using examples relevant to spatial  
1203 capture-recapture, and second, we find that most introductory texts are not accom-  
1204 panied by code that can be helpful to the novice. We therefore attempt to present

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

After covering some basic concepts of hierarchical modeling, we end the chapter by describing spatial capture-recapture models using hierarchical modeling notation. This makes the concepts outlined in the previous chapter more precise, and it provides the basis for describing variations on the basic them of combining models of the ecological process of interest with models describing how the observations are thought to arise conditional on the actual state of the system.

## 2.1 RANDOM VARIABLES AND PROBABILITY DISTRIBUTIONS

### 2.1.1 Stochasticity in ecology

Few ecological processes can be described using purely deterministic models, and thus we need a formal method for drawing conclusions from data while acknowledging the stochastic nature of ecological systems and their observation. This is the role of statistical inference, which is founded on the laws of probability. For our purposes, it suffices to be familiar with a small number of concepts from probability theory. The most important of which is the concept of a random variable, say  $X$ . We wish to know the probability that a realization of  $X$  takes on some value  $x$ ,  $\Pr(X = x)$ , or perhaps the probability that  $x$  lies within some range of values. An approach for achieving such goals is to develop a statistical model for  $X$  and then estimate the parameters of the model by fitting it to data collected during an experiment.

To clarify the concept of a random variable, let  $X$  be the number of American shad (*Alosa sapidissima*) caught after  $K = 20$  casts at the shad hole on Deerfield River in Massachusetts. Suppose that we had a good day and caught  $x = 7$  fish. If there were no random variation at play, we would say that the probability of catching a fish, denoted  $p$ , is  $p = 7/20 = 0.35$ , and therefore might think we would always catch 7 shad after 20 casts. In other words, our deterministic model is  $x = 0.35 \times K$ . In reality, however, we can be pretty sure that this deterministic model would not be very good. Even if we knew for certain  $p \equiv 0.35$ , we would expect some variation in the number of fish caught on repeated fishing outings. To describe this variation, we need a model that acknowledges uncertainty (i.e., stochasticity), and specifically we need a model that describes the probability of catching  $x$  fish given  $K$  and  $p$ ,  $\Pr(X = x|K, p)$ .

To specify a model for  $\Pr(X = x|K, p)$  we need a specific type of function known as a probability mass function (pmf). Or, in the case where  $X$  is a continuous random variable, we need a probability density function (pdf). We will generi-

1244 cally refer to these as probability distributions. Statisticians make things easier  
 1245 for themselves, and more complicated for everyone else, by using different notation  
 1246 for probability distributions. Sometimes you will see  $\Pr(X = x|K, p)$  expressed as  
 1247  $f(X|K, p)$  or  $f(X; K, p)$  or  $p(X|K, p)$  or  $\pi(X|K, p)$  or  $\mathbb{P}(X|K, p)$  or  $[X|K, p]$  or even  
 1248 just  $[X]!$  Just remember that these expressions all have the same meaning—they  
 1249 are all probability distributions that tell us the probability of observing any possible  
 1250 realization of the random variable  $X$ .

1251 In this book, we will almost always use bracket notation (the last two examples  
 1252 above) to represent arbitrary probability distributions. Hence, from here on out,  
 1253 when you see  $[X|K, p]$ , just remember that this is equivalent to the more traditional  
 1254 (but perhaps less popular) expression  $\Pr(X = x|K, p)$ . Note that this probability  
 1255 distribution could be virtually anything: Poisson, Gaussian, beta, etc ... In addition,  
 1256 from here on, to achieve a more concise presentation, we will no longer use  
 1257 uppercase letters to denote random variables and lowercase letters for realized values.  
 1258 Rather, we will define a random variable by some symbol ( $x, N, \theta$ , etc..) and  
 1259 let the context determine whether we are talking about the random variable itself,  
 1260 or realized values of it. In some, limited, cases, we will want upper- and lower-case  
 1261 letters to represent different variables. For example,  $N$  (population size) and  $n$   
 1262 (sample size) is a typical example.

1263 When we wish to be specific about a probability distribution, we will do so in  
 1264 one of two ways, one mathematically precise and one symbolic. Before explaining  
 1265 these two options, let's choose a specific distribution as a model for the data in our  
 1266 example. In this case, the natural choice for  $[x|K, p]$  is the binomial distribution,  
 1267 the mathematically precise representation of which is

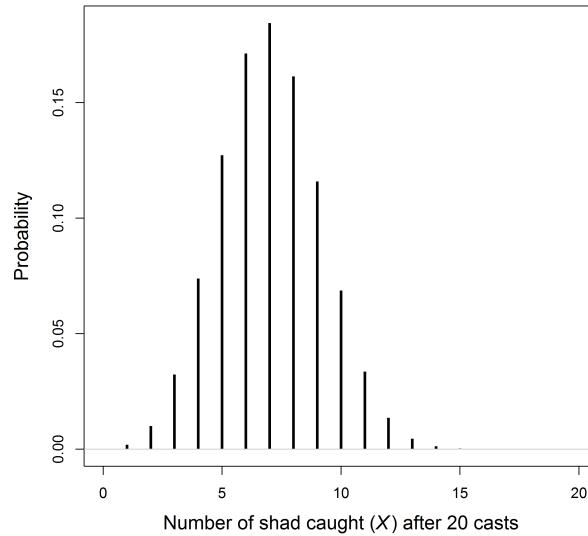
$$[x|K, p] = \binom{x}{K} p^x (1 - p)^{K-x}. \quad (2.1.1)$$

1268 The right-hand side of this equation is the binomial pmf (described in more detail  
 1269 in Sec. 2.2), and plugging in values for the parameters  $K$ , and  $p$  will return the  
 1270 probability of observing any realized value of the random variable  $x$ . This is precise,  
 1271 but it is also cumbersome to write repetitively, and it may make the eyes glaze over  
 1272 when seen too often. Thus, we will often simplify Eq. 2.1.1 using the symbolic  
 1273 notation:

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

1274 The “ $\sim$ ” symbol is meant to represent a stochastic relationship, and can be read  
 1275 “is distributed as.” Another reason for using this notation is that it resembles the  
 1276 syntax of the **BUGS** language, which we will frequently use to conduct Bayesian  
 1277 inference.

1278 Note that once we choose a probability distribution, we have chosen a model. In  
 1279 our example, we have specified our model as  $x \sim \text{Binomial}(K, p)$ , and because we  
 1280 are assuming that the parameters are known, we can make probability statements  
 1281 about future outcomes. Continuing with our fish example, we might want to know



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

what the probability of catching  $x = 7$  again on a future fishing outing assuming that we know  $p = 0.35$ . Evaluating the binomial pmf returns a probability of approximately 0.18, as shown using this bit of **R** code:

```
1285 > dbinom(7, 20, 0.35)
1286 [1] 0.1844012
```

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

```
1290 plot(0:20, dbinom(0:20, 20, 0.35), type="h", ylab="Probability",
1291 xlab="Number of shad caught (X)")
```

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

The purpose of this little example is to show that once we specify a model for the random variable(s) being studied, we can begin drawing conclusions, i.e. making inferences, about the processes of interest, even in the face of uncertainty. Probability distributions are essential to this process, and thus we need to understand 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	Mean	Variance
Discrete random variables				
Poisson	$x \sim \text{Pois}(\lambda)$	$\exp(-\lambda)\lambda^x/x!$	$\lambda$	$\lambda$
Bernoulli	$x \sim \text{Bern}(p)$	$p^x(1-p)^{1-x}$	$p$	$p(1-p)$
Binomial	$x \sim \text{Bin}(N, p)$	$\binom{x}{N}p^x(1-p)^{N-x}$	$Np$	$Np(1-p)$
Multinomial	$\mathbf{x} \sim \text{Multinom}(N, \boldsymbol{\pi})$	$\binom{N}{x_1 \dots x_k} \pi_1^{x_1} \dots \pi_k^{x_k}$	$N\pi_k$	$N\pi_k(1-\pi_k)$
Continuous random variables				
Normal	$x \sim N(\mu, \sigma^2)$	$\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$	$\mu$	$\sigma^2$
Uniform	$x \sim \text{Unif}(a, b)$	$1/(b-a)$	$(a+b)/2$	$(b-a)^2/12$
Beta	$x \sim \text{Beta}(a, b)$	$\frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1}(1-x)^{b-1}$	$a/(a+b)$	$\frac{ab}{(a+b)^2(a+b+1)}$
Gamma	$x \sim \text{Gamma}(a, b)$	$\frac{b^a}{\Gamma(a)} x^{a-1} \exp(-bx)$	$a/b$	$a/b^2$
Multivariate Normal	$\mathbf{x} \sim MVN(\boldsymbol{\mu}, \boldsymbol{\Sigma})$			

## 2.1.2 Properties of Probability Distributions

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

```
1298 > integrate(dnorm, -Inf, Inf, mean=0, sd=1)$value
1299 [1] 1
1300 > sum(dbinom(0:5, size=5, p=0.1))
1301 [1] 1
```

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

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

```
1311 > mean(rbinom(10000, 20, 0.3))
1312 [1] 6.9865
```

1323 For most probability distributions used in this book, the expected values are known  
 1324 exactly, as shown in Table ??, and thus we don't need to resort to such Monte Carlo  
 1325 approximations. For instance, the expected value of the binomial distribution is  
 1326 exactly  $\mathbb{E}(X) = Kp = 20 \times 0.35 = 7$ . In this case, it happens to take an integer  
 1327 value, but this is not a necessary condition, even for discrete random variables.

1328 A more formal definition of an expected value is the average of all possible  
 1329 values of the random variable, weighted by their probabilities. For continuous  
 1330 random variables, this weighted average is found by integration:

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

1331 For example, if  $[x]$  is normally distributed with mean 3 and unit variance, we could  
 1332 find the expected value using the following code.

```
1333 > integrate(function(x) x*dnorm(x, 3, 1), -Inf, Inf)
1334 3 with absolute error < 0.00033
```

1335 Of course, the mean *is* the expected value of the normal distribution, so we didn't  
 1336 need to compute the integral but, the point is, that Eq. 2.1.3 is generic. For  
 1337 discrete random variables, the expected value is found by summation rather than  
 1338 integration:

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

1339 where the summation is over all possible values of  $x$ . Earlier we approximated the  
 1340 expected value of the binomial distribution with  $K = 20$  trials and  $p = 0.35$  by  
 1341 taking a Monte Carlo average. Eq. 2.1.4 let's find the exact answer, using this bit  
 1342 of **R** code:

```
1343 > sum(dbinom(0:100, 20, 0.35)*0:100)
1344 [1] 7
```

1345 This is great. But of what use is it? One very important concept to understand is  
 1346 that when we fit models, we are often modeling changes in the expected value of  
 1347 some random variable. For example, in Poisson regression, we model its expected  
 1348 value, say  $\lambda$ , which may be a function of environmental variables.

1349 The ability to model the expected value of a random variable gets us very far,  
 1350 but we also need sometimes a model for the variance of the random variable. The  
 1351 variance describes the amount of variation around the expected value. Specifically,  
 1352  $\text{Var}(x) = \mathbb{E}((x - \mathbb{E}(x))^2)$ . Clearly, if the variance is zero, the variable is not random  
 1353 as there is no uncertainty in its outcome. For some distributions, notably the  
 1354 normal distribution, the variance is a parameter to be estimated. Thus, in ordinary  
 1355 linear regression, we estimate both the expected value  $\mu = \mathbb{E}(x)$ , which may be  
 1356 a function of covariates, and the variance  $\sigma^2$ , or similarly the residual standard

1357 error  $\sigma$ . For other distributions, the variance is not an explicit parameter to be  
 1358 estimated, and instead, the mean to variance ratio is fixed. In the case of the  
 1359 Poisson distribution, the mean is equal to the variance,  $\mathbb{E}(x) = \text{Var}(x) = \lambda$ . A  
 1360 similar situation is true for the binomial distribution—the variance is determined  
 1361 by the two parameters  $K$  and  $p$ ,  $\text{Var}(x) = Kp(1 - p)$ . Thus in our earlier example  
 1362 with  $K = 20$  and  $p = 0.35$ , the variance is 4.55. Toying around with these ideas  
 1363 using random number generators may be helpful. Here is some code to illustrate  
 1364 some of these basic concepts:

```
1365 > 20*0.35*(1-0.35)          # Exact variance, Var(x)
1366 [1] 4.55
1367 > x <- rbinom(100000, 20, 0.35)
1368 > mean((x-mean(x))^2)      # Monte Carlo approximation
1369 [1] 4.545525
```

## 2.2 COMMON PROBABILITY DISTRIBUTIONS

1370 We got a little ahead of ourselves in the previous sections by using the binomial  
 1371 and Poisson distributions without describing them in detail. A solid understanding  
 1372 of the binomial, Poisson, multinomial, uniform, and normal (or Gaussian) distri-  
 1373 butions is absolutely essential throughout the remainder of the book. We will  
 1374 occasionally make use of other distributions such as the beta, log-normal, gamma,  
 1375 Dirichlet, etc... that can be helpful when modeling capture-recapture data, but  
 1376 these distributions can be readily understood once you are comfortable with the  
 1377 more commonly used distributions described in this section.

### 1378 2.2.1 The Binomial Distribution

1379 The binomial distribution plays a critical role in ecology. It is used for purposes  
 1380 as diverse as modeling count data, survival probability, occurrence probability, and  
 1381 capture probability, just to name a few. To describe the properties of the binomial  
 1382 distribution, and related distributions, we will introduce a new example. Suppose  
 1383 we are conducting a bird survey at a site in which  $N = 10$  chestnut-sided warblers  
 1384 (*Setophaga pensylvanica*) occur, and each of these individuals has a detection prob-  
 1385 ability of  $p = 0.5$ . The binomial distribution is the natural choice for describing  
 1386 the number of individuals that we would expect to detect ( $n$ ) in this situation,  
 1387 and using our notation, we can write the model as:  $n \sim \text{Binomial}(10, 0.5)$ . Note  
 1388 that if  $p \equiv 1$  and we visit the site on  $K$  occasions, the observations  $\{n_1, \dots, n_K\}$   
 1389 would not be random outcomes—we would always observe  $n_k = 10$ . That is, the  
 1390 observed data would exactly equal the expected value and the variance would be  
 1391 zero. However, when  $p < 1$ , we can expect that we will observe a different number  
 1392 of warblers on each replicate visit. To see this, we can simulate data under this  
 1393 simple model with  $K = 3$  survey occasions

```

1394 > n <- rbinom(3, size=10, prob=0.5) # Generate 3 binomial outcomes
1395 > n                                     # Display the 3 values
1396 [1] 6 4 8

```

1397 The vector of counts will typically differ each time you issue this command; however,  
 1398 we know the probability of observing any value of  $n_k$  because it is defined by the  
 1399 binomial pmf. As we demonstrated earlier, in **R** this probability can be found using  
 1400 the **dbinom** function. For example, the probability of observing  $n_k = 5$  is given by:

```
1401 dbinom(5, 10, 0.5)
```

1402 This simply evaluates the function shown in Table 2.1. We could do the same more  
 1403 transparently, but less efficiently, using any of the following:

```

1404 n <- 5; N <- 10; p <- 0.5
1405 factorial(N)/(factorial(n)*factorial(N-n))*p^n*(1-p)^(N-n)
1406 exp(lgamma(N+1) - (lgamma(n+1) + lgamma(N-n+1)))*p^n*(1-p)^(N-n)
1407 choose(N, n)*p^n*(1-p)^(N-n)

```

1408 Note that the last three lines of code differ only in how they compute the binomial  
 1409 coefficient  $\binom{N}{n}$ , which is the number of different ways we could observe  $n = 5$  of  
 1410 the  $N = 10$  chestnut-sided warblers at the site. The binomial coefficient, which is  
 1411 read “ $N$  choose  $n$ ” is defined as

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

1412 Now that we know how to simulate binomial data and compute the probabilities  
 1413 of observing any particular outcome  $n$ , conditional on the parameters  $N$  and  $p$ ,  
 1414 we can contemplate the relevance of the binomial distribution in spatial capture-  
 1415 recapture models. One important application of the binomial distribution is as a  
 1416 model encounter frequencies. Indeed, one of the most important encounter models  
 1417 in SCR will be referred to as the “binomial encounter model”, in which the number  
 1418 of times individual  $i$  is captured at “trap”  $j$  after  $K$  survey occasions is modeled  
 1419 as  $y_{ij} \sim \text{Binomial}(K, p_{ij})$ . Here,  $p_{ij}$  is the encounter probability determined, in  
 1420 part, by the distance between an animal’s activity center and the trap location, as  
 1421 will be described more fully in subsequent chapters. The important point to note  
 1422 is that  $y_{ij} = 0$  if individual  $i$  is never encountered at trap  $j$ , and  $y_{ij} = 4$  if it is  
 1423 encountered on all four occasions. This binomial encounter model is described in  
 1424 detail in Sec. 5.2. Another important application of the binomial distribution is as  
 1425 a prior for the population size parameter in Bayesian analyses, as is discussed in  
 Chapt. 4.

### 1427 2.2.2 The Bernoulli Distribution

1428 Above, we showed three alternatives to **dbinom** for evaluating the binomial pmf.  
 1429 These three commands differed only in how they computed the binomial coefficient,

which we needed because of the numerous ways in which we could observe  $n = 5$  given  $N = 10$ . To conceptualize this, let  $y_i$  be a binary variable indicating if individual  $i$  was detected or not. Hence, given that 5 individuals were detected, the vector of individual detections could be something like  $\mathbf{y} = (0, 0, 1, 1, 1, 1, 1, 0, 0, 0)$ , which would indicate that we detected individuals 3-7 but not 1-2 or 8-10. For  $N = 10$  and  $n = 5$ , the binomial coefficient tells us that there are 252 possibilities of vectors  $\mathbf{y}$  that have 5 ones. However, when  $N \equiv 1$ , this term drops from the pmf and the result is the pmf for the Bernoulli distribution. That is, the Bernoulli distribution is simply the binomial distribution when  $N \equiv 1$ . Alternatively, we could say that the binomial distribution is the outcome of  $N$  iid Bernoulli trials. We use the standard term “*iid*” to mean *independent, identically distributed*.

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

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

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

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

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

---

 1464 **2.2.3 The Multinomial and Categorical Distributions**

1465 The binomial distribution is used when we are accumulating a binary response –  
 1466 that is, one in which there are two possible categories, such as success/failure or  
 1467 captured/not-captured. The multinomial distribution is a multivariate extension of  
 1468 the binomial used when there are  $G > 2$  categories. The multinomial distribution  
 1469 can be thought of as a model for placing  $N$  items in the  $G$  categories, which are  
 1470 also called bins or cells. Each bin has its own probability  $\pi_g$  and these probabilities  
 1471 must sum to one. In ecology,  $N$  is often population size or the number of individuals  
 1472 detected, but the definition of the  $G$  bins varies among applications. For example,  
 1473 in distance sampling, when the distance data are aggregated into intervals, the  
 1474 bins are the distance intervals, and the cell probabilities are functions of detection  
 1475 probability in each interval (Royle et al., 2004).

1476 The multinomial distribution is widely used to model data from traditional,  
 1477 non-spatial capture-recapture studies. Earlier we let  $y_{ik}$  denote a binary random  
 1478 variable indicating if warbler  $i$  was detected on survey  $k$ . The vector of observations  
 1479 for an individual,  $\mathbf{y}_i$ , is often referred to as the individual’s “encounter history”. The  
 1480 number of possible encounter histories depends on the number of survey occasions.  
 1481 Specifically, there are  $2^K$  possible encounter histories<sup>1</sup>. If we tabulate the number  
 1482 of individuals with each encounter history, the frequencies can be modeled using  
 1483 the multinomial distribution.

1484 Going back to our chestnut-sided warbler example, suppose the 10 individuals  
 1485 are marked and we make  $K = 2$  visits to the site such that there are  $2^K = 4$   
 1486 possible encounter histories: (11, 10, 01, 00), where, for example, “10” is the en-  
 1487 counter history for an individual detected on the first visit but not the second. If  
 1488  $p = 1$ , then the encounter history for each of the 10 individuals must be “11”.  
 1489 That is, we would detect each individual on both occasions. In this case, we could  
 1490 format our data as  $\mathbf{h} = (10, 0, 0, 0)$ . The corresponding cell probabilities would be  
 1491  $\boldsymbol{\pi} = (1, 0, 0, 0)$ . What about the situation where  $p < 1$ , e.g.  $p = 0.3$ ? In this case,  
 1492 the probability of observing the capture history “11” (detected on both occasions)  
 1493 is  $p \times p = 0.3 \times 0.3 = 0.09$ . The probability of observing “10” is  $p \times (1 - p) = 0.21$ .  
 1494 Following this logic, the vector of cell probabilities is  $\boldsymbol{\pi} = (0.09, 0.21, 0.21, 0.49)$ .  
 1495 We can simulate data under this model as follows:

```

1496 > caphist.probs <- c("11"=0.09, "10"=0.21, "01"=0.21, "00"=0.49)
1497 > drop(rmultinom(1, 10, caphist.probs))
1498 11 10 01 00
1499 0 3 2 5
  
```

1500 The result of our simulation is that zero individuals were observed with the capture  
 1501 history “11” and 5 individuals were observed with the capture history “00”. The

---

<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  $2^K - 1$

1502 other 5 individuals were observed one out of the two occasions. This is not such a  
 1503 surprising outcome given  $p = 0.3$ .

1504 As in non-spatial capture-recapture studies, the multinomial distribution turns  
 1505 out to be very important in spatial capture-recapture studies. However,  $N$  is not  
 1506 defined as population size. Rather, we use the multinomial distribution when an  
 1507 individual can only be captured in a single trap during an occasion. Thus  $N = 1$   
 1508 and the cell probabilities are the probabilities of being captured in each trap. A  
 1509 thorough discussion of this point can be found in Chapt. 9. Another application  
 1510 of the multinomial distribution in SCR models is discussed in Chapt. 13 where we  
 1511 discuss how to model the probability that an individual's activity center is located  
 1512 in one of the cells of a raster defining the spatial region of interest.

1513 Just as the Bernoulli distribution is the elemental form of the binomial distri-  
 1514 bution (being the case  $N = 1$ ), the categorical distribution is essentially equivalent  
 1515 to the multinomial distribution with size parameter  $N \equiv 1$ . The only difference is  
 1516 that, rather than returning a vector with a single element equal to 1, it returns the  
 1517 element number where the 1 occurs. For example, if  $\mathbf{y} = (0, 0, 1, 0)$  is an outcome  
 1518 of a multinomial distribution with  $N = 1$ , then the categorical outcome would be  
 1519 3 because the 1 is located in third position in the vector. Thus, in spatial capture-  
 1520 recapture models, we might use either the multinomial distribution with  $N = 1$   
 1521 or the categorical distribution. The various **BUGS** engines describe the categori-  
 1522 cal distribution by the declaration `dcat` and, in **R**, we can simulate categorical  
 1523 outcomes using the function `sample`.

#### 1524 2.2.4 The Poisson Distribution

1525 The Poisson distribution is the canonical model for count data in ecology. More  
 1526 generally, the Poisson distribution is a model for random variables taking on non-  
 1527 negative, integer values. Although it is a simple model having just one parameter,  
 1528  $\lambda = \mathbb{E}(x) = \text{Var}(x)$ , its applications are highly diverse, including as a model of  
 1529 spatial variation in abundance or as a model for the frequency of behaviors over  
 1530 time. Just as logistic regression is the standard generalized linear model (GLM)  
 1531 used to model binary data, Poisson regression is the default GLM for modeling  
 1532 count data and variation in  $\lambda$ .

1533 The Poisson distribution is related to both the binomial and multinomial distri-  
 1534 butions, and the following two bits of trivia are occasionally worth knowing. First,  
 1535 it is the limit of the binomial distribution as  $N \rightarrow \infty$  and  $p \rightarrow 0$ , which means that  
 1536 for high values of  $N$  and low values of  $p$ ,  $\text{Poisson}(N \times p)$  is approximately equal  
 1537 to  $\text{Binomial}(N, p)$ . Second, if  $\{n_1 \sim \text{Poisson}(\lambda_1), \dots, n_K \sim \text{Poisson}(\lambda_K)\}$  then the  
 1538 vector of counts is multinomial,  $\{n_1, \dots, n_K\} \sim \text{Multinomial}(\sum_k n_k, \{\frac{\lambda_1}{\sum_k \lambda_k}, \dots, \frac{\lambda_K}{\sum_k \lambda_k}\})$ .

1539 The Poisson distribution has two important uses in spatial capture-recapture  
 1540 models: (1) as a prior distribution for the population size parameter  $N$ , and (2) as  
 1541 a model for the frequency of captures in a trap. In the first context, the Poisson  
 1542 prior for  $N$  results in a Poisson point process for the location of the  $N$  activity

1543 centers in the region of interest. This topic is discussed in-depth in Chapt. 5 and  
 1544 Chapt 13. The second use of the Poisson distribution in spatial capture-recapture  
 1545 is to describe data from sampling methods in which an individual can be detected  
 1546 multiple times at a trap during a single occasion. For example, in camera trapping  
 1547 studies we might obtain multiple pictures of the same individual at a trap during  
 1548 a single sampling occasion. Thus,  $\lambda$  in this case would be defined as the expected  
 1549 number of detections or captures per occasion.

1550 **2.2.5 The Uniform Distribution**

1551 The lowly uniform distribution is a continuous distribution whose only two pa-  
 1552 rameters are the lower and upper bounds that restrict the possible values of the  
 1553 random variable  $x$ . These bounds are almost always known, so there is typically  
 1554 nothing to estimate. Nonetheless, the uniform distribution is one of the most widely  
 1555 used distributions, especially among Bayesians who frequently use it to as a “non-  
 1556 informative” prior distribution for a parameter. For example, if we have a capture  
 1557 probability parameter  $p$  that we wish to estimate, but we have no prior knowl-  
 1558 edge of what value it may take in the range  $[0,1]$ , we will often use the prior  
 1559  $p \sim \text{Uniform}(0,1)$ . This states that  $p$  is equally likely to take on any value between  
 1560 zero and one.

1561 Another common usage of the uniform distribution is as a prior for the coor-  
 1562 dinates of points in the real plane, i.e. in two-dimensional space. Such a use of  
 1563 the uniform distribution implies that a point process is “homogeneous”, meaning  
 1564 that the location of one point does not affect the location of another point and  
 1565 that the expected density of points is constant throughout the region. Thus, to  
 1566 simulate a realization from a homogeneous Poisson point process in the unit square  
 1567  $[0, 1] \times [0, 1]$ , we could use the following R code:

```
1568 D <- 100    # points per unit area
1569 A <- 1      # Area of unit square
1570 N <- rpois(1, D*A)
1571 plot(s <- cbind(runif(N), runif(N)))
```

1572 where  $s$  is a matrix of coordinates with  $N$  rows and 2 columns. We will often  
 1573 represent the uniform point process using the following notation:

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

1574 where  $\mathcal{S}$  is some specific unit of space called the state-space of the random variable  
 1575  $s$ . It would be more correct to somehow distinguish this two-dimensional uniform  
 1576 distribution for the univariate one. That is, it might be more clear to use notation  
 1577 such as  $s \sim \text{Uniform}_2(\mathcal{S})$  instead, but this is somewhat cumbersome, so we will opt  
 1578 for the former expression.

---

### 1579    2.2.6 Other Distributions

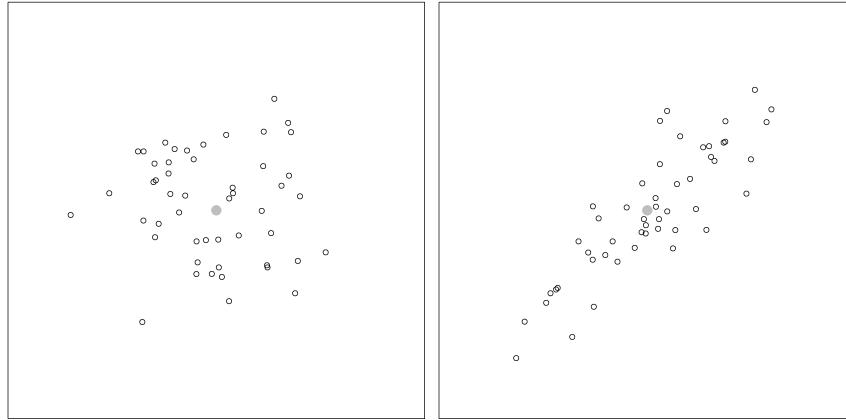
1580    The other continuous distributions that are regularly encountered in SCR models  
 1581    are primarily used as priors in Bayesian analyses, and thus we will avoid a lengthy  
 1582    discussion of their properties. The normal distribution, also called the Gaussian  
 1583    distribution, is perhaps the most widely recognized and applied probability model  
 1584    in statistics, but it plays only a minor role in SCR models although it has been  
 1585    used as a model for signal strength in acoustic SCR models (?Dawson and Efford,  
 1586    2009), and see Sec. 9.5. Nonetheless, it is the canonical prior for any continuous  
 1587    random variable with infinite support, and thus it is often used as a prior when  
 1588    applying Bayesian methods. One common usage is as a prior for the  $\beta$  coefficients  
 1589    of a linear model defining some parameter as a function of covariates (usually on a  
 1590    transformed scale). An example, including a cautionary note, is provided in Sec. ??.  
 1591    Be aware that although the normal distribution is typically parameterized in terms  
 1592    of the variance parameter  $\sigma^2$ , in the **BUGS** language, the inverse of the variance,  
 1593    or precision, is used instead,  $\tau = 1/\sigma^2$ .

1594    The bivariate normal distribution is a generalization of the normal distribution  
 1595    and a special case of the multivariate normal distribution whose pdf is shown in  
 1596    Table 2.1. The bivariate normal distribution is used to model two (possibly) depen-  
 1597    dent continuous variables whose symmetric variance-covariance matrix is denoted  
 1598     $\Sigma$ . In SCR models, we most often use this model as a rudimentary description of  
 1599    movement outcomes about a home range center. If there is no correlation,  
 1600    then the model reduces to two independent normal draws along the coordinate  
 1601    axes. The following code generates bivariate normal outcomes with no correlation  
 1602    ( $\rho = 0$ , the object `X1`) and with  $\rho = 0.9$ , the object `X2`.

```
1603 library(mvtnorm)
1604 set.seed(3)
1605 mu <- c(0,0)
1606 Sigma <- matrix(c(1, .9, .9, 1), 2, 2)
1607 X1 <- cbind(rnorm(50, mu[1], Sigma[1,1]), # No correlation (rho=0)
1608           rnorm(50, mu[2], Sigma[2,2]))
1609 X2 <- rmvnorm(50, mu, Sigma) # rho=0.9
```

1610 Fig. 2.2 shows the simulated points.

1611 Several of the parameters in capture-recapture models do not have infinite sup-  
 1612 port, but are instead are probabilities restricted to the range  $[0, 1]$ , or are positive  
 1613 valued living between zero and  $\infty$ . The beta distribution is the standard prior  
 1614 used for probabilities because it can be used to express either a lack of knowledge  
 1615 or very precise knowledge about a parameter. For example, a Beta(1, 1) distribu-  
 1616 tion is equivalent to a Uniform(0, 1) distribution. However, unlike the uniform  
 1617 distribution, the beta distribution can be used as an informative prior; for exam-  
 1618 ple if published estimates of detection probability exist we can choose parameters  
 1619 of the beta distribution to reflect that. To gain some familiarity with the beta



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

1620 distribution, execute the following **R** commands:

```
1621 curve(dbeta(x, 1, 1), col="black", ylim=c(0,5))
1622 curve(dbeta(x, 10, 10), col="blue", add=TRUE)
1623 curve(dbeta(x, 10, 20), col="darkgreen", add=TRUE)
```

1624 Other parameters in SCR models are continuous but positive-valued and can be  
 1625 modeled using the gamma distribution. As with the beta distribution, the gamma  
 1626 distribution is typically favored over the uniform distribution when one is interested  
 1627 in using an informative prior. It is also frequently used as a vague prior for the  
 1628 inverse of variance parameters, but it is wise to compare this prior to a uniform to  
 1629 assess its influence on the posterior.

### 2.3 STATISTICAL INFERENCE AND PARAMETER ESTIMATION

1630 If the parameters of a statistical model were known with absolute certainty, then it  
 1631 would be possible to use pdfs and pmfs to make direct probability statements about  
 1632 unknowns such as future outcomes. However, we almost never know the actual  
 1633 values of parameters, and instead we have to estimate them from observations  
 1634 (i.e., data). Our inferences must then acknowledge the uncertainty associated with  
 1635 our imperfect knowledge of the parameters. Doing so is most often accomplished  
 1636 using one of two approaches: classical (frequentist) inference or Bayesian inference.  
 1637 These two modes of inference regard the uncertainty about parameters in entirely  
 1638 different ways. In the next chapter, we will review some of the important concepts  
 1639 in Bayesian inference, so here, we will focus on the frequentist perspective.

1640 Suppose we count oak trees at  $J$  sites, and the resulting data  $\{y_1, \dots, y_J\}$  can  
 1641 be assumed to be iid outcomes from some distribution, such as the Poisson with

1642 unknown parameter  $\lambda$ . We want to estimate this parameter. In classical inference,  
 1643 the only uncertainty about  $\lambda$  is that attributable to sampling. For instance, we can  
 1644 imagine repeatedly sampling the population (sites in this example) and obtaining  
 1645 sample-specific estimates of  $\lambda$ . Typically, we entertain the idea that there are an  
 1646 infinite number of possible samples and so we could obtain an infinite number of  
 1647 estimates:  $\{\hat{\lambda}_1, \hat{\lambda}_2, \dots, \hat{\lambda}_\infty\}$ . If these estimates are produced using the method of  
 1648 maximum likelihood, the distribution of estimates, called the sampling distribution,  
 1649 will be normally distributed with  $\mathbb{E}(\hat{\lambda}) = \lambda$ . The standard deviation of the sampling  
 1650 distribution is called the standard error, which can be estimated as part of the  
 1651 maximum likelihood procedure. Given  $\hat{\lambda}$  and an estimate of its standard deviation,  
 1652 we can construct a confidence interval that will include the true value of  $\lambda$  with  
 1653 some prescribed coverage probability. Note that there is no uncertainty associated  
 1654 with the actual parameter—it is regarded as a fixed value, and hence probability  
 1655 is only used to characterize the estimator via its sampling distribution.

1656 Maximum likelihood is heuristically a method of finding the most “likely” value  
 1657 of  $\lambda$ , given the observed data, and of characterizing the variance of the sampling dis-  
 1658 tribution. Of course, it also applies to cases where the observations are multivariate,  
 1659 or the probability distribution is a function of multiple parameters. Endless num-  
 1660 bers of textbooks and online resources are available for those interested in a detailed  
 1661 explanation of maximum likelihood. For our purposes, we wish to keep it simple  
 1662 and focus on *how* to do it. The first step is to define the likelihood function, which  
 1663 is the joint distribution of the data regarded as a function of the parameter(s). If  
 1664 the joint distribution of the observations is denoted by  $[y_1, y_2, \dots, y_n | \lambda]$ , we usually  
 1665 denote the likelihood by flipping the arguments:  $\mathcal{L}(\lambda | \mathbf{y}) = [\lambda | y_1, y_2, \dots, y_n]$ .

1666 If the observations are *iid*, the likelihood simplifies to

$$\mathcal{L}(\lambda | \mathbf{y}) = \prod_i [y_i | \lambda]. \quad (2.3.1)$$

1667 where  $[y_i | \lambda]$  is a probability distribution, like those discussed in the previous sec-  
 1668 tions. For example, if  $y_i$  is Poisson distributed, then  $[y_i | \lambda] = \text{Poisson}(\lambda) = \frac{\lambda^{y_i} e^{-\lambda}}{y_i!}$ .  
 1669 Although likelihoods are typically shown on the natural scale, we almost always  
 1670 maximize the logarithm of the likelihood to avoid computational problems that  
 1671 arise when multiplying very small probabilities. Thus, we rewrite 2.3.1 as

$$\ell(\lambda | \mathbf{y}) = \sum_i \log(f(y_i | \lambda)) \quad (2.3.2)$$

1672 Here is some simple **R** code to simulate independent Poisson outcomes and esti-  
 1673 mate  $\lambda$  (as though we did not know it) using the method of maximum likelihood.  
 1674 Actually, we will minimize the negative log-likelihood because it is equivalent and  
 1675 is the default for **R**’s optimizers like `optim` and `nlm`.

```
1676 > lambda <- 3 # Actual parameter value
```

```

1677 > y1 <- rpois(100, lambda)    # Realized values (data)
1678 > negLogLike1 <- function(par) -sum(dpois(y1, par, log=TRUE))
1679 > starting.value <- c('lambda'=1)
1680 > optim(starting.value, negLogLike1)$par # MLE
1681   lambda
1682 3.039844
1683 Explicitly maximizing the likelihood, numerically, isn't actually necessary here be-
1684 cause the MLE of  $\lambda$  is given by the mean of the observations. A more interesting
1685 example is when there are covariates of  $\lambda$ . For example, suppose  $\lambda$  is a function of
1686 elevation and vegetation height according to:  $\log(\lambda_i) = \beta_0 + \beta_1 \text{ELEV}_i + \beta_2 \text{VEGHT}_i$ .
1687 This is a standard Poisson regression problem, with likelihood:
```

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

1688 This likelihood is almost identical to the previous one except that  $\lambda$  is now a  
1689 function, and so we need to estimate the parameters of the function, i.e. the  $\beta$ 's.  
1690 Some code to fit this model to simulated data is shown here:

```

1691 > nsites <- 100
1692 > elevation <- rnorm(100)
1693 > veght <- rnorm(100)
1694 > beta0 <- 1
1695 > beta1 <- -1
1696 > beta2 <- 0
1697 > lambda <- exp(beta0 + beta1*elevation + beta2*vegght)
1698 > y2 <- rpois(nsites, lambda)
1699 > negLogLike2 <- function(pars) {
1700   +   beta0 <- pars[1]
1701   +   beta1 <- pars[2]
1702   +   beta2 <- pars[3]
1703   +   lambda <- exp(beta0 + beta1*elevation + beta2*vegght)
1704   +   -sum(dpois(y2, lambda, log=TRUE))
1705 }
1706 > starting.values <- c('beta0'=0, 'beta1'=0, 'beta2'=0)
1707 > optim(starting.values, negLogLike2)$par
1708   beta0      beta1      beta2
1709 0.98457756 -1.03025173 -0.01218292
```

1710 We see that the maximum likelihood estimates (MLEs) are very close to the true  
1711 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:

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

1712 This is an example of a mixed effects model or a hierarchical model. How do  
 1713 we estimate the parameters of a model that includes random effects? Earlier the  
 1714 likelihood function was written as the product of probabilities determined by a  
 1715 single pmf or pdf,  $[y|\lambda]$ , but now we have an additional random variable, and we  
 1716 are forced to think about conditional relationships, because  $y$  depends upon  $\beta_i$  and  
 1717  $\beta_i$  depends upon other parameters, specifically  $\mu$  and  $\sigma^2$ . This type of conditional  
 1718 dependence among parameters is the essence of hierarchical models, and statistical  
 1719 analysis of hierarchical models requires that we discuss joint distributions, marginal  
 1720 distributions and conditional distributions.

## 2.4 JOINT, MARGINAL, AND CONDITIONAL DISTRIBUTIONS

1721 So far we have restricted our attention to situations in which we wish to make  
 1722 inference about a single random variable. However, in ecology, we often are inter-  
 1723 ested in multiple random variables and how they are related. Let  $Y$  be a random  
 1724 variable that may or may not be independent of  $X$  (here again we will distinguish  
 1725 between random variables and realized values for conceptual clarity). Inference  
 1726 about these two random variables can be made using the joint, marginal, or condi-  
 1727 tional distributions—or, we may make use of all of them depending on the question  
 1728 being asked. In the case of discrete random variables, the joint distribution is the  
 1729 probability that  $X$  takes on the value  $x$  and that  $Y$  takes on the value  $y$ , which  
 1730 is written  $[X = x, Y = y]$ . To clarify this concept, let's go back to our original  
 1731 example where  $X$  was the number of fish caught after 20 casts, which we said  
 1732 was an *iid* binomial random variable. Now, let's suppose that  $X$  depends on the  
 1733 random variable  $Y$ , which is the number of other fisherman at the hole. Specifi-  
 1734 cally, let's say that the probability of catching a fish  $p$  is related to  $X$  according  
 1735 to  $\text{logit}(p) = -0.6 + -2y$ . Furthermore, let's make the intuitive assumption that  
 1736 the number of fishermen at the hole is a Poisson random variable with mean 0.6,  
 1737 i.e.  $X \sim \text{Poisson}(0.6)$ . Our model is now fully specified, and so we can answer the  
 1738 question: “what is the probability of catching  $x$  fish and of there being  $y$  fishermen  
 1739 at the hole”. This joint distribution is given by the product of the binomial pmf  
 1740 (with  $p$  determined by  $y$ ) and the Poisson pmf with  $\lambda = 0.6$ . The following R code  
 1741 creates the joint distribution.

```
1742 > X <- 0:20 # All possible values of X
1743 > Y <- 0:10 # All possible values of Y
```

```

1744 > lambda <- 0.6
1745 > p <- plogis(-0.62 + -2*Y) # p as function of Y
1746 > round(p,2)
1747 [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
1748 > joint <- matrix(NA, length(X), length(Y))
1749 > rownames(joint) <- paste("X=", X, sep="")
1750 > colnames(joint) <- paste("Y=", Y, sep="")
1751 >
1752 > # Joint distribution [X,Y]
1753 > for(i in 1:length(Y)) {
1754 +   joint[,i] <- dbinom(X, 20, p[i]) * dpois(Y[i], lambda)
1755 +
1756 > round(joint,2)
1757     Y=0  Y=1  Y=2  Y=3  Y=4  Y=5  Y=6  Y=7  Y=8  Y=9  Y=10
1758 X=0  0.00 0.08 0.08 0.02  0  0  0  0  0  0  0
1759 X=1  0.00 0.12 0.02 0.00  0  0  0  0  0  0  0
1760 X=2  0.01 0.08 0.00 0.00  0  0  0  0  0  0  0
1761 X=3  0.02 0.04 0.00 0.00  0  0  0  0  0  0  0
1762 X=4  0.04 0.01 0.00 0.00  0  0  0  0  0  0  0
1763 X=5  0.07 0.00 0.00 0.00  0  0  0  0  0  0  0
1764 X=6  0.09 0.00 0.00 0.00  0  0  0  0  0  0  0
1765 X=7  0.10 0.00 0.00 0.00  0  0  0  0  0  0  0
1766 X=8  0.09 0.00 0.00 0.00  0  0  0  0  0  0  0
1767 X=9  0.06 0.00 0.00 0.00  0  0  0  0  0  0  0
1768 X=10 0.04 0.00 0.00 0.00  0  0  0  0  0  0  0
1769 X=11 0.02 0.00 0.00 0.00  0  0  0  0  0  0  0
1770 X=12 0.01 0.00 0.00 0.00  0  0  0  0  0  0  0
1771 X=13 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
1772 X=14 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
1773 X=15 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
1774 X=16 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
1775 X=17 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
1776 X=18 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
1777 X=19 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
1778 X=20 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0

```

1779 This matrix tells us the probability of all possible combinations of  $x$  and  $y$ , and  
 1780 we see that the most likely value is  $(X = 1, Y = 1)$ , i.e. we will catch 1 fish and  
 1781 there will be 1 other fisherman. This matrix also demonstrates the law of total  
 1782 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:

```
1788 > margX <- rowSums(joint)
1789 > round(margX, 2)
1790   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
1791 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
1792 X=15 X=16 X=17 X=18 X=19 X=20
1793 0.00 0.00 0.00 0.00 0.00
```

1794 Bad news—the most likely value is  $X = 0$ . However, the chances of catching 1 fish  
1795 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:

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

1796 That is, the conditional distribution of  $X$  given  $Y$  is the joint distribution divided  
1797 by the marginal distribution of  $Y$ .

```
1798 > XgivenY <- joint/matrix(margY, nrow(joint), ncol(joint), byrow=TRUE)
1799 > round(XgivenY, 2)
1800   Y=0  Y=1  Y=2  Y=3  Y=4  Y=5  Y=6  Y=7  Y=8  Y=9  Y=10
1801 X=0  0.00 0.25 0.82 0.97  1  1  1  1  1  1  1
1802 X=1  0.00 0.36 0.16 0.03  0  0  0  0  0  0  0
1803 X=2  0.01 0.25 0.02 0.00  0  0  0  0  0  0  0
1804 X=3  0.03 0.11 0.00 0.00  0  0  0  0  0  0  0
```

---

1805	X=4	0.07	0.03	0.00	0.00	0	0	0	0	0	0	0
1806	X=5	0.13	0.01	0.00	0.00	0	0	0	0	0	0	0
1807	X=6	0.17	0.00	0.00	0.00	0	0	0	0	0	0	0
1808	X=7	0.18	0.00	0.00	0.00	0	0	0	0	0	0	0
1809	X=8	0.16	0.00	0.00	0.00	0	0	0	0	0	0	0
1810	X=9	0.12	0.00	0.00	0.00	0	0	0	0	0	0	0
1811	X=10	0.07	0.00	0.00	0.00	0	0	0	0	0	0	0
1812	X=11	0.03	0.00	0.00	0.00	0	0	0	0	0	0	0
1813	X=12	0.01	0.00	0.00	0.00	0	0	0	0	0	0	0
1814	X=13	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
1815	X=14	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
1816	X=15	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
1817	X=16	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
1818	X=17	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
1819	X=18	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
1820	X=19	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
1821	X=20	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0

1822 Note that we have 11 probability distributions for  $X$ , one for each possible value of  
 1823  $Y$ , and each pmf sums to unity as it should. Note also that if you show up at the  
 1824 hole and there are  $> 2$  fisherman, your chance of catching a fish is very low. Go  
 1825 home. These concepts are explained in more detail in other texts such as Casella  
 1826 and Berger (2002) and Link and Barker (2010), but hopefully, the code shown here  
 1827 complements the equations and makes it easier for non-statisticians to understand  
 1828 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:

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

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

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

1829 From here on out, when you see such notation, you should immediately grasp  
 1830 the fact that  $Y$  is a random variable independent of  $X$ , but  $X$  depends upon  
 1831  $Y$  through  $p$ . Now you have the tools to make probability statements about the  
 1832 random variables in this system. The one caveat faced in reality is that we typically  
 1833 do not know the values of the parameters, and instead we have to estimate them.

## 2.5 HIERARCHICAL MODELS AND INFERENCE

1834 The term hierarchical modeling (or hierarchical model) has become something of  
 1835 a buzzword over the last decade with hundreds of papers published in ecological  
 1836 journals using that term. So then, what exactly is a hierarchical model, anyhow?  
 1837 Obviously, this term stems from the root “hierarchy” which means:

1838 Definition: *hierarchy* (noun) – a series of ordered groupings of people or things  
 1839 within a system;

1840 In the case of a hierarchical model (hierarchical being the adjective form of hi-  
 1841 erarchy), the “things” are probability distributions, and they are ordered according  
 1842 to their conditional probability structure. Thus, a hierarchical model is *an ordered*  
 1843 *series of models, ordered by their conditional probability structure.*

1844 If we declare that the random variable  $y = \text{number of times an individual is}$   
 1845  $\text{encountered in a trap out of } K = 10 \text{ days}$  has a  $\text{Binomial}(10, p)$  distribution then  
 1846 this is but a single model and, thus, not a hierarchical model. If, however, we  
 1847 declare that

$$y \sim \text{Binomial}(10, p)$$

1848 and

$$p \sim \text{Beta}(1, 1)$$

1849 which is the same as the previous model but with a “flat” prior distribution on  
 1850  $p$ , then this is kind of a cheap pedestrian hierarchical model according to our def-  
 1851 inition although it is barely more interesting than the previous non-hierarchical  
 1852 model. On the other hand, suppose we have some meaningful group structure in  
 1853 this problem such that the data arise by observing repeated Bernoulli trials on *in-*  
 1854 *dividuals*, e.g., they are eggs hatching from a common nest (or parentage). So let  
 1855  $y_i$  be the outcomes for individuals  $i = 1, 2, \dots, N$  with

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

1856 and

$$p_i \sim \text{Beta}(\mu, \tau).$$

1857 Because of the meaningful group structure, this is a more interesting hierarchical  
 1858 model. In fact, in the context of capture-recapture this is a specific version of  
 1859 “model  $M_h$ ” (see Chapt. 4 and Dorazio and Royle (2003)). We should consider  
 1860 this a type of a hierarchical model although we will make a further conceptual  
 1861 distinction shortly that further dichotomizes the space of hierarchical models.

1862 A canonical hierarchical model in ecology is this elemental model of species  
 1863 occurrence or distribution (MacKenzie et al., 2002; Tyre et al., 2003; Kéry, 2011):

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

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

1865 where  $y_i = \text{observation of presence/absence at a site } i$  and  $z_i = \text{occurrence status}$   
 1866 ( $z_i = 1$  if a species occurs at site  $i$  and  $z_i = 0$  if not). This model has an important  
 1867 conceptual distinction between the hierarchical model shown just previously (model  
 1868  $M_h$ ) and also other types of classical multi-level models such as repeated measures  
 1869 on subjects, in that  $z_i$  is an actual state of nature. In that sense,  $z$  is a random

variable that is the outcome of a “real” process. Royle and Dorazio (2008) used the term *explicit* hierarchical model to describe this type of model to distinguish from hierarchical models (*implicit* hierarchical models) where the latent variables don’t correspond to an actual state of nature – but rather just soak up variation that is unmodeled by explicit elements of the model. At best, latent variables in such models are surrogates for something of ecological relevance (“time effects”, “space effects” etc.).

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

1879

1880 **Definition: Hierarchical Model:** A model with explicit component models that de-  
1881 scribe variation in the data due to (spatial/temporal) variation in *ecological process*,  
1882 and due to *imperfect observation* of the process.

### 1883 **2.5.1 Spatial Capture-Recapture models as hierarchical models**

1884 Most models considered in this book describe the encounter of individuals condi-  
1885 tional on the “activity center” of the individual, which is a latent variable (i.e.,  
1886 unobserved random effect). The definition of an activity center will be context-  
1887 dependent as discussed in Chapt. ??, but often it can be thought of as an indi-  
1888 vidual’s home range center. The collection of these latent variables represents the  
1889 outcome of an ecological process describing how individuals distribute themselves  
1890 over the landscape. Moreover, how individuals are encountered in traps is, in some  
1891 cases, the result of a model governing movement. As such, these models are ex-  
1892 amples of hierarchical models that contain formal model components representing  
1893 both ecological process and also the observation of that process. That is, they  
1894 are explicit hierarchical models (Royle and Dorazio, 2008) as opposed to implicit  
1895 hierarchical models.

## 2.6 CHARACTERIZATION OF SCR MODELS

1896 For the purposes of this book, an SCR model is any “individual encounter model”  
1897 (not just “capture-recapture”!) where auxiliary spatial information is also obtained.  
1898 To be more precise we could as well use the term “Spatial capture and/or recap-  
1899 ture” but that is slightly unwieldy and, besides, it also abbreviates to SCR. The  
1900 class of SCR models includes traditional capture-recapture models with auxiliary  
1901 spatial information and even some models that do not even require “recapture”  
1902 (e.g., distance sampling). There is even a class of models (Chapt. 18) which don’t  
1903 require capture or unique identification of individuals.

1904 Conceptually, SCR models involve a collection of random variables,  $\mathbf{s}$ ,  $\mathbf{u}$  and  $y$   
1905 where  $\mathbf{s}$  is the activity or home range center,  $\mathbf{u}$  is the location of the individual at  
1906 the time of sampling, which we may think of as a realization from some movement

model, and  $y$  is the “response variable”—what the observer records. For example,  $y = 1$  means “detected” and  $y = 0$  means “not detected”, but many other types of responses are possible (Chapt 9). A broad class of models for estimating density are unified by a hierarchical model involving explicit models for animal activity centers  $\mathbf{s}$ , movement outcomes  $\mathbf{u}$ , and encounter data  $y$ . In some cases, we don’t observe  $y$  but rather summaries of  $y$ , say  $n(y)$ , yet it might be convenient in such cases to retain an explicit focus on  $y$  in terms of model construction. We thus introduce a sequence of models—a hierarchical model—to relate these random variables, which can be written as

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

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

Let’s begin with the model  $[\mathbf{s}]$  that describes the distribution of the activity centers of each animal in the spatial region  $\mathcal{S}$  (the state-space as we called it previously). As will be explained in Chapt. 5 and Chapt. 13,  $[\mathbf{s}]$  defines a spatial point process, which may be inhomogeneous if there exists spatial variation in density, or it may be homogeneous if density is constant throughout  $\mathcal{S}$ . In the later case, we can write  $[\mathbf{s}] = \text{Uniform}(\mathcal{S})$ , which is to say that the  $N$  activity centers are uniformly distributed in the polygon  $\mathcal{S}$ . A point process is also a model for the number of individuals in the population  $N$ . So we could write  $[\mathbf{s}|\mu]$  where  $\mu$  is an intensity parameter defined as the number of points per unit area. In other words,  $\mu$  is population density, and we often model population size as either  $N \sim \text{Poisson}(\mu A(\mathcal{S}))$ , where  $A(\mathcal{S})$  is the area of the state-space; or,  $N \sim \text{Binomial}(M, \psi)$  where  $\psi = \mu A(\mathcal{S})/M$  and  $M$  is some large integer used simply as a convenience measure when conducting Bayesian analysis. As it turns out, there is very little practical difference in the Poisson prior versus a binomial models for  $N$  (Chapt. 13).

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

The third component of the model,  $[y|\mathbf{u}]$ , describes how the observed data—the so-called capture-histories—arise conditional on the locations of animals. However, as mentioned previously, most SCR models do not contain a movement model, and thus, we typically entertain the model  $[y|\mathbf{s}]$  instead of  $[y|\mathbf{u}]$ . This encounter model

1948 generally has at least two parameters, say  $p_0$  and  $\sigma$ , describing the probability of  
 1949 capturing or detecting an individual given the distance between  $\mathbf{s}$  and the trap.  
 1950 The most basic model is often called the half-normal model, although we typically  
 1951 refer to it as the Gaussian model since, in two-dimensional space, it is the kernel  
 1952 of a bivariate normal distribution. The model is  $p_{ij} = p_0 \exp(-\|\mathbf{x}_j - \mathbf{s}_i\|/(2\sigma^2))$   
 1953 where  $p_0$  is the capture probability when the activity center occurs at the trap  
 1954 location  $\mathbf{x}_j$ , and  $\sigma$  is a spatial scale parameter determining how rapidly capture  
 1955 probability declines with distance. One common design leads to the model  $[y_{ij}|\mathbf{s}_i] =$   
 1956 Bernoulli( $p_{ij}$ ). Chapt. 5 and Chapt. 9 describe many other possible encounter  
 1957 models.

1958 When individuals are marked by biologists or have natural markings permitting  
 1959 individual recognition,  $y_{ij}$  is the observed data. However, some or all of the  
 1960 individuals cannot be uniquely identified, then we cannot record this individual-  
 1961 specific encounter history data. Instead, the data might be simply the number of  
 1962 detections at a trap or perhaps binary detection/non-detection data at each trap on  
 1963 each survey occasion. We call this reduced information data  $n(y)$ , and Chapt. ??  
 1964 and Chapt. ?? describe models for  $[n(y)|y]$  that still allow for density estimation.  
 1965 The basic strategy is to view  $y$  as “missing data” and to use the spatial correlation  
 1966 in the counts, or other sources of information, to provide information about these  
 1967 latent encounter histories.

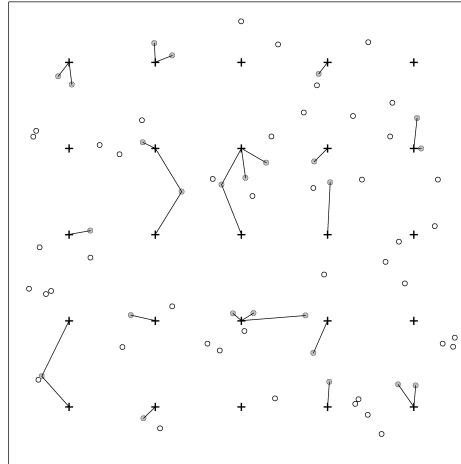
Eq. 2.6.1 is a compact description of the the basic components of a SCR model,  
 but it is also rather vague. The previous four paragraphs added enough extra detail  
 so that we can now describe a specific SCR model. Perhaps the simplest SCR model  
 is this:

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

1968 These “assumptions” are statistical statements of three basic hypotheses that (1)  
 1969 population size  $N$  is Poisson distributed (2) activity centers are uniformly dis-  
 1970 tributed in two-dimensional space, and (3) capture probability is a funciton of the  
 1971 distance between the activity and the trap. Each of these model components can  
 1972 be modified as needed to match specific hypotheses, study designs, and data struc-  
 1973 tures. For example, spatial variation in abundance or density can be easily modeled  
 1974 as a function of habitat covariates (Chapt. 13).

1975 We realize that many the model description in Eq. 2.6.2 may not be self-evident  
 1976 to some ecologists. However, it is absolutely essential that one can understand  
 1977 such a model description—not just for being able to read this book, but also for  
 1978 understanding any statistical model in ecology. One of the best ways of familiarizing  
 1979 oneself with this notation is to translate it into **R** code that simulates outcomes  
 1980 from the model. The following code is an example.

```
1981 set.seed(36372)
```



**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.

```

1982 Area <- 1                                # area of state-space (unit square)
1983 x <- cbind(rep(seq(.1,.9,.2), each=5),
1984           rep(seq(.1,.9,.2), times=5))
1985 p0 <- 0.3                                # baseline capture probability
1986 sigma <- 0.05                             # Gaussian scale parameter
1987 mu <- 50                                 # population density
1988 N <- rpois(1, mu*Area)                   # population size
1989 s <- cbind(runif(N, 0, 1),
1990           runif(N, 0, 1))                    # activity centers in unit square
1991 K <- 5
1992 y <- matrix(NA, N, nrow(x))              # capture data
1993 for(i in 1:N) {
1994   d.ij <- sqrt((x[,1] - s[i,1])^2 +
1995             (x[,2] - s[i,2])^2)               # distance between x and s[i]
1996   p.ij <- p0*exp(-d.ij^2 / (2*sigma^2)) # capture probability
1997   y[i,] <- rbinom(nrow(x), K, p.ij)      # capture history for animal i
1998 }

```

1999 Fig. 2.3 shows the results of this simulation from a basic, yet very useful, SCR  
2000 model.

---

**2.6.1 Variations on the SCR theme**

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

1. **Distance sampling.** The last 2 stages of the hierarchy are confounded (implicitly) and so analysis is based on the model  $[y|\mathbf{u}^*][\mathbf{u}^*]$ . The “process model” is that of “uniformity”:  $\mathbf{u}^* \sim \text{Uniform}(\mathcal{S})$ .
2. **Spatial capture-recapture model with a fixed array of traps.** SCR models appear to have little in common with distance sampling because observations are made only at a pre-defined set of discrete locations—where traps are placed. However, the models are closely related in terms of our hierarchical representation above. In SCR models based on fixed arrays, we cannot estimate both  $\Pr(y = 1|\mathbf{u})$  and  $\Pr(\mathbf{u}|\mathbf{s})$ —the probability that an individual “moves to  $\mathbf{u}$ ” cannot be separated from the probability that it is detected given that it moves to  $\mathbf{u}$ , because of the fact that the observation locations are fixed by design. Formally, such SCR models confound  $[y|\mathbf{u}]$  with  $[\mathbf{u}|\mathbf{s}]$  so that the observation model arises as:

$$[y|\mathbf{s}] = \int_{\mathbf{u}} [y|\mathbf{u}][\mathbf{u}|\mathbf{s}]d\mathbf{u}$$

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

3. **Search-encounter models.** What we call “search-encounter” models (Royle and Young, 2008; Royle et al., 2011a) are kind of a hybrid model combining features of SCR models and features of distance sampling. Like distance sampling they allow for encounters in continuous space which provide direct observations from  $[\mathbf{u}|\mathbf{s}]$ . Thus, the hierarchical model is fully identified.

## 2.7 SUMMARY AND OUTLOOK

Spatial capture-recapture models are hierarchical models, and hierarchical models are models of multiple random variables that are conditionally related. It is there-

2037 fore important that the basic rules of modeling random variables are understood,  
2038 and we hope that this chapter has made some of the basic concepts accessible to  
2039 ecologists with rudimentary background in statistics. If some of this material still  
2040 seems difficult to grasp, we recommend working with the provided **R** code, which  
2041 is perhaps the best way of making the equations more tangible.

2042 In some respects, it is possible to understand the jist of SCR without knowing  
2043 anything about marginal and conditional relationships. One can always fit models  
2044 using canned software and interpret the output without understanding the guts of  
2045 the model or the details of the estimation process. For some applied ecologists,  
2046 this may be perfectly fine, and this book is meant to be useful for both statistical  
2047 novices and ecologists with more advanced quantitative skills. In most chapters, we  
2048 begin with a basic conceptual discussion, then we explain the technical details that  
2049 require an understanding of the concepts in this chapter, and finally we end with  
2050 one or more worked examples. For those not interested in the technical details,  
2051 we recommend focusing on the chapter introductions and the examples. However,  
2052 taking the time to understand the concepts presented in this chapter can only  
2053 increase one's ability to tackle the unique and complex problems that often present  
2054 themselves when modeling spatial and temporal aspects of population dynamics.



## GLMS AND BAYESIAN ANALYSIS

2058 A major theme of this book is that spatial capture-recapture models are, for the  
2059 most part, just generalized linear models (GLMs) wherein the covariate, distance  
2060 between trap and home range center, is partially or fully unobserved – and therefore  
2061 regarded as a random effect. Outside of capture-recapture, such models are usually  
2062 referred to as Generalized Linear Mixed Models (GLMMs) and, therefore, SCR  
2063 models can be thought of as a specialized type of GLMM. Naturally then, we should  
2064 consider analysis of these slightly simpler models in order to gain some experience  
2065 and, hopefully, develop a better understanding of spatial capture-recapture models.

2066 In this chapter, we consider classes of GL(M)Ms – Poisson and binomial (i.e.,  
2067 logistic regression) models – that will prove to be enormously useful in the analysis  
2068 of capture-recapture models of all kinds. Many readers are likely familiar with these  
2069 models already because they are one of the most useful models in all of ecology and,  
2070 as such, have received considerable attention in many introductory and advanced  
2071 texts. We focus on them here in order to introduce the readers to the analysis of  
2072 such models in **R** and **WinBUGS** or **JAGS**, which we will translate directly to  
2073 the analysis of SCR models in subsequent chapters.

2074 Bayesian analysis is convenient for analyzing GL(M)Ms because it allows us to  
2075 work directly with the conditional model – i.e., the model that is conditional on  
2076 the random effects, using computational methods known as Markov chain Monte  
2077 Carlo (MCMC). Learning how to do Bayesian analysis of GLMs and GLMMs using  
2078 the **BUGS** language is, in part, the purpose of this chapter. We focus here on  
2079 the use of **WinBUGS** because it is the most popular “**BUGS** engine”. However,  
2080 later in the book we transition to another popular **BUGS** engine known as **JAGS**  
2081 (?) which stands for *Just Another Gibbs Sampler*. For most of our purposes,  
2082 the specification of models in either platform is the same, but **JAGS** is under  
2083 active development at the present time while **WinBUGS** no longer is, having

2084 transitioned to **OpenBUGS** (Lunn et al., 2009) which is still in active development.  
 2085 While we use **BUGS** of one sort or another to do the Bayesian computations, we  
 2086 organize and summarize our data and execute **WinBUGS** or **JAGS** from within **R**  
 2087 using the useful packages **R2WinBUGS** (Sturtz et al., 2005), **R2jags** (Su and Yajima,  
 2088 2011) or **rjags** (?). Kéry (2010), and Kéry and Schaub (2012) provide excellent  
 2089 and accessible introductions to the basics of Bayesian analysis and GLMs using  
 2090 **WinBUGS**. We don't want to be too redundant with those books and so we avoid  
 2091 a detailed treatment of Bayesian methodology and software usage - instead just  
 2092 providing a cursory overview so that we can move on and attack the problems  
 2093 we're most interested in related to spatial capture-recapture. In addition, there are  
 2094 a number of texts that provide general introductions to Bayesian analysis, MCMC,  
 2095 and their applications in ecology including McCarthy (2007), Kéry (2010), Link  
 2096 and Barker (2010), and King (2009).

2097 While this chapter is about Bayesian analysis of GL(M)Ms, such models are  
 2098 routinely analyzed using likelihood methods too. Later in this book, we will use  
 2099 likelihood methods to analyze SCR models but, for now, we concentrate on providing  
 2100 a basic introduction to Bayesian analysis because that is the approach we will  
 2101 use in a majority of cases in later chapters.

### 3.1 GLMS AND GLMMS

2102 We have asserted already that SCR models work out most of the time to be variations  
 2103 of GL(M)Ms. You might therefore ask: What are these GLM and GLMM  
 2104 models, anyhow? These models are covered extensively in many very good applied  
 2105 statistics books and we refer the reader elsewhere for a detailed introduction. The  
 2106 classical reference for GLMs is Nelder and Wedderburn (1972) and also McCullagh  
 2107 and Nelder (1989). In addition, we think Kéry (2010), Kéry and Schaub (2012),  
 2108 and Zuur et al. (2009) are all accessible treatments. Here, we'll give the 1 minute  
 2109 treatment of GL(M)Ms, not trying to be complete but rather only to preserve a  
 2110 coherent organization to the book.

2111 The generalized linear model (GLM) is an extension of standard linear models by  
 2112 allowing the response variable to have some distribution from the exponential family  
 2113 of distributions. This includes the normal distribution but also others such as the  
 2114 Poisson, binomial, gamma, exponential, and many more. In addition, GLMS allow  
 2115 the response variable to be related to the predictor variables (i.e., covariates) using  
 2116 a link function, which is usually nonlinear. The GLM consists of three components:

- 2117 1. A probability distribution for the dependent (or response) variable  $y$ , from the  
 2118 exponential family of probability distributions.
- 2119 2. A “linear predictor”  $\eta = \beta_0 + x\beta_1$ , where  $x$  is a predictor variable (i.e., a covariate).
- 2120 3. A link function  $g$  that relates the expected value of  $y$ ,  $\mathbb{E}(y)$ , to the linear predictor,  
 2121  $\mathbb{E}(y) = \mu = g^{-1}(\eta)$ . Therefore  $g(\mathbb{E}(y)) = \eta = \beta_0 + x\beta_1$ .

2123 A key aspect of GLMs is that  $g(\mathbb{E}(y))$  is assumed to be a linear function of the  
 2124 predictor variable(s), here  $x$ , with unknown parameters, here  $\beta_0$  and  $\beta_1$ , to be  
 2125 estimated. In standard GLMs, the variance of  $y$  is a function  $V$  of the mean of  $y$ :  
 2126  $\text{Var}(y) = V(\mu)$  (see below for examples). As an example, a Poisson GLM posits  
 2127 that  $y \sim \text{Poisson}(\lambda)$  with  $\mathbb{E}(y) = \lambda$  and usually the model for the mean is specified  
 2128 using the *log link function* by

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

2129 The variance function is  $V(y_i) = \lambda_i$ . To see how a Poisson GLM works, use the **R**  
 2130 code below to simulate some data and then estimate the parameters:

```
2131 set.seed(13)
2132 n=100          # set sample size
2133 beta0<- -2    # set intercept term
2134 beta1<- 1.5   # set coefficient
2135 x<-rnorm(n, 0,1) # generate a predictor variable, x
2136
2137 linpred<- beta0 + beta1*x  # calculate linear predictor of E[y]
2138 y<-rpois(n, exp(linpred)) # generate observations from model
```

2139 The **R** function `glm()` fits a GLM to the data we just generated and returns es-  
 2140 timates of  $\beta_0$  and  $\beta_1$ , which we see are fairly close to the data generating values  
 2141 above:

```
2142 glm(y ~ 1 + x, family='poisson')      # the fit model
```

2143 This produces the output:

```
2144 Call: glm(formula = y ~ 1 + x, family = "poisson")
2145
2146 Coefficients:
2147 (Intercept)           x
2148       -2.007        1.446
2149
2150 [... some output deleted ...]
```

2151 The binomial GLM posits that  $y_i \sim \text{Binomial}(K, p)$  where  $K$  is the fixed sample  
 2152 size parameter and  $\mathbb{E}(y_i) = K \times p_i$ . Usually the model for the mean is specified  
 2153 using the *logit link function* according to

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

2154 Where  $\text{logit}(p) = \log(p/(1-p))$ . The inverse-logit function, consequently, is  $\text{logit}^{-1}(p) =$   
 2155  $\exp(p)/(1 + \exp(p))$ .

2156 A GLMM is the extension of GLMs to accommodate “random effects”. Often  
 2157 this involves adding a normal random effect to the linear predictor. One simple  
 2158 example is using a random intercept,  $\alpha$ :

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

2159 where

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

2160 Many other probability distributions and formulations of the linear predictor might  
 2161 be considered. GLMMs are enormously useful in ecological modeling applications  
 2162 for modeling variation due to subjects, observers, spatial or temporal stratification,  
 2163 clustering, and dependence that arises from any kind of group structure and, of  
 2164 course, because SCR models prove to be a type of GLM with a random effect, but  
 2165 one that does not enter the mean linearly.

### 3.2 BAYESIAN ANALYSIS

2166 Bayesian analysis is unfamiliar to many ecological researchers because older co-  
 2167 horts of ecologists were largely educated in the classical statistical paradigm of  
 2168 frequentist inference. But advances in technology and increasing exposure to the  
 2169 benefits of Bayesian analysis are fast making Bayesians out of people or at least  
 2170 making Bayesian analysis an acceptable, general alternative to classical, frequentist  
 2171 inference.

2172 Conceptually, the main thing about Bayesian inference is that it uses proba-  
 2173 bility directly to characterize uncertainty about things we don’t know. “Things”,  
 2174 in this case, are parameters of models and, just as it is natural to characterize  
 2175 uncertain outcomes of stochastic processes using probability, it seems natural also  
 2176 to characterize information about unknown parameters using probability. At least  
 2177 this seems natural to us and, we think, most ecologists either explicitly adopt that  
 2178 view or tend to fall into that point of view naturally. Conversely, frequentists use  
 2179 probability in many different ways, but never to characterize uncertainty about  
 2180 parameters<sup>1</sup>. Instead, frequentists use probability to characterize the behavior of  
 2181 *procedures* such as estimators or confidence intervals (see below). It is surprising  
 2182 that people readily adopt a philosophy of statistical inference in which the things  
 2183 you don’t know (i.e., parameters) should *not* be regarded as random variables, so  
 2184 that, as a consequence, one cannot use probability to characterize one’s state of  
 2185 knowledge about them.

#### 2186 3.2.1 Bayes’ Rule

2187 As its name suggests, Bayesian analysis makes use of Bayes’ rule in order to make  
 2188 direct probability statements about model parameters. Given two random variables

---

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

2189  $z$  and  $y$ , Bayes' rule relates the two conditional probability distributions  $[z|y]$  and  
2190  $[y|z]$  by the relationship:

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

2191 Bayes' rule itself is a mathematical fact and there is no debate in the statistical  
2192 community as to its validity and relevance to many problems. Generally speaking,  
2193 these distributions are characterized as follows:  $[y|z]$  is the conditional probability  
2194 distribution of  $y$  given  $z$ ,  $[z]$  is the marginal distribution of  $z$  and  $[y]$  is the marginal  
2195 distribution of  $y$ . In the context of Bayesian inference we usually associate specific  
2196 meanings in which  $[y|z]$  is thought of as “the likelihood”,  $[z]$  as the “prior” and so  
2197 on. We leave this for later because here the focus is on this expression of Bayes'  
2198 rule as a basic fact of probability.

2199 As an example of a simple application of Bayes' rule, consider the problem  
2200 of determining species presence at a sample location based on imperfect survey  
2201 information. Let  $z$  be a binary random variable that denotes species presence  
2202 ( $z = 1$ ) or absence ( $z = 0$ ), let  $\Pr(z = 1) = \psi$  where  $\psi$  is usually called occurrence  
2203 probability, “occupancy” (MacKenzie et al., 2002) or “prevalence”. Let  $y$  be the  
2204 *observed* presence ( $y = 1$ ) or absence ( $y = 0$ ) (or, strictly speaking, detection  
2205 and non-detection), and let  $p$  be the probability that a species is detected in a  
2206 single survey at a site given that it is present. Thus,  $\Pr(y = 1|z = 1) = p$ . The  
2207 interpretation of this is that, if the species is present, we will only observe it with  
2208 probability  $p$ . In addition, we assume here that  $\Pr(y = 1|z = 0) = 0$ . That is, the  
2209 species cannot be detected if it is not present which is a conventional view adopted  
2210 in most biological sampling problems (but see Royle and Link (2006)). If we survey  
2211 a site  $K$  times but never detect the species, then this clearly does not imply that the  
2212 species is not present ( $z = 0$ ) at this site (we could just fail to observe it every single  
2213 time). Rather, our degree of belief in  $z = 0$  should be made with a probabilistic  
2214 statement, namely the conditional probability  $\Pr(z = 1|y_1 = 0, \dots, y_K = 0)$ . If the  
2215  $K$  surveys are independent so that we might regard  $y_k$  as *iid* Bernoulli trials, then  
2216 the total number of detections, say  $y$ , is Binomial with probability  $p$ , and we can  
2217 use Bayes' rule to compute the probability that the species is present given that  
2218 it is not detected in  $K$  samples, i.e.,  $\Pr(z = 1|y_1 = 0, \dots, y_K = 0)$ . In words, the  
2219 expression we seek is:

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

2220 Mathematically, this is

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

2221 To apply this, suppose that  $K = 2$  surveys are done at a wetland for a species of frog,  
 2222 and the species is not detected there. Suppose further that  $\psi = 0.8$  and  $p = 0.5$  are  
 2223 obtained from a prior study. Then the probability that the species is present at this  
 2224 site, even though it was not detected, is  $(1 - 0.5)^2 \times 0.8 / ((1 - 0.5)^2 \times 0.8 + (1 - 0.8)) =$   
 2225 0.5. That is, there is a 50/50 chance that the site is occupied despite the fact that  
 2226 the species wasn't observed there.

2227 In summary, Bayes' rule provides a simple linkage between the conditional prob-  
 2228 abilities  $[y|z]$  and  $[z|y]$ , which is useful whenever we need to deduce one from the  
 2229 other.

2230 **3.2.2 Principles of Bayesian Inference**

2231 Bayes' rule as a basic fact of probability is not disputed. What is controversial to  
 2232 some is the scope and manner in which Bayes' rule is applied by Bayesian analysts.  
 2233 Bayesian analysts assert that Bayes' rule is relevant, in general, to all statistical  
 2234 problems by regarding all unknown quantities of a model as realizations of ran-  
 2235 dom variables - this includes data, latent variables, and also parameters. Classical  
 2236 (non-Bayesian) analysts sometimes object to regarding parameters as outcomes of  
 2237 random variables. Classically, parameters are thought of as "fixed but unknown"  
 2238 (using the terminology of classical statistics). Indeed, a common misunderstanding  
 2239 on the distinction between Bayesian and frequentist inference goes something like  
 2240 this "in frequentist inference parameters are fixed but unknown but in a Bayesian  
 2241 analysis parameters are random." At best this is a sad caricature of the distinction  
 2242 and at worst it is downright wrong. In Bayesian analysis the parameters are also  
 2243 unknown and, in fact, there is a single data-generating value of each parameter,  
 2244 and so they are also fixed. The difference is that the fixed but unknown values are  
 2245 regarded as having been generated from some probability distribution. Specifica-  
 2246 tion of that probability distribution is necessary to carryout Bayesian analysis, but  
 2247 it is not required in classical frequentist inference.

2248 To see the general relevance of Bayes' rule in the context of statistical inference,  
 2249 let  $y$  denote observations - i.e., data - and let  $[y|\theta]$  be the observation model (often  
 2250 colloquially referred to as the "likelihood"). Suppose  $\theta$  is a parameter of interest  
 2251 having (prior) probability distribution  $[\theta]$  (also simply referred to as the prior).  
 2252 These are combined to obtain the posterior distribution using Bayes' rule, which  
 2253 is:

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

2254 Asserting the general relevance of Bayes' rule to all statistical problems, we can  
 2255 conclude that the two main features of Bayesian inference are that: (1) parameters,  
 2256  $\theta$ , are regarded as realizations of a random variable and, as a result, (2) inference is  
 2257 based on the probability distribution of the parameters given the data,  $[\theta|y]$ , which  
 2258 is called the posterior distribution. This is the result of using Bayes' rule to combine  
 2259 the "likelihood" and the prior distribution. The key concept is regarding parameters

as realizations of a random variable because, once you admit this conceptual view, this leads directly to the posterior distribution, a very natural quantity upon which to base inference about things we don't know - including parameters of statistical models. In particular,  $[\theta|y]$  is a probability distribution for  $\theta$  and therefore we can make direct probability statements to characterize uncertainty about  $\theta$ .

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

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

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

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

### 3.2.3 Prior distributions

The prior distribution  $[\theta]$  is an important feature of Bayesian inference. As a conceptual matter, the prior distribution characterizes “prior beliefs” or “prior

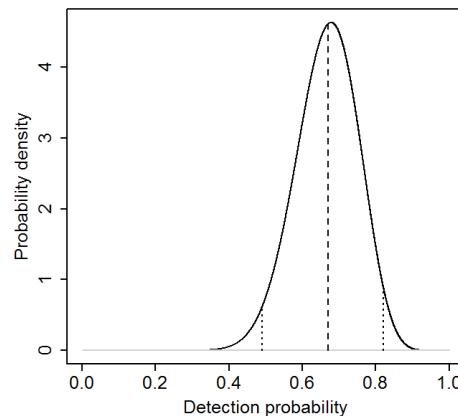
2301 information” about a parameter. Indeed, an oft-touted benefit of Bayesian analysis  
2302 is the ease with which prior information can be included in an analysis. However,  
2303 more commonly, the prior is chosen to express a lack of prior information, even if  
2304 previous studies have been done and even if the investigator does in fact know quite  
2305 a bit about a parameter. This is because the manner in which prior information  
2306 is embodied in a prior (and the amount of information) is usually very subjective  
2307 and thus the result can wind up being very contentious, e.g., different investigators  
2308 might report different results based on subjective assessments of prior information.  
2309 Thus it is usually better to “let the data speak” and use priors that reflect absence of  
2310 information beyond the data set being analyzed. An example for an uninformative  
2311 prior is a  $\text{Uniform}(0, 1)$  for a probability, or a  $\text{Uniform}(-\infty, \infty)$  (also called a “flat”  
2312 or “improper” prior) for an unbounded continuous parameter. Alternatively, people  
2313 use “diffuse priors”; these contain some information, but (ideally) not enough to  
2314 exert meaningful influence on the posterior. An example for a diffuse prior could  
2315 be a Normal distribution with a large standard deviation.

2316 But still the need occasionally arises to embody prior information or beliefs  
2317 about a parameter formally into the estimation scheme. In SCR models we often  
2318 have a parameter that is closely linked to “home range size” and thus auxiliary  
2319 information on the home range size of a species can be used as prior information,  
2320 which may improve parameter estimation (e.g., see Chandler and Royle (In press);  
2321 also Chapt. 18).

2322 At times the situation arises where a prior can inadvertently impose substantial  
2323 effect on the posterior of a parameter, and that is not desirable. For example, we use  
2324 data augmentation to deal with the fact that the population size  $N$  is an unknown  
2325 parameter (Royle et al., 2007) which is equivalent to imposing a  $\text{Binomial}(M, \psi)$   
2326 prior on  $N$  for some integer  $M$  (see Sec. 4.2). One has to take care to make sure  
2327 that  $M$  is sufficiently large so as to not affect the posterior distribution on  $N$  (see  
2328 Fig. 14.6, and also Kéry and Schaub (2012, Ch. 5)). Another situation that we  
2329 have to be careful of is that prior distributions are *not* invariant to transformation of  
2330 the parameter, and therefore neither are posterior distributions (Link and Barker,  
2331 2010, Sec. 6.2.1). Thus, a prior that is ostensibly non-informative on one scale,  
2332 may be very informative on another scale. For example, if we have a flat prior  
2333 on  $\text{logit}(p)$  for some probability parameter  $p$ , this is very different from having a  
2334  $\text{Uniform}(0, 1)$  prior on  $p$ . We show an example where this makes a difference in  
2335 Chapt. 5. Nonetheless, it is always possible to assess the influence of prior choice,  
2336 and it is often the case (with sufficient data and a structurally identifiable model)  
2337 that the influence of priors is negligible.

### 2338 3.2.4 Posterior Inference

2339 In Bayesian inference, we are not focusing on estimating a single point or interval  
2340 but rather on characterizing a whole distribution – the posterior distribution –  
2341 from which one can report any summary of interest. A point estimate might be



**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

the posterior mean, median, mode, etc.. In many applications in this book, we will compute 95% Bayesian confidence intervals using the 2.5% and 97.5% quantiles of the posterior distribution. For such intervals, it is correct to say  $\Pr(L < \theta < U) = 0.95$ . That is, “the probability that  $\theta$  lies between  $L$  and  $U$  is 0.95”.

As an example, suppose we conducted a Bayesian analysis to estimate detection probability ( $p$ ) of some species at a study site, and we obtained a posterior distribution of  $\text{beta}(20,10)$  for the parameter  $p$ . The following **R** commands demonstrate how we make inferences based upon summaries of the posterior distribution:

```
2342 > post.median <- qbeta(0.5, 20, 10)
2343 [1] 0.6704151
2344 > post.95ci <- qbeta(c(0.025, 0.975), 20, 10)
2345 [1] 0.4916766 0.8206164
```

Thus, we can state that there is a 95% probability that  $\theta$  lies between 0.49 and 0.82. Fig. 3.1 shows the posterior along with the summary statistics.

It is not a subtle thing that such statements cannot be made using frequentist methods, although people tend to say it anyway and not really understand why it is wrong or even that it is wrong. This is because frequentists have not been successful in convincing people to over-ride their natural inclination on how to use probability. As a frequentist, you simply cannot use probability in the manner that you would like to.

**2362 3.2.5 Small sample inference**

2363 The posterior distribution is an exhaustive summary of the state-of-knowledge  
2364 about an unknown quantity. It is *the* posterior distribution - not an estimate of  
2365 that thing. It is also not, usually, an approximation except to within Monte Carlo  
2366 error (in cases where we use simulation to calculate it, see Sec. 3.5.2). One of the  
2367 great virtues of Bayesian analysis which is not widely appreciated is that posterior  
2368 inference is not “asymptotic”, which is to say, valid in a limiting sense as the sample  
2369 size tends to infinity. Rather, posterior inference is valid for *any* sample size and, in  
2370 particular, *the* sample size on-hand. Conversely, almost all frequentist procedures  
2371 are based on asymptotic approximations to the procedure which is being employed.

2372 There seems to be a prevailing view in statistical ecology that classical likelihood-  
2373 based procedures are virtuous because of the availability of simple formulas and  
2374 procedures for carrying out inference, such as calculating standard errors, doing  
2375 model selection by AIC, and assessing goodness-of-fit. In large samples, this may  
2376 be an important practical benefit, but the theoretical validity of these procedures  
2377 cannot be asserted in most situations involving small samples. This is not a mi-  
2378 nor issue because it is typical in many wildlife sampling problems – especially in  
2379 surveys of carnivores or rare/endangered species – to wind up with a small, some-  
2380 times extremely small, data set, that is nevertheless extremely valuable (Foster and  
2381 Harmsen, 2012). For examples: A recent paper (Hawkins and Racey, 2005) on the  
2382 fossa (*Cryptoprocta ferox*), estimated an adult density of 0.18 adults per sq. km  
2383 based on a sample size of 20 animals captured over 3 years. Sepúlveda et al. (2007)  
2384 estimated density of the endangered southern river otter (*Lontra provocax*) based  
2385 on 12 individuals captured over 3 years, Gardner et al. (2010a) estimated density  
2386 from a study of the Pampas cat (*Leopardus colocolo*), a species for which very lit-  
2387 tle is known, based on only 22 captured individuals over a two year study period,  
2388 Trolle and Kéry (2005) reported only 9 individual ocelots captured and Jackson  
2389 et al. (2006) captured 6 individual snow leopards (*Panthera uncia*) using camera  
2390 trapping. Thus, almost all likelihood-based analysis of data on rare and/or secre-  
2391 tive carnivores necessarily and flagrantly violate one of Le Cam’s Basic Principles:  
2392 “If you need to use asymptotic arguments, do not forget to let your number of  
2393 observations tend to infinity.”(Le Cam, 1990).

2394 The biologist thus faces a dilemma with such data. On one hand, these datasets,  
2395 and the resulting inference, are often criticized as being poor and unreliable. Or,  
2396 even worse<sup>2</sup>, “the data set is so small, this is a poor analysis.” On the other hand,  
2397 such data may be all that is available for species that are extraordinarily important  
2398 for conservation and management. The Bayesian framework for inference provides  
2399 a valid, rigorous, and flexible framework that is theoretically justifiable in arbitrary  
2400 sample sizes. This is not to say that one will obtain precise estimates of density  
2401 or other parameters, just that your inference is coherent and justifiable from a  
2402 conceptual and technical statistical point of view. That is, for example when we es-

---

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

2403 timate the density  $D$  of some animal population, we report the posterior probability  
 2404  $\Pr(D|data)$  which is easily interpretable and just what it is advertised to be and we  
 2405 don't need to do a simulation study to evaluate how well the reported  $\Pr(D|data)$   
 2406 deviates from the "true"  $\Pr(D|data)$  because they are the same quantity.

### 3.3 CHARACTERIZING POSTERIOR DISTRIBUTIONS BY MCMC SIMULATION

2407 In practice, it is not really feasible to ever compute the marginal probability distribution  
 2408  $[y]$ , the denominator resulting from application of Bayes' rule (Eq. 3.2.1).  
 2409 For decades (even centuries!) this impeded the adoption of Bayesian methods by  
 2410 practitioners. Or, the few Bayesian analyses done were based on asymptotic normal  
 2411 approximations to the posterior distribution. While this was useful from a  
 2412 theoretical and technical standpoint and, practically, it allowed people to make the  
 2413 probability statements that they naturally would like to make, it was kind of a bad  
 2414 joke around the Bayesian water-cooler to, on one hand, criticize classical statistics  
 2415 for being, essentially, completely ad hoc in their approach to things but then, on  
 2416 the other hand, have to devise various approximations to what they were trying  
 2417 to characterize. The advent of Markov chain Monte Carlo (MCMC) methods has  
 2418 made it easier to calculate posterior distributions for just about any problem to  
 2419 sufficient levels of precision.

2420 Broadly speaking, MCMC is a class of methods for drawing random samples  
 2421 (i.e., simulating from or just "sampling") from the target posterior distribution.  
 2422 Thus, even though we might not recognize the posterior as a named distribution or  
 2423 be able to analyze its features analytically, e.g., devise mathematical expressions  
 2424 for the mean and variance, we can use these MCMC methods to obtain a large  
 2425 sample from the posterior and then use that sample to characterize features of  
 2426 the posterior. What we do with the sample depends on our intentions – typically  
 2427 we obtain the mean or median for use as a point estimate, and take a confidence  
 2428 interval based on Monte Carlo estimates of the quantiles.

#### 3.3.1 What Goes on Under the MCMC Hood

2429 We will develop and apply MCMC methods in some detail for spatial capture-  
 2430 recapture models in Chapt. 14. Here we provide a simple illustration of some basic  
 2431 ideas related to the practice of MCMC.

2432 A type of MCMC method relevant to most problems is Gibbs sampling (Geman  
 2433 and Geman, 1984) which we address in more detail in Chapt. 14. Gibbs sampling  
 2434 involves iterative simulation from the "full conditional" distributions (also called  
 2435 conditional posterior distributions). The full conditional distribution for an un-  
 2436 known quantity is the conditional distribution of that quantity given every other  
 2437 random variable in the model - the data and all other parameters (see Sec. 3.3.2 for  
 2438 rules of how to construct full conditionals). For example, for a normal regression

2440 model<sup>3</sup> with  $y \sim \text{Normal}(\beta_0 + \beta_1(x - \bar{x}), \sigma^2)$  where lets say  $\sigma^2$  is known, the full  
2441 conditionals are, in symbolic terms, and using the “bracket notation”,

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

2442 and

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

2443 We might use our knowledge of probability to identify these mathematically. In  
2444 particular, by Bayes' Rule,  $[\beta_0|y, \beta_1] = [y|\beta_0, \beta_1][\beta_0|\beta_1]/[y|\beta_1]$  and similarly for  
2445  $[\beta_1|y, \beta_0]$ . For example, if we have priors for  $[\beta_0] = \text{Normal}(\mu_{\beta_0}, \sigma_{\beta_0}^2)$  and  $[\beta_1] =$   
2446  $\text{Normal}(\mu_{\beta_1}, \sigma_{\beta_1}^2)$  then some algebra reveals that

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

2447 where  $\tau = 1/\sigma^2$  and  $\tau_{\beta_0} = 1/\sigma_{\beta_0}^2$  (the inverse of the variance is sometimes called  
2448 *precision*), and  $w = \tau n / (\tau n + \tau_{\beta_0})$ . We see in this case that the posterior mean  
2449 is a *precision-weighted* sum of the sample mean  $\bar{y}$  and the prior mean  $\mu_{\beta_0}$ , and  
2450 the posterior *precision* is the sum of the precision of the likelihood and that of the  
2451 prior. These results are typical of many classes of problems. In particular, note  
2452 that as the prior precision tends to 0, i.e.,  $\tau_{\beta_0} \rightarrow 0$ , then the posterior of  $\beta_0$  tends  
2453 to  $\text{Normal}(\bar{y}, \sigma^2/n)$ . We recognize the variance of this distribution as that of the  
2454 variance of the sampling distribution of  $\bar{y}$  and its mean is in fact the MLE of  $\beta_0$  for  
2455 this model. The conditional posterior of  $\beta_1$  has a very similar form:

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

2456 which might look slightly unfamiliar, but note that if  $\tau_{\beta_1} = 0$ , then the mean of this  
2457 distribution is the familiar  $\hat{\beta}_1$ , and the variance is, in fact, the sampling variance of  
2458  $\beta_1$ . The MCMC algorithm for this model has us simulate in succession, repeatedly,  
2459 from those two distributions. See Gelman et al. (2004) for more examples of Gibbs  
2460 sampling for the normal model, and we also provide another example in Chapt.  
2461 14. A conceptual representation of the MCMC algorithm for this simple model is  
2462 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
}

```

2463<sup>3</sup>We center the independent variable here so that things look more intuitive in the result

As we just saw for this simple “normal-normal” model it is sometimes possible to specify the full conditional distributions analytically. In general, when certain so-called conjugate prior distributions are used, which have an analytic form that, in a statistical sense, “matches” the likelihood, then the form of full conditional distributions is also similar to that of the observation model. In this normal-normal case, the normal distribution for the mean parameters is the conjugate prior for the normal observation model, and thus the full-conditional distributions are also normal. This is convenient because, in such cases, we can simulate directly from them using standard methods (or **R** functions). But, in practice, we don’t really ever need to know such things because most of the time we can get by using a simple algorithm, called the Metropolis-Hastings (henceforth “MH”) algorithm, to obtain samples from these full conditional distributions without having to recognize them as specific, named, distributions. This gives us enormous freedom in developing models and analyzing them without having to resolve them mathematically because to implement the MH algorithm we need only identify the full conditional distribution up to a constant of proportionality, that being the marginal distribution in the denominator (e.g.,  $[y|\beta_1]$  above).

We will talk about the Metropolis-Hastings algorithm shortly, and we will use it extensively in the analysis of SCR models (e.g., Chapt. 14).

### 3.3.2 Rules for constructing full conditional distributions

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

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

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

In the context of our simple linear regression model that we’ve been working with, to characterize  $[\beta_0|y, \beta_1]$  we first apply step 1 and identify the model components as:  $[y|\beta_0, \beta_1]$ , with prior distributions  $[\beta_0]$  and  $[\beta_1]$ . Step 2 has us write  $[\beta_0|y, \beta_1] \propto [y|\beta_0, \beta_1][\beta_0][\beta_1]$ . Step 3: We note that  $[\beta_1]$  is not a function of  $\beta_0$  and therefore we remove it to obtain  $[\beta_0|y, \beta_1] \propto [y|\beta_0, \beta_1][\beta_0]$ . Similarly, applying step

2504 2 and 3 for  $\beta_1$  we obtain  $[\beta_1|y, \beta_0] \propto [y|\beta_0, \beta_1][\beta_1]$ . We apply step 4 and manipulate  
 2505 these algebraically to arrive at the result (which we provided in Eqs. 3.3.1  
 2506 and 3.3.2) or, alternatively, we can sample them indirectly using the Metropolis-  
 2507 Hastings algorithm, which we discuss now.

2508 **3.3.3 Metropolis-Hastings algorithm**

2509 The Metropolis-Hastings (MH) algorithm is a completely generic method for sam-  
 2510 pling from any distribution, say  $[\theta]$ . In our applications,  $[\theta]$  will typically be the full  
 2511 conditional distribution of  $\theta$ . While we sometimes use Gibbs sampling, we seldom  
 2512 use “pure” Gibbs sampling because full conditionals do not always take the form  
 2513 of known distributions we can sample from directly. In such cases, we use MH to  
 2514 sample from the full conditional distributions. When the MH algorithm is used to  
 2515 sample from full conditional distributions of a Gibbs sampler the resulting hybrid  
 2516 algorithm is called *Metropolis-within-Gibbs*. In Sec. 3.6.3 we will actually construct  
 2517 such an algorithm for a simple class of models. We discuss both the Gibbs and the  
 2518 MH algorithm, as well as their hybrid in more depth in Chapt. 14.

2519 The MH algorithm generates candidate values for the parameter(s) we want  
 2520 to estimate from some proposal or candidate-generating distribution that may be  
 2521 conditional on the current value of the parameter, denoted by  $h(\theta^*|\theta^{t-1})$ . Here,  
 2522  $\theta^*$  is the *candidate* or proposed value and  $\theta^{t-1}$  is the value of  $\theta$  at the previous  
 2523 time step, i.e., at iteration  $t - 1$  of the MCMC algorithm. The proposed value is  
 2524 accepted with probability

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

2525 which is called the MH acceptance probability. This ratio can sometimes be  $> 1$  in  
 2526 which case we set it equal to 1. It is useful to note that  $h()$  can be anything at all.

2527 In the context of using the MH algorithm to do MCMC (in which case the  
 2528 target distribution is a full-conditional or posterior distribution), an important fact  
 2529 is, no matter the choice of  $h()$ , we can compute the MH acceptance probability  
 2530 directly because the marginal distribution of  $y$  cancels from both the numerator  
 2531 and denominator of  $r$ . This is the magic of the MH algorithm.

### 3.4 BAYESIAN ANALYSIS USING THE BUGS LANGUAGE

2532 We won’t be too concerned with devising our own MCMC algorithms for every  
 2533 analysis although we will do that a few times for fun. More often, we will rely on  
 2534 the freely available software package **WinBUGS** or **JAGS** for doing this. We will  
 2535 always execute these **BUGS** engines from within **R** using the **R2WinBUGS** (Sturtz  
 2536 et al., 2005) or, for **JAGS**, the **R2jags** (Su and Yajima, 2011) or **rjags** (?) pack-  
 2537 ages. **WinBUGS** and **JAGS** are MCMC black boxes that take a pseudo-code  
 2538 description (i.e., written in the **BUGS** language) of all of the relevant stochastic

and deterministic elements of a model and generate an MCMC algorithm for that model. But you never get to see the algorithm. Instead, **WinBUGS/JAGS** will run the algorithm and return the Markov chain output - the posterior samples of model parameters.

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

While we normally use **WinBUGS**, we note that **OpenBUGS** is the current active development tree of the **BUGS** project. See Kéry (2010) and Kéry and Schaub (2012, especially App. 1) for more on practical analysis in **WinBUGS**. Those books should be consulted for a more comprehensive introduction to using **WinBUGS**. Recently we have migrated many of our analyses to **JAGS** (?), which we adopt later in the book. You can refer to Hobbs (2011) for an ecological introduction to **JAGS**. Next, we provide an example of a Bayesian analysis using **WinBUGS**

### 3.4.1 Linear Regression in WinBUGS

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

```
2557 x<-rnorm(10)
2558 mu<- -3.2 + 1.5*x
2559 y<-rnorm(10,mu, sd=4)
```

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

```
2560 cat("
2561   model {
2562     for (i in 1:10) {
2563       y[i] ~ dnorm(mu[i],tau)           # the likelihood
2564       mu[i]<- beta0 + beta1*x[i]      # the linear predictor
2565     }
2566     beta0 ~ dnorm(0,.01)             # prior distributions
2567     beta1 ~ dnorm(0,.01)
2568     sigma ~ dunif(0,100)
2569     tau<-1/(sigma*sigma)          # tau is a derived parameter
2570   }
2571   ",file="normal.txt")
```

2567 Alternatively, you can write the model specifications directly within a text file and  
 2568 save it in your current working directory, but we do not usually take that approach  
 2569 in this book.

2570 The **BUGS** dialects<sup>4</sup> parameterize the normal distribution in terms of the mean  
 2571 and inverse-variance, called the precision. Thus, `dnorm(0,.01)` implies a variance  
 2572 of 100. We typically use diffuse normal priors for mean parameters,  $\beta_0$  and  $\beta_1$   
 2573 in this case, but sometimes we might use uniform priors with suitable bounds -  
 2574  $B$  and  $+B$ . Also, we typically use a  $\text{Uniform}(0, B)$  prior on standard deviation  
 2575 parameters (Gelman, 2006). But sometimes we might use a gamma prior on the  
 2576 precision parameter  $\tau$ . In a **BUGS** model file, every variable referenced in the  
 2577 model description has to be either data, which will be input (see below), a random  
 2578 variable which must have a probability distribution associated with it using the  
 2579 tilde character (aka “twiddle”) “ $\sim$ ” or it has to be a derived parameter connected  
 2580 to variables and data using an array: “ $<-$ ”.

2581 To fit the model, we need to describe various data objects to **WinBUGS**.  
 2582 In particular, we create an **R** list object called `data` which are the data objects  
 2583 identified in the **BUGS** model file. In the example, the data consist of two objects  
 2584 which exist as  $y$  and  $x$  in the **R** workspace and also in the **WinBUGS** model  
 2585 definition. We also create an **R** function that produces a list of starting values,  
 2586 `inits`, that get sent to **WinBUGS**. In general, starting values are optional. We  
 2587 recommend to always provide reasonable starting values where possible, both for  
 2588 structural parameters and also random effects<sup>5</sup>. Finally, we identify the names of  
 2589 the parameters (labeled correspondingly in the **WinBUGS** model specification)  
 2590 that we want **WinBUGS** to save the MCMC output for. In this example, we will  
 2591 “monitor” the parameters  $\beta_0$ ,  $\beta_1$ ,  $\sigma$  and  $\tau$ . **WinBUGS** is executed using the **R**  
 2592 command `bugs()`. We set the option `debug=TRUE` if we want the **WinBUGS** GUI  
 2593 to stay open (useful for analyzing MCMC output and looking at the **WinBUGS**  
 2594 error log). Also, we set `working.dir=getwd()` so that **WinBUGS** output files and  
 2595 the log file are saved in the current **R** working directory (note that sometimes you  
 2596 will need to specify the place where you installed **WinBUGS** within the `bugs()`  
 2597 call, using the `bugs.directory` argument). All of these activities together look like  
 2598 this:

```
2599 library("R2WinBUGS")    # "attach" the R2WinBUGS library
2600 data <- list ( "y","x")
2601 inits <- function()
2602   list ( beta1=rnorm(1),beta0=rnorm(1),sigma=runif(1,0,2) )
2603 parameters <- c("beta0","beta1","sigma","tau")
2604 out<-bugs (data, inits, parameters, "normal.txt", n.thin=1, n.chains=2,
2605             n.burnin=2000, n.iter=6000, debug=TRUE,working.dir=getwd())
```

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

<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 V; we will come back to this issue when we use **JAGS**

2606 A common question is “how should my data be formatted?” That depends on  
 2607 how you describe the model in the **BUGS** language, and how your data are in-  
 2608 put into **R**. There is no unique way to describe any particular model and so you  
 2609 have some flexibility. We talk about data format further in the context of capture-  
 2610 recapture models and SCR models in Chapt. 5 and elsewhere. Note that the  
 2611 previously created objects defining data, initial values and parameters to monitor  
 2612 are passed to the function `bugs()`. In addition, various other things are declared:  
 2613 The number of parallel Markov chains (`n.chains`), the thinning rate (`n.thin`),  
 2614 the number of burn-in iterations (`n.burnin`) and the total number of iterations  
 2615 (`n.iter`). To develop a detailed understanding of the various parameters and set-  
 2616 tings used for MCMC, consult a basic reference such as Kéry (2010). We also come  
 2617 back to these issues in the follwong section (3.5) and in Chapt. 14.

2618 You should execute all of the commands given above and then look at the  
 2619 resulting output (summarized in table ??). Close the **WinBUGS** GUI and the  
 2620 data will be read back into **R** (or specify `debug=FALSE`). We don’t want to give  
 2621 instructions on how to navigate and use the GUI - but you can fire up **WinBUGS**  
 2622 and read the help files, or see Ch. 4 from Kéry (2010) for a brief introduction. The  
 2623 print command applied to the object `out` prints some basic summary output (this  
 2624 is slightly edited):

```
2625 > print(out,digits=2)
2626 Inference for Bugs model at "normal.txt", fit using WinBUGS,
2627 2 chains, each with 6000 iterations (first 2000 discarded)
2628 n.sims = 8000 iterations saved
2629      mean   sd  2.5%   25%   50%   75% 97.5% Rhat n.eff
2630 beta0    -6.62 1.64 -9.77 -7.63 -6.64 -5.63 -3.29    1  4200
2631 beta1     0.81 1.20 -1.63  0.09  0.80  1.54  3.24    1  5100
2632 sigma     4.99 1.56  2.93  3.92  4.66  5.70  8.85    1  8000
2633 tau       0.05 0.03  0.01  0.03  0.05  0.07  0.12    1  8000
2634 deviance 58.72 3.21 55.06 56.35 57.85 60.26 67.15    1  6200
2635
2636 For each parameter, n.eff is a crude measure of effective sample size,
2637 and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
2638
2639 DIC info (using the rule, pD = Dbar-Dhat)
2640 pD = 2.5 and DIC = 61.3
```

2641 In the **WinBUGS** output you see a column called “Rhat”, as well as one  
 2642 called “n.eff”. These are convergence diagnostics (the  $\hat{R}$  or Brooks-Gelman-Rubin  
 2643 statistic and the effective sample size) and we will discuss those in the following  
 2644 section, 3.5.2. DIC is the “deviance information criterion” (Spiegelhalter et al.,  
 2645 2002) (see section 3.9) which some people use in a manner similar to AIC although  
 2646 it is recognized to have some problems in hierarchical models (Millar, 2009). We  
 2647 evaluate this in the context of SCR models in Chapt. 7.

### 3.5 PRACTICAL BAYESIAN ANALYSIS AND MCMC

2648 The mere execution of a Bayesian analysis using the **BUGS** language, as demon-  
 2649 strated with the linear regression example, is fairly straight forward. There are,  
 2650 however, a number of really important practical issues to be considered in any  
 2651 Bayesian analysis and we cover some of these briefly here before we move on to  
 2652 implementing slightly more complex GL(M)Ms in a Bayesian framework.

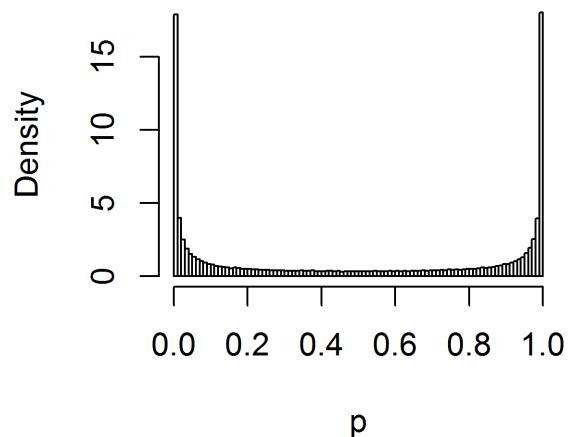
2653 **3.5.1 Choice of prior distributions**

2654 Bayesian analysis requires that we choose prior distributions for all of the structural  
 2655 parameters of the model (we use the term structural parameter to mean all param-  
 2656 eters that aren't customary thought of as latent variables). We will strive to use  
 2657 priors that are meant to express little or no prior information - default or customary  
 2658 "non-informative" or diffuse priors. This will be  $\text{Uniform}(a, b)$  priors for parameters  
 2659 that have a natural bounded support and, for parameters that live on the real line  
 2660 we use either (1) diffuse normal priors, as we did in the linear regression example  
 2661 above; (2) improper uniform priors which have unbounded support, e.g.,  $[\theta] \propto 1$ ,  
 2662 or (3) sometimes even a bounded  $\text{Uniform}(a, b)$  prior, if that greatly improves the  
 2663 performance of **WinBUGS** or other software doing the MCMC for us. In **Win-**  
 2664 **BUGS** a prior with low precision,  $\tau$ , where  $\tau = 1/\sigma^2$ , such as  $\text{Normal}(0, .01)$  will  
 2665 typically be used. Of course  $\tau = 0.01$  ( $\sigma^2 = 100$ ) might be very informative for a  
 2666 regression parameter depending on its magnitude and scaling of  $x$ . Therefore, we  
 2667 recommend that predictor variables *always* be standardized to have mean 0 and  
 2668 variance 1.

2669 **Lack of invariance of priors to transformation.** Clearly there are a lot of  
 2670 choices for ostensibly non-informative priors, and the degree of non-informativeness  
 2671 depends on the parameterization. For example, a natural non-informative prior for  
 2672 the intercept of a logistic regression

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

2673 would be a very diffuse normal prior,  $[\beta_0] = \text{Normal}(0, \text{Large})$  or even  $\beta_0 \sim$   
 2674  $\text{Uniform}(-\text{Large}, \text{Large})$ . However, we might also use a prior on the parameter  
 2675  $p_0 = \text{logit}^{-1}(\beta_0)$ , which is  $\Pr(y=1)$  for the value  $x=0$ . Since  $p_0$  is a probability a  
 2676 natural choice is  $p_0 \sim \text{Uniform}(0, 1)$ . These priors are very different in their impli-  
 2677 cations. For example, if we choose the normal prior for  $\beta_0$  with variance  $\text{Large} = 5^2$   
 2678 and look at the implied prior for  $p_0$  we have the result shown in Fig. 3.2 which  
 2679 looks nothing like a  $\text{Uniform}(0, 1)$  prior. These two priors can affect results (see  
 2680 Sec. 4.4.2 for an illustration of this for a real data set), yet they are both sensible  
 2681 non-informative priors. Despite this, it is often the case that priors will have little  
 2682 or not impact on the results. Choice of priors and parameterization is very much  
 2683 problem-specific and often largely subjective. Moreover, it also affects the behavior  
 2684 of MCMC algorithms and therefore the analyst needs to pay some attention to this



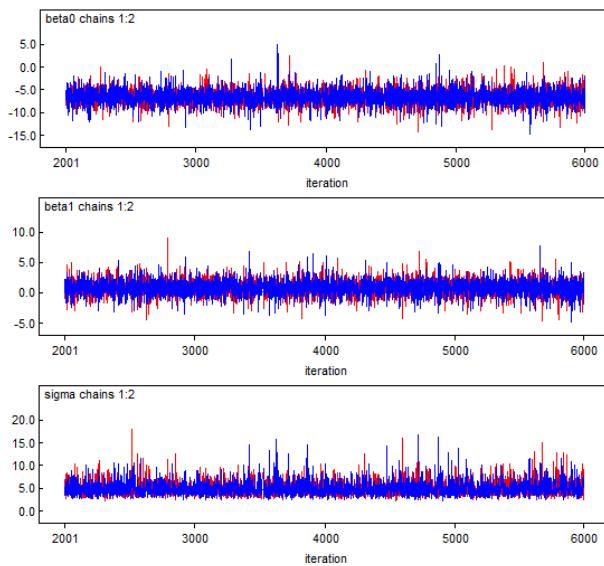
**Figure 3.2.** Implied prior for  $p_0 = \exp(\beta_0)/(1 + \exp(\beta_0))$  if  $\beta_0 \sim \text{Normal}(0, 5^2)$ .

2685 issue and possibly try different things out. Most standard Bayesian analysis books  
2686 address issues related to specification and effect of prior distribution choice in some  
2687 depth. Some good references include Kass and Wasserman (1996), Gelman (2006)  
2688 and Link and Barker (2010).

2689 **3.5.2 Convergence and so-forth**

2690 Once we have carried-out an analysis by MCMC, there are many other practical  
2691 issues that we have to confront. One characteristic of MCMC sampling is that  
2692 Markov chains take some time to converge to their stationary distribution - in our  
2693 case the posterior distribution for some parameter given data,  $[\theta|y]$ . Only when the  
2694 Markov chain has reached its stationary distribution, the generated samples can be  
2695 used to characterize the posterior distribution. Thus, one of the most important  
2696 issues we need to address is “have the chains converged?” Since we do not know  
2697 what the stationary posterior distribution of our Markov chain should look like  
2698 (this is the whole point of doing an MCMC approximation), we effectively have  
2699 no means to assess whether it has truly converged to this desired distribution or  
2700 not. Most MCMC algorithms only guarantee that, eventually, the samples being  
2701 generated will be from the target posterior distribution, but no-one can tell us how  
2702 long this will take. Also, you only know the part of your posterior distribution  
2703 that the Markov chain has explored so far – for all you know the chain could be  
2704 stuck in a local maximum, while other maxima remain completely undiscovered.  
2705 Acknowledging that there is truly nothing we can do to ever prove convergence of  
2706 our MCMC chains, there are several things we can do to increase the degree of  
2707 confidence we have about the convergence of our chains. Some problems are easily  
2708 detected using simple plots, such as a time-series plot, where parameter values of  
2709 each MCMC iteration are plotted against the number of iterations. Fig. 3.3 shows  
2710 the time series plots for the three parameters -  $\beta_0$ ,  $\beta_1$  and  $\sigma$  - from our linear  
2711 regression example, taken from the **WinBUGS** GUI before closing it to return to  
2712 **R**.

2713 Typically a period of transience is observed in the early part of the MCMC  
2714 algorithm, and this is usually discarded as the “burn-in” period. In our linear  
2715 regression example, within the **bugs** call we set the burn-in period as 2000 iterations  
2716 so these are automatically removed by **WinBUGS** and are not part of the output  
2717 (but Fig. 3.6 shows a time-series plot that starts at iteration 0 with a clearly visible  
2718 burn-in period). The quick diagnostic to whether convergence has been achieved  
2719 is that your Markov chains look “grassy” – this seems a reasonable statement  
2720 for the plots in Fig. 3.3. Another way to check convergence is to update the  
2721 parameters some more and see if the posterior changes. Yet another option, and one  
2722 generally implemented in **WinBUGS**, is to run several Markov chains and to start  
2723 them off at different initial values that are overdispersed relative to the posterior  
2724 distribution. Such initial values help to explore different areas of the parameter  
2725 space simultaneously; if after a while all chains oscillate around the same average



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

2726 value, chances are good that they indeed converged to the posterior distribution.  
2727 Gelman and Rubin came up with the so-called “R-hat” statistic ( $\hat{R}$ ) or Brooks-  
2728 Gelman-Rubin statistic that essentially compares within-chain and between-chain  
2729 variance to check for convergence of multiple chains (Gelman et al., 1996).  $\hat{R}$  should  
2730 be close to 1 if the Markov chains have converged and sufficient posterior samples  
2731 have been obtained. For the linear regression example, we ran two parallel chains  
2732 (also specified in the `bugs` call) and **WinBUGS** returns the  $\hat{R}$  statistic for us as  
2733 part of the summary model output. If you look back to Sec. 3.4.1 you see that  
2734  $\hat{R} = 1$  for all parameters of the linear model. In practice,  $\hat{R} = 1.2$  is probably  
2735 good enough for some problems. For some models you can’t actually realize a low  
2736  $\hat{R}$ . E.g., if the posterior is a discrete mixture of distributions then you can be  
2737 misled into thinking that your Markov chains have not converged when in fact the  
2738 chains are just jumping back and forth in the posterior state-space. This happens  
2739 in some of indicator variable model selection discussed in Chapt. 7. Often, when  
2740 there is little information about a parameter in the data, or when parameters are  
2741 on the boundary of the parameter space, convergence will appear to be poor also.  
2742 These kinds of situations are normally ok and you need to think really hard about  
2743 the context of the model and the problem before you conclude that your MCMC  
2744 algorithm is ill-behaved.

2745 Some models exhibit “poor mixing” of the Markov chains (or “slow conver-  
2746 gence”) in which case the samples might well be from the posterior (i.e., the Markov  
2747 chains have converged to the proper stationary distribution) but simply mix or move  
2748 around the posterior rather slowly. Anyway, poor mixing can happen for many rea-  
2749 sons – when parameters are highly correlated (even confounded), or barely identified  
2750 from the data, or the algorithms are very terrible and probably other reasons as  
2751 well.

2752 Slow mixing equates to high autocorrelation in the Markov chain - the successive  
2753 draws are highly correlated, and thus we need to run the MCMC algorithm much  
2754 longer to get an effective sample size that is sufficient for estimation, or to reduce the  
2755 MC error (see below) to a tolerable level. A strategy often used to reduce autocor-  
2756 relation is “thinning” - i.e., keep only every  $m^{th}$  value of the Markov chain output.  
2757 However, thinning is necessarily inefficient from the stand point of inference - you  
2758 can always get more precise posterior estimates by using all of the MCMC output  
2759 regardless of the level of autocorrelation (MacEachern and Berliner, 1994; Link and  
2760 Eaton, 2011). Practical considerations might necessitate thinning, even though it  
2761 is statistically inefficient. For example, in models with many parameters or other  
2762 unknowns being tabulated, the output files might be enormous and unwieldy to  
2763 work with. In such cases, thinning is perfectly reasonable. In many cases, how well  
2764 the Markov chains mix is strongly influenced by parameterization, standardization  
2765 of covariates, and the prior distributions being used. Some things work better than  
2766 others, and the investigator should experiment with different settings and remain  
2767 calm when things don’t work out perfectly.

2768 **Is the posterior sample large enough?** The subsequent samples generated

from a Markov chain are not *iid* 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).

```
2788 > bugs.log('log.txt')$stats
2789 $stats
2790      mean     sd   MCerror    2.5%   median   97.5% start sample
2791 beta0    -6.64700 1.60300 0.0179400 -9.7140 -6.70800 -3.2730 2001 8000
2792 beta1     0.82100 1.19000 0.0116800 -1.4900 0.82560 3.1800 2001 8000
2793 deviance 58.66000 3.08800 0.0506800 55.0700 57.93000 66.8400 2001 8000
2794 sigma     4.96800 1.52300 0.0248300 2.9350 4.68100 8.7410 2001 8000
2795 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 always be different. Thus, as a good rule of thumb, you should avoid reporting MCMC results to more than 2 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,  $\theta^{(t-1)}$ .

2805 **3.5.3 Bayesian confidence intervals**

2806 The 95% Bayesian confidence interval based on percentiles of the posterior is not  
 2807 a unique interval - there are many of them - and the so-called “highest posterior  
 2808 density” (HPD) interval is the narrowest interval that contains *at least* 95% of the  
 2809 posterior mass. As a result (of the *at least* clause), for discrete parameters, the 95%  
 2810 HPD is not often exactly 95% but usually slightly more conservative than nominal.

2811 **3.5.4 Estimating functions of parameters**

2812 A benefit of analysis by MCMC is that we can seamlessly estimate functions of  
 2813 parameters by simply tabulating the desired function of the simulated posterior  
 2814 draws. For example, if  $\theta$  is the parameter of interest and let  $\theta^{(i)}$  for  $i = 1, 2, \dots, M$   
 2815 be the posterior samples of  $\theta$ . Let  $\eta = \exp(\theta)$ , then a posterior sample of  $\eta$  can  
 2816 be obtained simply by computing  $\exp(\theta^{(i)})$  for  $i = 1, 2, \dots, M$ . Almost all SCR  
 2817 models in this book involve at least 1 derived parameter. For example, density  $D$   
 2818 is a derived parameter, being a function of population size  $N$  and the area  $A$  of the  
 2819 underlying state-space of the point process (see Chapt. 5).

2820 **Example: Finding the optimum value of a covariate.** As another example  
 2821 of estimating functions of model parameters, suppose that the normal regression  
 2822 model from Sec. 3.4.1 had a quadratic response function of the form

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

2823 Then the optimum value of  $x$ , i.e., that corresponding to the optimal expected  
 2824 response, can be found by setting the derivative of this function to 0 and solving  
 2825 for  $x$ . We find that

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

2826 yields that  $x_{opt} = -\beta_1/(2 * \beta_2)$ . We can just take our posterior draws for  $\beta_1$  and  
 2827  $\beta_2$  and obtain a posterior sample of  $x_{opt}$  by this simple calculation applied to the  
 2828 posterior output. As an exercise, take the normal model above and simulate a  
 2829 quadratic response and then describe the posterior distribution of  $x_{opt}$ .

**3.6 POISSON GLMS**

2830 The Poisson GLM (also known as “Poisson regression”) is probably the most rel-  
 2831 evant and important class of models in all of ecology. The basic model assumes  
 2832 observations  $y_i; i = 1, 2, \dots, n$  follow a Poisson distribution with mean  $\lambda$  which we  
 2833 write

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

2834 Commonly  $y_i$  is a count of animals or plants at some point in space  $i$ , and  $\lambda$  might  
 2835 depend on  $i$ . For example,  $i$  might index point count locations in a forest, BBS  
 2836 route centers, or sample quadrats, or similar. If covariates are available it is typical

2837 to model them as linear effects on the log mean. If  $x_i$  is some measured covariate  
 2838 associated with observation  $i$ . Then,

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

2839 While we only specify the mean of the Poisson model directly, the Poisson model  
 2840 (and all GLMs) has a “built-in” variance which is directly related to the mean. In  
 2841 this case,  $\text{Var}(y) = \mathbb{E}(y) = \lambda$ . Thus the model accommodates a linear increase in  
 2842 variance with the mean.

### 2843 3.6.1 Example: Breeding Bird Survey Data

2844 As an example we consider a classical situation in ecology where counts of an  
 2845 organism are made at a collection of spatial locations. In this particular example,  
 2846 we have mourning dove (*Zenaida macroura*) counts made along North American  
 2847 Breeding Bird Survey (BBS) routes in Pennsylvania, USA. A route consists of 50  
 2848 stops separated by 0.5 miles. For the purposes here we are defining  $y_i$  = route total  
 2849 count and the sample location will be marked by the center point of the BBS route.  
 2850 The survey is run annually and the data set we have is 1966-1998. BBS data can  
 2851 be obtained online at <http://www.pwrc.usgs.gov/bbs/>, but the particular chunk  
 2852 of data we will be using here is also included in the **scrbook** package (**bbsdata**).  
 2853 We will make use of the whole data set shortly but for now we’re going to focus  
 2854 on a specific year of counts – 1990 – for the sake of building a simple model. For  
 2855 1990 there were 77 active routes, where rows index the unique route, column 1 is  
 2856 the route ID, columns 2-3 are the route coordinates (longitude/latitude), column  
 2857 4 is a habitat covariate “forest cover” (standardized, see below) and the remaining  
 2858 columns are the yearly counts. Years for which a route was not run are coded as  
 2859 “NA” in the data matrix. We imagine that this will be a typical format for many  
 2860 ecological studies, perhaps with more columns representing covariates. To read in  
 2861 the data and display the first few elements of the data frame containing the counts,  
 2862 do this:

```
2863 data(bbsdata) #loads data frame 'bbs'  

2864 bbsdata$counts[1:2,1:6]  

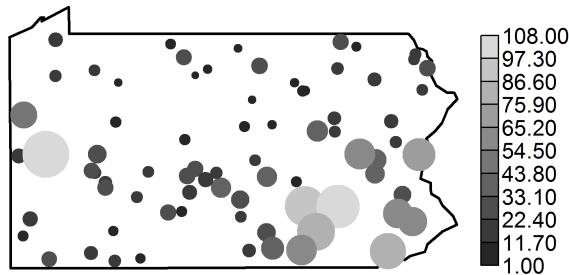
2865      X     lon     lat   habitat X66 X67  

2866 1 72002 -80.445 41.501 -0.3871372 NA  24  

2867 2 72003 -80.347 41.214 -1.0171629 NA  NA
```

2868 It is useful to display the spatial pattern in the observed counts. For that we  
 2869 use a spatial dot plot – where we plot the coordinates of the observations and mark  
 2870 the color of the plotting symbol based on the magnitude of the count. We have a  
 2871 special plotting function for that which is called **spatial.plot()** and it is available  
 2872 with the supplemental **R** package **scrbook**. Actually, what we want to do here is  
 2873 plot the log-counts (+1 of course) which (Fig. 3.4) display a notable pattern that  
 2874 could be related to something. The **R** commands for obtaining this figure are:

```
2875 library("scrbook")
```



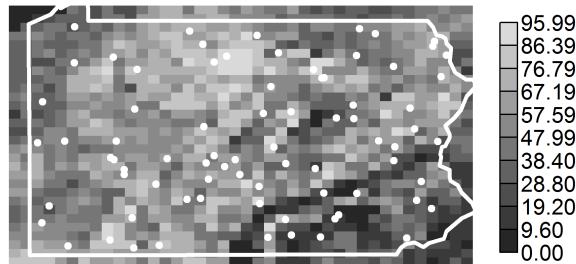
**Figure 3.4.** Plot mourning doves along North American Breeding Bird Survey routes in Pennsylvania (year = 1990). Plot symbol shading and circle size is proportional to raw count.

```

2876 data(bbsdata)
2877 library("maps")
2878
2879 y<-bbsdata$counts[, "X90"] # pick out 1990
2880 notna<-!is.na(y)
2881 y<-y[notna]
2882 locs<-bbsdata$counts[notna,c("lon","lat")]
2883 sz<- y/max(y)
2884
2885 par(mar=c(3,3,3,6))
2886 plot(locs,pch=" ",axes=FALSE,xlim=range(locs[,1])+c(-.3,.3),
2887       ylim=c(range(locs[,2]) + c(-.6,.6)), xlab=" ",ylab=" ")
2888 map('state',regions='pennsylvania',add=TRUE,lwd=2)
2889 spatial.plot(bbsdata$counts[notna,2:3],y,cx=1+sz*6,add=TRUE)

```

We can ponder the potential effects that might lead to dove counts being high....corn fields, telephone wires, barn roofs along with misidentification of pigeons, these could all 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 (provided in the data frame “`bbsdata$habitat`”) which can be plotted using the `spatial.plot` function with the following **R** commands



**Figure 3.5.** Forest cover (percent deciduous) in Pennsylvania.

```

2897 habdata<-bbsdata$habitat
2898 map('state',regions="penn",lwd=2)
2899 I<-matrix(NA,nrow=30,ncol=40)
2900 I<- matrix(habdata[,"dfor"],ncol=40,byrow=FALSE)
2901 ux<-unique(habdata[,2])
2902 uy<-sort(unique(habdata[,3]))
2903
2904 par(mar=c(3,3,3,6))
2905 plot(locs,pch=" ",axes=FALSE,xlim=range(locs[,1])+c(-.3,+.3),
2906       ylim=c(range(locs[,2]) + c(-.6,.6)),xlab=" ",ylab=" ")
2907 image(ux,uy,rot(I),add=TRUE,col=gray(seq(3,17,,10)/20) )
2908 map('state',regions='pennsylvania',add=TRUE,lwd=3,col="white")
2909 image.scale(I,col=gray(seq(3,17,,10)/20) )
2910 points(locs,pch=20,col="white")

```

2911 where the result appears in Fig. 3.5. We see a prominent pattern that indicates  
 2912 high forest coverage in the central part of the state and low forest cover in the SE.  
 2913 Inspecting the previous figure of the raw counts suggests a relationship between  
 2914 counts and forest cover which is perhaps not surprising.

2915 **3.6.2 Doing it in WinBUGS**

2916 Here we demonstrate how to fit a Poisson GLM in **WinBUGS** using the covariate  
 2917  $x_i$  = forest cover along BBS route  $i$ . It is advisable that  $x_i$  be standardized in most  
 2918 cases as this will improve mixing of the Markov chains. We have pre-standardized  
 2919 the forest cover covariate for the BBS route locations, and so we don't have to  
 2920 worry about that here. To read the BBS data into **R** and get things set up for  
 2921 **WinBUGS** we issue the following commands:

```
2922 library("scrbook")
2923 data(bbsdata)
2924
2925 y<-bbsdata$counts[, "X90"] # pick out 1990
2926 notna<-!is.na(y)
2927 y<-y[notna]
2928 ## forest cover already standardized here:
2929 habitat<-bbsdata$counts[notna,"habitat"]
2930 M<-length(y)
2931
2932 library("R2WinBUGS")           # load R2WinBUGS
2933 data <- list ( "y","M","habitat") # bundle data for WinBUGS
```

2934 Now we write out the Poisson model specification in **WinBUGS** pseudo-code,  
 2935 provide initial values, identify parameters to be monitored and then execute **Win-**  
**BUGS**:

```
cat("
model {
  for (i in 1:M){
    y[i] ~ dpois(lam[i])
    log(lam[i])<- beta0+beta1*habitat[i]
  }
  beta0 ~ dunif(-5,5)
  beta1 ~ dunif(-5,5)
}
",file="PoissonGLM.txt")

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

2937 The **WinBUGS** output can be viewed in **R** using the `print` command:

```
2938 print(out,digits=2)
2939 Inference for Bugs model at "PoissonGLM.txt", fit using WinBUGS,
2940 2 chains, each with 6000 iterations (first 2000 discarded), n.thin = 2
2941 n.sims = 4000 iterations saved
2942             mean     sd    2.5%     25%     50%     75%   97.5% Rhat n.eff
```

---

```

2943 beta0      3.15 0.02   3.10   3.13   3.15   3.17   3.20   1  4000
2944 beta1      -0.50 0.02  -0.54  -0.51  -0.50  -0.48  -0.46   1  4000
2945 deviance 1116.56 1.95 1115.00 1115.00 1116.00 1117.00 1122.00   1  4000

```

2946 **3.6.3 Constructing your own MCMC algorithm**

2947 At this point it might be helpful to suffer through an example building a custom MCMC  
 2948 algorithm. Here, we develop an MCMC algorithm for the Poisson regression model, using  
 2949 a Metropolis-within-Gibbs sampling framework. Building MCMC algorithms is covered in  
 2950 more detail in Chapt. 14 where you can also find step-by-step instructions for Metropolis-  
 2951 within-Gibbs samplers, should the following section move through all this material too  
 2952 quickly.

2953 We will assume that the two parameters,  $\beta_0$  and  $\beta_1$ , have diffuse normal priors, say  
 2954  $[\beta_0] = \text{Normal}(0, 100)$  and  $[\beta_1] = \text{Normal}(0, 100)$  where each has *standard deviation* 100  
 2955 (recall that **WinBUGS** parameterizes the normal in terms of  $1/\sigma^2$ ). We need to assem-  
 2956 ble the relevant elements of the model which are these two prior distributions and the  
 2957 likelihood  $[\mathbf{y}|\beta_0, \beta_1] = \prod_i [y_i|\beta_0, \beta_1]$  which is, mathematically, the product of the Poisson  
 2958 pmf evaluated at each  $y_i$ , given particular values of  $\beta_0$  and  $\beta_1$ . Next, we need to identify  
 2959 the full conditionals  $[\beta_0|\beta_1, \mathbf{y}]$  and  $[\beta_1|\beta_0, \mathbf{y}]$ . We use the all-purpose rule for constructing  
 2960 full conditionals (section 3.3.2) to discover that:

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

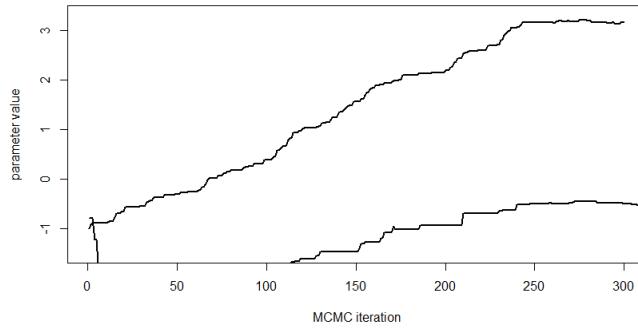
2961 Mathematically, the full conditional is of the form

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

2962 which you can program as an **R** function with arguments  $\beta_0$ ,  $\beta_1$  and  $\mathbf{y}$  without difficulty.  
 2963 The full-conditional for  $\beta_1$  is:

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

2964 which has a similar mathematical representation except the prior is expressed in terms  
 2965 of  $\beta_1$  instead of  $\beta_0$ . Remember, we could replace the “ $\propto$ ” with “=” if we put  $[y|\beta_1]$  or  
 2966  $[y|\beta_0]$  in the denominator. But, in general,  $[y|\beta_0]$  or  $[y|\beta_1]$  will be quite a pain to compute  
 2967 and, more importantly, it is a constant as far as the operative parameters ( $\beta_0$  or  $\beta_1$ ,  
 2968 respectively) are concerned. Therefore, the MH acceptance probability will be the ratio of  
 2969 the full-conditional evaluated at a candidate draw to that evaluated at the current value,  
 2970 and so the denominator required to change  $\propto$  to  $=$  winds up canceling from the MH  
 2971 acceptance probability. Here we will use the so-called random walk candidate generator,  
 2972 which is a Normal proposal distribution, so that, for example,  $\beta_0^* \sim \text{Normal}(\beta_0^t, \delta)$  where  
 2973  $\delta$  is the standard-deviation of the proposal distribution, which is just a tuning parameter  
 2974 that is set by the user and adjusted to achieve efficient mixing of chains (see Sec. 14.2.2).  
 2975 We remark also that calculations are often done on the log-scale to preserve numerical



**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  $\delta = 0.05$ .

integrity of things when quantities evaluate to small or large numbers, so keep in mind, for example,  $a * b = \exp(\log(a) + \log(b))$  for two positive numbers  $a$  and  $b$ . The “Metropolis within Gibbs” algorithm for a Poisson regression turns out to be remarkably simple and is given in Panel 3.1. It is also part of the `scrbook` package and you can run 1000 iterations of it by calling `PoisGLMBBS(y=y, habitat=habitat, niter=1000)` (note that  $y$  = point count data and `habitat` = forest cover have to be defined in your **R** workspace as shown in the previous analysis of these data).

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

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

### 3.7 POISSON GLM WITH RANDOM EFFECTS

In most of this book, we will be dealing with random effects in GLM-like models - similar to what are usually referred to as generalized linear mixed models (GLMMs). We provide

---

```

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

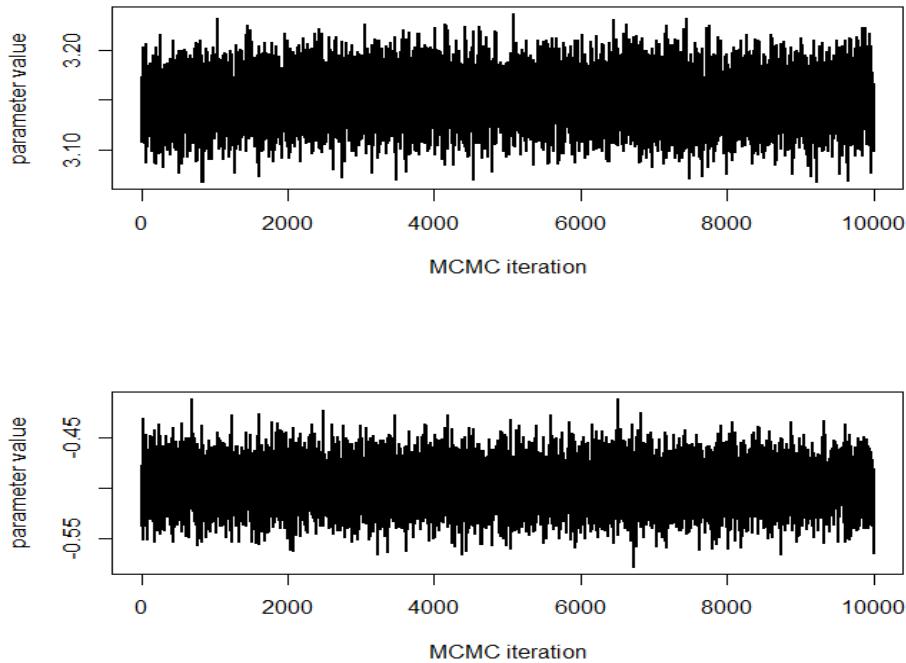
  out[i,]<-c(beta0,beta1) # save the current values
}

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 3.7.** 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  $\delta = 0.05$ .

2999 a brief introduction of such a model by way of example, extending our Poisson regression  
 3000 model to include a random effect.

3001 **The Log-Normal mixture:** The classical situation involves a GLM with a normally  
 3002 distributed random effect that is additive on the linear predictor. For the Poisson case,  
 3003 we have:

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

3004 where  $\eta_i \sim \text{Normal}(0, \sigma^2)$ . In this context,  $\eta$  could represent an error term capturing  
 3005 variation in  $\lambda_i$  not accounted for by the covariates, or overdispersion. It is really amazingly  
 3006 simple to express this model in the **BUGS** language and have **WinBUGS** (or **JAGS**,  
 3007 etc..) draw samples from the posterior distribution. The code for analysis of the BBS  
 3008 dove counts is given as follows:

```
library("scrbook")
data(bbsdata)
```

**Table 3.1.** Inference for Poisson GLM with habitat effect for mourning dove counts across BBS routes in PA, 1990, 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

```

### grab the BBS Data as before
###
### set random seed so that results are repeatable
set.seed(2013)

cat("
model {
  for (i in 1:M){
    y[i] ~ dpois(lam[i])
    log(lam[i])<- beta0+ beta1*habitat[i] + eta[i]
    frog[i]<-beta1*habitat[i] + eta[i]
    eta[i] ~ dnorm(0,tau)
  }

  beta0 ~ dunif(-5,5)
  beta1 ~ dunif(-5,5)
  sigma ~ dunif(0,10)
  tau<-1/(sigma*sigma)
}

",file="model.txt")

data <- list ("y","M","habitat")
inits <- function()
  list ( beta0=rnorm(1),beta1=rnorm(1),sigma=runif(1,0,4))
parameters <- c("beta0","beta1","sigma","tau")
library("R2WinBUGS")                                # load R2WinBUGS library

out<-bugs (data, inits, parameters, "model.txt", n.thin=2,n.chains=2,
n.burnin=1000,n.iter=5000,debug=TRUE)

```

3009        This produces the posterior summary statistics given in table 3.1. One thing we notice  
 3010        is that the posterior standard deviations of the regression parameters are much higher,  
 3011        a result of the extra-Poisson variation allowed for by this model. We would also notice  
 3012        much less precise predictions of hypothetical new observations.

### 3.8 BINOMIAL GLMS

3013 Another extremely important class of models in ecology are binomial models. We use  
 3014 binomial models for count data whenever the observations are counts or frequencies and  
 3015 it is natural to condition on a “sample size”, say  $K$ , the maximum frequency possible in  
 3016 a sample. The random variable,  $y \leq K$ , is then the frequency of occurrences out of  $K$   
 3017 “trials”. The parameter of the binomial models is  $p$ , often called “success probability”  
 3018 which is related to the expected value of  $y$  by  $\mathbb{E}(y) = pK$ . Usually we are interested in  
 3019 modeling covariates that affect the parameter  $p$ , and such models are called binomial GLMs  
 3020 , binomial regression models or logistic regression, although logistic regression really only  
 3021 applies when the logistic link is used to model the relationship between  $p$  and covariates  
 3022 (see below).

3023 One of the most typical binomial GLMs occurs when the sample size equals 1 and the  
 3024 outcome,  $y$ , is “presence” ( $y = 1$ ) or “absence” ( $y = 0$ ) of a species. This is a classical  
 3025 species distribution modeling situation. A special situation occurs when presence/absence  
 3026 is observed with error (MacKenzie et al., 2002; Tyre et al., 2003). In that case,  $K > 1$   
 3027 samples are usually needed for effective estimation of model parameters.

3028 In standard binomial regression problems the sample size is fixed by design but in-  
 3029 teresting models also arise when the sample size is itself a random variable. These are  
 3030 the  $N$ -mixture models (Royle, 2004a; Kéry et al., 2005; Royle and Dorazio, 2008; Kéry,  
 3031 2010) and related models (in this case,  $N$  being the sample size, which we labeled  $K$   
 3032 above)<sup>7</sup>. Another situation in which the binomial sample size is “fixed” is closed popula-  
 3033 tion capture-recapture models in which a population of individuals is sampled  $K$  times.  
 3034 The number of times each individual is encountered is a binomial outcome with parameter  
 3035 (encounter probability)  $p$ , based on a sample of size  $K$ . In addition, the total number of  
 3036 unique individuals observed,  $n$ , is also a binomial random variable based on population  
 3037 size  $N$ . We consider such models in Chapt. 4.

#### 3038 3.8.1 Binomial regression

3039 In binomial models, covariates are modeled on a suitable transformation (the link function)  
 3040 of the binomial success probability,  $p$ . Let  $x_i$  denote some measured covariate for sample  
 3041 unit  $i$  and let  $p_i$  be the success probability for unit or subject  $i$ . The standard choice is the  
 3042 logit link function (3.1) but there are many other possible link functions. We sometimes use  
 3043 the complementary log-log (= “cloglog”) link function in ecological applications because  
 3044 it is natural in some cases when the response should scale in relation to area or effort  
 3045 (Royle and Dorazio, 2008, p. 150). As an example, the “probability of observing a count  
 3046 greater than 0” under a Poisson model is  $\Pr(y > 0) = 1 - \exp(-\lambda)$ . In that case, for the  
 3047  $i^{th}$  observation,

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

3048 so that if you have covariates in your linear predictor for  $\mathbb{E}(y)$  under a Poisson model then  
 3049 they are linear on the complementary log-log link of  $p$ . In models of species occurrence  
 3050 it seems natural to view occupancy as being derived from local abundance  $N$  (Royle

<sup>7</sup>Some of the jargon is actually a little bit confusing here because the binomial index is cus-  
 3048 tomarily referred to as “sample size” but in the context of  $N$ -mixture models  $N$  is actually the  
 3049 “population size”

3051 and Nichols, 2003; Royle and Dorazio, 2006; Dorazio, 2007). Therefore, models of local  
 3052 abundance in which  $N_i \sim \text{Poisson}(A_i\lambda_i)$  for a habitat patch of area  $A_i$  implies a model  
 3053 for occupancy  $\psi_i$  of the form

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

3054 We will use the cloglog link in some analyses of SCR models in Chapt. 5 and elsewhere.

### 3055 3.8.2 Example: Waterfowl Banding Data

3056 The standard binomial modeling problem in ecology is that of modeling species distri-  
 3057 butions, where  $K = 1$  and the outcome is occurrence ( $y = 1$ ) or not ( $y = 0$ ) of some  
 3058 species. Such examples abound in books (e.g., Royle and Dorazio (2008, ch. 3); Kéry  
 3059 (2010, ch. 21); Kéry and Schaub (2012, ch. 13)) and in the literature. Therefore, instead,  
 3060 we will consider an example involving band returns of waterfowl in the upper great plains  
 3061 including some Canadian provinces, which were analyzed by Royle and Dubovsky (2001).

3062 For these data,  $y_{it}$  is the number of mallard (*Anas platyrhynchos*) bands recovered out  
 3063 of  $B_{it}$  birds banded at some location  $s_i$  in year  $t$ . In this case  $B_{it}$  is fixed. Thinking about  
 3064 recovery rate as being proportional to harvest rate, we use these data to explore geographic  
 3065 gradients in recovery rate resulting from variability in harvest pressure experienced by  
 3066 different populations. As such, we fit a basic binomial GLM with a linear response to  
 3067 geographic coordinates (including an interaction term). Here we provide the part of the  
 3068 script for creating the model and fitting the model in **WinBUGS**. There are few structural  
 3069 differences between this model and the Poisson GLM fitted previously. The main things  
 3070 are due to the data structure (we have a matrix here instead of a vector) and otherwise  
 3071 we change the distributional assumption to binomial (specified with **dbin**) and then use  
 3072 the **logit** function to relate the parameter  $p_{it}$  to the covariates.

3073 **Dummy variables in BUGS:** In Chapt. 2 we introduced the concept of categorical  
 3074 variables and how to display them in model formulas in the form of “dummy variables”.  
 3075 In the mallard example, we model the band recovery probability  $p_{it}$  not only as a linear  
 3076 function (on the logit scale) of geographic location, but also allow for variation in  $p_{it}$  with  
 3077 year,  $t$ ;  $t = 1, 2, \dots, T$ . In this particular example there are  $T = 5$  years of data and if we  
 3078 wanted to describe the full mallard model with a formula adopting the dummy variable  
 3079 format, it would look like this:

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

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

3080 Here,  $x_2$  to  $x_5$  are the dummy variable vectors of length  $T$  that take on the value of 1  
 3081 when  $t$  corresponds to the respective year and 0 otherwise.  $\beta_0$  is the common intercept  
 3082 term and correspond to  $t = 1$ ;  $\beta_1 - \beta_4$  describe the difference in  $p_{it}$  for each  $t$  relative to  
 3083  $t = 1$ .

3084 There is a more concise way of implementing such a model with a categorical covariate  
 3085 in **BUGS**, namely, by using indexing instead of dummy variables<sup>8</sup>. Essentially, instead of

<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)

3086 estimating the difference in  $p$  relative to category 1, we estimate a separate intercept term  
 3087 for each category, so that we have 5 different  $\beta_0$  parameters indexed by  $t$ . This reduces  
 3088 the linear predictor to:

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

3089 The model can be implemented in the **BUGS** language for the mallard banding data  
 3090 using the following **R** script, provided in the **scrbook** package (see **help(mallard)**):

```
library("scrbook")
data(mallard)      # load mallard data

cat("
model {
  for(t in 1:5){
    for (i in 1:nobs){
      y[i,t] ~ dbin(p[i,t], B[i,t])
      pl[i,t]<-beta0[t]+beta1*X[i,1]+beta2*X[i,2]+beta3*X[i,1]*X[i,2]
      p[i,t]<-exp(pl[i,t])/(1+exp(pl[i,t]))
    }
  }
  beta1 ~ dnorm(0,.001)
  beta2 ~ dnorm(0,.001)
  beta3 ~ dnorm(0,.001)
  for(t in 1:5){
    beta0[t] ~ dnorm(0,.001)
  }
}
",file="BinomialGLM.txt")

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

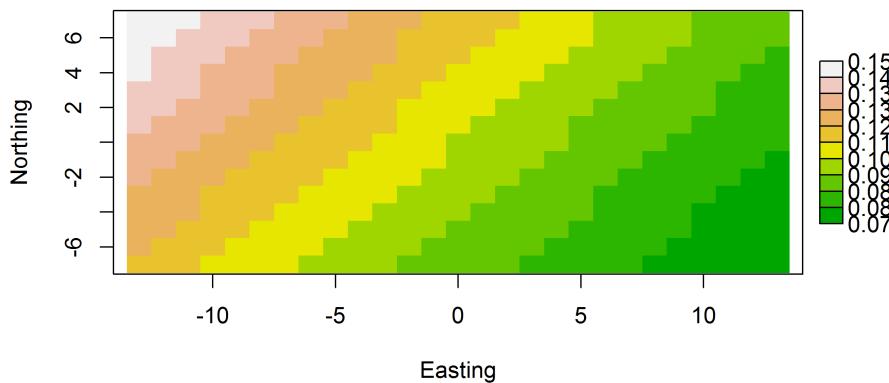
3091 Look at the posterior summaries of model parameters in Table 3.2. The basic result  
 3092 suggests a negative east-west gradient and a positive south to north gradient of band  
 3093 recovery probabilities, but no interaction. A map of the response surface is shown in Fig.  
 3094 3.8.

### 3.9 MODEL CHECKING AND SELECTION

3095 In general terms model checking – or assessing the adequacy of the model – and model  
 3096 selection are quite thorny issues and, despite contrary and, sometimes, strongly held belief  
 3097 among practitioners, there are not really definitive, general solutions to either problem.

**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
beta0[1]	-2.346	0.036	-2.417	-2.346	-2.277	1.001	1500
beta0[2]	-2.356	0.032	-2.420	-2.356	-2.292	1.001	1500
beta0[3]	-2.220	0.035	-2.291	-2.219	-2.153	1.001	1500
beta0[4]	-2.144	0.039	-2.225	-2.143	-2.068	1.000	1500
beta0[5]	-1.925	0.034	-1.990	-1.924	-1.856	1.004	570
beta1	-0.023	0.003	-0.028	-0.023	-0.018	1.001	1500
beta2	0.020	0.006	0.009	0.020	0.031	1.001	1500
beta3	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



**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.

3098 We're against dogma on these issues and think people need to be open-minded about  
 3099 such things and recognize that models can be useful whether or not they pass certain  
 3100 statistical tests. Some models are intrinsically better than others because they make more  
 3101 biological sense or foster understanding or achieve some objective that some bootstrap or  
 3102 other goodness-of-fit test can't decide for you. That said, it gives you some confidence if  
 3103 your model seems adequate in a purely statistical sense. We provide a very brief overview  
 3104 of concepts here, but provide more detailed coverage in Chapt. 7. See also coverage of  
 3105 these topics in Kéry (2010) and Link and Barker (2010) for specific context related to  
 3106 Bayesian model checking and selection.

3107 **3.9.1 Goodness-of-fit**

3108 Goodness-of-fit testing is an important element of any analysis because our model rep-  
 3109 presents a general set of hypotheses about the ecological and observation processes that  
 3110 generated our data. Thus, if our model "fits" in some statistical or scientific sense, then  
 3111 we believe it to be consistent with the hypotheses that went into the model. More for-  
 3112 mally, we would conclude that the data are *not inconsistent* with the hypotheses, or that  
 3113 the model appears adequate. If we have enough data, then of course we will reject any  
 3114 set of statistical hypotheses. Conversely, we can always come up with a model that fits  
 3115 by making the model extremely complex. Despite this paradox, it seems to us that sim-  
 3116 ple models that you can understand should usually be preferred even if they don't fit,  
 3117 for example if they embody essential mechanisms central to our understanding of things,  
 3118 or if we think that some contributing factors to lack-of-fit are minor or irrelevant to the  
 3119 scientific context and intended use of the model. In other words, models can be useful  
 3120 irrespective of whether they fit according to some formal statistical test of fit. Yet the  
 3121 tension is there to obtain fitting models, and this comes naturally at the expense of models  
 3122 that can be easily interpreted and studied and effectively used. Unfortunately, conduct-  
 3123 ing a goodness-of-fit test is not always so easy to do. And, moreover, it is never really  
 3124 easy (or especially convenient) to decide if your goodness-of-fit test is worth anything. It  
 3125 might have 0 power! Despite this, we recommend attempting to assess model fit in real  
 3126 applications, as a general rule, and we provide some basic guidance here and some more  
 3127 specific to SCR models in Chapt. 7.

3128 To evaluate goodness-of-fit in Bayesian analyses, we will most often use the Bayesian  
 3129 p-value (Gelman et al., 1996). The basic idea is to define a fit statistic or "discrepancy  
 3130 measure" and compare the posterior distribution of that statistic to the posterior predictive  
 3131 distribution of that statistic for hypothetical perfect data sets for which the model is known  
 3132 to be correct. For example, with count frequency data, a standard measure of fit is the  
 3133 sum of squares of the "Pearson residuals",

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

3134 The fit statistic based on the squared residuals computed from the observations is

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

3135 which can be computed at each iteration of a MCMC algorithm given the current values  
 3136 of parameters that determine the response distribution. At the same time (i.e., at each

3137 MCMC iteration), the equivalent statistic is computed for a “new” data set, say  $\mathbf{y}^{new}$ ,  
 3138 simulated using the current parameter values. From the new data set, we compute the  
 3139 same fit statistic:

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

3140 and the Bayesian p-value is simply the posterior probability  $\Pr(T(\mathbf{y}^{new}) > T(\mathbf{y}))$  which  
 3141 should be close to 0.50 for a good model – one that “fits” in the sense that the observed  
 3142 data set is consistent with realizations simulated under the model being fitted to the  
 3143 observed data. In practice we judge “close to 0.50” as being “not too close to 0 or 1” and,  
 3144 as always, closeness is somewhat subjective. We’re happy with anything  $> .1$  and  $< .9$   
 3145 but might settle for  $> .05$  and  $< 0.95$ . Another useful fit statistic is the Freeman-Tukey  
 3146 statistic, in which

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

3147 (Brooks et al., 2000), where  $y_i$  is the observed value of observation  $i$  and  $\mathbb{E}(y_i)$  its expected  
 3148 value. In contrast to a Chi-square discrepancy, the Freeman-Tukey statistic removes the  
 3149 need to pool cells with small expected values. In summary, you can see that the Bayesian  
 3150 p-value is easy to compute, and it is widely used as a result.

### 3151 3.9.2 Model Selection

3152 In ecology, scientific hypotheses are often manifest as different models or parameters of  
 3153 a model, and so evaluating the importance of different models is fundamental to many  
 3154 ecological studies. For model selection we typically use three different methods: First is,  
 3155 let’s say, common sense. If a variable should plausibly be relevant to explaining the data-  
 3156 generating processes, and it has posterior mass concentrated away from 0, then it seems  
 3157 like it should be regarded as important - that is, it is “significant.” This approach seems  
 3158 to have fallen out of favor in ecology over the last 10 or 15 years but in many situations  
 3159 it is a reasonable thing to do.

3160 For regression problems we sometimes use the indicator variable method of Kuo and  
 3161 Mallick (1998), in which we introduce a set of binary variables  $I_k$  for variable  $k$ , and  
 3162 express the model as, e.g., for a single covariate model:

$$\mathbb{E}(y_i) = \beta_0 + I_1 \beta_1 x_i$$

3163 where  $I_1$  is given a Bernoulli prior distribution with some prescribed probability. E.g.,  
 3164  $I_1 \sim \text{Bernoulli}(0.50)$  to provide a prior probability of 0.50 that variable  $x$  should be an  
 3165 element of the linear predictor. The posterior probability of the event  $I_1 = 1$  is a gage of  
 3166 the importance of the variable  $x$ . i.e., high values of  $\Pr(I_1 = 1)$  indicate stronger evidence  
 3167 to support that “ $x$  is in the model” whereas values of  $\Pr(I_1 = 1)$  close to 0 suggest that  
 3168  $x$  is less important. Expansion of the model to include the binary variable  $I_1$  defines a  
 3169 set of 2 distinct models for which we can directly compute the posterior probabilities for,  
 3170 merely by tallying up the posterior frequency of  $I_1$ . See Royle and Dorazio (2008, Chapt.  
 3171 3) for an example in the context of logistic regression.

3172 This approach seems to even work sometimes with fairly complex hierarchical models  
 3173 of a certain form. E.g., Royle (2008) applied it to a random effects model to evaluate the  
 3174 importance of the random effect component of the model. The main problem, which is

3175 really a general problem in Bayesian model selection, is that its effectiveness and results  
3176 will typically be highly sensitive to the prior distribution on the structural parameters  
3177 (e.g., see Royle and Dorazio (2008, table 3.6)). The reason for this is obvious: If  $I_1 = 0$   
3178 for the current iteration of the MCMC algorithm, so that  $\beta$  is sampled from the prior  
3179 distribution, and the prior distribution is very diffuse, then extreme values of  $\beta$  are likely.  
3180 Consequently, when the current value of  $\beta$  is far away from the mass of the posterior when  
3181  $I_1 = 1$ , then the Markov chain may only jump from  $I_1 = 0$  to  $I_1 = 1$  infrequently. One  
3182 seemingly reasonable solution to this problem is to fit the full model to obtain posterior  
3183 distributions for all parameters, and then use those as prior distributions in a “model  
3184 selection” run of the MCMC algorithm (Aitkin, 1991). This seems preferable to more-or-  
3185 less arbitrary restriction of the prior support to improve the performance of the MCMC  
3186 algorithm.

3187 A third method that we advocate is subject-matter context. It seems that there are  
3188 some situations – some models – where one should not have to do model selection because  
3189 it is necessitated by the specific context of the problem, thus rendering a formal hypothesis  
3190 test pointless (Johnson, 1999). Certain aspects of SCR models are such an example. In  
3191 SCR models, we will see that “spatial location” of individuals is an element of the model.  
3192 The simpler, reduced, model is an ordinary capture-recapture model which is not spatially  
3193 explicit (i.e., Chapt. 4), but it seems silly and pointless to think about actually using the  
3194 reduced model even if we could concoct some statistical test to refute the more complex  
3195 model. The simpler model is manifestly wrong but, more importantly, not even a plausible  
3196 data-generating model! Other examples are when effort, area or sample rate is used as a  
3197 covariate. One might prefer to have such things in models regardless of whether or not  
3198 they pass some statistical litmus test.

3199 Many problems can be approached using one of these methods but there are also  
3200 broad classes of problems that can't and, for those, you're on your own. In later chapters  
3201 (especially Chapt. 7) we will address model selection in specific contexts and we hope  
3202 those will prove useful for a majority of the situations you might encounter.

### 3.10 SUMMARY AND OUTLOOK

3203 GLMs and GLMMs are the most useful statistical methods in all of ecology. The principles  
3204 and procedures underlying these methods are relevant to nearly all modeling and analysis  
3205 problems in every branch of ecology. Therefore, understanding how to analyze these mod-  
3206 els is an essential skill for the quantitative ecologist to possess. If you understand and can  
3207 conduct classical likelihood and Bayesian analysis of Poisson and binomial GLM(M)s, then  
3208 you will be successful analyzing and understanding more complex classes of models that  
3209 arise. We will see shortly that spatial capture-recapture models are a type of GLMM and  
3210 thus having a basic understanding of the conceptual origins and formulation of GLM(M)s  
3211 and their analysis is extremely useful.

3212 We note that GLM(M)s are routinely analyzed by likelihood methods but we have fo-  
3213 cused on Bayesian analysis here in order to develop the tools that are less familiar to most  
3214 ecologists, and that we will apply in much of the remainder of the book. In particular,  
3215 Bayesian analysis of models with random effects is relatively straightforward because the  
3216 models are easy to analyze conditional on the random effect, using methods of MCMC.  
3217 Thus, we will often analyze SCR models in later chapters by MCMC, explicitly adopting a

3218 Bayesian inference framework. In that regard, the various **BUGS** engines (**WinBUGS**,  
3219 **OpenBUGS**, **JAGS**; see also V) are enormously useful because they provide an accessible  
3220 platform for carrying out analyses by MCMC by just describing the model, and not  
3221 having to worry about how to actually build MCMC algorithms. That said, the **BUGS**  
3222 language is more important than just to the extent that it enables one to do MCMC - it  
3223 is useful as a modeling tool because it fosters understanding, in the sense that it forces  
3224 you to become intimate with your model. You have to think about and write down all  
3225 of the probability assumptions, and the relationships between observations and latent  
3226 variables and parameters in a way that is ecologically sensible and statistically coherent.  
3227 Because of this, that it focuses your thinking on *model construction*, as M. Kéry says in  
3228 his **WinBUGS** book (Kéry, 2010), “**WinBUGS** frees the modeler in you.”

3229 While we have emphasized Bayesian analysis in this chapter, and make primary use of  
3230 it through the book, we will provide an introduction to likelihood analysis in Chapt. 6  
3231 and use those methods also from time to time. Before getting to that, however, it will be  
3232 useful to talk about more basic, conventional closed population capture-recapture models  
3233 and such models are the topic of the next chapter.



3234  
3235

# 4

3236

## CLOSED POPULATION MODELS

3237 In this chapter we introduce ordinary *non-spatial* capture-recapture (CR) models for esti-  
3238 mating population size in closed populations. A closed population is one whose abundance  
3239  $N$  does not change during the study. Two forms of closure are often discussed: demo-  
3240 graphic closure, meaning that no births or deaths occur, and geographic closure, which  
3241 states that no individuals move onto or off of the sampled area during the study. Few  
3242 populations are actually closed except during very short time intervals, but closed popula-  
3243 tion CR models serve as the basis for the development of the rest of the models presented  
3244 in this book, including the models for open populations discussed in Chapt.15.

3245 We will see that classical closed population CR models are closely related to binomial  
3246 (or logistic) regression-type models. In fact, when  $N$  is known, they are precisely logistic  
3247 regression models. We consider some important extensions of ordinary closed population  
3248 models that accommodate various types of “individual effects” — either in the form of  
3249 explicit, observed covariates (sex, age, body mass) or unstructured “heterogeneity” in the  
3250 form of an individual random effect, which represent effects of unobserved or unmeasured  
3251 covariates. In general, these models are variations of generalized linear or generalized linear  
3252 mixed models (GLMs and GLMMs, respectively). Because of the paramount importance  
3253 of this concept, we focus mainly on fairly simple models in which the observations are  
3254 individual encounter frequencies,  $y_i$  = the number of encounters of individual  $i$  out of  $K$   
3255 replicate samples of the population which, for the models we consider here, is the outcome  
3256 of a binomial random variable. We begin with the most basic capture-recapture model,  
3257 colloquially referred to as “model  $M_0$ ” (Otis et al., 1978), in which encounter probability  
3258 is strictly constant in all respects (across individuals, and replicates). We then consider  
3259 extensions of that model to include individual heterogeneity and individual covariates. A  
3260 special type of individual covariate models is distance sampling, which could be thought  
3261 of as the most primitive spatial capture-recapture model. In this chapter, we hope to  
3262 establish the methodological linkage between non-spatial and spatial capture-recapture  
3263 models which we formalize in the next chapter. Along the way, we review some of the  
3264 informal methods of estimating density using CR methods, and consider some of their  
3265 limitations.

3266 We emphasize Bayesian analysis of capture-recapture models and we accomplish this  
 3267 using a method related to classical “data augmentation” from the statistics literature (e.g.,  
 3268 Tanner and Wong, 1987). This is a general concept in statistics but, in the context of  
 3269 capture-recapture models where  $N$  is unknown, it has a consistent implementation across  
 3270 classes of capture-recapture models and one that is really convenient from the standpoint  
 3271 of doing MCMC (Royle et al., 2007; Royle and Dorazio, 2012). We use data augmentation  
 3272 throughout this book and thus emphasize its conceptual and technical origins and demon-  
 3273 strate applications to closed population models. We refer the reader to Kéry and Schaub  
 3274 (2012, ch. 6) for an accessible and complementary development of Bayesian analysis of  
 3275 ordinary, i.e., nonspatial closed population models.

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

3276 To start looking at the simplest capture-recapture model, let’s suppose there exists a pop-  
 3277 ulation of  $N$  individuals which we subject to repeated sampling, say over  $K$  “occasions”,  
 3278 such as trap nights, where individuals are captured, marked, released, and subsequently  
 3279 recaptured. We suppose that individual encounter histories are obtained, and these are of  
 3280 the form of a sequence of 0’s and 1’s indicating capture ( $y = 1$ ) or not ( $y = 0$ ) during any  
 3281 sampling occasion. As an example, suppose  $K = 5$  sampling occasions, then an individual  
 3282 captured during occasion 2 and 3 but not otherwise would have an encounter history of  
 3283 the form  $\mathbf{y} = (0, 1, 1, 0, 0)$ . Thus, the observation  $\mathbf{y}_i$  for each individual ( $i = 1, 2, \dots, N$ )  
 3284 is a vector having elements denoted by  $y_{ik}$  for  $k = 1, 2, \dots, K$ . Usually this is organized  
 3285 as a row of a matrix with elements  $y_{ik}$ , see Table 4.1. Except where noted explicitly,  
 3286 we suppose that observations are independent within individuals and among individuals.  
 3287 Formally, this allows us to say that  $y_{ik}$  are independent and identically distributed (“iid”)  
 3288 Bernoulli random variables and we may write  $y_{ik} \sim \text{Bern}(p)$ . Consequently, for this very  
 3289 simple model in which  $p$  is constant (i.e., there are no individual or temporal covariates  
 3290 that affect  $p$ ) the original binary detection variables can be aggregated into total encounter  
 3291 frequencies for each individual (total number of captures),  $y_i = \sum_k y_{ik}$ , and the obser-  
 3292 vation model changes from a Bernoulli distribution to a binomial distribution based on a  
 3293 sample of size  $K$ . That is

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

3294 for every individual in the population  $i = 1, 2, \dots, N$ , where  $N$  is the number of individuals  
 3295 in the population (i.e., population size).

3296 We emphasize the central importance of the basic Bernoulli encounter model – an  
 3297 individual is either encountered in a sample, or not – which forms the cornerstone of  
 3298 almost all of classical capture-recapture models, including many spatial capture-recapture  
 3299 models discussed in this book.

3300 Evidently, the basic capture-recapture model is a simplistic version of a logistic-  
 3301 regression model with only an intercept term ( $\text{logit}(p) = \text{constant}$ ). To say that all  
 3302 capture-recapture models are just logistic regressions is only slightly inaccurate. In fact,  
 3303 we are proceeding here as if we knew  $N$ . In practice we don’t, of course, and estimating  $N$   
 3304 is actually the central objective. But, by proceeding as if  $N$  were known, we can specify  
 3305 a simple model and then deal with the fact that  $N$  is unknown using standard methods  
 3306 that you are already familiar with (i.e., GLMs - see Chapt. 3).

**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

Assuming individuals in the population are observed independently, the joint probability distribution of the observations is the product of  $N$  binomials

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

We emphasize that this expression is conditional on  $N$ , in which case we get to observe the  $y_i = 0$  observations and the resulting data are just *iid* binomial counts. Because this is a binomial regression model of the variety described in Chapt. 3, fitting this model using a **BUGS** engine poses no difficulty.

Equation 4.1.1 can be simplified even further if we reformat the observations as capture frequencies, which are the sufficient statistics under this model. Specifically, let  $n_k$  denote the number of individuals captured exactly  $k$  times after  $K$  survey occasions,  $n_k = \sum_{i=1}^N I(y_i = k)$  where  $I$  is the indicator function evaluating to 1 if its argument is true and 0 otherwise. For sake of illustration, we converted the data from Table 4.1 to this format (Table 4.2). What is important to note is that if we know  $N$ , then we known  $n_0$ , i.e. the number of individuals not captured at all. In this case, an alternative and equivalent expression to Eq. 4.1.1 is

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

where  $\pi_k = \Pr(y = k)$  under the binomial model with parameter  $p$  and sample size  $K$ .

**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

The essential problem in capture-recapture, however, is that  $N$  is *not* known because the number of uncaptured individuals ( $n_0$ ) is unknown. Consequently, the observed

3324 capture frequencies  $n_k$  are no longer independent because  $n_0$  is a function of the other  
 3325 frequencies,  $n_0 = N - \sum_{k=1}^K n_k$ . Hence, their joint distribution is multinomial (e.g., see  
 3326 Illian et al. (2008, p. 61)):

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

3327 We give a general overview of the multinomial distribution in Sec. 2.2. The multinomial  
 3328 distribution is the standard model for discrete responses that can fall into a fixed num-  
 3329 ber ( $K + 1$  in this case) of possible categories. In the context of capture-recapture, the  
 3330 multinomial posits a population of  $N$  individuals with  $K + 1$  possible outcomes defined  
 3331 by the possible encounter frequencies: captured  $y = 1, 2, \dots, K$  times or not captured at  
 3332 all. These possible outcomes occur with probabilities  $\pi_k$ , which we refer to as “cell prob-  
 3333 abilities” or in the specific context of capture-recapture, encounter history probabilities.

3334 To fit the model in which  $N$  is *unknown*, we can regard  $n_0$  as a parameter and maximize  
 3335 the multinomial likelihood directly. Direct likelihood analysis of the multinomial model is  
 3336 straightforward, but that is not always sufficiently useful in practice because we seldom  
 3337 are concerned with models for the aggregated encounter history frequencies, which entail  
 3338 that capture probabilities are the same for all individuals. In many instances, including  
 3339 for spatial capture-recapture (SCR) models, we require a formulation of the model that  
 3340 can accommodate individual-level covariates to account for differences in detection among  
 3341 individuals which we address subsequently in this chapter and also in Chapt. 8.

### 3342 4.1.1 The Core Capture-Recapture Assumptions

3343 This basic capture-recapture model – model  $M_0$  – comes with it a host of specific biological  
 3344 and statistical assumptions. In addition to the basic assumption of population closure,  
 3345 Otis et al. (1978) list the following:

- 3346 1. animals do not lose their marks during the experiment,
- 3347 2. all marks are correctly noted and recorded at each trapping occasion, and
- 3348 3. each animal has a constant and equal probability of capture on each trapping oc-  
 3349 casion.

3350 The remainder of their classic work is dedicated to relaxing assumption 3. While assump-  
 3351 tions 1 and 2 are undoubtedly necessary for inference from basic CR methods to be valid,  
 3352 and while they are also assumed by most of the models we present in the following chap-  
 3353 ters, we refrain from repeatedly making such statements. Our opinion is that all model  
 3354 assumptions are apparent when a model is clearly specified, and it is both redundant and  
 3355 impossible to list all the things not allowed by the model. For example, closed population  
 3356 models also assume that other sources of data entry do not occur, but it is not necessary  
 3357 to enumerate each possibility. Rather, it is necessary to make clear statements such as

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

3358 This simple model description carries a tremendous amount of information, and it leaves  
 3359 very little left to say with respect to assumptions. Although we will not always show  
 3360 the *iid* symbol, it will be assumed unless otherwise noted, and this assumption is critical  
 3361 for valid inference. It implies that the encounter of one individual does not affect the

3362 encounter of another individual. Under this assumption, it is easy to write down the  
 3363 likelihood of the parameters and obtain parameter estimates; however, whether or not it  
 3364 is true depends upon biological and sampling issues. If this assumption is deemed false,  
 3365 the model can be discarded in favor of a more realistic alternative. However, once we have  
 3366 settled on our model, statistical inference proceeds by assuming the model is truth—not  
 3367 an approximation to truth—but actual truth.

3368 In spite of the fact that we assume that all models are truth, but we acknowledge that  
 3369 all models are wrong due to their assumptions, assumptions should not be viewed as a  
 3370 necessary evil. In fact, one way to view assumptions is as embodiments of our ecological  
 3371 hypotheses. If we make these assumptions too complex or too specific, then we will never  
 3372 be able to study general phenomenon that hold true across space and time. Furthermore,  
 3373 in practice, we will rarely have enough data to estimate the parameters of highly complex  
 3374 models.

#### 3375 4.1.2 Conditional likelihood

3376 We saw that the closed population model is a simple logistic regression model if  $N$  is known  
 3377 and, when  $N$  is unknown, the model is multinomial with index or sample size parameter  
 3378  $N$ . This multinomial model, being conditional on  $N$ , is sometimes referred to as the “joint  
 3379 likelihood” the “full likelihood” or the “unconditional likelihood” (sometimes “model” in  
 3380 place of “likelihood”) (Sanathanan, 1972; Borchers et al., 2002). This formulation differs  
 3381 from the so-called “conditional likelihood” approach in which the likelihood of the observed  
 3382 encounter histories is devised conditional on the event that an individual is captured at  
 3383 least once. To construct this likelihood, we have to recognize that individuals appear or not  
 3384 in the sample based on the value of the random variable  $y_i$ , that is, we capture them if and  
 3385 only if  $y_i > 0$ . The observation model is therefore based on  $\Pr(y|y > 0)$ . For the simple  
 3386 case of model  $M_0$ , the resulting conditional distribution is a “zero truncated” binomial  
 3387 distribution which accounts for the fact that we cannot observe the value  $y = 0$  in the  
 3388 data set. Both the conditional and unconditional models are legitimate modes of analysis  
 3389 in all capture-recapture types of studies, and they provide equally valid descriptions of  
 3390 the data and for many practical purposes provide equivalent inferences, at least in large  
 3391 sample sizes (Sanathanan, 1972).

3392 In this book we emphasize Bayesian analysis of capture-recapture models using data  
 3393 augmentation (described in sec. 4.2 below), which produces yet a third distinct formu-  
 3394 lation of capture-recapture models based on the zero-*inflated* binomial distribution that  
 3395 we describe in the next section. Thus, there are 3 distinct formulations of the model – or  
 3396 modes of analysis – for analyzing all capture-recapture models based on the (1) binomial  
 3397 model for the joint or unconditional specification; (2) zero-truncated binomial that arises  
 3398 “conditional on  $n$ ”; and (3) the zero-inflated binomial that arises under data augmen-  
 3399 tation. Each formulation has distinct model parameters (shown in Table 4.3 for model  
 3400  $M_0$ ).

## 4.2 DATA AUGMENTATION

3401 We consider a method of analyzing closed population models using parameter-expanded  
 3402 data augmentation (PX-DA), which we abbreviate to “data augmentation” or DA, which

**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

is useful for Bayesian analysis and, in particular, analysis of models using the various BUGS engines and other Bayesian model fitting software. Data augmentation is a general statistical concept that is widely used in statistics in many different settings. The classical reference is Tanner and Wong (1987), but see also Liu and Wu (1999). Data augmentation can be adapted to provide a very generic framework for Bayesian analysis of capture-recapture models with unknown  $N$ . This idea was introduced for closed populations by Royle et al. (2007), and has subsequently been applied to a number of different contexts including individual covariate models (Royle, 2009a), open population models (Royle and Dorazio, 2008, 2012; Gardner et al., 2010a), spatial capture-recapture models (Royle and Young, 2008; Royle, 2010; Gardner et al., 2009), and many others. Kéry and Schaub (2012, Chapt. 6) provides a good introduction to data augmentation in the context of closed population models.

Conceptually, the technique of data augmentation represents a reparameterization of the “complete data” model – i.e., that conditional on  $N$ . The reparameterization is achieved by embedding this data set into a larger data set having  $M > N$  “rows” (individuals) and re-expressing the model conditional on  $M$  instead of  $N$ . The great thing about data augmentation is that we do not need to know  $N$  for this reparameterization. Although this has a whiff of arbitrariness or even outright ad hockery to it, in the choice of  $M$ , it is always possible, in practice, to choose  $M$  pretty easily for a given problem and context and results will be insensitive to choice of  $M^1$ . Then, under data augmentation, analysis is focused on the “augmented data set.” That is, we analyze the bigger data set – the one having  $M$  rows – with an appropriate model that accounts for the augmentation. This is achieved by a Bernoulli sampling process that determines whether an individual in  $M$  is also a member of  $N$ . Inference is focused directly on estimating the proportion  $\psi = E[N]/M$ , instead of directly on  $N$ , where  $\psi$  is the “data augmentation parameter.”

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

There is a close correspondence between so-called “occupancy” models and closed population models Royle and Dorazio (see 2008, sec. 5.6). In occupancy models (MacKenzie et al., 2002; Tyre et al., 2003) the sampling situation is that  $M$  sites, or patches, are sampled multiple times to assess whether a species occurs at the sites. This yields encounter data such as that illustrated in the left panel of Table 4.4. The important problem is that a species may occur at a site, but go undetected, yielding an all-zero encounter history

<sup>1</sup>Unless the data set is sufficiently small that parameters are weakly identified

3435 for the site, which in the case of occupancy studies, are *observed*. However, some of the  
 3436 zero vectors will typically correspond to sites where the species in fact *does* occur. Thus,  
 3437 while the zeros are observed, there are too many of them and, in a sense, the inference  
 3438 problem is to partition the zeros into “structural” (fixed) and “sampling” (or stochastic)  
 3439 zeros, where the former are associated with unoccupied and the latter with occupied sites  
 3440 where the species went undetected. More formally, inference is focused on the parameter  
 3441  $\psi$ , the probability that a site is occupied.

3442 In contrast to occupancy studies, in classical closed population studies, we observe a  
 3443 data set as in the middle panel of Table 4.4 where *no* zeros are observed. The inference  
 3444 problem is, essentially, to estimate how many sampling zeros there are – or should be – in  
 3445 a “complete” data set. This objective (how many sampling zeros?) is precisely the same  
 3446 for both types of problems if an upper limit  $M$  is specified for the closed population model.  
 3447 The only distinction being that, in occupancy models,  $M$  is set by design (i.e., the number  
 3448 of sites in the sample), whereas a natural choice of  $M$  for capture-recapture models may  
 3449 not be obvious. However, the choice of  $M$  induces a uniform prior for  $N$  on the integers  
 3450  $[0, M]$  (Royle et al., 2007). Then, one can analyze capture-recapture models by adding  
 3451  $M - n$  all-zero encounter histories to the data set and regarding the augmented data  
 3452 set, essentially, as a site-occupancy data set, where the occupancy or data augmentation  
 3453 parameter ( $\psi$ ) takes the place of the abundance parameter ( $N$ ).

3454 Thus, the heuristic motivation of data augmentation is to fix the size of the data  
 3455 set by adding *too many* all-zero encounter histories to create the data set shown in the  
 3456 right panel of Table 4.4, and then analyze the augmented data set using an occupancy  
 3457 type model which includes both “unoccupied sites” (in capture-recapture, augmented  
 3458 individuals that are not members of the real population that was sampled) as well as  
 3459 “occupied sites” (in capture-recapture, individuals that are members of the population  
 3460 but that were undetected by sampling) at which detections did not occur. We call these  
 3461  $M - n$  all-zero histories “potential individuals” because they exist to be recruited (in a  
 3462 non-biological sense) into the population, for example during an analysis by MCMC.

3463 To analyze the augmented data set, we recognize that it is a zero-inflated version of  
 3464 the known- $N$  data set. That is, some of the augmented all-zero rows are sampling zeros  
 3465 (corresponding to actual individuals that were missed) and some are “structural” zeros,  
 3466 which do not correspond to individuals in the population. For a basic closed-population  
 3467 model, the resulting likelihood under data augmentation – that is, for the data set of size  
 3468  $M$  – is a simple zero-inflated binomial likelihood. The zero-inflated binomial model can be  
 3469 described “hierarchically”, by introducing a set of binary latent variables,  $z_1, z_2, \dots, z_M$ ,  
 3470 to indicate whether each individual  $i$  is ( $z_i = 1$ ) or is not ( $z_i = 0$ ) a member of the  
 3471 population of  $N$  individuals exposed to sampling. We assume that  $z_i \sim \text{Bernoulli}(\psi)$   
 3472 where  $\psi$  is the probability that an individual in the data set of size  $M$  is a member of the  
 3473 sampled population – in the sense that  $1 - \psi$  is the probability of realizing a “structural  
 3474 zero” in the augmented data set. The zero-inflated binomial model which arises under  
 3475 data augmentation can be formally expressed by the following set of assumptions (we

3476 include typical priors for a Bayesian analysis):

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

3477 for  $i = 1, \dots, M$ , where  $1(y=0)$  is a point mass at  $y=0$ . It is sometimes convenient to  
3478 express the conditional-on- $z$  observation model concisely in just one step:

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

3479 and we understand this to mean, if  $z_i = 0$ , then  $y_i$  is necessarily 0 because its success  
3480 probability is  $z_i p = 0$ .

3481 Note that, under data augmentation,  $N$  is no longer an explicit parameter of this  
3482 model. In its place, we estimate  $\psi$  and functions of the latent variables  $z$ . In particular,  
3483 under the assumptions of the zero-inflated model,  $z_i \stackrel{iid}{\sim} \text{Bern}(\psi)$ ; therefore,  $N$  is a function  
3484 of these latent variables:

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

3485 Further, we note that the latent  $z_i$  parameters *can be* removed from the model by inte-  
3486 gration, in which case the joint probability of the data is

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

3487 Interpreted as a likelihood, we can directly maximize this expression to obtain the MLEs of  
3488 the structural parameters  $\psi$  and  $p$  or those of other more complex models (e.g., see Royle,  
3489 2006). We could estimate these parameters and then use them to obtain an estimator of  
3490  $N$  using the so-called “Best unbiased predictor” (see Royle and Dorazio, 2012). Normally,  
3491 however, we will analyze the model in its “conditional-on- $z$ ” form using methods of MCMC  
3492 either in the **BUGS** engines or using our own MCMC algorithms (see Chapt. 14).

### 3493 4.2.2 Model $M_0$ in BUGS

3494 It is helpful to understand data augmentation by seeing what its effect is on implementing  
3495 model  $M_0$ . For this model, in which we can aggregate the encounter data to individual-  
3496 specific encounter frequencies, the augmented data are given by the vector of frequencies  
3497  $(y_1, \dots, y_n, 0, 0, \dots, 0)$  where the augmented values of  $y = 0$  represent the encounter fre-  
3498 quency for pseudo-individuals  $y_{n+1}, \dots, y_M$ . The zero-inflated model of the augmented  
3499 data combines the model of the latent variables,  $z_i \sim \text{Bern}(\psi)$  with the conditional-on- $z$   
3500 binomial model:

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

3501 so that, if  $z_i = 0$ , the success probability of the binomial distribution is identically 0  
3502 whereas, if  $z_i = 1$ , the success probability is  $p$ . This is useful in describing the model in

**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	.	0	1	0	3	1	0	1
4	1	0	1	.	1	0	1	4	1	0	1
5	0	1	1	.	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	M	0	0	0

3503 the **BUGS** language, as shown in Panel 4.1. Note the last line of the model specification  
 3504 provides the expression for computing  $N$  from the data augmentation variables  $z_i$ . Note  
 3505 that, to improve readability of code snippets (especially of large ones), we will sometimes  
 3506 deviate from our standard notation a bit. In this case we use `nind` for  $n$  (the number of  
 3507 encountered individuals), and  $M = nind + nz$  is the total size of the augmented data set.  
 3508 In other cases we might also use `nperiods` in place of  $K$  and `ntraps` in place of  $J$ . We  
 3509 find that word definitions make code easier to understand, especially without having to  
 3510 read surrounding text.

3511 Specification of a more general model in terms of the individual encounter observations  
 3512  $y_{ik}$  is not much more difficult than for the individual encounter frequencies. We define  
 3513 the observation model by a double loop and change the indexing of quantities accordingly,  
 3514 i.e.,

```
3515 for(i in 1:(nind+nz)) {
  3516   z[i]~dbern(psi)
  3517   for(k in 1:K){
  3518     mu[i,k]<-z[i]*p
  3519     y[i,k]~dbin(mu[i,k],1)
  3520   }
  3521 }
```

```

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  $y$ ,  $K$ ,  $n$  and  $\text{nz}$  are provided as data. The population size parameter  $N$  is computed as a function of the data augmentation variables  $z$ .

3522 In this manner, it is straightforward to incorporate covariates on  $p$  for both individuals  
 3523 and sampling occasions (see discussion of this below and also Chapt. 8) as well as to devise  
 3524 other extensions of the model, including models for open populations (see Chapt. 15).

### 3525 4.2.3 Formal development of data augmentation (DA)

3526 Use of parameter-expanded data augmentation (PX-DA), or DA for short, for solving  
 3527 inference problems with unknown  $N$  can be justified as originating from the choice of a  
 3528 uniform prior on  $N$ . The Uniform( $0, M$ ) prior for  $N$  is innocuous in the sense that the  
 3529 posterior associated with this prior is equal to the likelihood for sufficiently large  $M$ . One  
 3530 way of inducing the Uniform( $0, M$ ) prior on  $N$  is by assuming the following hierarchical  
 3531 prior:

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

3532 which includes a new model parameter  $\psi$  (note that we have seen  $\psi$  in the previous section  
 3533 as the proportion  $E[N]/M$ ). This parameter denotes the probability that an individual in  
 3534 the super-population of size  $M$  is a member of the population of  $N$  individuals exposed  
 3535 to sampling. The model assumptions, specifically the multinomial model (Eq. 4.1.3) and  
 3536 Eq. 4.2.2, may be combined to yield a reparameterization of the conventional model that  
 3537 is appropriate for the augmented data set of known size  $M$ :

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

3538 This expression arises by removing  $N$  from Eq. 4.1.3 by integrating over the binomial  
 3539 prior distribution for  $N$ . Thus, the models we analyze under data augmentation arise  
 3540 formally by removing the parameter  $N$  from the ordinary closed-population model—the  
 3541 model conditional on  $N$ —by integrating over a binomial prior distribution for  $N$ .

3542 Note that the  $M-n$  unobserved individuals in the augmented data set have probability  
 3543  $\psi\pi(0) + (1-\psi)$ , indicating that these unobserved individuals are a mixture of individuals  
 3544 that are sampling zeros ( $\psi\pi_0$ ), and belong to the population of size  $N$ , and others that  
 3545 are “structural zeros” (occurring in the augmented data set with probability  $1-\psi$ ). In  
 3546 Eq. 4.2.3  $N$  has been eliminated as a formal parameter of the model by marginalization  
 3547 (integration) and replaced with the new parameter  $\psi$ , the data augmentation parameter.  
 3548 However, the full likelihood containing both  $N$  and  $\psi$  can also be analyzed (see Royle  
 3549 et al., 2007).

#### 3550 4.2.4 Remarks on Data Augmentation

3551 Data augmentation may seem like a strange and mysterious black-box, and likely it is un-  
 3552 familiar to most people, even to many of those with substantial experience with capture-  
 3553 recapture models. However, it really is just a formal reparameterization of capture-  
 3554 recapture models in which  $N$  is marginalized out of the ordinary (conditional-on- $N$ ) model  
 3555 (by summation). As a result, we could refer to the resulting model as the “binomial-  
 3556 integrated likelihood” to reflect that an estimator could be obtained from the ordinary  
 3557 likelihood, integrated over a binomial prior. Other such “integrated likelihood” models are  
 3558 sensible. For example, we could place a Poisson prior on  $N$  with mean  $\Lambda$  and marginalize  
 3559  $N$  over the Poisson prior. This produces a likelihood in which  $\Lambda$  replaces  $N$ , instead of  $\psi$   
 3560 replacing  $N$ . We note that this type of marginalization over a Poisson prior is what the  
 3561 R package **secr** does in its analysis of spatial capture-recapture models (see Sec. 6.5.3).

3562 We emphasize the motivation for data augmentation being that it produces a data set  
 3563 of fixed size, so that the parameter dimension in any capture-recapture model is also fixed.  
 3564 As a result, MCMC is a relatively simple proposition using standard Gibbs Sampling.  
 3565 And, in particular, capture-recapture models become trivial to implement in **BUGS**.  
 3566 Consider the simplest context—analyzing model  $M_0$  using the occupancy-type model.  
 3567 In this case, DA converts model  $M_0$  to a basic occupancy model and the parameters  $p$   
 3568 and  $\psi$  have known full-conditional distributions (in fact, beta distributions) that can be  
 3569 sampled from directly. Furthermore, the data augmentation variables, i.e., the collection  
 3570 of  $z$ 's, can be sampled from Bernoulli full conditionals. MCMC is not much more difficult  
 3571 for complicated models—sometimes the hyperparameters need to be sampled using a  
 3572 Metropolis-Hastings step (e.g., Chapt. 14), but nothing more sophisticated than that is  
 3573 required.

3574 Potential sensitivity of parameter estimates (especially of  $N$ ) might be cause for some  
 3575 concern. The guiding principle is that it should be chosen large enough so that the  
 3576 posterior for  $N$  is not truncated, but it should not be too large due to the increased  
 3577 computational burden. It seems likely that the properties of the Markov chains should  
 3578 be affected by  $M$  and so some optimal choice of  $M$  might exist (Gopalaswamy, 2012).  
 3579 Formal analysis of this is needed.

3580 There are other approaches to analyzing models with unknown  $N$ , using reversible  
 3581 jump MCMC (RJMCMC) or other so-called “trans-dimensional” (TD) algorithms (King

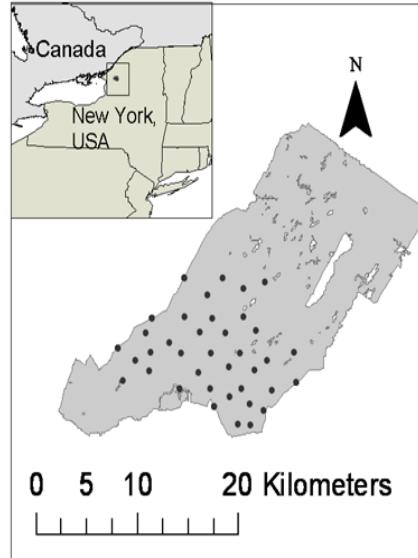
and Brooks, 2001; Durban and Elston, 2005; King et al., 2008; Schofield and Barker, 2008; Wright et al., 2009). What distinguishes DA from RJMCMC and related TD methods is that DA is used to create a distinctly new model that is unconditional on  $N$  and we (usually) analyze the unconditional model. 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., 2012b) which we consider in Chapt. 16. 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.

#### 3594 4.2.5 Example: Black Bear Study on Fort Drum

3595 To illustrate the analysis of model  $M_0$  using data augmentation, we use a data set collected  
 3596 at Fort Drum Military Installation in upstate New York by P.D. Curtis and M.T Wegan of  
 3597 Cornell University and their colleagues at the Fort Drum Military Installation. These data  
 3598 have been analyzed in various forms by Wegan (2008); Gardner et al. (2009) and Gardner  
 3599 et al. (2010b). The specific data used here are encounter histories on 47 individuals  
 3600 obtained from an array of 38 baited “hair snares” (Fig. 4.1) during June and July 2006.  
 3601 Barbed wire traps were baited and checked for hair samples each week for eight weeks,  
 3602 thus we distinguished  $K = 8$  weekly sample intervals. The data are provided in the R  
 3603 package **scrbook**, can be loaded by typing `load(beardata)`, and the analysis can be set  
 3604 up and run as follows (see `?beardata` for the commands to do the analysis). Here, the  
 3605 data were augmented with 128 all-zero encounter histories, resulting in a total sample size  
 3606 of  $M = 175$ .

```
3607 library(scrbook)
3608 data(beardata)
3609 trapmat<-beardata$trapmat
3610 nind<-dim(beardata$bearArray)[1]
3611 K<-dim(beardata$bearArray)[3]
3612 ntraps<-dim(beardata$bearArray)[2]
3613
3614 M=175
3615 nz<-M-nind
3616 Yaug <- array(0, dim=c(M,ntraps,K))
3617
3618 Yaug[1:nind,,]<-beardata$bearArray
3619 y<- apply(Yaug,c(1,3),sum) # summarize by ind x rep
3620 y[y>1]<- 1 # toss out multiple encounters b/c
3621 # traditional CR models ignore space
```

3622 The raw data object, `beardata$bearArray` is a 3-dimensional array  $nind \times ntraps \times K$   
 3623 of individual encounter events (i.e.,  $y_{ijk} = 1$  if individual  $i$  was encountered in trap  $j$  during  
 3624 occasion  $k$ , and 0 otherwise). For fitting model  $M_0$  (or  $M_h$ , see below), it is sufficient to



**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.

3625 reduce the data to individual encounter frequencies which we have re-labeled  $y$  above.  
 3626 The **BUGS** model file along with commands to fit the model are as follows:

```

3627 set.seed(2013)                      # to obtain the same results each time
3628 library(R2WinBUGS)
3629 data0<-list(y=y,M=M,K=K)
3630 params0<-list('psi','p','N')
3631 zst=c(rep(1,nind),rbinom(M-nind, 1, .5))
3632 inits =  function() { list(z=zst, psi=rnorm(1), p=rnorm(1)) }
3633
3634 cat("
3635 model {
3636
3637   psi~dunif(0, 1)
3638   p~dunif(0,1)
3639
3640   for (i in 1:M){
3641     z[i]~dbin(psi)
3642     for(k in 1:K){
3643       tmp[i,k]<-p*z[i]
3644       y[i,k]~dbin(tmp[i,k],1)
3645     }
  
```

```

3646     }
3647 N<-sum(z[1:M])
3648 }
3649 ",file="modelM0.txt")
3650
3651 fit0 = bugs(data0, inits, params0, model.file="modelM0.txt",n.chains=3,
3652   n.ITER=2000, n.burnin=1000, n.thin=1,debug=TRUE,working.directory=getwd())

```

3653 This produces the following posterior summary statistics:

```

3654 > print(fit0,digits=2)
3655 Inference for Bugs model at "modelM0.txt", fit using WinBUGS,
3656 3 chains, each with 2000 iterations (first 1000 discarded)
3657 n.sims = 3000 iterations saved
3658      mean    sd   2.5%   25%   50%   75%  97.5% Rhat n.eff
3659 psi      0.29  0.04   0.22   0.26   0.29   0.31   0.36    1 3000
3660 p       0.30  0.03   0.25   0.28   0.30   0.32   0.35    1 3000
3661 N       49.94 1.99  47.00  48.00  50.00  51.00  54.00    1 3000
3662 deviance 489.05 11.28 471.00 480.45 488.80 495.40 513.70    1 3000
3663
3664 [... some output deleted ...]

```

3665 **WinBUGS** did well in choosing an MCMC algorithm for this model – we have  $\hat{R} = 1$   
 3666 for each parameter, and an effective sample size of 3000, equal to the total number of  
 3667 posterior samples<sup>2</sup>. We see that the posterior mean of  $N$  under this model is 49.94 and  
 3668 a 95% posterior interval is (48, 54). We revisit these data later in the context of more  
 3669 complex models.

3670 In order to obtain an estimate of density,  $D$ , we need an area to associate with the  
 3671 estimate of  $N$ , and in Chapt. 1 we already went through a number of commonly used  
 3672 procedures to conjure up such an area, including buffering the trap array by the home range  
 3673 radius, often estimated by the mean maximum distance moved (MMDM) (Parmenter  
 3674 et al., 2003), 1/2 MMDM (Dice, 1938) or directly from telemetry data (Wallace et al.,  
 3675 2003) Typically, the trap array is defined by the convex hull around the trap locations, and  
 3676 this is what we applied a buffer to. We computed the buffer by using an estimate of the  
 3677 mean female home range radius (2.19 km) estimated from telemetry studies (Bales et al.,  
 3678 2005) instead of using an estimate based on our relatively more sparse recapture data.  
 3679 For the Fort Drum study, the convex hull has an area of 157.135 km<sup>2</sup>, and the buffered  
 3680 convex hull has an area of 277.011 km<sup>2</sup>. To create this we used functions contained in  
 3681 the **R** package **rgeos** and created a utility function **bcharea** which is in our **R** package  
 3682 **scrbook**. The commands are as follows:

```

3683 library(rgeos)
3684
3685 bcharea<-function(buff,traplocs){
3686 p1<-Polygon(rbind(traplocs,traplocs[1,]))
3687 p2<-Polygons(list(p1=p1),ID=1)

```

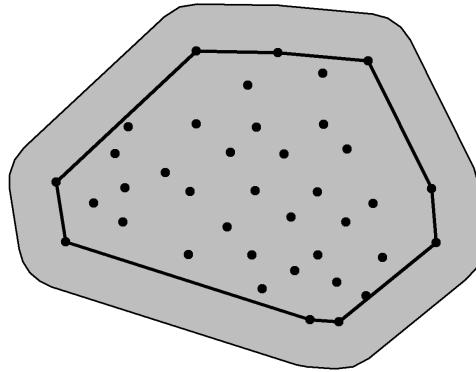
<sup>2</sup>This is even a little suspicious....

```

3688 p3<-SpatialPolygons(list(p2=p2))
3689 p1ch<-gConvexHull(p3)
3690 bp1<-(gBuffer(p1ch, width=buf))
3691 plot(bp1, col='gray')
3692 plot(p1ch, border='black', lwd=2, add=TRUE)
3693 gArea(bp1)
3694 }
3695
3696 bcharea(2.19,traplocs=trapmat)

```

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



**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).

```

3697
3698 To conjure up a density estimate under model  $M_0$ , we compute the appropriate pos-
3699 terior summary of the ratio of  $N$  and the prescribed area ( $277.011 \text{ km}^2$ ):

```

```

3700 > summary(fit0$sims.list$N/277.011)
3701   Min. 1st Qu. Median Mean 3rd Qu. Max.
3702 0.1697 0.1733 0.1805 0.1803 0.1841 0.2130
3703
3704 > quantile(fit0$sims.list$N/277.011,c(0.025,0.975))
3705   2.5% 97.5%
3706 0.1696684 0.1949381

```

3707 which yields a density estimate of about 0.18 ind/km<sup>2</sup>, and a 95% Bayesian confidence  
 3708 interval of (0.170, 0.195).

3709 Our estimate of density should be reliable if we have faith in our stated value of  
 3710 the “sample area”. Clearly though this is largely subjective, and not something we can  
 3711 formally evaluate (or estimate) from the data. More importantly, what exactly is the  
 3712 meaning of this area – in what quantitative sense is it the “effective sample area”? – and,  
 3713 in this context, how do we gauge bias and/or variance of “estimators” of it? These are  
 3714 questions that, to the best of our knowledge, have not been addressed in any generality,  
 3715 if at all<sup>3</sup>.

### 4.3 TEMPORALLY VARYING AND BEHAVIORAL EFFECTS

3716 The purpose of this chapter is mainly to emphasize the central importance of the bino-  
 3717 mial model in capture-recapture and so we have considered models for individual encounter  
 3718 frequencies—the number of times individuals are captured out of  $K$  samples. Sometimes  
 3719 we can’t aggregate the encounter data for each individual, such as when encounter proba-  
 3720 bility varies over time among samples. Time-varying responses that are relevant in many  
 3721 capture-recapture studies are “effort” such as amount of search time, number of observers,  
 3722 or trap nights, or when encounter probability varies over time or as a function of date or  
 3723 season due to species behavior (Kéry et al., 2010). A common situation in many animal  
 3724 studies is that in which there exists a “behavioral response” to trapping (even if the animal  
 3725 is not physically trapped).

3726 Behavioral response is an important concept in animal studies because individuals  
 3727 might learn to come to baited traps or avoid traps due to trauma related to being encoun-  
 3728 tered. There are a number of ways to parameterize a behavioral response to encounter.  
 3729 The distinction between persistent and ephemeral was made by Yang and Chao (2005)  
 3730 who considered a general behavioral response model of the form:

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

3731 where  $x_{ik}$  is a covariate indicator variable of previous capture (i.e.,  $x_{ik} = 1$  if captured  
 3732 in any previous period). Therefore, encounter probability changes depending on whether  
 3733 an individual was captured in the immediate previous period (a Markovian or ephemeral  
 3734 behavioral response; (Yang and Chao, 2005)), described by the term  $\alpha_1 y_{i,k-1}$  or in *any*  
 3735 previous period (persistent behavioral response), described by the term  $\alpha_2 x_{ik}$ . Because  
 3736 spatial capture-recapture models allow us to include trap-specific covariates, we can de-  
 3737 scribe a 3rd type of behavioral response—a local behavioral response that is trap-specific  
 3738 (Royle et al., 2011b). In this local behavioral response, the encounter probability is mod-  
 3739 ified for an individual trap depending on previous capture in that trap. Models with  
 3740 temporal effects are easy to describe and analyze in the **BUGS** language and we provide  
 3741 a number of examples in Chapt. 8 and elsewhere.

<sup>3</sup>We note that Karanth and Nichols (1998) and possibly others have computed the variance of an estimated buffer strip, but do not provide a quantitative definition of effective sample area.

#### 4.4 MODELS WITH INDIVIDUAL HETEROGENEITY

3742 Models in which encounter probability varies by individual, say  $p_i$ , have a long history in  
3743 capture-recapture and, indeed, this so-called “model  $M_h$ ” is one of the elemental capture-  
3744 recapture models in (Otis et al., 1978). Conceptually, we imagine that the individual-  
3745 specific encounter probability parameters,  $p_i$ , are random variables distributed according  
3746 to some probability distribution,  $[\theta]$ . We denote this basic model assumption as  $p_i \sim [\theta]$ .  
3747 This type of model is similar in concept to extending a GLM to a GLMM but in the  
3748 capture-recapture context  $N$  is unknown. The basic class of models is often referred to as  
3749 “model  $M_h$ ” ( $h$ ’ for heterogeneity), but really this is a broad class of models, each being  
3750 distinguished by the specific distribution assumed for  $p_i$ . There are many different varieties  
3751 of model  $M_h$  including parametric and various non-parametric approaches (Burnham and  
3752 Overton, 1978; Norris III and Pollock, 1996; Pledger, 2000). One important practical  
3753 matter is that estimates of  $N$  can be extremely sensitive to the choice of heterogeneity  
3754 model (Fienberg et al., 1999; Dorazio and Royle, 2003; Link, 2003). Indeed, Link (2003)  
3755 showed that in some cases it’s possible to find models that yield precisely the same expected  
3756 data, yet produce wildly different estimates of  $N$ . In that sense,  $N$  for most practical  
3757 purposes is not identifiable across classes of different heterogeneity models, and this should  
3758 be understood before fitting any such model. One solution to this problem is to seek to  
3759 model explicit factors that contribute to heterogeneity, e.g., using individual covariate  
3760 models (See 4.5 below). Indeed, spatial capture-recapture models seek to do just that, by  
3761 modeling heterogeneity due to the spatial organization of individuals in relation to traps  
3762 or other encounter mechanism. For additional background and applications of model  $M_h$   
3763 see Royle and Dorazio (2008, Chapt. 6) and Kéry and Schaub (2012, Chapt. 6).

3764 Model  $M_h$  has important historical relevance to spatial capture-recapture situations  
3765 (Karanth, 1995) because investigators recognized that the juxtaposition of individuals  
3766 with the array of trap locations should yield heterogeneity in encounter probability, and  
3767 thus it became common to use some version of model  $M_h$  in spatial trapping arrays to  
3768 estimate  $N$ . While this doesn’t resolve the problem of not knowing the area relevant to  
3769  $N$ , it does yield an estimator that accommodates the heterogeneity in  $p$  induced by the  
3770 spatial aspect of capture-recapture studies.

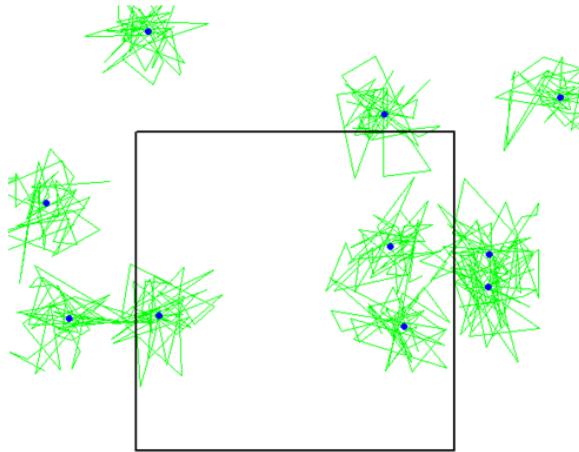
3771 To see how this juxtaposition induces heterogeneity, we have to understand the relevance  
3772 of movement in capture-recapture models. Imagine a quadrat that can be uniformly  
3773 searched by a crew of biologists for some species of reptile (see Royle and Young (2008)).  
3774 Figure 4.3 shows a sample quadrat searched repeatedly over a period of time. Further,  
3775 suppose that the species exhibits some sense of spatial fidelity in the form of a home range  
3776 or territory, and individuals move about their home range (home range centroids are given  
3777 by the blue dots) in some kind of random fashion. Heuristically, we imagine that each  
3778 individual in the vicinity of the study area is liable to experience variable exposure to  
3779 encounter due to the overlap of its home range with the sampled area - essentially the  
3780 long-run proportion of times the individual is within the sample plot boundaries, say  $\phi$ .  
3781 We might model the exposure of an individual to capture by supposing that  $z_i = 1$  if  
3782 individual  $i$  is available to be captured (i.e., within the survey plot) during any sample,  
3783 and 0 otherwise. Then,  $\Pr(z_i = 1) = \phi$ . In the context of spatial studies, it is natural  
3784 that  $\phi$  should depend on *where* an individual lives, i.e., it should be individual-specific  
3785  $\phi_i$  (Chandler et al., 2011). This system describes, precisely, that of “random temporary  
3786 emigration” (Kendall et al., 1997) where  $\phi_i$  is the individual-specific probability of being

3787 “available” for capture.

3788 Conceptually, SCR models aim to deal with this problem of variable exposure to sam-  
3789 pling due to movement in the proximity of the trapping array explicitly and formally with  
3790 auxiliary spatial information. If individuals are detected with probability  $p_0$ , *conditional*  
3791 on  $z_i = 1$ , then the marginal probability of detecting individual  $i$  is

$$p_i = p_0 \phi_i$$

3792 so we see clearly that individual heterogeneity in encounter probability is induced as a re-  
3793 sult of the juxtaposition of individuals (i.e., their home ranges) with the sample apparatus  
3794 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.

3795 We will work with a specific type of model  $M_h$  here which is a natural extension of  
3796 the basic binomial observation model of model  $M_0$  so that

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

3797 where  $\mu$  is a fixed parameter (the mean) to be estimated, and  $\eta_i$  is an individual random  
3798 effect assumed to be normally distributed:

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

3799 We could as well combine these two steps and write  $\text{logit}(p_i) \sim \text{Normal}(\mu, \sigma_p^2)$ . This  
 3800 “logit-normal mixture” was analyzed by Coull and Agresti (1999) and elsewhere. It is  
 3801 a natural extension of the basic model with constant  $p$ , as a mixed GLMM, and similar  
 3802 models occur throughout statistics. It is also natural to consider a beta prior distribution  
 3803 for  $p_i$  (Dorazio and Royle, 2003) and so-called “finite-mixture” models are also popular  
 3804 (Norris III and Pollock, 1996; Pledger, 2000). In the latter, individuals are assumed to  
 3805 belong to a finite number of latent classes, each of which has its own capture probability.

#### 3806 4.4.1 Analysis of Model $M_h$

3807 If  $N$  is known, it is worth taking note of the essential simplicity of model  $M_h$  as a binomial  
 3808 GLMM. This is a type of model that is widely applied throughout statistics using standard  
 3809 methods of inference based either on integrated likelihood (Laird and Ware, 1982; Berger  
 3810 et al., 1999), which we discuss in Chapt. 6, or standard Bayesian methods. However,  
 3811 because  $N$  is not known, inference is somewhat more challenging. We address that here  
 3812 using Bayesian analysis based on data augmentation. Although we use data augmentation  
 3813 in the context of Bayesian methods here, we note that heterogeneity models formulated  
 3814 under DA are easily analyzed by conventional likelihood methods as zero-inflated binomial  
 3815 mixtures (Royle, 2006) and more traditional analysis of model  $M_h$  based on integrated  
 3816 likelihood, without using data augmentation, has been considered by Coull and Agresti  
 3817 (1999), Dorazio and Royle (2003), and others.

3818 As with model  $M_0$ , we have the Bernoulli model for the zero-inflation variables:  $z_i \sim$   
 3819  $\text{Bern}(\psi)$  and the model of the observations expressed conditional on these latent variables  
 3820  $z_i$ . For  $z_i = 1$ , we have a binomial model with individual-specific  $p_i$ :

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

3821 and otherwise  $y_i | z_i = 0 \sim 1(y = 0)$ . Further, we prescribe a distribution for  $p_i$ . Here we  
 3822 assume

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

3823 For prior distributions we assume  $p_0 = \text{logit}^{-1}(\mu) \sim \text{Uniform}(0, 1)$  and, for the variance  
 3824 component  $\sigma \sim \text{Uniform}(0, B)$  for some large  $B$ . Another common default prior is to  
 3825 assume  $\tau = 1/\sigma^2 \sim \text{Gamma}(1, 1)$ , although we usually choose the  $\sigma \sim \text{Uniform}(0, B)$ .

#### 3826 4.4.2 Analysis of the Fort Drum data with model $M_h$

3827 Here we provide an analysis of the Fort Drum bear survey data using the logit-normal  
 3828 heterogeneity model, and we used data augmentation to produce a data set of  $M = 700$   
 3829 individuals. We have so far mostly used **WinBUGS** but for most of our operational  
 3830 analyses now we are transitioning to the use of **JAGS** run from within **R** using the useful  
 3831 packages **R2jags** or **rjags**. The function **jags** from the **R2jags** package runs essentially  
 3832 like the **bugs** function which we demonstrate as follows for setting up and running model  
 3833  $M_h$  for the Fort Drum bear data:

```
3834 [... get data as before ...]
3835
3836 set.seed(2013)
```

```

3837
3838 cat("
3839 model{
3840 p0 ~ dunif(0,1)      # prior distributions
3841 mup<- log(p0/(1-p0))
3842 sigmap ~ dunif(0,10)
3843 taup<- 1/(sigmap*sigmap)
3844 psi~dunif(0,1)

3845 for(i in 1:(nind+nz)){
3846   z[i]~dbern(psi)    # zero inflation variables
3847   lp[i] ~ dnorm(mup,taup) # individual effect
3848   logit(p[i])<-lp[i]
3849   mu[i]<-z[i]*p[i]
3850   y[i]~dbin(mu[i],K)  # observation model
3851 }
3852

3853 N<-sum(z[1:(nind+nz)])
3854 }
3855 ",file="modelMh.txt")
3856

3857 data1<-list(y=y, nz=nz, nind=nind,K=K)
3858 params1= c('p0','sigmap','psi','N')
3859 inits = function() {list(z=as.numeric(y>=1), psi=.6, p0=runif(1),
3860   sigmap=runif(1,.7,1.2),lp=rnorm(M,-2)) }
3861 library(R2jags)
3862 wbout = jags(data1, inits, params1, model.file = "modelMh.txt", n.chains = 3,
3863   n.iter = 1010000, n.burnin =10000, working.directory = getwd())

```

3865 We provide an **R** function `modelMhBUGS` in the package `scrbook` which will fit the  
 3866 model using either **JAGS** or **WinBUGS** as specified by the user. In addition, for fun,  
 3867 we construct our own MCMC algorithm using a Metropolis-within-Gibbs algorithm for  
 3868 model  $M_h$  in Chapt. 14, where we also develop MCMC algorithms for spatial capture-  
 3869 recapture models. Using the **WinBUGS** option in `modelMhBUGS`, we ran 3 chains of 1  
 3870 million iterations (mixing is poor for this model and this data set), which produced the  
 3871 posterior distribution for  $N$  shown in Fig. 4.4. Posterior summaries of parameters are  
 3872 given as follows<sup>4</sup>:

```

3873 > print(wbout,digits=3)
3874 Inference for Bugs model at "modelMh.txt", fit using WinBUGS,
3875 3 chains, each with 1010000 iterations (first 10000 discarded)
3876 n.sims = 2970000 iterations saved
3877          mean     sd   2.5%   25%   50%   75%  97.5% Rhat n.eff
3878 p0      0.072  0.056  0.002  0.027  0.060  0.106  0.203 1.008   540
3879 sigmap  2.096  0.557  1.215  1.694  2.025  2.424  3.373 1.003   820
3880 psi     0.176  0.101  0.084  0.117  0.147  0.198  0.458 1.006   650

```

<sup>4</sup>The reported `n.sims` printed appears to be an error in the `R2WinBUGS` package

```

3881 N      122.695 69.897 62.000 82.000 102.000 137.000 319.000 1.006 630
3882 deviance 187.089 17.998 155.400 174.400 185.900 198.500 225.700 1.002 3600
3883
3884 [... output deleted ...]

```

3885 We used  $M = 700$  for this analysis and we note that while the posterior mass of  $N$  is  
 3886 concentrated away from this upper bound (Fig. 4.4), the posterior has an extremely long  
 3887 right tail, with some MCMC draws at the upper boundary  $N = 700$ , suggesting that an  
 3888 even higher value of  $M$  may be called for. To characterize the posterior distribution of  
 3889 density we produce the relevant summaries of the posterior distribution of  $D = N/277.11$   
 3890 (recall the buffered area of the convex hull is  $277.11 \text{ km}^2$ ):

```

3891 > summary(wbout$sims.list$N/277.11)
3892   Min. 1st Qu. Median Mean 3rd Qu. Max.
3893 0.1696 0.2959 0.3681 0.4428 0.4944 2.5260
3894
3895 > quantile(wbout$sims.list$N/277.11,c(0.025,0.50,0.975))
3896   2.5% 50% 97.5%
3897 0.2237379 0.3680849 1.1511674

```

3898 so the point estimate, characterized by the posterior median, is around 0.37 bears per  
 3899 square km and a 95% Bayesian credible interval is (0.224, 1.151).

#### 3900 4.4.3 Comparison with MLE

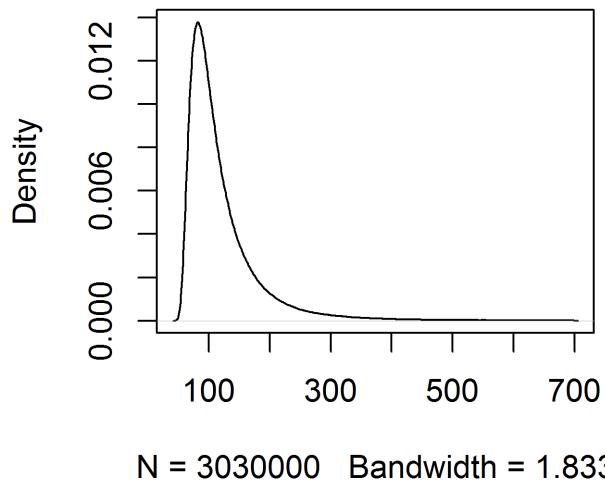
3901 The posterior of  $N$  is highly skewed; therefore, we see that the posterior mean ( $N = 122.7$ )  
 3902 is considerably higher than the posterior median ( $N = 102$ ). Further, it may be surprising  
 3903 that these posterior summaries do not compare well with the MLE. We used the **R** code  
 3904 contained in Panel 6.1 from Royle and Dorazio (2008) to obtain the MLE of  $\log(n_0)$ ,  
 3905 the logarithm of the number of uncaptured individuals, is  $\widehat{\log(n_0)} = 3.86$  and therefore  
 3906  $\hat{N} = \exp(3.86) + 47 = 94.47$ , much higher than the mode shown in Fig. 4.4. To see this,  
 3907 we compute the posterior mode by finding the posterior value of  $N$  with the highest mass.  
 3908 Because  $N$  is discrete, we can use the **table()** function in **R** and find the most frequent  
 3909 value<sup>5</sup>. If we want to smooth out some of the Monte Carlo error a bit, we can use a  
 3910 smoother of some sort applied to the tabled posterior frequencies of  $N$ . Here we use a  
 3911 smoothing spline (**R** function **smooth.spline**) with the degree of smoothing chosen by  
 3912 cross-validation:

```

3913 > N<-table(jout$BUGSoutput$sims.list$N)
3914 > xg<-as.numeric(names(N))
3915
3916 > sp<- smooth.spline(xg,N,cv=TRUE)
3917
3918 > sp
3919

```

<sup>5</sup>For a continuous random variable we can use the function **density()** to smooth the posterior samples and obtain the mode.



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

```

3920 Call:
3921 smooth.spline(x = xg, y = N, cv = TRUE)
3922
3923 Smoothing Parameter spar= 0.09339815 lambda= 8.201724e-09 (17 iterations)
3924 Equivalent Degrees of Freedom (Df): 121.1825
3925 Penalized Criterion: 2544481
3926 PRESS: 5903.4

```

3927 We obtain the mode of the smoothed frequencies as follows:

```

3928 sp$x[sp$y==max(sp$y)]
3929 [1] 82

```

3930 We don't dwell too much on the difference, but we do note here that the posterior  
 3931 distribution for the parameters of this model, for the Fort Drum data set, are very sensitive  
 3932 to the prior distributions. While MLEs are invariant to transformation of the parameters,  
 3933 the posterior distribution definitely is *not* invariant. In the present case, the use of a  
 3934 Uniform(0, 1) prior for  $p_0 = \text{logit}^{-1}(\mu)$  is somewhat informative—in particular, it is not at  
 3935 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. That specific prior suggests the possibility that the posterior distribution may be improper for that prior specification. 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

A standard situation in capture-recapture models is when a covariate which is thought to influence encounter probability is measured for each individual. As with other closed population models, we begin with the basic binomial observation model:

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

and we assume also a model for encounter probability according to:

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

where  $x_i$  is the covariate value for individuals  $i$  and the parameters  $\alpha$  are the regression coefficients. Classical examples of covariates influencing detection probability are type of animal (juvenile/adult or male/female), a continuous covariate such as body mass (Royle and Dorazio, 2008, ch. 6), or a discrete covariate such as group or cluster size. For example, in models of aerial survey data, it is natural to model the detection probability of a group as a function of the observation-level individual covariate, “group size” (Royle, 2008, 2009a; Langtimm et al., 2011).

Such “individual covariate models” are 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 individual covariate model, where the covariate describes *where* the individual is located in relation to the trapping array. Specifically, they are such models, but where the individual covariate is a partially observed latent variable for captured individuals. Unlike model  $M_h$ , in individual covariate models (and SCR models) 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 individual covariate models is achieved using methods based on ideas of unequal probability sampling (i.e., Horvitz-Thompson estimation<sup>6</sup>; Huggins (1989), Alho (1990) and Borchers et al. (2002)). An estimator of  $N$  is

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

<sup>6</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)

3967 where  $\tilde{p}_i$  is the probability that individual  $i$  appeared in the sample. That is,  $\tilde{p}_i = \Pr(y_i > 0)$  where, in closed population capture-recapture models,

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

3969 where  $p_i$  is a function of parameters  $\alpha_0$  and  $\alpha_1$  according to Eq. 4.5.1. In practice, pa-  
3970 rameters are estimated from the conditional-likelihood of the observed encounter histories  
3971 which is, for observation  $y_i$ ,

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

3972 This derives from a straightforward application of the law of total probability. Conceptu-  
3973 ally, 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  
3975  $\Pr(y|y > 0) = \Pr(y)/\Pr(y > 0)$  which, in the specific case where  $\Pr(y)$  is the binomial  
3976 pmf produces Eq. 4.5.2.

3977 Here we take a formal model-based approach to Bayesian analysis of such models  
3978 based on the joint likelihood using data augmentation (Royle, 2009a). Classical likelihood  
3979 analysis of the so-called “full likelihood” is covered by Borchers et al. (2002). For Bayesian  
3980 analysis of individual covariate models, because the individual covariate is unobserved for  
3981 the  $n_0 = N - n$  uncaptured individuals, we require a model to describe variation among  
3982 individuals, essentially allowing the sample to be extrapolated to the population. For  
3983 example, if we have a continuous trait measured on each individual, then we might assume  
3984 that  $x$  has a normal distribution:

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

3985 Data augmentation can be applied directly to this class of models. In particular, refor-  
3986 mulation of the model under DA yields a basic zero-inflated binomial model of the form:

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

3987 Fully spatial capture-recapture models use this formulation with a latent covariate that  
3988 is directly related to the individual detection probability (see next section). As with  
3989 the previous models, implementation is trivial in the **BUGS** language. The **BUGS**  
3990 specification is very similar to that for model  $M_h$ , but we require the distribution of the  
3991 covariate to be specified, along with priors for the parameters of that distribution.

### 3992 4.5.1 Example: Location of capture as a covariate.

3993 Here we consider a special type of individual covariate model that is especially relevant to  
3994 spatial capture-recapture. Intuitively, some measure of distance from home range center  
3995 to traps for an individual should be a reasonable covariate to explain heterogeneity in  
3996 encounter probability, i.e., individuals with more exposure to traps should have higher

3997 encounter probabilities and vice versa. So we can imagine *estimating* such a quantity,  
 3998 say average distance from home range center to “the trap array”, and then using it as  
 3999 an individual covariate in capture-recapture models. A version of this idea was put forth  
 4000 by Boulanger and McLellan (2001) (see also Ivan (2012)), but using the Huggins-Alho  
 4001 estimator and with covariate “distance from home range center to edge” of the trapping  
 4002 array, where the home range center is estimated by the average capture location. This  
 4003 is intuitively appealing because we can imagine, in some kind of an ideal situation where  
 4004 we have a dense grid of traps over some geographic region, that the average location of  
 4005 capture would be a decent estimate (heuristically) of an individual’s home range center.  
 4006 We provide an example of this type of heuristically motivated approach using a fully  
 4007 model-based individual covariate model described above analyzed by data augmentation.  
 4008 We take a slightly different approach than that adopted by Boulanger and McLellan  
 4009 (2001). By analyzing the full likelihood and placing a prior distribution on the individual  
 4010 covariate, we will resolve the problem of having an ill-defined sampled area. After you read  
 4011 later chapters of this book, it will be apparent that SCR models represent a formalization  
 4012 of this heuristic procedure.

4013 For our purposes here, we define the individual covariate  $x_i = \|\mathbf{s}_i - \mathbf{x}_0\|$  where  $\mathbf{s}_i$  is  
 4014 the average encounter location of individual  $i$  and  $\mathbf{x}_0$  is the centroid of the trap array. In  
 4015 practice, people have used distance from edge of the trap array but that is less easy to  
 4016 quantify, as “edge” itself is not precisely defined. Conceptually, individuals in the middle  
 4017 of the array should have a higher probability of encounter and, as  $x_i$  increases,  $p_i$  should  
 4018 therefore decrease. We note that we have defined  $\mathbf{s}_i$  in terms of a sample quantity—the  
 4019 observed mean—which is ad hoc but consistent with the use of individual covariate models  
 4020 in the literature. For an expansive, dense trapping grid we might expect the sample mean  
 4021 encounter location to be a good estimate of home range center but, clearly this is biased for  
 4022 individuals that live around the edge (or off) the trapping array. A key point is that  $\mathbf{s}_i$  is  
 4023 missing for each individual that is not encountered and so  $x_i$  is also missing. Therefore, it  
 4024 is a latent variable, or random effect, and we need to specify a probability distribution for  
 4025 it. As a measurement of distance we know it must be positive-valued, and it seems sensible  
 4026 that an individual located extremely far from the array of traps would not be captured.  
 4027 Therefore, let’s assume that  $x_i$  is uniformly distributed from 0 to some large number, say  
 4028  $B$ , beyond which it would be difficult to imagine an individual being captured by the trap  
 4029 array. For example,  $B$  should be at a home range diameter past the furthest trap from  
 4030 the center. As such, we use this distribution for the individual covariate “distance from  
 4031 center of the trap array”

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

4032 where  $B$  is a specified constant, which we may choose to be arbitrarily large.

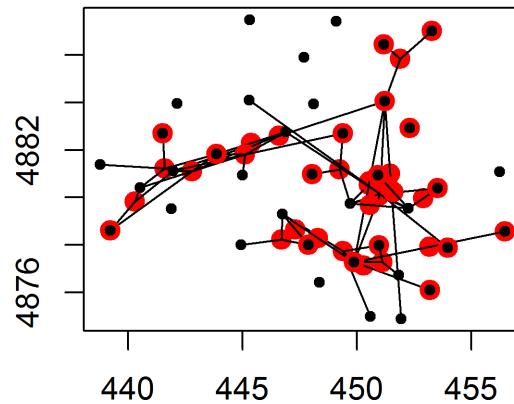
#### 4033 4.5.2 Fort Drum Bear Study

4034 We have to do a little bit of data processing to fit this individual covariate model to the  
 4035 Fort Drum data. We need to compute the individual covariate  $\mathbf{x}_i$  (distance from the  
 4036 centroid of the trapping array) using the **R** function `spiderplot` provided in `scrbook`.  
 4037 This function also produces the keen plot shown in Fig. 4.5 which we call a “spider plot”.  
 4038 The **R** commands for obtaining the individual covariate “distance from trap centroid” and  
 4039 making the spider plot are as follows:

```

4040 library(scrbook)
4041 data(beardata)
4042 toad<- spiderplot(beardata$bearArray,beardata$trapmat)
4043 xcent<-toad$xcent

```



**Figure 4.5.** Spider plot of the Fort Drum study data. The black dots represent the 47 trap locations with larger red dots 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. **XXX note to Andy:** change one of the dots to an X or something XXXXX

4044 For the analysis of these data using the individual covariate “distance from centroid”  
 4045 we used  $x_i \sim \text{Uniform}(0, B)$  with  $B = 11.5 \text{ km}^2$ , which is about the distance from the  
 4046 array center to the furthest trap. Once we choose a value for  $B$ , the direct implication is  
 4047 that the population size parameter,  $N$ , applies to the area within 11.5 units of the trap  
 4048 centroid. We will see shortly that  $N$  does, in fact, scale with our choice of  $B$  to reflect  
 4049 the changing area over which the  $N$  individuals of the model reside. The **BUGS** model  
 4050 specification and **R** commands to package the data and fit the model are as follows:

```

4051 cat("
4052 model{

```

```

4053 p0 ~ dunif(0,1)      # prior distributions
4054 alpha0<- log(p0/(1-p0))
4055 psi~dunif(0,1)
4056 beta~dnorm(0,.01)
4057
4058 for(i in 1:(nind+nz)){
4059   xcent[i]~dunif(0,B)
4060   z[i]~dbern(psi)      # DA variables
4061   lp[i] <- alpha0 + beta*xcent[i] # individual effect
4062   logit(p[i])<-lp[i]
4063   mu[i]<-z[i]*p[i]
4064   y[i]~dbin(mu[i],K)  # observation model
4065 }
4066 N<-sum(z[1:(nind+nz)])
4067 }
4068 ",file="modelMcov.txt")
4069
4070 data2<-list(y=y,nz=nz,nind=nind,K=K,xcent=xcent,B=11.5)
4071 params2<-list('p0','psi','N','beta')
4072 inits = function() {list(z=zst, psi=psi, p0=rnorm(1),beta=rnorm(1) ) }
4073 fit2 = bugs(data2, inits, params2, model.file="modelMcov.txt",
4074 n.chains=3, n.iter=11000, n.burnin=1000, n.thin=1)

```

4075 This produces the following posterior summary statistics:

```

4076 Inference for Bugs model at "modelMcov.txt", fit using WinBUGS,
4077 3 chains, each with 11000 iterations (first 1000 discarded)
4078 n.sims = 30000 iterations saved
4079          mean    sd  2.5%   25%   50%   75% 97.5% Rhat n.eff
4080 p0       0.54  0.07  0.40  0.50  0.54  0.59  0.67    1  1100
4081 psi      0.34  0.05  0.25  0.31  0.34  0.37  0.44    1  3500
4082 N        58.92 5.49 50.00 55.00 58.00 62.00 71.00    1  1900
4083 beta     -0.25 0.06 -0.36 -0.29 -0.25 -0.21 -0.12    1   780
4084 deviance 459.51 13.21 435.80 450.20 458.80 467.90 487.40    1  2600

```

4085 We note that the estimated  $N$  is much lower than obtained by model  $M_h$  but there  
 4086 is a good explanation for this, discussed subsequently. That issue notwithstanding, it is  
 4087 worth pondering how this model could be an improvement (conceptually or technically)  
 4088 over some other model/estimator including  $M_0$  and  $M_h$  considered previously. Well, for  
 4089 one, we have accounted formally for heterogeneity due to spatial location of individuals  
 4090 relative to exposure to the trap array, characterized by the centroid of the array. Moreover,  
 4091 we have done so using a model that is based on an explicit mechanism, as opposed to a  
 4092 phenomenological one such as Model  $M_h$ . Moreover, importantly, using our new model,  
 4093 *the estimated  $N$  applies to an explicit area which is defined by our prescribed value of  $B$ .*  
 4094 That is, this area is a fixed component of the model and the parameter  $N$  therefore has  
 4095 explicit spatial context, as the number of individuals with home range centers less than  
 4096  $B$  from the centroid of the trap array. As such, the implied “effective area” of the trap

4097 array for a given  $B$  is a precisely defined quantity—it is that of a circle with radius  
 4098  $B$ .

4099 **4.5.3 Extension of the Model**

4100 The model developed in the previous section is not a very good model for one important  
 4101 reason: Imposing a uniform prior distribution on  $x$  implies that density is *not constant*  
 4102 over space. In particular, this model implies that density *decreases* as we move away from  
 4103 the centroid of the trap array. That is,  $x_i \sim \text{Uniform}(0, B)$  implies constant  $N$  in each  
 4104 distance band from the centroid but obviously the *area* of each distance band is increasing.  
 4105 This is one reason we have a lower estimate of density than that obtained previously from  
 4106 model  $M_h$  (Sec. 4.4.2) and also why, if we were to increase  $B$ , we would see density  
 4107 continue to decrease.

4108 Fortunately, we are not restricted to use of this specific distribution for the individual  
 4109 covariate. Clearly, it is a bad choice and, therefore, we should think about whether we  
 4110 can choose a better distribution for  $B$ —one that doesn’t imply a decreasing density as  
 4111 distance from the centroid increases. Conceptually, what we want to do is impose a prior  
 4112 on distance from the centroid,  $x$ , such that density is proportional to the amount of area in  
 4113 each successive distance band as you move farther away from the centroid. In fact, theory  
 4114 exists which tells us what the correct distribution of  $x$  is:  $2x/B^2$ . This can be derived by  
 4115 noting that  $F(x) = \Pr(X < x) = (\pi x^2)/(\pi * B^2)$ . Then,  $f(x) = dF/dx = 2 * x/(B^2)$ .  
 4116 This is a sort of triangular distribution in density induced because the incremental area in  
 4117 each additional distance band increases linearly with radius (i.e., distance from centroid).  
 4118 This can be verified empirically as follows:

```
4119 u<-runif(10000,-1,1)
4120 v<-runif(10000,-1,1)
4121 d<- sqrt(u*u+v*v)
4122 hist(d[d<1])
4123 hist(d[d<1],100)
4124 hist(d[d<1],100,probability=TRUE)
4125 abline(0,2)
```

4126 It would be useful if we could describe this distribution directly in **BUGS** but there  
 4127 is not a built-in way to do so. However, we can implement a discrete version of the pdf<sup>7</sup>.  
 4128 To do this, we break  $B$  into  $L$  distance classes of width  $\delta$ , with probabilities proportional  
 4129 to  $2 * x$ . In particular, if we denote the cut-points by  $g_1 = 0, g_2, \dots, g_{L+1} = B$  and the  
 4130 interval midpoints are  $m_i = g_{i+1} - \delta$ . Then the interval probabilities are, approximately,  
 4131  $p_i = 2m_i\delta/(B^2)$ , which we can compute once and then pass them to **BUGS** as data. The  
 4132 **R** commands for doing all of this (noting that we have already loaded and processed the  
 4133 Fort Drum bear data) are given in the following **BUGS** code:

```
4134 delta<- .2
4135 xbin<-xcent%/%delta + 1
4136 Dgrid<- seq(delta,Dmax,delta)
```

<sup>7</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.

```

4137 xprobs<- delta*(2*Dgrid/(Dmax*Dmax))
4138 xprobs<-xprobs/sum(xprobs)
4139
4140 cat("
4141 model{
4142 p0 ~ dunif(0,1)      # prior distributions
4143 alpha0<- log(p0/(1-p0))
4144 psi~dunif(0,1)
4145 beta~dnorm(0,.01)
4146
4147 for(i in 1:(nind+nz)){
4148   xbin[i]~dcat(xprobs[])
4149   z[i]~dbeern(psi)           # zero inflation variables
4150   lp[i] <- alpha0 + beta*xbin[i]*delta # individual effect
4151   logit(p[i])<-lp[i]
4152   mu[i]<-z[i]*p[i]
4153   y[i]~dbin(mu[i],K)  # observation model
4154 }
4155
4156 N<-sum(z[1:(nind+nz)])
4157 }
4158 ",file="modelMcov.txt")

```

4159     In the model description the variable  $x$  (observed distance from centroid of the trap  
4160 array) has been rounded or binned so that the discrete version of the pdf of  $x$  can be used  
4161 as described previously. The new variable labeled **xbin** is then the *integer category* in  
4162 units of  $\delta$  from 0. Thus, to convert back to distance in the expression for  $lp[i]$ , **xbin[i]**  
4163 has to be multiplied by  $\delta$ . To fit the model, keeping in mind that the data objects required  
4164 below have been defined in previous analyses of this chapter, we do this:

```

4165 data2<-list(y=y,nz=nz,nind=nind,K=K,xbin=xbin,xprobs=xprobs,delta=delta)
4166 params2<-list('p0','psi','N','beta')
4167 inits = function() {list(z=z, psi=psi, p0=rnorm(1),beta=rnorm(1) ) }
4168 fit = bugs(data2, inits, params2, model.file="modelMcov.txt",
4169   working.directory=getwd(), debug=FALSE, n.chains=3, n.iter=11000,
4170   n.burnin=1000, n.thin=2)

```

4171     By specification of  $B$ , this model induces a clear definition of area in which the popu-  
4172 lation of  $N$  individuals reside. The parameter  $N$  of the model is the population size that  
4173 applies to the particular value of  $B$  and, as such, we will see that  $N$  scales with our choice  
4174 of  $B$ . This might be disconcerting to some—we can get whatever value of  $N$  we want  
4175 by changing  $B$ ! Fortunately, we find empirically, that while  $N$  is highly sensitive to the  
4176 prescribed value of  $B$ , density appears invariant to  $B$  as long as  $B$  is sufficiently large.  
4177 We fit the model for a random of values of  $B$  from  $B = 12$  (restricting values of  $x$  to be  
4178 in close proximity to the trap array) on up to 20. The results are given in Table 4.5.

4179     We see that the posterior mean and SD of density (individuals per square km) appear  
4180 insensitive to choice of  $B$  once we reach about  $B = 17$  or so. The estimated density of  
4181 0.25 per  $\text{km}^2$  is actually quite a bit lower than we reported using model  $M_h$  for which no

**Table 4.5.** 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 and SD is the posterior standard deviation.

$B$	Density	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

relevant “area” quantity is explicit in the model, and so we had to make one up. Using MLEs of  $N$  in conjunction with buffer strips (see Tab. 1.1) our estimates were in the range of  $0.32 - 0.43$  and see Sec. 4.4 above. On the other hand our estimate of  $\hat{D} = 0.25$  here (based on the posterior mean) is higher than that reported from model  $M_0$  using the buffered area (0.18). There is no basis really for comparing or contrasting these various estimates. In particular, application of models  $M_0$  and  $M_h$  are distinctly *not* spatially explicit models—the area within which the population resides is not defined under either model. There is therefore no reason at all to think that the estimates produced under either closed population model, based on a buffered “trap area”, are justifiable by any theory. In fact, we would get exactly the same estimate of  $N$  no matter what we declare the area to be. On the other hand, the individual covariate model uses an explicit model for “distance from centroid” that is a reasonable and standard null model—it posits, in the absence of direct information, that individual home range centers are randomly distributed in space and that probability of detection depends on the distance between home range center and the centroid of the trap array. Under this definition of the system, we see that density is invariant to the choice of area, which seems like a desirable feature.

#### 4.5.4 Invariance of density to $B$

Under this individual covariate model, and also under models that we consider in later chapters, a general property of the estimators is that while  $N$  increases with the prescribed “area of the modeled population” (equivalent to  $B$  in this case)<sup>8</sup>, we expect that density estimators should be invariant to this area. In the model used above, we note that  $\text{Area}(B) = \pi B^2$  and  $\mathbb{E}(N(B)) = \lambda \text{Area}(B)$  and thus  $\mathbb{E}(\text{Density}(B)) = \lambda$ , i.e., constant. This should be interpreted as the *prior* density. Absent data, then realizations under the model will have density  $\lambda$  regardless of what  $B$  is prescribed to be. As we verified empirically above, posterior summaries of density are also invariant to  $B$  as long as the prescribed area is sufficiently large.

<sup>8</sup>define this earlier in the section

---

#### 4208 4.5.5 Toward Fully Spatial Capture-recapture Models

4209 While the individual covariate model resolves two important problems inherent in almost  
 4210 all capture-recapture studies (induced heterogeneity and absence of a precise relationship  
 4211 between  $N$  and area), is not ideal for all purposes because it does not make full use of  
 4212 the spatial information in the data set, i.e., the trap locations and the locations of each  
 4213 individual encounter, so that we cannot use this model to model trap-specific effects (e.g.,  
 4214 trap effort or type). Moreover, we applied this model for “data” being the average observed  
 4215 encounter location, and equated that summary to the home range center  $\mathbf{s}_i$ . Intuitively,  
 4216 taking the average encounter location as an estimate of home range center makes sense  
 4217 but more so when the trapping grid is dense and expansive relative to typical home range  
 4218 sizes which might not be reasonable in practice. Moreover, this approach also ignored the  
 4219 variable precision with which each  $\mathbf{s}_i$  is estimated. Finally, it ignores that estimates of  
 4220  $\mathbf{s}_i$  around the “edge” (however we define that) are biased because the observations are  
 4221 truncated—we can only observe locations interior to the array.

4222 However, there is hope to extend this model in order to resolve these remaining defi-  
 4223 ciencies. In the next chapter we provide a further extension of this individual covariate  
 4224 model that definitively resolves the *ad hoc* nature of the approach we took here. In that  
 4225 chapter we build a model in which  $\mathbf{s}_i$  are regarded as latent variables and the observation  
 4226 locations (i.e., trap specific encounters) are linked to those latent variables with an explicit  
 4227 model. We note that the model fitted previously could be adapted easily to deal with  $\mathbf{s}_i$   
 4228 as a latent variable, simply by adding a prior distribution for  $\mathbf{s}_i$ .

## 4.6 DISTANCE SAMPLING: A PRIMITIVE SCR MODEL

4229 Distance sampling is a class of methods for estimating animal density from measurements  
 4230 of distance from an observer to individual animals (or groups). The basic assumption  
 4231 is that detection probability is a function of distance. Distance sampling is one of the  
 4232 most popular methods for estimating animal abundance (Burnham et al., 1980; Buckland  
 4233 et al., 2001; Buckland, 2004) because, unlike ordinary closed population models, distance  
 4234 sampling provides explicit estimates of *density*. In terms of methodological context, the  
 4235 distance sampling model is a special case of a closed population model with an individual  
 4236 covariate. The covariate in this case,  $x_i$ , is the distance between an individual’s location  
 4237 say  $\mathbf{u}$  and the observation location or transect. In fact, the model underlying distance  
 4238 sampling is precisely the same model as that which applies to the individual-covariate  
 4239 models, except that observations are made at only  $K = 1$  sampling occasion. Thus, in that  
 4240 sense, distance sampling is a spatial capture-recapture model, but without the “recapture.”  
 4241 Distance sampling eliminates the need to explicitly identify individuals (except they need  
 4242 to be *distinguished* from other individuals) repeatedly and so distance sampling can be  
 4243 applied to unmarked populations. This first and most basic spatial capture-recapture  
 4244 model has been used routinely for decades and, formally, it is a spatially-explicit model  
 4245 in the sense that it describes, explicitly, the spatial organization of individual locations  
 4246 (although this is not always stated explicitly) and, as a result, somewhat general models  
 4247 of how individuals are distributed in space can be specified (Hedley et al., 1999; Royle  
 4248 et al., 2004; Johnson, 2010; Niemi and Fernández, 2010; Sillett et al., 2012).

4249 As with other models we’ve encountered in this chapter, the distance sampling model,  
 4250 under data augmentation, includes a set of  $M$  zero-inflation variables  $z_i$  and a binomial

4251 observation model expressed conditional on  $z$  (binomial for  $z = 1$ , and fixed zeros for  
 4252  $z = 0$ ). In distance sampling we pay for having only a single sample occasion (i.e.,  $K = 1$ )  
 4253 by requiring constraints on the model of detection probability, normally imposed as the  
 4254 assumption that detection probability is 1.0 when distance equals 0. A standard model  
 4255 for detection probability is the “half-normal” model:

$$\log(p_i) = \alpha_1 x_i^2$$

4256 for  $\alpha_1 < 0$ , where  $x_i$  denotes the distance at which the  $i$ th individual is detected relative  
 4257 to some reference location where perfect detectability ( $p = 1$ ) is assumed. This encounter  
 4258 probability model is more often written with  $\alpha_1 = 1/2\sigma^2$ . If  $K > 1$  then an intercept in  
 4259 this model, say  $\alpha_0$ , is identifiable and such models are usually called “capture-recapture  
 4260 distance sampling” (Alpízar-Jara and Pollock, 1996; Borchers et al., 1998).

4261 As with previous examples, we require a distribution for the individual covariate  $x_i$ .  
 4262 The customary choice is

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

4263 wherein  $B > 0$  is a known constant, being the upper limit of data recording by the observer  
 4264 (i.e., the point count radius, or transect half-width). In practice, this is sometimes asserted  
 4265 to be infinity, but in such cases the distance data are usually truncated. Specification of  
 4266 this distance sampling model in the **BUGS** language is shown in Panel 4.2, taken from  
 4267 Royle and Dorazio (2008).

---

```

alpha1~dunif(0,10)
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 MODEL
  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)])
D<- N/striparea  # area of transects

```

---

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

4268 As with the individual covariate model in the previous section, the distance sampling  
 4269 model can be equivalently specified by putting a prior distribution on individual *location*  
 4270 instead of distance between individual and observation point (or transect). Thus we can  
 4271 write the general distance sampling model as

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

4272 along with

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

4273 where  $\mathbf{x}_0$  is a fixed point (or line) and  $\mathbf{u}_i$  is the individual's location, which is observed for  
 4274 the sample of  $n$  individuals. In practice it is easier to record distance instead of location.  
 4275 Basic math can be used to argue that if individuals have a uniform distribution in space,  
 4276 then the distribution of Euclidean distance is also uniform. In particular, if a transect of  
 4277 length  $L$  is used and  $x$  is distance to the transect then  $F(x) = \Pr(X \leq x) = L*x/L*B =$   
 4278  $x/B$  and  $f(x) = dF/dx = (1/B)$ . For measurements of radial distance, we provided the  
 4279 analogous argument in the previous section.

4280 The preceding paragraph makes it clear that distance sampling is a special case of  
 4281 spatial capture-recapture models, such as those derived from model  $M_x$  of the previous  
 4282 section, where the encounter probability is related directly to *distance*, which is a reduced  
 4283 information summary of *location*,  $\mathbf{u}$ . Some intermediate forms of SCR/DS models can  
 4284 be described (Royle et al., 2011a). In the context of our general characterization of SCR  
 4285 models (Chapt. 2.6), we suggested that every SCR model can be described, conceptually,  
 4286 by a hierarchical model of the form:

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

4287 Distance sampling ignores the part of the model pertaining to  $\mathbf{s}$ , and deals only with the  
 4288 model components for the observed data  $\mathbf{u}$ <sup>9</sup>. Thus, we are left with a hierarchical model  
 4289 of the form

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

4290 In contrast, as we will see in the next chapters, many SCR models (Chapt. 5) ignore  $\mathbf{u}$   
 4291 and condition on  $\mathbf{s}$ , which is not observed:

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

4292 Since  $[\mathbf{u}]$  and  $[\mathbf{s}]$  are both assumed to be uniformly distributed, these are equivalent models!  
 4293 The main differences have to do with interpretation of model components and whether or  
 4294 not the latent variables are observable (in distance sampling they are).

4295 So why bother with SCR models when distance sampling yields density estimates and  
 4296 accounts for spatial heterogeneity in detection? For one, imagine trying to collect distance  
 4297 sampling data on species such as jaguars or tigers! Clearly, distance sampling requires  
 4298 that one can collect large quantities of distance data, which is not always possible. For  
 4299 tigers, it is much easier, efficient, and safer to employ camera traps or tracking plates  
 4300 and then apply SCR models. Furthermore, as we will see in Chapt. 17, SCR models can  
 4301 use distance data, allowing us to study distribution, movement, and density. Thus, SCR  
 4302 models are more general and versatile than distance sampling models (which clearly are  
 4303 a special case), and can accommodate data from virtually all animal survey designs.

#### 4304 4.6.1 Example: Sonoran Desert Tortoise Study

4305 We illustrate the application of distance sampling models using data on the Sonoran desert  
 4306 tortoise (*Gopherus agassizii*), shown in Fig. 4.6, collected along transects in southern

---

<sup>9</sup>Equivalently, we could also say that  $[\mathbf{u}]$  in the distance sampling model is  $[\mathbf{u}] = \int [\mathbf{u}|\mathbf{s}][\mathbf{s}]ds$

4307 Arizona (see Zylstra et al. (2010) for details). The data are from 120 square transects  
 4308 having four 250 m sides, although we ignore this detail in our analysis here and regard  
 4309 them as 1 km transects, and we pooled the detection data from all 120 transects. The  
 histogram of encounter distances from the 65 encounter individuals is shown in Fig. 4.7

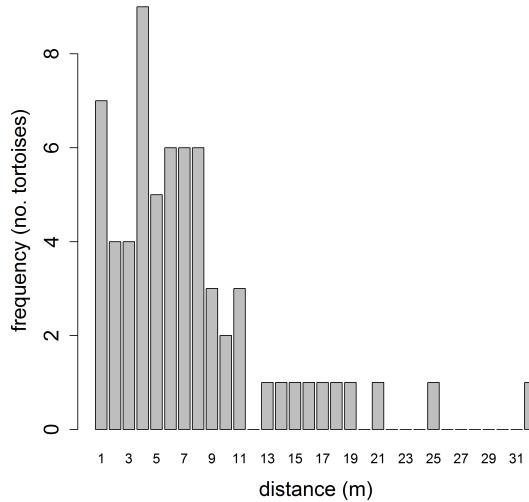


**Figure 4.6.** Desert tortoise in its native habitat (*Photo credit: Erin Zylstra, Univ. of Arizona*).

4310  
 4311 Commands for reading in and organizing the data for analysis using **WinBUGS**, fol-  
 4312 lowed by writing the model to a text file, are given below. Note that the total sampled area  
 4313 of the transects **striparea** is input as data, and computed as: 120 (transects) multiplied  
 4314 by the length (1000 m) and width ( $B = 40$  m), then multiplied by 2, and then divided  
 4315 by 10000 to convert to units of individuals per ha. The full script is provided with the  
 4316 **scrbook R** package (see `?tortoise`), where we also give some commands for analyzing  
 4317 the data with **unmarked** (Fiske and Chandler, 2011) using hierarchical distance sampling  
 4318 models (Royle et al., 2004).

```
4319 data(toroise)
4320 x<-tortoise[,"Dist"]
4321 nind<-sum(!is.na(x))

4322
4323 y<-rep(1,nind)      # create encounter vector
4324 nz<-700               # data augmentation
4325 y<-c(y,rep(0,nz))    # add 0s to the encounter vector
4326 x<-c(x,rep(NA,nz))   # pad distance vector with NA
4327 z<-y                  # starting vals for data augmentation variables
```



**Figure 4.7.** Distance histogram of Sonoran desert tortoise detections from 120 km of survey transect.

```

4328
4329 cat("
4330 model{
4331 alpha1~dunif(0,10)
4332 sigma<- sqrt(1/(2*alpha1))
4333 psi~dunif(0,1)

4334 for(i in 1:(nind+nz)){
4335   z[i]~dbern(psi)      # DA Variables
4336   x[i]~dunif(0,B)      # B=strip width
4337   p[i]<-exp(logp[i])  # DETECTION MODEL
4338   logp[i]<- -alpha1*(x[i]*x[i])
4339   mu[i]<-z[i]*p[i]
4340   y[i]~dbern(mu[i])    # OBSERVATION MODEL
4341 }
4342
4343 N<-sum(z[1:(nind+nz)])
4344 D<- N/striparea  # area of transects
4345 }
4346 ",file="dsamp.txt")
4347
4348 After creating the model file, next we bundle the data, provide initial values, indicate

```

---

4349 which parameters to monitor, and then pass those things to **WinBUGS**:  
 4350 `library(R2WinBUGS)`  
 4351  
 4352 `# density to be reported in units of ind/ha`  
 4353 `data<-list(y=y,x=x,nz=nz,nind=nind,B=40,striparea=(120*1000*40*2/10000))`  
 4354 `params<-list('alpha1','sigma','N','D','psi')`  
 4355 `inits = function() {list(z=z, psi=runif(1), alpha1=runif(1,0,.02) )}`  
 4356 `fit = bugs(data, inits, params, model.file="dsamp.txt",working.directory=getwd(),`  
 4357 `debug=FALSE, n.chains=3, n.iter=3000, n.burnin=1000, n.thin=2)`

4358 Posterior summaries for the tortoise data are given in Tab. 4.6. Estimated density  
 4359 (posterior mean) is 0.54 individuals per ha and the estimated scale parameter of the  
 4360 distance function (posterior mean) is  $\sigma = 9.12$  meters. The Rhat statistics of around 1.02  
 4361 suggest that slightly longer MCMC simulations might be useful. The posterior mass of  
 4362 the data augmentation parameter  $\psi$  is located away from the upper-bound  $\psi = 1$  and so  
 4363 the degree of data augmentation appears sufficient.

**Table 4.6.** 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
$\alpha_1$	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
$N$	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
$\psi$	0.61	0.07	0.49	0.61	0.75	1.02	96

## 4.7 SUMMARY AND OUTLOOK

4364 Traditional closed population capture-recapture models are closely related to binomial  
 4365 generalized linear models. Indeed, the only real distinction is that in capture-recapture  
 4366 models, the population size parameter  $N$  (corresponding also to the size of a hypothetical  
 4367 “complete” data set) is unknown. This requires special consideration in the analysis of  
 4368 capture-recapture models. The classical approach to inference recognizes that the observa-  
 4369 tions don’t have a standard binomial distribution but, rather, a truncated binomial (from  
 4370 which which the so-called *conditional likelihood* derives) since we only have encounter  
 4371 frequency data on observed individuals. If instead we analyze the models using data  
 4372 augmentation, which arises under a Uniform(0,  $M$ ) prior for  $N$ , the observations can be  
 4373 modeled using a zero-inflated binomial distribution. When we deal with the unknown- $N$   
 4374 problem using data augmentation then we are left with zero-inflated GLM and GLMMs in-  
 4375 stead of ordinary GLM or GLMMs. The analysis of such zero-inflated models is practically  
 4376 convenient, especially using the **BUGS** variants.

4377 Spatial capture-recapture models that we will consider in the rest of the chapters of  
 4378 this book are closely related to individual covariate models. Naturally, spatial capture-  
 4379 recapture models arise by defining individual covariates based on observed locations of

4380 individuals—we can think of using some function of mean encounter location as an in-  
4381 dividual covariate. We did this in a novel way, by using distance to the centroid of the  
4382 trapping array as a covariate. We analyzed the *full likelihood* using data augmentation,  
4383 and placed a prior distribution on the individual covariate which was derived from an  
4384 assumption that individual locations are, *a priori*, uniformly distributed in space. This  
4385 assumption provides for invariance of the density estimator to the choice of population  
4386 size area (induced by maximum distance from the centroid of the trap array). The model  
4387 addressed some important problems in the use of closed population models: it allows for  
4388 heterogeneity in encounter probability due to the spatial context of the problem and it also  
4389 provides a direct estimate of density because area is a feature of the model (via the prior  
4390 on the individual covariate). The model is still not completely general because it does not  
4391 make use of the fully spatial encounter histories, which provide direct information about  
4392 the locations and density of individuals. A specific individual covariate model that is in  
4393 widespread use is classical distance sampling. The model underlying distance sampling is  
4394 precisely a special kind of SCR model—but one without replicate samples. Understanding  
4395 distance sampling and individual covariate models more broadly provides a solid basis for  
4396 understanding and analyzing spatial capture-recapture models.



4397

## Part II

4398

---

4399

## Extension to SCR Models



4400  
4401

---

# 5

4402

## FULLY SPATIAL CAPTURE-RECAPTURE 4403 MODELS

4404 In the previous chapter, we discussed models that could be viewed as primitive spatial  
4405 capture-recapture models. We looked at a basic distance sampling model and we also con-  
4406 sidered a classical individual covariate modeling approach in which we defined a covariate  
4407 to be the distance from the (estimated) home range center to the center of the trap array.  
4408 These were spatial in the sense that they included some characterization of where individ-  
4409 uals live but, on the other hand, only a primitive or no characterization of trap location.  
4410 That said, there is only a small step from these two models to spatial capture-recapture  
4411 models that we consider in this chapter, which fully recognize the spatial attribution of  
4412 both individual animals *and* the locations of encounter devices.

4413 Capture-recapture models must accommodate the spatial organization of individuals  
4414 and the encounter devices because the encounter process occurs at the level of individual  
4415 traps. Failure to consider the trap-specific data is the key deficiency with classical ad-  
4416 hoc approaches which aggregate encounter information to the resolution of the entire  
4417 trap array. We have previously addressed some problems that this induces including  
4418 induced heterogeneity in encounter probability, imprecise notation of “sample area” and  
4419 not being able to accommodate trap-specific effects or trap-specific missing values. In  
4420 this chapter we resolve these issues by developing our first fully spatial capture-recapture  
4421 model. This model is not too different from that considered in sec. 4.5 but, instead of  
4422 defining the individual covariate to be distance to the centroid of the array we define  
4423  $J$  individual covariates - the distance to *each* trap. And, instead of using estimates of  
4424 individual locations  $\mathbf{s}$ , we consider a fully hierarchical model in which we regard  $\mathbf{s}$  as a  
4425 latent variable and impose a prior distribution on it.

4426 In the following sections of this chapter we investigate the basic spatial capture-  
4427 recapture model, which we refer to as “model SCR0”, and address some important con-  
4428 siderations related to its analysis in **BUGS**. We demonstrate how to summarize posterior  
4429 output for the purposes of producing density maps or spatial predictions of density. The  
4430 key aspect of the SCR models considered in this chapter is the formulation of a model

for encounter probability that is a function of distance between individual home range center and trap locations. We also discuss how encounter probability models are related to explicit models of space usage or “home range area.” Understanding this allows us to compute, for example, the area used by an individual some prescribed percent of the time. While it is intuitive that SCR models should be related to a basic model of space usage, this has not been discussed much in the literature (but see Royle and Chandler (2012) which we address further in Chapt. 11).

## 5.1 SAMPLING DESIGN AND DATA STRUCTURE

In our development here, we will assume a standard sampling design in which an array of  $J$  traps is operated for  $K$  sample occasions (say, nights) producing encounters of  $n$  individuals. Because sampling occurs by traps and also over time, the most general data structure yields encounter histories for *each individual* that are temporally *and* spatially indexed. Thus a typical data set will include an encounter history *matrix* for each individual indicating which trap the individual was captured, during each sample occasion. For example, suppose we observe 6 individuals in sampling at 4 traps over 3 nights of sampling then a plausible data set for a single individual captured one time in trap 1 on the first night and one time in trap 3 on the 3rd night is:

```
night1 night2 night3
trap1    1    0    0
trap2    0    0    0
trap3    0    0    1
trap4    0    0    0
```

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

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

The statistical assumptions we make to build a model for these data are that individual 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).

**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

4472 These basic (but admittedly at this point somewhat imprecise) assumptions define the ba-  
 4473 sis spatial capture-recapture model, which we will refer to as “SCR0” so that we may use  
 4474 that model as a point of reference without having to provide a long-winded enumeration  
 4475 of assumptions and sampling design each time we do. We will make things more precise  
 4476 as we develop a formal statistical definition of the model shortly.

## 5.2 THE BINOMIAL ENCOUNTER MODEL

4477 We begin by considering the simple model in which there are no time-varying covariates  
 4478 that influence encounter, there are no explicit individual-specific covariates, and there are  
 4479 no covariates that influence density. In this case, we can aggregate the binary encounters  
 4480 over the  $K$  sample occasions and record the total number of encounters out of  $K$ . We will  
 4481 denote these individual- and trap-specific encounter frequencies by  $y_{ij}$  for  $i = 1, 2, \dots, n$   
 4482 captured individuals and  $j = 1, 2, \dots, J$  traps. For example, suppose we observe 6 indi-  
 4483 viduals in sampling at 4 traps over 3 nights of sampling then a plausible data set is the  
 4484 6  $\times$  4 matrix of encounters, out of 3, of the form shown in Table 5.1. We assume that  $y_{ij}$   
 4485 are mutually independent outcomes of a binomial random variable which we express as:

$$y_{ij} \sim \text{Binomial}(K, p_{ij}) \quad (5.2.1)$$

4486 This is the basic model underlying standard closed population models (Chapt. 4) except  
 4487 that, in the present case, the encounter frequencies are individual- *and* trap-specific, and  
 4488 encounter probability  $p_{ij}$  depends on both individual *and* trap.

4489 As we did in sec. 4.5, we will make explicit the notion that  $p_{ij}$  is defined conditional  
 4490 on *where* individual  $i$  lives. Naturally, we think about defining an individual home range  
 4491 and then relating  $p_{ij}$  explicitly to a summary of its location relative to each trap. For  
 4492 example, the centroid of the individuals home range, or its center of activity (Efford, 2004;  
 4493 Borchers and Efford, 2008; Royle and Young, 2008). In what follows, we define  $\mathbf{s}_i$ , a two-  
 4494 dimensional spatial coordinate, to be the home range or activity center of individual  $i$ .  
 4495 Then, the SCR model postulates that encounter probability,  $p_{ij}$ , is a decreasing function  
 4496 of distance between  $\mathbf{s}_i$  and the location of trap  $j$ ,  $\mathbf{x}_j$ . Naturally, if we think of modeling  
 4497 binomial counts using logistic regression, with a model for  $p_{ij}$  such as:

$$\text{logit}(p_{ij}) = \alpha_0 + \alpha_1 \|\mathbf{x}_j - \mathbf{s}_i\| \quad (5.2.2)$$

4498 where, here,  $\|\mathbf{x}_j - \mathbf{s}_i\|$  is the distance between  $\mathbf{s}_i$  and  $\mathbf{x}_j$ . We sometimes write  $\|\mathbf{s}_i - \mathbf{x}_j\| =$   
 4499  $\text{dist}(\mathbf{s}_i, \mathbf{x}_j) = d_{ij}$ . We probably expect that the parameter  $\alpha_1$  in Eq. 5.2.2 or 15.3.1 should

4500 be negative, so that the probability of encounter decreases with distance between the trap  
 4501 and individual home range center. Alternatively, a popular model is

$$p_{ij} = p_0 \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{x}_j - \mathbf{s}_i\|^2\right) \quad (5.2.3)$$

4502 which is similar to the “half-normal” model in distance sampling, except with an intercept  
 4503  $p_0 \leq 1$  which can be estimated in SCR studies. Because it is the kernel of a bivariate  
 4504 normal, or Gaussian, probability density function we will refer to it as the “(bivariate)  
 4505 normal” or “Gaussian” model although the distance sampling term “half-normal” is widely  
 4506 used. However, we adopt the Gaussian/bivariate normal nomenclature here because the  
 4507 model is the kernel of a bivariate density for the random variable “individual location”.  
 4508 In the context of 2-dimensional space, the model is clearly interpretable as a primitive  
 4509 model of movement outcomes or space usage (we discuss this in Sec. 5.3).

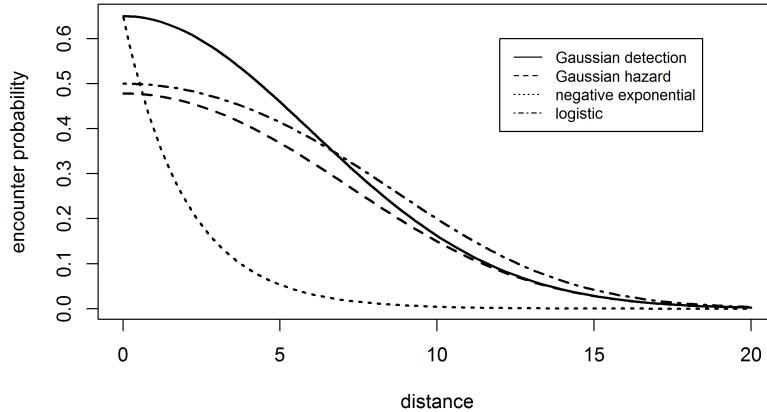
4510 There are a large number of standard detection models commonly used (see Chapt. 8).  
 4511 All other standard models that relate encounter probability to  $\mathbf{s}$  will also have a parameter  
 4512 that multiplies distance in some non-linear function. To be consistent with parameter  
 4513 naming across models, we will sometimes parameterize any encounter probability model  
 4514 so that the coefficient on distance (or distance squared) is  $\alpha_1$ . So, for the Gaussian model,  
 4515  $\alpha_1 = 1/(2\sigma^2)$ . A characteristic of the common parametric forms is they are monotone  
 4516 decreasing in distance, but vary in their characteristic behavior as they approach distance  
 4517 = 0. We show the standard Gaussian, Gaussian hazard, negative exponential and logistic  
 4518 models in Fig. 5.1. The negative exponential model has  $p_{ij} = p_0 \exp(-\alpha_1 \|\mathbf{x}_j - \mathbf{s}_i\|)$  and  
 4519 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  
 4520 kernel. Whatever model we choose for encounter probability, we should always keep in  
 4521 mind that the activity center for individual  $i$ ,  $\mathbf{s}_i$ , is an unobserved random variable. To  
 4522 be precise about this in the model, we should express the observation model as

$$y_{ij} | \mathbf{s}_i \sim \text{Binomial}(K, p(\mathbf{s}_i; \alpha_1))$$

4523 but sometimes, for notational simplicity, we abbreviate this by omitting some of the  
 4524 arguments to  $p$ .

### 4525 5.2.1 Definition of home range center

4526 We define an individual’s home range as *the area used by an organism during some time*  
 4527 *period* which has a clear meaning for most species regardless of their biology. We therefore  
 4528 define the home range center (or activity center) to be the center of the space that indi-  
 4529 vidual was occupying (or using) during the period in which traps were active. Thinking  
 4530 about it in that way, it could even be observable (almost) as the centroid of a very large  
 4531 number of radio fixes over the course of a survey period or a season. Thus, this practical  
 4532 version of a home range center in terms of space usage is a well-defined construct regardless  
 4533 of whether one thinks the home range itself is a meaningful concept, even if individuals  
 4534 are not particularly territorial. We use the terms home range center and activity center  
 4535 interchangeably, and we recognize that this is a transient thing which applies only to a  
 4536 well-defined period of study.



**Figure 5.1.** Some common encounter probability models showing the characteristic monotone decrease of encounter probability with distance between activity center and trap location.  
 XXXX Suggest capitalizing axis labels. Why do they have different intercepts?  
 XXXX

### 4537 5.2.2 Distance as a latent variable

4538 If we knew precisely every  $s_i$  in the population (and population size  $N$ ), then the model  
 4539 specified by Eqs. 5.2.1 and 5.2.2 would be just an ordinary logistic regression-type of  
 4540 a model (with covariate  $d_{ij}$ ) which we learned how to fit using **WinBUGS** previously  
 4541 (Chapt. 3). However, the activity centers are unobservable even in the best possible  
 4542 circumstances. In that case,  $d_{ij}$  is an unobserved variable, analogous to the situation in  
 4543 classical random effects models. We need to therefore extend the model to accommodate  
 4544 these random variables with an additional model component – the random effects distri-  
 4545 bution. The customary assumption is the so-called “uniformity assumption,” which is to  
 4546 assume that the  $s_i$  are uniformly distributed over space (the obvious next question “which  
 4547 space?” is addressed below). This uniformity assumption amounts to a uniform prior  
 4548 distribution on  $s_i$ , i.e., the pdf of  $s_i$  is constant, which we may express

$$\Pr(s_i) \propto \text{constant} \quad (5.2.4)$$

4549 As it turns out, this assumption is usually not precise enough to fit SCR models in practice  
 4550 for reasons we discuss in the shortly. We will give another way to represent this prior  
 4551 distribution that is more concrete, but depends on specifying the “state-space” of the  
 4552 random variable  $s_i$ . The term state-space is a technical way of saying “the space of all  
 4553 possible outcomes” of the random variable.

**5.2.3 The Core SCR Assumptions**

4555 It's always a good idea to sit down and reflect on the meaning of any particular model,  
4556 its various assumptions, and what they mean in a specific context. From the statistician's  
4557 point of view, the basic assumption, the omnibus assumption, as in all of statistics, and  
4558 for every statistical model, is that "the model is correctly specified". So, naturally, that  
4559 precludes everything that isn't explicitly addressed by the model. To point this out to  
4560 someone seems to cause a lot of anxiety, so we enumerate here what we think are the most  
4561 important statistical assumptions of the basic SCR0 model:

- 4562 • **Demographic closure.** The model does not allow for demographic processes. There  
4563 is no recruitment or entry into the sampled population. There is no mortality or exit  
4564 from the sampled population.
- 4565 • **Geographic closure.** Strictly speaking this is not assumed – animals may move around  
4566 in space and sometimes be exposed to trapping and sometimes not. The whole point  
4567 of SCR models is to accommodate this dynamic. In ordinary capture-recapture models  
4568 we have to assume geographic closure to interpret  $N$  in a meaningful way.
- 4569 • **Activity centers are randomly distributed.** That is, uniformity and independence  
4570 of the underlying point process  $s_1, \dots, s_N$ .
- 4571 • **Detection is a function of distance.** A detection model that describes how encounter  
4572 probability declines as a function of distance from an individual's home range center.
- 4573 • **Independence of encounters** among individuals. Encounter of any individual is  
4574 independent of encounter of each other individual.
- 4575 • **Independence of encounters** of the same individual. Encounter of an individual  
4576 in any trap is independent of its encounter in any other trap, and subsequent sample  
4577 occasion.

4578 It's easy to get worried and question the whole SCR enterprise just on the grounds that  
4579 these assumptions combine to form such a simplistic model, one that surely can't describe  
4580 the complexity of real populations. On this sentiment, a few points are worth making.  
4581 First, you don't have inherently fewer assumptions by using an ordinary capture-recapture  
4582 model but, rather, the SCR model relaxes a number of important assumptions compared  
4583 to the non-spatial counterpart. For one, here, we're not assuming that  $p$  is constant for all  
4584 individuals but rather that  $p$  varies substantially as a matter of the spatial juxtaposition of  
4585 individuals with traps. So maybe the manner in which  $p$  varies isn't quite right, but that's  
4586 not an argument that supports doing less modeling. Fundamentally a distance-based  
4587 model for  $p$  has some basic biological justification in virtually every capture-recapture  
4588 study. Secondly, for some of these core assumptions such as uniformity, and independence  
4589 of individuals and of encounters, we expect a fair amount of robustness to departures.  
4590 They function primarily to allow us to build a model and an estimation scheme, we don't  
4591 usually think they represent real populations (of course, no model does!). Third, we can  
4592 extend these assumptions in many different ways and we do that to varying extents in  
4593 this book and more work remains to be done in this regard. Forth, we can also evaluate  
4594 the reasonableness of the assumptions formally in some cases using standard methods of  
4595 assessing model fit (Chapt. 7).

4596 Finally, we return back to our sentiment about the omnibus assumptions which is  
4597 that the model is properly specified. This precludes *everything* that isn't in the model.  
4598 Sometimes you see in capture-recapture literature things like "we assume no marks are

4599 lost”, “marks are correctly identified” and similar things. We might as well also assume  
 4600 that, a shopping mall is not built, or a meteor does not crash down into our study area,  
 4601 the sun does not go super-nova, and things like that. Our point is that we should separate  
 4602 statistical assumptions about model parameters or aspects of the probability model from  
 4603 what are essentially logistical or operational assumptions about how we interpret our data,  
 4604 or based on our ability to conduct or carry-on the study. It is pointless to enumerate all  
 4605 of the possible explanations for apparent *departures*, because there are an infinity of such  
 4606 cases.

### 5.3 THE IMPLIED MODEL OF SPACE USAGE

4607 We developed the basic SCR model in terms of a latent variable,  $\mathbf{s}$ , the home range center  
 4608 or activity center. Surely then, the encounter probability model which relates encounter  
 4609 of individuals in specific traps to  $\mathbf{s}$  must somehow imply a certain model for home range  
 4610 geometry and size – “space usage”. Here we explore the nature of that relationship and  
 4611 we argue that any given detection model implies a model of space usage – i.e., the amount  
 4612 and extent of area used some prescribed percentage of the time. So we might say, for  
 4613 example, 95% of animal movements are within some distance from an individual’s activity  
 4614 center. While we have used the term “home range” or similar, what we really mean to  
 4615 imply is something that would be more clearly identified as resource selection or space  
 4616 usage (the latter term meaning resource selection, when the resource is only homogeneous  
 4617 space).

4618 Intuitively, the detection function of SCR models is related to home range area or  
 4619 space usage of individuals. Indeed, it is natural to interpret the detection model as the  
 4620 composite of two processes: movement of an individual about its home range i.e., how it  
 4621 uses space within its home range (“space usage”), and detection *conditional on use* in the  
 4622 vicinity of a trapping device. It is natural to decompose encounter probability according  
 4623 to:

$$\Pr(\text{encounter at } \mathbf{x}|\mathbf{s}) = \Pr(\text{encounter}|\text{usage of } \mathbf{x}, \mathbf{s}) \Pr(\text{usage of } \mathbf{x}|\mathbf{s}).$$

4624 In practice it might make sense to think about the first component, i.e.,  $\Pr(\text{encounter}|\text{usage of } \mathbf{x}, \mathbf{s})$   
 4625 as being a constant (e.g., if traps are located within arbitrarily small grid cells) and then,  
 4626 in that case, the encounter probability model is directly proportional to this model for  
 4627 individual movements about their home range center determining the use frequency of  
 4628 each  $\mathbf{x}$ . This is a sensible heuristic model for what ecologists would call a central place  
 4629 forager although we have stated previously that it may be meaningful as a description of  
 4630 transient space usage as well.

4631 To motivate a specific model for space usage, imagine some kind of perfect observation  
 4632 device (e.g., continuous telemetry) so that we observe every time an individual moves into  
 4633 a pixel. After a long period of time we observe an enormous sample size of  $\mathbf{x}$  values. We  
 4634 tally those up into each pixel, producing the frequency  $m(\mathbf{x}, \mathbf{s})$ , which is something like the  
 4635 “true” usage of pixel  $\mathbf{x}$  by individual  $\mathbf{s}$ . So, then, the usage model should be regarded as  
 4636 a probability mass function for these counts and, naturally, we regard the counts  $m(\mathbf{x}, \mathbf{s})$   
 4637 as a multinomial observation with probabilities  $\pi(\mathbf{x}|\mathbf{s})$  and prescribe a suitable model for  
 4638  $\pi(\mathbf{x}|\mathbf{s})$  that describes how use events should accumulate in space. A natural null model  
 4639 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  
 4640 more time close to their activity centers than far away. We can regard points used by

4641 the individual with activity center  $\mathbf{s}$  as the realization of a point process with conditional  
 4642 intensity:

$$\pi(\mathbf{x}|\mathbf{s}) = \frac{k(\mathbf{x}, \mathbf{s})}{\int k(\mathbf{x}, \mathbf{s}) d\mathbf{x}}$$

4643 where  $k(\mathbf{x}, \mathbf{s})$  is any positive function. If we represent our landscape by discrete pixels,  
 4644 then this is the equivalent form is:

$$\pi(\mathbf{x}|\mathbf{s}) = \frac{k(\mathbf{x}, \mathbf{s})}{\sum_x k(\mathbf{x}, \mathbf{s})}. \quad (5.3.1)$$

4645 Clearly the space used by an individual will be proportional to whatever kernel,  $k(\mathbf{x}, \mathbf{s})$ , we  
 4646 plug-in for the numerator. If we use a negative exponential function, then this produces a  
 4647 standard resource selection function (RSF) model (e.g., Manly et al., 2002, Ch. 8). But,  
 4648 here we use a Gaussian kernel, i.e.,

$$k(\mathbf{x}, \mathbf{s}) = \exp(-d(\mathbf{x}, \mathbf{s})^2/(2\sigma^2))$$

4649 so that contours of the probability of space usage resemble a bivariate normal or Gaussian  
 4650 probability distribution function.

4651 To apply this model of space-usage to SCR problems we allow for imperfect detection  
 4652 by introducing a constant “thinning rate” of the true counts  $m(\mathbf{x}, \mathbf{s})$ . Then, this yields,  
 4653 precisely, our Gaussian encounter probability model. The main take-away point here is  
 4654 that underlying every SCR model is some kind of model of space-usage, implied by the  
 4655 specific choice of  $k(\mathbf{x}, \mathbf{s})$ . Whether or not we have perfect sampling devices, the function we  
 4656 use in the encounter probability model equates to some conditional distribution of points,  
 4657 a utilization distribution, as in Eq. 5.3.1, from which we can compute effective home range  
 4658 area, i.e., the area that contains some percent of the mass of a probability distribution  
 4659 proportional to  $k(\mathbf{x}, \mathbf{s})$ ; e.g., 95% of all space used by an individual with activity center  $\mathbf{s}$ .

### 4660 5.3.1 Bivariate normal case

4661 One encounter model that allows direct analytic computation of home range area is the  
 4662 Gaussian encounter probability model

$$p(\mathbf{x}, \mathbf{s}) = p_0 \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{x} - \mathbf{s}\|^2\right).$$

4663 For this model, encounter probability is proportional to the kernel of a bivariate normal  
 4664 (Gaussian) pdf and so the natural interpretation is that in which movement outcomes (or  
 4665 successive locations of an individual) are draws from a bivariate normal distribution with  
 4666 standard deviation  $\sigma$ . We say that use of this model implies a bivariate normal model of  
 4667 space usage. Under this model we can compute precisely the effective home range area. In  
 4668 particular, if use outcomes are bivariate normal, then  $\|\mathbf{x} - \mathbf{s}\|^2$  has a chi-square distribution  
 4669 with 2 d.f. and the quantity  $B(\alpha)$  that encloses  $(1 - \alpha)\%$  of all realized distances i.e.,  
 4670  $\Pr(d \leq B(\alpha)) = 1 - \alpha$ , is  $B(\alpha) = \sigma * \sqrt{q(\alpha, 2)}$  where  $q(\alpha, 2)$  is the 0.05 chi-square  
 4671 critical value on 2 df. For example, to compute  $q(.05, 2)$  in R we execute the command  
 4672 `qchisq(.95, 2)` which is  $q(2, \alpha) = 5.99$ . Then, for  $\sigma = 1$ ,  $B(\alpha) = 1 * \sqrt{5.99} = 2.447$ .  
 4673 Therefore 95% of the points used will be within 2.447 (standard deviation) units of the

4674 home range center. So, in practice, we can estimate  $\sigma$  by fitting the bivariate normal  
 4675 encounter probability model to some SCR data, and then use the estimated  $\sigma$  to compute  
 4676 the “95% radius”, say  $r_{.95} = \sigma\sqrt{5.99}$ , and convert this to the 95% *use area* – the area  
 4677 around  $\mathbf{s}$  which contains 95% of the movement outcomes – according to  $A_{.95} = \pi r_{.95}^2$ .

4678 An alternative bivariate normal model is the bivariate normal hazard rate model:

$$p(\mathbf{x}, \mathbf{s}) = 1 - \exp(-\lambda_0 * \exp(-\frac{1}{2\sigma^2} \|\mathbf{x} - \mathbf{s}\|^2)) \quad (5.3.2)$$

4679 We use  $\lambda_0$  here because this parameter, the baseline encounter *rate*, can be  $> 1$ . This arises  
 4680 by assuming the latent “use frequency”  $m(\mathbf{x}, \mathbf{s})$  is a Poisson random variable with intensity  
 4681  $\lambda_0 k(\mathbf{x}, \mathbf{s})$ . The model is distinct from our Gaussian encounter model  $p(\mathbf{x}, \mathbf{s}) = p_0 k(\mathbf{x}, \mathbf{s})$   
 4682 used previously, although we find that they produce similar results in terms of estimates  
 4683 of density or 95% use area, as long as baseline encounter probability is low. We discuss  
 4684 these two formulations of the bivariate normal model further in Chapt. 9.

### 4685 5.3.2 Empirical analysis

4686 For any encounter model we can compute space usage quantiles empirically by taking a fine  
 4687 grid of points and either simulating movement outcomes with probabilities proportional to  
 4688  $p(\mathbf{x}, \mathbf{s})$  and accumulating area around  $\mathbf{s}$ , or else we can do this precisely by varying  $B(\alpha)$   
 4689 to find that value within which 95% of all movements are concentrated, i.e., the set of all  
 4690  $\mathbf{x}$  such that  $\|\mathbf{x} - \mathbf{s}\| \leq B(q)$ . Under any detection model, movement outcomes will occur  
 4691 in proportion to  $p(\mathbf{x}, \mathbf{s})$ , as long as the probability of encounter is constant, *conditional on*  
 4692 *use*, and so we can define our space usage distribution according to:

$$\pi(\mathbf{x}|\mathbf{s}) = \frac{p(\mathbf{x}, \mathbf{s})}{\sum_x p(\mathbf{x}, \mathbf{s})}$$

4693 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

$$\sum_{\mathbf{x} s.t. \|\mathbf{x} - \mathbf{s}\| \leq B(q)} \pi(\mathbf{x}, \mathbf{s}) \leq 1 - q$$

4694 We have a function called **hra** in the **scrbook** package that computes the home range area  
 4695 for any encounter model and prescribed parameter values. The help file for **hra** has an  
 4696 example of simulating some data. The following commands illustrate this calculation for  
 4697 two different bivariate normal models of space usage:

```
4698 ##
4699 ## Define encounter probability model as R function
4700 ##
4701 pGauss2<-function(parms,Dmat){
4702   a0<-parms[1]
4703   sigma<-parms[2]
4704   lp<-  parms[1] -(1/(2*parms[2]*parms[2]))*Dmat*Dmat
4705   p<- 1-exp(-exp(lp))
4706   p
4707 }
```

```

4708
4709 pGauss1<-function(parms,Dmat){
4710   a0<-parms[1]
4711   sigma<-parms[2]
4712   p<- plogis(parms[1])*exp( -(1/(2*parms[2]*parms[2]))*Dmat*Dmat )
4713   p
4714 }
4715
4716 xlim<-c(0,6)
4717 ylim<-c(0,6)
4718
4719 ## execute hra with sigma = .3993
4720 ##
4721 > hra(pGauss1,parms=c(-2,.3993),plot=FALSE,xlim,ylim,ng=500,tol=.0005)
4722 [1] 0.9784019
4723 radius to achieve 95% of area: 0.9784019
4724 home range area: 3.007353
4725 [1] 3.007353
4726 >
4727 >
4728 ## analytic solution
4729 ## true sigma to produce area of 3
4730 > sqrt(3/pi)/sqrt(5.99)
4731 [1] 0.3992751

```

4732 What this means is that  $B(q) = 0.978$  is the radius that encloses about 95% of all  
4733 movements under the standard bivariate normal encounter model. Therefore, the area is  
4734 about  $\pi * 0.978 * 0.978 = 3.007$  spatial units. You can change the intercept of the model  
4735 and find that it has no effect. The true (analytic) value of  $\sigma$  that produces a home range  
4736 area of 3.0 is 0.3993 which is the value we initially plugged in to the **hra** function. We  
4737 can improve on the numerical approximation to home range area (get it closer to 3.0) by  
4738 increasing the resolution of our spatial grid (increase the **ng** argument) along with the **tol**  
4739 argument.

4740 We can also reverse this process, and find, for any detection model, the parameter  
4741 values that produce a certain  $(1 - q)\%$  home range area, which we imagine would be  
4742 useful for doing simulation studies. The function **hra** will compute the value of the scale  
4743 parameter that achieves a certain target  $(1 - q)\%$  home range area, by simply providing a  
4744 non-null value of the variable **target.area**. Here we use **target.area = 3.00735** (from  
4745 above) to obtain a close approximation to the value  $\sigma$  we started with (the parameter  
4746 argument is meaningless here):

```

4747 > hra(pGauss1,parms=c(-2,.3993),plot=FALSE,xlim,ylim,ng=500,
4748       target.area=3.00735,tol=.0005)
4749
4750 Value of parm[2] to achieve 95% home range area of 3.00735: 0.3993674

```

**4751 5.3.3 Relevance of Understanding Space Usage**

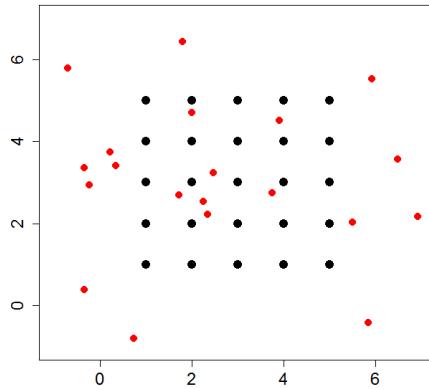
4752 One important reason that we need to be able to deduce “home range area” from a  
4753 detection model is so that we can compare different models with respect to a common  
4754 biological currency. Many encounter probability models have some “scale parameter”,  
4755 which we might call  $\sigma$  no matter the model, but this relates to 95% area in a different  
4756 manner under each model. Therefore, we want to be able to convert different models  
4757 to the same currency. Another reason to understand the relationship between models of  
4758 encounter probability and space usage is that it opens the door to combining traditional  
4759 resource selection data from telemetry with spatial capture-recapture data. In Chapt. 11  
4760 we consider this problem, for the case in which a sample of individuals produces encounter  
4761 history data suitable for SCR models and, in addition, we have telemetry relocations on a  
4762 sample of individuals. This is achieved by regarding the two sources of data as resulting  
4763 from the same underlying process of space usage but telemetry data produce “perfect”  
4764 observations, like always-on camera traps blanketing a landscape. We use this idea to  
4765 model the effect of a measured covariate at each pixel, say  $z(\mathbf{x})$ , on home range size and  
4766 geometry and, hence, the probability of encounter in traps.

**4767 5.4 THE BINOMIAL POINT PROCESS MODEL**

4768 The collection of individual activity centers  $\mathbf{s}_1, \dots, \mathbf{s}_N$  represents a realization of a *binomial*  
4769 *point process* (Illian et al., 2008, p. 61). The binomial point process (BPP) is analogous  
4770 to a Poisson point process in the sense that it represents a “random scatter” of points  
4771 in space - except that the total number of points is *fixed*, whereas, in a Poisson point  
4772 process, it is random (having a Poisson distribution). As an example, we show in Fig. 5.2  
4773 locations of 20 individual activity centers (black dots) in relation to a grid of 25 traps.  
4774 For a Poisson point process the number of such points in the prescribed state-space would  
4775 be random whereas often we will simulate fixed numbers of points, e.g., for evaluating the  
4776 performance of procedures such as how well does our estimator perform of  $N = 50$ ?

4777 It is natural to consider a binomial point process in the context of capture-recapture  
4778 models because it preserves  $N$  in the model and thus preserves the linkage directly with  
4779 closed population models. In fact, under the binomial point process model, model  $M_0$   
4780 and other closed models are simple limiting cases of SCR models, i.e., as the coefficient on  
4781 distance ( $\alpha_1$  above) tends to 0. In addition, use of the BPP model allows us to use data  
4782 augmentation, our preferred tool, for Bayesian analysis of the models as in Chapt. 4, thus  
4783 yielding a methodologically coherent approach to analyzing the different classes of models.  
4784 Despite this, making explicit assumptions about  $N$ , such as Poisson, is convenient in some  
4785 cases (see Chapt. 16).

4786 One consequence of having fixed  $N$  in the BPP model is that the model is not  
4787 strictly a model of “complete spatial randomness”. This is because if one forms counts  
4788  $n(A_1), \dots, n(A_k)$  in any set of disjoint regions say  $A_1, \dots, A_k$ , then these counts are *not*  
4789 independent. In fact, they have a multinomial distribution (see Illian et al., 2008, p. 61).  
4790 Thus, the BPP model introduces a slight bit of dependence in the distribution of points.  
4791 However, in most situations this will have no practical effect on any inference or analysis  
4792 and, as a practical matter, we will usually regard the BPP model as one of spatial in-  
4793 dependence among individual activity centers because each activity center is distributed  
4794 independently of each other activity center. Despite this implicit independence we see



**Figure 5.2.** Realization (small dots) of a binomial point process with  $N = 20$ . The large dots represent trap locations.

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. 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. 13.

#### 5.4.1 The state-space of the point process

Shortly we will focus on Bayesian analysis of this model with  $N$  known so that we can gain some basic experience with important elements of the model. To do this, we note that the individual activity centers  $\mathbf{s}_1, \dots, \mathbf{s}_N$  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 convenient 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

4814 statement of the “uniformity assumption” is

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

4815 where  $\mathcal{S}$  is a precisely defined region. e.g., in Fig. 5.2,  $\mathcal{S}$  is the square defined by  $[-1, 7] \times$   
 4816  $[-1, 7]$ . Thus each of the  $N = 20$  points were generated by randomly selecting each  
 4817 coordinate on the line  $[-1, 7]$ . When points are distributed uniformly over some region,  
 4818 the point process is usually called a *homogeneous point process*.

### 4819 Prescribing the state-space

4820 Evidently, we need to define the state-space,  $\mathcal{S}$ . How can we possibly do this objectively?  
 4821 Prescribing any particular  $\mathcal{S}$  seems like the equivalent of specifying a “buffer” which we  
 4822 have criticized as being ad hoc. How is it, then, that the choice of a state-space is *not* ad  
 4823 hoc? As we observed in Chapt. 4, it is true that  $N$  increases with  $\mathcal{S}$ , but only at the same  
 4824 rate as the area of  $\mathcal{S}$  increases under the prior assumption of constant density. As a result,  
 4825 we say that density is invariant to  $\mathcal{S}$  as long as  $\mathcal{S}$  is sufficiently large. Thus, while  
 4826 choice of  $\mathcal{S}$  is (or can be) essentially arbitrary, once  $\mathcal{S}$  is chosen, it defines the population  
 4827 being exposed to sampling, which scales appropriately with the size of the state-space.

4828 For our simulated system developed previously in this chapter, we defined the state-  
 4829 space to be a square within which our trap array was centered. For many practical  
 4830 situations this might be an acceptable approach to defining the state-space, i.e., just a  
 4831 rectangle around the trap array. Although defining the state-space to be a regular polygon  
 4832 has computational advantages (e.g., we can implement this more efficiently in **BUGS** and  
 4833 cannot for irregular polygons), a regular polygon induces an apparent problem of admitting  
 4834 into the state-space regions that are distinctly non-habitat (e.g., oceans, large lakes, ice  
 4835 fields, etc.). It is difficult to describe complex regions in mathematical terms that can be  
 4836 used in **BUGS**. As an alternative, we can provide a representation of the state-space as  
 4837 a discrete set of points which the **R** package **secr** (Efford, 2011) permits (**secr** uses the  
 4838 term “mask” for what we call the state-space). Defining the state-space by a discrete set  
 4839 of points is handy because it allows specific points to be deleted or not, depending on  
 4840 whether they represent available or suitable habitat (see sec. 5.9). We can also define the  
 4841 state-space as an arbitrary collection of polygons stored as a GIS shapefile which can be  
 4842 analyzed easily by MCMC in **R** (see sec. 14.5), but not so easily in the **BUGS** engines.  
 4843 In what follows below we provide an analysis of the wolverine camera trapping data, in  
 4844 which we define the state-space to be a regular continuous polygon (a rectangle).

### 4845 Invariance to the state-space

4846 We will assert for all models we consider in this book that density is invariant to the size  
 4847 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  
 4848 decreasing function of distance. We can prove this easily by drawing an analogy with a 1-d  
 4849 case involving distance sampling. Let  $y_j$  be the number of individuals captured in some  
 4850 interval  $[d_{j-1}, d_j]$ , and define  $d_J = B$  for some large value of  $B$ . The observations from a  
 4851 survey are  $y_1, \dots, y_J$  and the likelihood is a multinomial likelihood, so the log-likelihood  
 4852 is of the form

$$\text{logL}(y_1, \dots, y_J) = \sum_{j=1}^J y_j \log(\pi_j)$$

4853 where  $\pi_j$  is the probability of detecting an individual in distance class  $j$ , which depends on  
 4854 parameters of the detection function (the manner of which is not relevant for the present

4855 discussion). Choosing  $B$  sufficiently large guarantees that  $\mathbb{E}(y_J) = 0$  and therefore the  
 4856 observed frequency in the “last cell” contributes nothing to the likelihood, in regular  
 4857 situations in which the detection function decays monotonically with distance and prior  
 4858 density is constant. We can think of  $B$  as being related to the state-space in an SCR  
 4859 model;  $B \times L$ ,  $L$  being the length of the transect, defines a rectangular state-space,  $\mathcal{S}$ .  
 4860 Thus, if we choose  $B$  large enough, then we ensure that the expected trap-frequencies  
 4861 beyond  $B$  will be 0, and thus contribute nothing to the likelihood.

4862 Sometimes our estimate of density can be affected by choosing  $\mathcal{S}$  too small. However,  
 4863 this might be sensible if  $\mathcal{S}$  is naturally well-defined. As we discussed in Chapt. 1, **choice**  
 4864 **of  $\mathcal{S}$  is part of the model**, and thus it is sensible that estimates of density might be  
 4865 sensitive to its definition in problems where it is natural to restrict  $\mathcal{S}$ . One could imagine,  
 4866 however, that in specific cases where you’re studying a small population with well-defined  
 4867 habitat preferences, that a problem could arise because changing the state-space around  
 4868 based on differing opinions and GIS layers might have substantial changes on the density  
 4869 estimates and hence those of population size. But this is a real biological problem and a  
 4870 natural consequence of the spatial formalization of capture-recapture models – a feature,  
 4871 not a bug or some statistical artifact – and it should be resolved with better information,  
 4872 research, and thinking. For situations where there is not a natural choice of  $\mathcal{S}$ , we should  
 4873 default to choosing  $\mathcal{S}$  to be very large in order to achieve invariance or, otherwise, evaluate  
 4874 sensitivity of density estimates by trying a couple of different choices of  $\mathcal{S}$ . This is a  
 4875 standard “sensitivity to prior” argument that Bayesians always have to be conscious of.  
 4876 We demonstrate this in our analysis of section 5.8 below. As an additional practical  
 4877 consideration, we note that the area of the state-space  $\mathcal{S}$  affects data augmentation. If  
 4878 you increase the size of  $\mathcal{S}$ , then there are more individuals to account for and therefore the  
 4879 size of the augmented data set  $M$  must increase. This has computational implications.

#### 4880 5.4.2 Connection to Model $M_h$ and Distance Sampling

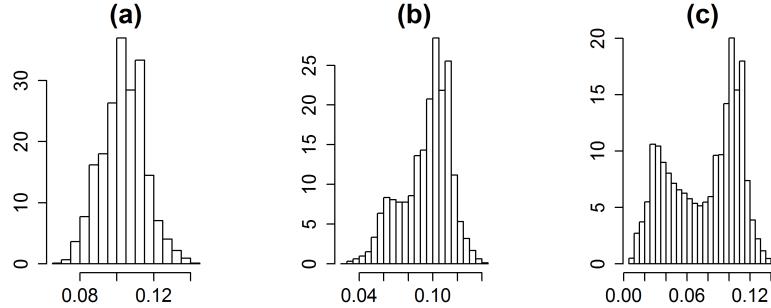
4881 SCR models are closely related to “model  $M_h$ ” and also distance sampling. In SCR  
 4882 models, heterogeneity in encounter probability is induced by both the effect of distance in  
 4883 the model for detection probability and also from specification of the state-space. Hence,  
 4884 the state-space is an explicit element of the model. To understand this, suppose activity  
 4885 centers have the uniform distribution:

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

4886 and encounter probability is a function of  $\mathbf{s}$ , denoted by  $p(\mathbf{s}) = p(y = 1|\mathbf{s})$ . For example,  
 4887 under Eq. 5.2.2 we have that

$$p(\mathbf{s}) = \text{logit}^{-1}(\alpha_0 + \alpha_1 \|\mathbf{s}_i - \mathbf{x}_j\|)$$

4888 and we can work out, either analytically or empirically, what is the implied distribution  
 4889 of  $p$  for a population of individuals. We show an illustration in Fig. 5.3 which shows  
 4890 a histogram of  $p$  for a hypothetical population of 100000 individuals on a state-space  
 4891 enclosing our  $5 \times 5$  trap array above, under the logistic model for distance given by Eq.  
 4892 5.2.2. The histogram shows the encounter probability under buffers of 0.2, 0.5 and 1.0.  
 4893 We see the mass shifts to the left as the buffer increases, implying more individuals with



**Figure 5.3.** Implied population 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.

4894 lower encounter probabilities, as their home range centers increase in distance from the  
 4895 trap array.

4896 Another way to understand this is by representing  $\mathcal{S}$  as a set of discrete points on a  
 4897 grid. In the coarsest possible case where  $\mathcal{S}$  is a single arbitrary point, then every individual  
 4898 has exactly the same  $p$ . As we increase the number of points in  $\mathcal{S}$ , more distinct values  
 4899 of  $p$  are possible. Indeed, when  $\mathcal{S}$  is characterized by discrete points then SCR models  
 4900 are precisely a type of finite-mixture model (Norris III and Pollock, 1996; Pledger, 2000),  
 4901 except, in the case of SCR models, we have some information about which group an  
 4902 individual belongs (i.e., where their activity center is), as a result of which traps they're  
 4903 captured in.

4904 It is also worth re-emphasizing that the basic SCR encounter model is a binomial  
 4905 encounter model in which distance is a covariate. As such, it is strikingly similar to classical  
 4906 distance sampling models (Buckland et al., 2001). Both have distance as a covariate but  
 4907 in classical distance sampling problems the focus is on the distance between the observer  
 4908 and the animal at an instant in time, not the distance between a trap and an animal's  
 4909 home range center. As a practical matter, in distance sampling, "distance" is *observed*  
 4910 for those individuals that appear in the sample. Conversely, in SCR problems, it is  
 4911 only imperfectly observed (we have partial information in the form of trap observations).  
 4912 Clearly, it is preferable to observe distance if possible, but distance sampling requires field  
 4913 methods that are not practical in many situations, e.g. when studying carnivores such as  
 4914 bears or large cats. Furthermore, SCR models allow us to relax many of the assumptions  
 4915 made in classical distance sampling, such as perfect detection at distance zero, and SCR  
 4916 models allow for estimates of quantities other than density, such as home range size, and  
 4917 space usage (see Chaps. 11 and 12).

## 5.5 SIMULATING SCR DATA

4918 It is always useful to simulate data because it allows you to understand the system that  
 4919 you're modeling and also calibrate your understanding with specific values of the model  
 4920 parameters. That is, you can simulate data using different parameter values until you  
 4921 obtain data that "look right" based on your knowledge of the specific situation that  
 4922 you're interested in. Here we provide a simple script to illustrate how to simulate spatial  
 4923 encounter history data. In this exercise we simulate data for 100 individuals and a 25 trap  
 4924 array laid out in a  $5 \times 5$  grid of unit spacing. The specific encounter model is the Gaussian  
 4925 model given above and we used this code to simulate data used in subsequent analyses.  
 4926 The 100 activity centers were simulated on a state-space defined by a  $8 \times 8$  square within  
 4927 which the trap array was centered (thus the trap array is buffered by 2 units). Therefore,  
 4928 the density of individuals in this system is fixed at  $100/64$ .

```

4929 set.seed(2013)
4930 # create 5 x 5 grid of trap locations with unit spacing
4931 traplocs<- cbind(sort(rep(1:5,5)),rep(1:5,5))
4932 Dmat<-e2dist(traplocs,traplocs)
4933 ntraps<-nrow(traplocs)

4934
4935 # define state-space of point process. (i.e., where animals live).
4936 # "buffer" just adds a fixed buffer to the outer extent of the traps.
4937 buffer<-2
4938 xlim<-c(min(traplocs[,1] - buffer),max(traplocs[,1] + buffer))
4939 ylim<-c(min(traplocs[,2] - buffer),max(traplocs[,2] + buffer))

4940
4941 N<-100 # population size
4942 K<- 20 # number nights of effort

4943
4944 sx<-runif(N,xlim[1],xlim[2]) # simulate activity centers
4945 sy<-runif(N,ylim[1],ylim[2])
4946 S<-cbind(sx,sy)
4947 D<- e2dist(S,traplocs) # distance of each individual from each trap

4948
4949 alpha0<- -2.5      # define parameters of encounter probability
4950 sigma<- 0.5        # scale parameter of half-normal
4951 alpha1<- 1/(2*sigma*sigma) # convert to coefficient on distance

4952
4953 probcap<- plogis(-2.5)*exp( - alpha1*D*D) # probability of encounter
4954 # now generate the encounters of every individual in every trap
4955 Y<-matrix(NA,nrow=N,ncol=ntraps)
4956 for(i in 1:nrow(Y)){
4957   Y[i,]<-rbinom(ntraps,K,probcap[i,])
4958 }
```

4959 We remind the reader that, in presenting **R** or other code snippets throughout the  
 4960 book, we will deviate from our standard variable expressions for some quantities. In

particular, we sometimes substitute words for integer variable designations: `nind` (for  $n$ ), `ntraps` (for  $J$ ), and `nperiods` (for  $K$ ). In our opinion this leaves less to be inferred by the reader in trying to understand code snippets.

Subsequently we will generate data using this code packaged in an **R** function called `simSCR0` in the package `scrbook` which takes a number of arguments including `discard0` which, if `TRUE`, will return only the encounter histories for captured individuals. A second argument is `array3d` which, if `TRUE`, returns the 3-d encounter history array instead of the aggregated  $nind \times ntraps$  encounter frequencies (see below). Finally we provide a random number seed, `rnd = 2013` to ensure repeatability of the analysis here. We obtain a data set as above using the following command:

```
4971 data<-simSCR0(discard0=TRUE,array3d=FALSE,rnd=2013)
```

4972 The **R** object `data` is a list, so let's take a look at what's in the list and then harvest some  
4973 of its elements for further analysis below.

```
4974 > names(data)
4975 [1] "Y"          "traplocs" "xlim"       "ylim"       "N"          "alpha0"     "beta"
4976 [8] "sigma"      "K"
4977
4978 ## grab encounter histories from simulated data list
4979 > Y<-data$Y
4980 ## grab the trap locations
4981 > traplocs<-data$traplocs
```

### 4982 5.5.1 Formatting and manipulating real data sets

4983 Conventional capture-recapture data are easily stored and manipulated as a 2-dimensional  
4984 array, an  $nind \times K$  (individuals by sample occasions) matrix, which is maximally informative  
4985 for any conventional capture-recapture model, but not for spatial capture-recapture  
4986 models. For SCR models we must preserve the spatial information in the encounter history  
4987 information. We will routinely analyze data from 3 standard formats:

- 4988 (1) The basic 2-dimensional data format, which is an `nind`  $\times$  `ntraps` encounter frequency  
4989 matrix such as that simulated previously. These are the total number of encounters in  
4990 each trap, summed over the  $K$  sample occasions.
- 4991 (2) The maximally informative 3-dimensional array, for which we establish here the convention  
4992 that it has dimensions `nind`  $\times$  `K`  $\times$  `ntraps`.
- 4993 (3) We use a compact format - the “encounter data file” – which we describe below in  
4994 section 5.8.

4995 To simulate data in the most informative format - the “3-d array” - we can use the **R**  
4996 commands given previously but replace the last 4 lines with the following:

```
4997 Y<-array(NA,dim=c(N,K,ntraps))
4998
4999 for(i in 1:nrow(Y)){
5000   for(j in 1:ntraps){
5001     Y[i,1:K,j]<-rbinom(K,1,probcap[i,j])}
```

```
5002 }
5003 }
```

5004 We see that a collection of  $K$  binary encounter events are generated for *each* individual  
 5005 and for *each* trap. The probabilities of those Bernoulli trials are computed based on the  
 5006 distance from each individual's home range center and the trap (see calculation above),  
 5007 and those are housed in the matrix `probcap`. Our data simulator function `simSRC0` will  
 5008 return the full 3-d array if `array3d=TRUE` is specified in the function call. To recover the  
 5009 2-d matrix from the 3-d array, and subset the 3-d array to individuals that were captured,  
 5010 we do this:

```
5011 # sum over the ‘‘replicates’’ dimension (2nd margin of the array)
5012 Y2d<- apply(Y,c(1,3),sum)
5013     # compute how many times each individual was captured
5014 ncaps<-apply(Y2d,1,sum)
5015     # keep those individuals that were captured
5016 Y<-Y[ncaps>0,,]
```

## 5.6 FITTING MODEL SCR0 IN BUGS

5017 Clearly if we somehow knew the value of  $N$  then we could fit this model directly because,  
 5018 in that case, it is a special kind of logistic regression model, one with a random effect (`s`)  
 5019 that enters into the model in a peculiar fashion, and also with a distribution (uniform)  
 5020 which we don't usually think of as standard for random effects models. So our aim here  
 5021 is to analyze the known- $N$  problem, using our simulated data, as an incremental step in  
 5022 our progress toward fitting more generally useful models.

5023 To begin, we use our simulator to grab a data set and then harvest the elements of  
 5024 the resulting object for further analysis.

```
5025 data<-simSRC0(discard0=FALSE,rnd=2013)
5026 y<-data$Y
5027 traplocs<-data$traplocs
5028     # in this case nind=N because we're doing the known-N problem
5029 nind<-nrow(y)
5030 X<-data$traplocs
5031 J<-nrow(X)    # number of traps
5032 K<-data$K
5033 xlim<-data$xlim
5034 ylim<-data$ylim
```

5035 Note that we specify `discard0 = FALSE` so that we have a “complete” data set, i.e.,  
 5036 one with the all-zero encounter histories corresponding to uncaptured individuals. Now,  
 5037 within an **R** session, we can create the **BUGS** model file and fit the model using the  
 5038 following commands.

```
cat("
model {
alpha0 ~ dnorm(0,.1)
```

---

```

logit(p0)<- alpha0
alpha1 ~ dnorm(0,.1)
sigma<- sqrt(1/(2*alpha1))
for(i in 1:N){ # note N here because N is KNOWN in this example
  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]<- p0*exp(- alpha1*d[i,j]*d[i,j])
  }
}
",file = "SCR0a.txt")

```

5039     This model describes the half-normal detection model but it would be trivial to modify  
 5040     that to various others including the logistic described above. One consequence of using  
 5041     the half-normal is that we have to constrain the encounter probability to be in [0, 1] which  
 5042     we do here by defining **alpha0** to be the logit of the intercept parameter **p0**. Note that the  
 5043     distance covariate is computed within the **BUGS** model specification given the matrix of  
 5044     trap locations, **X**, which is provided to **WinBUGS** as data.

5045     Next we do a number of organizational activities including bundling the data for **Win-**  
 5046 **BUGS**, defining some initial values, the parameters to monitor and some basic MCMC  
 5047 settings. We choose initial values for the activity centers **s** by generating uniform random  
 5048 numbers in the state-space but, for the observed individuals, we replace those values by  
 5049 each individual's mean trap coordinate for all encounters

```

5050 sst<-cbind(runif(nind,xlim[1],xlim[2]),runif(nind,ylim[1],ylim[2])) # starting values for s
5051 for(i in 1:nind){
5052 if(sum(y[i,])==0) next
5053 sst[i,1]<- mean( X[y[i,>0,1] )
5054 sst[i,2]<- mean( X[y[i,>0,2] )
5055 }
5056
5057 data <- list (y=y,X=X,K=K,N=nind,J=J,xlim=xlim,ylim=ylim)
5058 inits <- function(){
5059   list (alpha0=rnorm(1,-4,.4),alpha1=runif(1,1,2),s=sst)
5060 }
5061
5062 library(R2WinBUGS)
5063 parameters <- c("alpha0","alpha1","sigma")
5064 nthin<-1
5065 nc<-3
5066 nb<-1000
5067 ni<-2000
5068 out <- bugs (data, inits, parameters, "SCR0a.txt", n.thin=nthin, n.chains=nc,
5069               n.burnin=nb,n.iter=ni,debug=TRUE,working.dir=getwd())

```

5070 There is little to say about the preceding operations other than to suggest that you  
 5071 might explore the output and investigate additional analyses by running the `simSCR0`  
 5072 script provided in the **R** package `scrbook`.

5073 For purposes here, we ran 1000 burn-in and 1000 post-burn-in iterations, and 3 chains,  
 5074 to obtain 3000 posterior samples. Because we know  $N$  for this particular data set we only  
 5075 have 2 parameters of the detection model to summarize (`alpha0` and `alpha1`), along with  
 5076 the derived parameter  $\sigma$ , the scale parameter of the Gaussian kernel, i.e.,  $\sigma = \sqrt{1/(2\alpha_1)}$ .  
 5077 When the object `out` is produced we print a summary of the results as follows:

```
5078 > print(out,digits=2)
5079 Inference for Bugs model at "SCR0a.txt", fit using WinBUGS,
5080   3 chains, each with 2000 iterations (first 1000 discarded)
5081   n.sims = 3000 iterations saved
5082     mean    sd   2.5%   25%   50%   75% 97.5% Rhat n.eff
5083 alpha0  -2.50  0.22 -2.95 -2.65 -2.48 -2.34 -2.09 1.01  190
5084 alpha1   2.44  0.42  1.64  2.15  2.44  2.72  3.30 1.00  530
5085 sigma    0.46  0.04  0.39  0.43  0.45  0.48  0.55 1.00  530
5086 deviance 292.80 21.16 255.60 277.50 291.90 306.00 339.30 1.01  380
5087
5088 ...
5089 [some output deleted]
5090 ...
```

5091 We know the data were generated with `alpha0` =  $-2.5$  and `alpha1` =  $2$ . The estimates  
 5092 look reasonably close to those data-generating values and we probably feel pretty good  
 5093 about the performance of the Bayesian analysis and MCMC algorithm that **WinBUGS**  
 5094 cooked-up based on our sample size of 1 data set. It is worth noting that the `Rhat`  
 5095 statistics indicate reasonable convergence but, as a practical matter, we might choose to  
 5096 run the MCMC algorithm for additional time to bring these closer to 1.0 and to increase  
 5097 the effective posterior sample size (`n.eff`). Other summary output includes “deviance”  
 5098 and related things including the deviance information criterion (DIC). We discuss general  
 5099 issues of convergence and other MCMC considerations in Chapt. 14, and DIC and model  
 5100 selection in Chapt. 7.

## 5.7 UNKNOWN N

5101 In all real applications  $N$  is unknown. We handled this important issue in Chapt. 4  
 5102 using the method of data augmentation (DA) which we apply here to achieve a realistic  
 5103 analysis of model SCR0. As with the basic closed population models considered previously,  
 5104 we formulate the problem by augmenting our observed data set with a number of “all-  
 5105 zero” encounter histories - what we referred to in Chapt. 4 as potential individuals. If  
 5106  $n$  is the number of observed individuals, then let  $M - n$  be the number of potential  
 5107 individuals in the data set. For the basic  $y_{ij}$  data structure ( $n$  individual  $\times$   $J$  traps  
 5108 encounter frequencies) we simply add additional rows of all-zero observations to that data  
 5109 set. Because such “individuals” are unobserved, they therefore necessarily have  $y_{ij} = 0$  for  
 5110 all  $j$ . A data set, say with 4 traps and 6 individuals, augmented with 4 pseudo-individuals  
 5111 therefore might look like this:

```

5112      trap1 trap2 trap3 trap4
5113      [1,]    1    0    0    0
5114      [2,]    0    2    0    0
5115      [3,]    0    0    0    1
5116      [4,]    0    1    0    0
5117      [5,]    0    0    1    1
5118      [6,]    1    0    1    0
5119      [7,]    0    0    0    0
5120      [8,]    0    0    0    0
5121      [9,]    0    0    0    0
5122      [10,]   0    0    0    0

```

5123     We typically have more than 4 traps and, if we're fortunate, many more individuals  
 5124    in our data set.

5125    For the augmented data, we introduce a set of binary latent variables (the data aug-  
 5126    mentation variables),  $z_i$ , and the model is extended to describe  $\Pr(z_i = 1)$  which is, in  
 5127    the context of this problem, the probability that an individual in the augmented data set  
 5128    is a member of the population of size  $N$  that was exposed to sampling. In other words,  
 5129    if  $z_i = 1$  for one of the all-zero encounter histories, this is implied to be a sampling zero  
 5130    whereas observations for which  $z_i = 0$  are "structural zeros" under the model. Under DA,  
 5131    we also express the Binomial observation model *conditional on  $z_i$*  as follows:

$$y_{ij} \sim \text{Binomial}(K, z_i p_{ij})$$

5132    where we see that the binomial probability evaluates to 0 if  $z_i = 0$  (so  $y_{ij}$  is a fixed 0 in  
 5133    that case) and evaluates to  $p_{ij}$  if  $z_i = 1$ .

5134    How big does the augmented data set have to be? We discussed this issue in Chapt. 4  
 5135    where we noted that the size of the data set is equivalent to the upper limit of a uniform  
 5136    prior distribution on  $N$ . Practically speaking, it should be sufficiently large so that the  
 5137    posterior distribution for  $N$  is not truncated. On the other hand, if it is too large then  
 5138    unnecessary calculations are being done. An approach to choosing  $M$  by trial-and-error  
 5139    is indicated. Do a short MCMC run and then consider whether you need to increase  
 5140     $M$ . See Chapt. 14 for an example of this. Kéry and Schaub (2012, ch. 6) provide an  
 5141    assessment of choosing  $M$  in closed population models. The useful thing about DA is that  
 5142    it removes  $N$  as an explicit parameter of the model. Instead,  $N$  is a derived parameter,  
 5143    computed by  $N = \sum_{i=1}^M z_i$ . Similarly, *density*,  $D$ , is also a derived parameter computed  
 5144    as  $D = N/\text{area}(\mathcal{S})$ .

### 5145    5.7.1 Analysis using data augmentation in WinBUGS

5146    We provide a complete R script for simulating and organizing a data set, and analyzing  
 5147    the data in **WinBUGS**. As before we begin by obtaining a data set using our `simSCRO`  
 5148    routine and then harvesting the required data objects from the resulting data list. Note  
 5149    that we use the `discard0=TRUE` option this time so that we get a "real looking" data set  
 5150    with no all-zero encounter histories:

```
5151  ## Simulate the data and extract the required objects
```

```

5152 data<-simSCR0(discard0=TRUE,rnd=2013)
5153 y<-data$Y
5154 nind<-nrow(y)
5155 X<-data$traplocs
5156 K<-data$K
5157 J<-nrow(X)
5158 xlim<-data$xlim
5159 ylim<-data$ylim

```

5160 After harvesting the data we augment the data matrix  $y$  with  $M - n$  all-zero encounter  
5161 histories, and create starting values for the variables  $z_i$  and also the activity centers  $s_i$   
5162 of which, for each, we require  $M$  values. One thing to take care of in using the **BUGS**  
5163 engines is the starting values for the activity centers. It is usually helpful to start the  $s_i$   
5164 for each observed individual at or near the trap(s) it was captured. All of this happens as  
5165 follows:

```

5166 ## Data augmentation
5167 M<-200
5168 y<-rbind(y,matrix(0,nrow=M-nind,ncol=ncol(y)))
5169 z<-c(rep(1,nind),rep(0,M-nind))

5170
5171 # starting values for s
5172 sst<-cbind(runif(M,xlim[1],xlim[2]),runif(M,ylim[1],ylim[2]))
5173 for(i in 1:nind){
5174   sst[i,1]<- mean( X[y[i,]>0,1] )
5175   sst[i,2]<- mean( X[y[i,]>0,2] )
5176 }

```

5177 Next, we write out the **BUGS** model specification and save it to an external file called  
5178 **SCR0b.txt**. The model specification which now includes  $M$  encounter histories including  
5179 the augmented potential individuals, the data augmentation parameters  $z_i$ , and the data  
5180 augmentation parameter  $\psi$ :

```

5181 cat("
5182 model{
5183 alpha0 ~ dnorm(0,.1)
5184 logit(p0)<- alpha0
5185 alpha1 ~ dnorm(0,.1)
5186 sigma<- sqrt(1/(2*alpha1))
5187 psi ~ dunif(0,1)

5188
5189 for(i in 1:M){
5190   z[i] ~ dbern(psi)
5191   s[i,1] ~ dunif(xlim[1],xlim[2])
5192   s[i,2] ~ dunif(ylim[1],ylim[2])
5193   for(j in 1:J){
5194     d[i,j]<- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
5195     y[i,j] ~ dbin(p[i,j],K)

```

---

```

5196     p[i,j]<- z[i]*p0*exp(- alpha1*d[i,j]*d[i,j])
5197   }
5198 }
5199 N<-sum(z[])
5200 D<-N/64
5201 }
5202 ",file = "SCR0b.txt")

```

5203     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:

```

5206 data <- list (y=y,X=X,K=K,M=M,J=J,xlim=xlim,ylim=ylim)
5207 inits <- function(){
5208   list (alpha0=rnorm(1,-4,.4),alpha1=runif(1,1,2),s=sst,z=z) }
5209
5210 library(R2WinBUGS)
5211 parameters <- c("alpha0","alpha1","sigma","N","D")
5212 out <- bugs (data, inits, parameters, "SCR0b.txt", n.thin=1,n.chains=3,
5213   n.burnin=1000,n.iter=2000,debug=TRUE,working.dir=getwd())

```

5214     Note the differences in this new **WinBUGS** model with that appearing in the known-  
5215  $N$  version – there is not much! The loop over individuals goes up to  $M$  now, and there is a  
5216 model component for the DA variables  $z$ . We are also computing some derived parameters:  
5217 population size  $N(\mathcal{S})$  is computed by summing up all of the data augmentation variables  
5218  $z_i$  (as we've done previously in Chapt. 4) and density,  $D$ , is also a derived parameter,  
5219 being a function of  $N$ . The input data has changed slightly too, as the augmented data  
5220 set has more rows to include excess all-zero encounter histories. Previously we knew that  
5221  $N = 100$  but in this analysis we pretend not to know  $N$ , but think that  $N = 200$  is a  
5222 good upper bound; This analysis can be run directly using the **SCR0bayes** function once  
5223 the **scrbook** package is loaded, by issuing the following commands:

```

5224 library(scrbook)
5225 data<-simSCR0(discard0=TRUE,rnd=2013)
5226 out1<-SCR0bayes(data,M=200,engine="winbugs",ni=2000,nb=1000)

```

5227 Summarizing the output from **WinBUGS** produces:

```

5228 > print(out1,digits=2)
5229 Inference for Bugs model at "SCR0b.txt", fit using WinBUGS,
5230 3 chains, each with 2000 iterations (first 1000 discarded)
5231 n.sims = 3000 iterations saved
5232        mean    sd   2.5%   25%   50%   75% 97.5% Rhat n.eff
5233 alpha0   -2.57  0.23  -3.04  -2.72  -2.56  -2.41  -2.15 1.01  320
5234 alpha1    2.46  0.42   1.63   2.16   2.46   2.73   3.33 1.02  120
5235 sigma     0.46  0.04   0.39   0.43   0.45   0.48   0.55 1.02  120
5236 N        113.62 15.73  86.00 102.00 113.00 124.00 147.00 1.01  260
5237 D         1.78  0.25   1.34   1.59   1.77   1.94   2.30 1.01  260

```

**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
$\alpha_0$	-2.57	-2.60	-1.51	-0.47
$\alpha_1$	2.46	2.56	3.59	3.86
N	113.62	114.16	119.69	118.29
D	1.78	1.78	1.87	1.85
hra	3.85	3.78	5.51	2.64

```

5238 deviance 302.60 23.67 261.19 285.47 301.50 317.90 354.91 1.00 1400
5239
5240 [...some output deleted...]
5241

```

5242 The **Rhat** statistic (discussed in secs. 3.5.2 and 14.4.5) for this analysis indicates  
5243 satisfactory convergence. We see that the estimated parameters ( $\alpha_0$  and  $\alpha_1$ ) are comparable  
5244 to the previous results obtained for the known- $N$  case, and also not too different  
5245 from the data-generating values. The posterior of  $N$  overlaps the data-generating value  
5246 substantially.

### 5.7.2 Implied Home Range Area

5247 Here we apply the method described in sec. 5.3 to compute the effective home range area  
5248 under some different encounter probability models fit to simulated data. We simulated  
5249 data from the Gaussian kernel model as in sec. 5.7 and then we fitted 4 models to it: (1)  
5250 the true data-generating model; (2) the “hazard” or complementary log-log link model  
5251 (Eq. 5.3.2); (3) the negative exponential model and (4) the logit model (Eq. 5.2.2). We  
5252 modified the function **SCR0bayes** for this purpose which you should be able to do with  
5253 little difficulty. We fit each model to the same simulated data set using **WinBUGS**, based  
5254 only on 1000 post-burn-in samples and 3 chains, which produced the posterior summaries  
5255 given in Table 5.2. The main thing we see is that, while the implied home range area can  
5256 vary substantially, there are smaller differences in the estimated  $N$  and hence  $D$ .

### 5.7.3 Use of other BUGS engines: JAGS

5258 There are two other popular **BUGS** engines in widespread use: **OpenBUGS** (Thomas  
5259 et al., 2006) and **JAGS** (Plummer, 2003). Both of these are easily called from **R**. **Open-**  
5260 **BUGS** can be used instead of **WinBUGS** by changing the package option in the **bugs**  
5261 call to **package='OpenBUGS'**. **JAGS** can be called using the function **jags()** in package  
5262 **R2jags** which has nearly the same arguments as **bugs()**. Or, it can be executed from the  
5263 **R** package **rjags** (?) which has a slightly different implementation that we demonstrate  
5264 here as we reanalyze the simulated data set in the previous section (note: the same **R**  
5265 commands are used to generate the data and package the data, inits and parameters to  
5266 monitor). The function **jags.model** is used to initialize the model and run the MCMC  
5267 algorithm for an adaptive period during which tuning of the MCMC algorithm might take

5269 place. These samples cannot be used for inference. Then the Markov chains are updated  
 5270 using `coda.samples()` to obtain posterior samples for analysis, as follows:

```
5271 jinit<- jags.model("SCR0b.txt", data=data, inits=inits,
5272   n.chains=3, n.adapt=1000)
5273 jout <- coda.samples(jinit, parameters, n.iter=1000, thin=1)
```

5274 These commands can be executed using the function `SCR0bayes` provided with the **R**  
 5275 package `scrbook`. Hobbs (2011) provides a good introduction to ecological modeling with  
 5276 **JAGS** which we recommend.

## 5.8 WOLVERINE CAMERA TRAPPING STUDY

5277 We provide an analysis here of camera trapping data from a study of wolverines *Gulo gulo*  
 5278 (Magoun et al., 2011; Royle et al., 2011b). The study took place in SE Alaska (Fig. 5.4)  
 5279 where 37 cameras were operational for variable periods of time (min = 5 days, max = 108  
 5280 days, median = 45 days). A consequence of this is that the binomial sample size  $K$  (see  
 5281 Eq. 5.2.1) is variable for each camera. Thus, we must provide a matrix of sample sizes as  
 5282 data to **BUGS** and modify the model specification in sec. 5.7 accordingly. Our treatment  
 5283 of the data here is based on the analysis of Royle et al. (2011b).

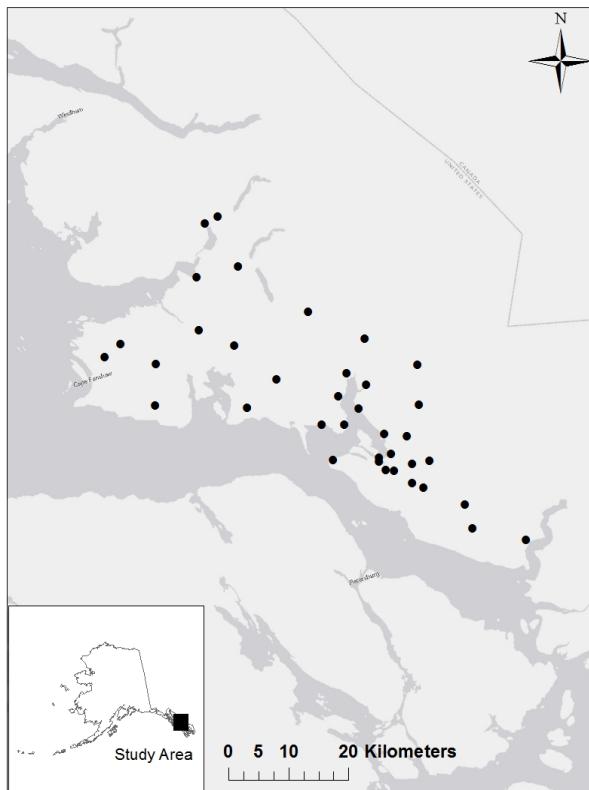
### 5.8.1 Practical Data Organization

5285 To carry-out an analysis of these data, we require the matrix of trap coordinates and the  
 5286 encounter history data. We usually store data in 2 distinct data files which contain all  
 5287 the information needed for an analysis. These files are

- 5288 • The encounter data file (EDF) containing a record of which traps and when each individual encounter occurred.
- 5289 • The trap deployment file (TDF) which contains the coordinates of each trap, along with information indicating which sample occasions each trap was operating.

5292 **Encounter Data File (EDF)** – We store the encounter data in the an efficient file format  
 5293 which is easily manipulated in **R**) and easy to create in Excel and other spreadsheets  
 5294 which are widely used for data management. The file structure is a simple matrix with 4  
 5295 columns, those being: (1) `sessID`: the trap session which usually corresponds to a year or  
 5296 a primary period in the context of a Robust Design situation, but it could also correspond  
 5297 to a distinct spatial unit (see sec. 6.5.4 and Chapt. 16). For a single-year study (as  
 5298 considered here) this should be an integer that is constant for all records; (2) `indID`: the  
 5299 individual identity, being an integer from 1 to  $n$  (repeated for multiple captures of the  
 5300 same individual) indicating which individual the record (row) of the matrix belongs to;  
 5301 (3) `sampID`: The integer sample occasion which generated the record, and (4) `trapID`: the  
 5302 trap identity, an integer from 1 to  $J$ , the number of traps. The structure of the EDF  
 5303 is the same as used in the `secr` package (Efford, 2011) and similar to that used in the  
 5304 **SPACECAP** (Gopalaswamy et al., 2012a), and **SCRbayes** (Russell et al., 2012) packages,  
 5305 both of which have a 3-column format (`trapID`, `indID`, `sampID`).

5306 To illustrate this format, the wolverine data are available in the package `scrbook` by  
 5307 typing:



**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.

```

5308 data(wolverine)
5309 which contains a list having elements wcaps (the EDF) and wtraps (the TDF). We see
5310 that wcaps has 115 rows, each representing a unique encounter event including the trap
5311 identity, the individual identity and the sample occasion index (sample). The first 5 rows
5312 of wcaps are:
5313 > wolverine$wcaps[1:5,]
5314   year individual day trap
5315 [1,]    1          2 127   1
5316 [2,]    1          2 128   1
5317 [3,]    1          2 129   1
5318 [4,]    1         18 130   1

```

---

```
5319 [5,]    1      3 106    2
```

5320 To reiterate the structure, the variable `year` is an integer indicating the year or session  
 5321 of the encounter. All these data come from a single year (2008) and so `year` is set to 1.  
 5322 Variable `individual` is an integer identity of each individual captured, `day` is the sample  
 5323 occasion of capture (in this case, the sample occasions correspond to days), and `trap` is  
 5324 the integer trap identity. Often, the variable `trapid` will have to correspond to the row  
 5325 of a matrix containing the trap coordinates - in this case the TDF file `wtraps` which we  
 5326 describe further below.

5327 Note that the information provided in this encounter data file `wcaps` does not represent  
 5328 a completely informative summary of the data. For example, if no individuals were  
 5329 captured in a certain trap or during a certain period, then this compact data format will  
 5330 have no record. Thus we will need to know  $J$ , the number of traps, and  $K$ , the number of  
 5331 sample occasions when reformatting this SCR data format into a 2-d encounter frequency  
 5332 matrix or 3-d array. In addition, the encounter data file does not provide information  
 5333 about which periods each trap was operated. This additional information is also necessary  
 5334 as the trap-specific sample sizes must be passed to **BUGS** as data. We provide this  
 5335 information along with trap coordinates, in the “trap deployment file” (TDF) which is  
 5336 described below.

5337 For our purposes we need to convert the `wcaps` file into the  $n \times J$  array of binomial  
 5338 encounter frequencies, although more general models might require an encounter-history  
 5339 formulation of the model which requires a full 3-d array. To obtain our encounter frequency  
 5340 matrix, we do this the hard way by first converting the encounter data file into a 3-d array  
 5341 and then summarize to trap totals. We have a handy function `SCR23darray` which takes  
 5342 the compact encounter data file, and converts it to a 3-d array, and then we use the **R**  
 5343 function `apply` to summarize over the sample occasion dimension (by convention here,  
 5344 this is the 2nd dimension). To apply this to the wolverine data in order to compute the  
 5345 3-d array we do this:

```
5346 y3d <-SCR23darray(wolverine$wcaps,wolverine$wtraps)
5347 y <- apply(y3d,c(1,3),sum)
```

5348 See the help file for more information on `SCR23darray`. The 3-d array is necessary to  
 5349 fit certain types of models (e.g., behavioral response) and this is why we sometimes will  
 5350 require this maximally informative 3-d data format but, here, we analyze the summarized  
 5351 data.

5352 **Trap Deployment File (TDF)** – The other important information needed to fit SCR  
 5353 models is the “trap deployment file” (TDF) which provides the additional information not  
 5354 contained in the encounter data file. The traps file has  $K + 3$  columns. The first column is  
 5355 assumed to be a trap identifier, columns 2 and 3 are the easting and northing coordinates  
 5356 (assumed to be in a Euclidean coordinate system), and columns 4 to  $K + 3$  are binary  
 5357 indicators of whether each trap was operational during each sample occasion. The first 10  
 5358 rows (out of 37) and 10 columns (out of 167) of the trap deployment file for the wolverine  
 5359 data are shown as follows:

```
5360 wolverine$wtraps[1:10,1:10]
5361
```

```

5362      Easting Northing 1 2 3 4 5 6 7 8
5363 1 632538 6316012 0 0 0 0 0 0 0 0
5364 2 634822 6316568 1 1 1 1 1 1 1 1
5365 3 638455 6309781 0 0 0 0 0 0 0 0
5366 4 634649 6320016 0 0 0 0 0 0 0 0
5367 5 637738 6313994 0 0 0 0 0 0 0 0
5368 6 625278 6318386 0 0 0 0 0 0 0 0
5369 7 631690 6325157 0 0 0 0 0 0 0 0
5370 8 632631 6316609 0 0 0 0 0 0 0 0
5371 9 631374 6331273 0 0 0 0 0 0 0 0
5372 10 634068 6328575 0 0 0 0 0 0 0 0

```

5373 This tells us that trap 2 was operated in periods (days) 1-7 but the other traps were  
5374 not operational during those periods. It is extremely important to recognize that each  
5375 trap was operated for a variable period of time and thus the binomial “sample size” is  
5376 different for each, and this needs to be accounted for in the **BUGS** model specification.  
5377 To compute the vector of sample sizes  $K$ , and extract the trap locations, we do this:

```

5378 traps<- wolverine$wtraps
5379 traplocs<- traps[,1:2]
5380 K<- apply(traps[,3:ncol(traps)],1,sum)

```

5381 This results in a matrix `traplocs` which contains the coordinates of each trap and a vector  
5382  $K$  containing the number of days that each trap was operational. We now have all the  
5383 information required to fit a basic SCR model in **BUGS**.

5384 Summarizing the data for the wolverine study, we see that 21 unique individuals were  
5385 captured a total of 115 times. Most individuals were captured 1-6 times, with 4, 1, 4, 3, 1,  
5386 and 2 individuals captured 1-6 times, respectively. In addition, 1 individual was captured  
5387 each 8 and 14 times and 2 individuals each were captured 10 and 13 times. The number  
5388 of unique traps that captured a particular individual ranged from 1-6, with 5, 10, 3, 1,  
5389 1, and 1 individual captured in each of 1-6 traps, respectively, for a total of 50 unique  
5390 wolverine-trap encounters. These numbers might be hard to get your mind around whereas  
5391 some tabular summary is often more convenient. For that it seems natural to tabulate  
5392 individuals by trap and total encounter frequencies. The spatial information in SCR data  
5393 is based on multi-trap captures<sup>1</sup>, and so, it is informative to understand how many unique  
5394 traps each individual is captured in. At the same time, it is useful to understand how  
5395 many total captures we have of each individual because this is, in an intuitive sense, the  
5396 effective sample size. So, we reproduce Table 1 from Royle et al. (2011b) which shows, in  
5397 Table 5.3, the trap and total encounter frequencies.

### 5398 5.8.2 Fitting the model in WinBUGS

5399 For illustrative purposes here we fit the simplest SCR model with the Gaussian distance  
5400 function although we revisit these data with more complex models in later chapters. The  
5401 model is summarized by the following 4 elements:

<sup>1</sup>I will add more context here on revision about spatial recaptures, lost recaptures, ordinary recaptures. Function `SCRsm` in `scrbook`. **XXXX THIS NEEDS WRITTEN XXXXX**

**Table 5.3.** Individual frequencies of capture for wolverines captured in camera traps in South-east 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	1	0	0

- 5402 (1)  $y_{ij}|\mathbf{s}_i \sim \text{Binomial}(K, z_i p_{ij})$   
 5403 (2)  $p_{ij} = p_0 \exp(-\alpha_1 ||\mathbf{x}_j - \mathbf{s}_i||^2)$   
 5404 (3)  $\mathbf{s}_i \sim \text{Uniform}(\mathcal{S})$   
 5405 (4)  $z_i \sim \text{Bernoulli}(\psi)$

5406 We assume customary flat priors on the structural (hyper-) parameters of the model,  
 5407  $\alpha_0 = \text{logit}(p_0)$ ,  $\alpha_1$  and  $\psi$ .

5408 It remains to define the state-space  $\mathcal{S}$ . For this, we nested the trap array (Fig. 5.4) in  
 5409 a rectangular state-space extending 20 km beyond the traps in each cardinal direction. We  
 5410 also considered larger state-spaces up to 50 km to evaluate that choice. The buffer of the  
 5411 state space should be large enough so that individuals beyond the state-space boundary  
 5412 are not likely to be encountered (sec. 5.4.1). Thus, some knowledge of typical space usage  
 5413 patterns of the species is useful to establish the model state-space. For the analysis, we  
 5414 scaled the coordinate system so that a unit distance was equal to 10 km, producing a  
 5415 rectangular state-space of dimension  $9.88 \times 10.5$  units ( $\text{area} = 10374 \text{ km}^2$ ) within which  
 5416 the trap array was nested. As a general rule, we recommend scaling the state-space so  
 5417 that it is defined near the origin  $(x, y) = (0, 0)$ . While the scaling of the coordinate system  
 5418 is theoretically irrelevant, a poorly scaled coordinate system can produce Markov chains  
 5419 that mix poorly. For the scaled coordinate system we fit models for various choices of  
 5420 a rectangular state-space based on buffers from 1.0 to 5.0 units on the scaled coordinate  
 5421 system (10 km to 50 km). In the **R** package **scrbook** we provide a function **wolvSCR0**  
 5422 which will fit the basic SCR model. For example, to fit the model in **WinBUGS** using  
 5423 data augmentation with  $M = 300$  potential individuals, using 3 Markov chains each of  
 5424 12000 total iterations, discarding the first 2000 as burn-in, we execute the following **R**  
 5425 commands:

```
5426 library(scrbook)
5427 data(wolverine)
5428 traps<-wolverine$wtraps
5429 y3d <-SCR23darray(wolverine$wcaps,wolverine$wtraps)
5430 toad<-wolvSCR0(y3d,traps,nb=2000,ni=12000,buffer=1,M=300)
```

5431 The argument **buffer** determines the buffer size of the state-space. Note that this  
 5432 analysis takes between 1-2 hours on many machines so we recommend trying it out with

**Table 5.4.** Posterior summaries of SCR model parameters for the wolverine camera trapping data from SE Alaska. 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	
55	0.64	0.05	1600	169.28	35.81	260	5.73	1.21	260	

**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
$\psi$	0.20	0.05	0.12	0.17	0.20	0.23	0.30	1
$\alpha_1$	1.26	0.21	0.87	1.11	1.25	1.40	1.71	1
$\sigma$	0.64	0.06	0.54	0.60	0.63	0.67	0.76	1
$p_0$	0.06	0.01	0.04	0.05	0.06	0.06	0.08	1
$N$	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

lower values of  $M$  and fewer iterations. The posterior summaries are shown in Table 5.8.2.

### 5.8.3 Summary of the Wolverine Analysis

We see that the estimated density is roughly consistent as we increase the state-space buffer from 15 to 55 km. We do note that the data augmentation parameter  $\psi$  (and, correspondingly,  $N$ ) increase with the size of the state space in accordance with the deterministic relationship  $N = D * A$ . However, density is more or less constant as we increase the size of the state-space beyond a certain point. For the 10 km state-space buffer, we see a slight effect on the posterior distribution of  $D$  because the state-space is not sufficiently large. The full results from the analysis based on 20 km state-space buffer are given in Table 5.5.

Our point estimate of wolverine density from this study, using the posterior mean from the state-space based on the 20 km buffer, is approximately 5.77 individuals/1000 km<sup>2</sup> with a 95% posterior interval of [3.86, 8.29]. Density is estimated imprecisely which might not be surprising given the low sample size ( $n = 21$  individuals!). This seems to be a

5447 basic feature of carnivore studies although it should not (in our view) preclude the study  
 5448 of their populations by capture-recapture nor attempts to estimate density or vital rates.

5449 It is worth thinking about this model, and these estimates, computed under a rect-  
 5450 angular state space roughly centered over the trapping array (Fig. 5.4). Does it make  
 5451 sense to define the state-space to include, for example, ocean? What are the possible  
 5452 consequences of this? What can we do about it? There's no reason at all that the state  
 5453 space has to be a regular polygon – we defined it as such here strictly for convenience and  
 5454 for ease of implementation in **WinBUGS** where it enables us to specify the prior for the  
 5455 activity centers as uniform priors for each coordinate. While it would be possible to define  
 5456 a more realistic state-space using some general polygon GIS coverage, it might take some  
 5457 effort to implement that in the **BUGS** language but it is not difficult to devise custom  
 5458 MCMC algorithms to do that (see Chapt. 14). Alternatively, we recommend using a  
 5459 discrete representation of the state-space – i.e., approximate  $\mathcal{S}$  by a grid of  $G$  points. We  
 5460 discuss this in sec. 5.9.

#### 5461 5.8.4 Wolverine Space Usage

5462 The parameter  $\alpha_1$  is related to the home range radius (sec. 5.3). For the Gaussian kernel  
 5463 model we interpret the scale parameter  $\sigma$ , related to  $\alpha_1$  by  $\alpha_1 = 1/(2\sigma^2)$ , as the radius  
 5464 of a bivariate normal model of space usage. In this case  $\sigma = 0.64$  standardized units (10  
 5465 km), which corresponds to  $0.64 \times 10 = 6.4$  km. It can be argued then that 95% of space  
 5466 used by an individual is within  $6.4\text{km} \times \sqrt{5.99} = 15.66$  km of the home range center. The  
 5467 effective “home range area” is then the area of this circle, which is  $\pi \times 15.66^2 = 770.4 \text{ km}^2$   
 5468 Using our handy function **hra** we do this:

```
5469 hra(pGauss1,parms=c(-2,1/(2*.64*.64)),xlim=c(-1,7),ylim=c(-1,7))
5470 [1] 7.731408
```

5472 which is in units of  $100 \text{ km}^2$ , so 773.1. The difference in this case is due to numerical  
 5473 approximation of our all-purpose tool **hra**. This home range size is relatively huge for  
 5474 measured home ranges, which range between 100 and  $535 \text{ km}^2$  (Whitman et al., 1986).

5475 In Royle et al. (2011b), they expressed the model like  $\text{cloglog}(p_{ij}) = \alpha_0 - (1/\sigma^2) * d_{ij}^2$ ,  
 5476 but the estimates of  $\sigma$  reported in their Table 2 are actually based on the model according  
 5477 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  
 5478 are consistent to what we report here except based on the complementary log-log link  
 5479 instead of the Gaussian model. Their estimates for  $\sigma$  are  $6.3 - 9.8$  km depending on the  
 5480 model, which isn't too different than here. However, these estimates are larger than the  
 5481 typical home range sizes suggested in the literature. One possible explanation is that if a  
 5482 wolverine is using traps as a way to get yummy chicken, so it's moving from trap to trap  
 5483 instead of adhering to “normal” space usage patterns, then the implied home range size  
 5484 might not be worth much biologically. Thus, interpretation of detection models in terms  
 5485 of home range area depends on some additional context or assumptions, such as that traps  
 5486 don't effect individual space usage patterns. As such, we caution against direct biological  
 5487 interpretations of home range area based on  $\sigma$ .

5488 We can calibrate the desired size of the state-space by looking at the estimated home  
 5489 range radius of the species. We should target a buffer of width  $2 - 3 \times \sigma$  in order that the

5490 probability of encountering an individual is very close to 0 beyond the prescribed state-space.  
5491 Essentially, by specifying a state-space, we're setting  $p = 0$  for individuals beyond  
5492 the prescribed state-space.

## 5.9 USING A DISCRETE HABITAT MASK

5493 The SCR model developed previously in this chapter assumes that individual activity  
5494 centers are distributed uniformly over the prescribed state-space. Clearly this will not  
5495 always be a reasonable assumption. In Chapt. 13 we talk about developing models that  
5496 allow explicitly for non-uniformity of the activity centers by modeling covariate effects  
5497 on density. A simplistic method of affecting the distribution of activity centers, which  
5498 we address here, is to modify the shape and organization of the state-space explicitly.  
5499 For example, we might be able to classify the state-space into distinct blocks of habitat  
5500 and non-habitat. In that case we can remove the non-habitat from the state-space and  
5501 assume uniformity of the activity centers over the remaining portions judged to be suitable  
5502 habitat. There are several ways to approach this: We can use a grid of points to represent  
5503 the state-space, i.e., by the set of coordinates  $s_1, \dots, s_G$ , and assign equal probabilities to  
5504 each possible value. Alternatively, we can retain the continuous formulation of the state-  
5505 space but attempt to describe constraints analytically, or we can use polygon clipping  
5506 methods to enforce constraints on the state-space in the MCMC analysis. We focus here  
5507 on the formulation of the basic SCR model in terms of a discrete state-space but in Chapt.  
5508 14 we demonstrate the latter approach based on using polygon operations to define an  
5509 irregular state-space. Use of a discrete state-space can be computationally expensive in  
5510 **WinBUGS**. That said, it isn't too difficult to perform the MCMC calculations in **R**  
5511 (discussed in Chapt. 14). The **R** package **SPACECAP** (Gopalaswamy et al., 2012a) arose  
5512 from the **R** implementation of the SCR model in Royle et al. (2009a).

5513 While clipping out non-habitat seems like a good idea, we think investigators should  
5514 go about this very cautiously. We might prefer to do it when non-habitat represents a  
5515 clear-cut restriction on the state-space such as a reserve boundary or a lake, ocean or  
5516 river. But, having the capability to do this also causes people to start defining "habitat"  
5517 vs. "non-habitat" based on their understanding of the system whereas it can't be known  
5518 whether the animal being studied has the same understanding. Moreover, differentiating  
5519 the landscape by habitat or habitat quality must affect the geometry and morphology of  
5520 home ranges (see Chapt. 11) much more so than the plausible locations of activity centers.  
5521 That is, a home range centroid could, in actual fact, occur in a shopping mall parking lot  
5522 if there is pretty good habitat around the shopping mall, so there is probably no sense  
5523 preclude it as the location for an activity center. It would generally be better to include  
5524 some definition of habitat quality in the model for the detection probability (Royle et al.,  
5525 2012c) which we address in Chaps. 11 and 12.

### 5.9.1 Evaluation of Coarseness of Discrete Approximation

5526 The coarseness of the state-space should not really have much of an effect on estimates  
5527 if the grain is sufficiently fine relative to typical animal home range sizes. Why is this?  
5528 We have two analogies that can help us understand. First is the relationship to model  
5529  $M_h$ . As noted in sec. 5.4.2 above, we can think about SCR models as a type of finite

5531 mixture (Norris III and Pollock, 1996; Pledger, 2000) where we are fortunate to be able to  
 5532 obtain direct information about which group individuals belong to (group being location  
 5533 of activity center). In the standard finite mixture models we typically find that a small  
 5534 number of groups (e.g., 2 or 3 at the most) can explain high levels of heterogeneity and  
 5535 are adequate for most data sets of small to moderate sample sizes. We therefore expect a  
 5536 similar effect in SCR models when we discretize the state-space. We can also think about  
 5537 discretizing the state-space as being related to numerical integration where we find (see  
 5538 Chapt. 6) that we don't need a very fine grid of support points to evaluate the integral to  
 5539 a reasonable level of accuracy. We demonstrate this here by reanalyzing simulated data  
 5540 using a state-space defined by a different numbers of support points. We provide an **R**  
 5541 script called **SCR0bayesDss** in the **R** package **scrbook**. We note that for this comparison  
 5542 we generated the actual activity centers as a continuous random variable and thus the  
 5543 discrete state-space is, strictly speaking, an approximation to truth. That said, we regard  
 5544 all state-space specifications as approximations to truth in the sense that they represent  
 5545 a component of the SCR model.

5546 As with our **R** function **SCR0bayes**, the modification **SCR0bayesDss** will use either  
 5547 **WinBUGS** or **JAGS**. In addition, it requires a grid resolution argument (**ng**) which  
 5548 is the dimension of 1 side of a square state-space. To execute this function we do, for  
 5549 example:

```
5550 library(scrbook)
5551 data<-simSCR0(discard0=TRUE,rnd=2013)    # generate data set
5552     # run with JAGS
5553 out1<-SCR0bayesDss(data,ng=8,M=200,engine="jags",ni=2000,nb=1000)
5554     # run with WinBUGS
5555 out2<-SCR0bayesDss(data,ng=8,M=200,engine="winbugs",ni=2000,nb=1000)
```

5556 We fit this model to the same simulated data set for  $6 \times 6$ ,  $9 \times 9$ ,  $12 \times 12$ ,  $15 \times 15$   
 5557 state-space grids. For **WinBUGS**, we used 3 chains of 5000 total length with 1000 burn-in,  
 5558 which yields 12000 total posterior samples. Summary results are shown in Table 5.6.  
 5559 The results are broadly consistent except for the  $6 \times 6$  case. We see that the run time  
 5560 increases with the size of the state-space grid (not unexpected), such that we imagine it  
 5561 would be impractical to run models with more than a few hundred state-space grid points.  
 5562 We found (not shown here) that the runtime of **JAGS** is much faster and, furthermore,  
 5563 relatively *constant* as we increase the grid size. We suspect that **WinBUGS** is evaluating  
 5564 the full-conditional for each activity center at all  $G$  possible values whereas it may be  
 5565 that **JAGS** is evaluating the full-conditional only at a subset of values or perhaps using  
 5566 previous calculations more effectively. While this might suggest that one should always  
 5567 use **JAGS** for this analysis, we found in our analysis of the wolverine (next section) that  
 5568 **JAGS** could be extremely sensitive to starting values, producing MCMC algorithms that  
 5569 often simply do not work for some problems, so be careful when using **JAGS**. To improve  
 5570 its performance, always start the latent activity centers at values near where individuals  
 5571 were captured. The performance of either should improve if we compute the full distance  
 5572 matrix outside of **BUGS** and pass it as data, although we haven't fully evaluated this  
 5573 approach.

**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	Mean	SD	NaiveSE	Time-seriesSE	runtime (sec)
6	111.6699	16.61414	0.1516657	0.682008	2274
9	114.2294	17.99109	0.1642355	0.833291	4300
12	115.9806	17.3843	0.1586964	0.762756	7100
15	115.379	17.93721	0.1637436	0.832483	13010

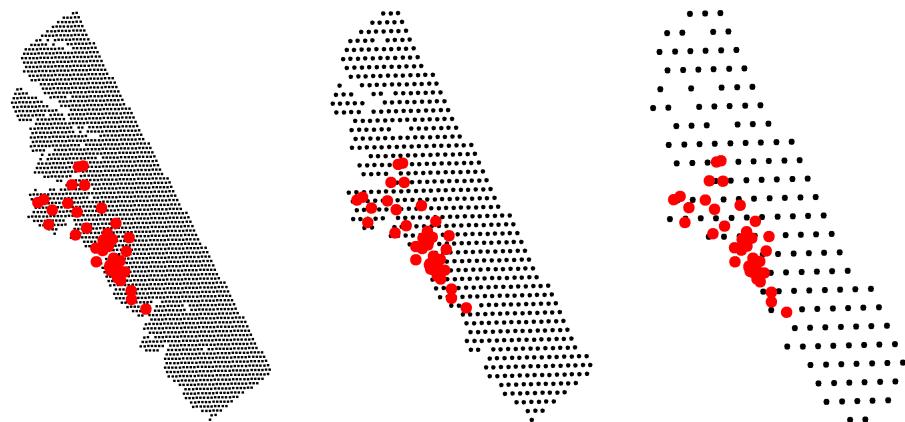
### 5574 5.9.2 Analysis of the wolverine camera trapping data

5575 We reanalyzed the wolverine data using discrete state-space grids with points spaced by  
 5576 2, 4 and 8 km (see Fig. 5.5). These were constructed from a 40 km buffered state-space,  
 5577 and deleting the points over water (see Royle et al., 2011b). Our interest in doing this  
 5578 was to evaluate the relative influence of grid resolution on estimated density because the  
 5579 coarser grids will be more efficient from a computational stand-point and so we would  
 5580 prefer to use them, but only if there is no strong influence on estimated density. The  
 5581 density estimates are quite a bit larger than obtained in our analysis (Table 5.8.2) based  
 5582 on a rectangular, continuous state-space. We also see that there are slight differences  
 5583 depending on the resolution of the state-space grid. Interestingly, the effectiveness of the  
 5584 MCMC algorithms, as measured by effective sample size (`n.eff`) is pretty remarkably  
 5585 different. Furthermore, the finest grid resolution (2 km spacing) took about 6 days to run  
 5586 and thus it would not be practical for large problems or with many models.

```

5587 > print(out.2km,digits=2)
5588 Inference for Bugs model at "modelfile.txt", fit using WinBUGS,
5589 3 chains, each with 11000 iterations (first 1000 discarded)
5590 n.sims = 30000 iterations saved
5591      mean     sd   2.5%   25%   50%   75%  97.5% Rhat n.eff
5592 psi    0.43  0.09  0.27  0.37  0.43  0.49  0.63  1.00  560
5593 sigma  0.62  0.05  0.54  0.59  0.62  0.65  0.73  1.01  160
5594 lam0   0.05  0.01  0.04  0.04  0.05  0.06  0.07  1.01  320
5595 p0    0.05  0.01  0.03  0.04  0.05  0.05  0.06  1.01  320
5596 N     86.56 16.94 57.00 75.00 85.00 97.00 124.00 1.00  510
5597 D     8.78  1.72  5.78  7.60  8.62  9.83  12.57 1.00  510
5598
5599 For each parameter, n.eff is a crude measure of effective sample size,
5600 and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
5601
5602 > print(out.4km,digits=2)
5603 Inference for Bugs model at "modelfile.txt", fit using WinBUGS,
5604 3 chains, each with 11000 iterations (first 1000 discarded)
5605 n.sims = 30000 iterations saved
5606      mean     sd   2.5%   25%   50%   75%  97.5% Rhat n.eff
5607 psi    0.45  0.09  0.28  0.38  0.44  0.50  0.64    1  1300
5608 sigma  0.61  0.04  0.53  0.58  0.61  0.64  0.71    1  1600

```



**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.

```

5609 lam0    0.05  0.01  0.04  0.05  0.05  0.06  0.07    1  2500
5610 p0      0.05  0.01  0.03  0.04  0.05  0.05  0.07    1  2500
5611 N       89.25 17.44 59.00 77.00 88.00 100.00 127.00   1  1100
5612 D       9.01   1.76  5.96  7.77  8.88  10.10  12.82   1  1100
5613
5614 For each parameter, n.eff is a crude measure of effective sample size,
5615 and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
5616
5617 > print(out.8km,digits=2)
5618 Inference for Bugs model at "modelfile.txt", fit using WinBUGS,
5619 3 chains, each with 11000 iterations (first 1000 discarded)
5620 n.sims = 30000 iterations saved
5621      mean     sd   2.5%   25%   50%   75%  97.5% Rhat n.eff
5622 psi     0.42   0.09   0.26   0.36   0.41   0.47   0.61 1.00   940
5623 sigma   0.68   0.05   0.59   0.64   0.67   0.71   0.77 1.01   220
5624 lam0    0.05   0.01   0.03   0.04   0.05   0.05   0.06 1.00   560
5625 p0      0.05   0.01   0.03   0.04   0.04   0.05   0.06 1.00   560
5626 N       83.18 16.14 56.00 72.00 82.00 93.00 119.00 1.00   700
5627 D       8.28   1.61   5.57   7.17   8.16   9.26   11.84 1.00   700
5628
5629 For each parameter, n.eff is a crude measure of effective sample size,
5630 and Rhat is the potential scale reduction factor (at convergence, Rhat=1).

```

## 5.10 SUMMARIZING DENSITY AND ACTIVITY CENTER LOCATIONS

5631 One of the most useful aspects of SCR models is that they are parameterized in terms  
 5632 of individual locations – i.e., *where* each individual lives – and, thus, we can compute  
 5633 many useful and interesting summaries of the activity centers obtained from an MCMC  
 5634 simulation, including maps of density (the number of activity centers per unit area),  
 5635 estimates of  $N$  for any well-defined polygon, or estimates of where the activity centers  
 5636 for specific individuals reside. In Bayesian analysis, obtaining such summaries entails no  
 5637 added calculations, when we do analysis by MCMC, because we need only post-process the  
 5638 output for the individual activity centers to obtain the desired summaries. We demonstrate  
 5639 that in this section. Note that you have to be sure to retain the MCMC history for the **s**  
 5640 variables and also the data augmentation variables **z** in order to do some of the following  
 5641 analyses.

### 5642 5.10.1 Realized vs. Expected Density

5643 In some problems, there will be interest in distinguishing between 3 distinct estimates of  
 5644 population size or density: (1) the size of *the* specific population under study; (2) the  
 5645 size of some specific population that we didn't study; (3) Less often we care about the  
 5646 expected population size for some population. [some of this is covered by (Efford and  
 5647 Fewster, 2012)]. Efford and Fewster called (1) the realized population size and they also  
 5648 used the term expected population size in an ambiguous manner.

5649 Imagine 2 populations of tigers in 2 reserves. You do a camera trapping study on one  
 5650 of the reserves to estimate  $N_1$  and you think the reserves are similar and homoegeneous  
 5651 so you're willing to apply a density estimate based on  $N_1$  to the 2nd reserve.

5652 that we're studying, then we probably want to make an inference about  $N$ , the realized  
 5653 population size. If, on the other hand, we wish to make a guess at the size of some  
 5654 population that wasn't sampled, then we probably want an estimate of  $\mathbb{E}(N)$  for that  
 5655 population, possibly scaled by area. Thus, the same distinction between realized and  
 5656 expected value applies to *density* as well, although it is not usually thought of as such.  
 5657 Realized density is  $N/\text{area}$  whereas the expected density is  $\mathbb{E}(N)/\text{area}$ .

5658 In the model we've developed so far, we suppose that our population size is  $N$  and  
 5659 this is meant to be the realized population size – i.e., of the population that was sampled.

5660 Think of it this way: Suppose  $N \sim \text{Poisson}(\lambda)$  and  $\lambda$  is known with certainty. If we  
 5661 have some patch of landscape with area  $A$  then our best guess at the expected number  
 5662 of guys out there is  $A\lambda$  and there is no variation in that at all. If we have estimated  $\lambda$   
 5663 somehow then ther eis some estimation error involved. On the other hand suppose we  
 5664 really care about some realized population  $N$  on a reserve that we never sampled but we  
 5665 suppose our estimate of  $\lambda$  applies to that reserve. Then we care about predicting a new  
 5666  $N$ , say  $N_{\text{new}}$  just to be clear, and there is extra variation in that prediction. In particular,  
 5667  $\text{Var}(N_{\text{new}}) = \text{Var}(N_{\text{new}}|\lambda) + \text{Var}(\lambda)$

```
5668 Var(E Nnew|lambda) + E Var Nnew|lambda =
5669 Var(\hat{lambda}) + \hat{lambda}
5670 ### need to put an A in there.
```

5671 Now suppose we care about a population that we just sampled, then  $N = n + n_0$  and  
 5672 because we observe a component of that population ( $n$ ) there is no variation about that,

5673 and we only have variation about the  $n_0$  part. The variance of this realized  $N$  should  
 5674 be less than the variation of predicting a new  $N$  because a portion of the population is  
 5675 observed and, hence, has no uncertainty with it (see Chapt. 13 for more on this). For  
 5676 studying a single population it probably only makes sense to ever consider the realized  
 5677 population of individuals.

5678 Making a density map can be either of these things but with constant  $N$  as we've  
 5679 considered so far there is no point to estimate "expected" density. See chapt13.

5680 The expected population size, using the data augmentation formulation of the model,  
 5681 is  $\mathbb{E}(N) = \psi M$  which can be converted directly to expected density by dividing by area.

5682 A question that might arise is, what is the size of some new population that we didn't  
 5683 sample? For the sake of elaboration, suppose we fitted our model to a grid of  $10 \times 10$   
 5684 traps of some sort and we want to predict population size on a grid that we might use  
 5685 for sampling in some adjacent landscape. In that case we might choose to estimate the  
 5686 size of this new population by the quantity  $\psi M$ . On the other hand, that under-states  
 5687 the variability of that prediction because there should be some binomial variation added  
 5688 in.....

### 5689 5.10.2 Constructing Density Maps

5690 Because SCR models are spatially-explicit, it is natural to want to summarize the results  
 5691 of fitting a model by producing a map of density. Using Bayesian analysis by MCMC, it is  
 5692 most easy to make a map of *realized* density. We can do this by tallying up the number of  
 5693 activity centers  $s_i$  in pixels of arbitrary size and then producing a nice multi-color spatial  
 5694 plot of the result. Specifically, let  $B(\mathbf{x})$  indicate a pixel centered at  $\mathbf{x}$  then

$$N(\mathbf{x}) = \sum_{i=1}^M I(s_i \in B(\mathbf{x}))$$

5695 (here,  $I(arg)$  is the indicator function which evaluates to 1 if  $arg$  is true, and 0 otherwise)  
 5696 is the population size of pixel  $B(\mathbf{x})$ , and  $D(\mathbf{x}) = N(\mathbf{x})/\|B(\mathbf{x})\|$  is the local density. Note  
 5697 that these  $N(\mathbf{x})$  parameter are just "derived parameters" as we normally obtain from  
 5698 posterior output using the appropriate Monte Carlo average (see Chapt. 3).

5699 One thing to be careful about, in the context of models in which  $N$  is unknown, is that,  
 5700 for each MCMC iteration  $m$ , we only tabulate those activity centers which correspond to  
 5701 individuals in the sampled population, i.e., for which the data augmentation variable  
 5702  $z_i = 1$ . In this case, we take all of the output for MCMC iterations  $m = 1, 2, \dots, n_{iter}$   
 5703 and compute this summary:

$$N(\mathbf{x}, m) = \sum_{i: z_{i,m}=1} I(s_{i,m} \in B(\mathbf{x}))$$

5704 Thus,  $N(\mathbf{x}, 1), N(\mathbf{x}, 2), \dots$ , is the Markov chain for parameter  $N(\mathbf{x})$ . In what follows we  
 5705 will provide a set of **R** commands for doing this calculation and making a basic image  
 5706 plot from the MCMC output.

5707 **Step 1:** Define the center points of each pixel  $B(\mathbf{x})$ , or point at which local density will  
 5708 be estimated:

```
5709 xg<-seq(xlim[1],xlim[2],,50)
5710 yg<-seq(ylim[1],ylim[2],,50)
```

5711 **Step 2:** Extract the MCMC histories for the activity centers and the data augmentation  
 5712 variables. Note that these are each  $N \times \text{niter}$  matrices. Here we do this assuming that  
 5713 **WinBUGS** was run producing the **R** object named **out**:

```
5714 Sxout<-out$sims.list$s[,1]
5715 Syout<-out$sims.list$s[,2]
5716 z<-out$sims.list$z
```

5717 **Step 3:** We associate each coordinate with the proper pixel using the **R** command **cut()**.  
 5718 Note that we keep only the activity centers for which  $z = 1$  (i.e., individuals that belong  
 5719 to the population of size  $N$ ):

```
5720 Sxout<-cut(Sxout[z==1],breaks=xg,include.lowest=TRUE)
5721 Syout<-cut(Syout[z==1],breaks=yg,include.lowest=TRUE)
```

5722 **Step 4:** Use the **table()** command to tally up how many activity centers are in each  
 5723  $B(\mathbf{x})$ :

```
5724 Dn<-table(Sxout,Syout)
```

5725 **Step 5:** Use the **image()** command to display the resulting matrix.

```
5726 image(xg,yg,Dn/nrow(z),col=terrain.colors(10))
```

5727 It is worth emphasizing here that density maps will not usually appear uniform despite  
 5728 that we have assumed that activity centers are uniformly distributed. This is because  
 5729 the observed encounters of individuals provide direct information about the location of  
 5730 the  $i = 1, 2, \dots, n$  activity centers and thus their “estimated” locations will be affected  
 5731 by the observations. In a limiting sense, were we to sample space intensely enough,  
 5732 every individual would be captured a number of times and we would have considerable  
 5733 information about all  $N$  point locations. Consequently, the uniform prior would have  
 5734 almost no influence at all on the estimated density surface in this limiting situation.  
 5735 Thus, in practice, the influence of the uniformity assumption decreases as the fraction of  
 5736 the population encountered, and the total number of encounters per individual, increases.

5737 **On the non-intuitiveness of `image()`** – the **R** function **image()**, invoked for a  
 5738 matrix  $M$  by **image(M)**, might not be very intuitive to some – it plots  $M[1, 1]$  in the lower  
 5739 left corner. If you want  $M[]$  to be plotted “as you look at it” then  $M[1, 1]$  should be in the  
 5740 upper left corner. We have a function **rot()** which does that. If you do **image(rot(M))**  
 5741 then it puts it on the monitor as if it was a map you were looking at. You can always  
 5742 specify the  $x$  and  $y$ - labels explicitly as we did above.

5743 **Spatial dot plots** – A cruder version of the density map can be made using our  
 5744 “spatial dot map” function **spatial.plot** (in **scrbook**). This function requires, as input,  
 5745 point locations and the value to be displayed. A simplified version of this function is as  
 5746 follows:

---

```

5747 > spatial.plot<- function(x,y){
5748   nc<-as.numeric(cut(y,20))
5749   plot(x,pch=" ")
5750   points(x,pch=20,col=topo.colors(20)[nc],cex=2)
5751   image.scale(y,col=topo.colors(20))
5752 }
5753 #
5754 # To execute the function do this:
5755 #
5756 > spatial.plot(cbind(xg,yg), Dn/nrow(z))

```

### 5.10.3 Example: Wolverine density map

We return to the wolverine study which took place in 2008 in SE Alaska (Fig. 5.4) and we produce a density map of wolverines from that analysis. We include the function `SCRdensity` which requires a specific data structure as shown below. In particular, we have to package up the MCMC history for the activity centers and the data augmentation variables  $z$  into a list. This also requires that we add those variables to the parameters-to-be-monitored list when we pass things to **BUGS**.

We used the posterior output from the wolverine model fitted previously to compute a relatively coarse version of a density map, using 100 pixels in a  $10 \times 10$  grid (Fig. 5.6 top panel) and using 900 pixels arranged in a  $30 \times 30$  grid (Fig. 5.6 lower panel) for a fine-scale map. The **R** commands for producing such a plot (for short MCMC run) are as follows:

```

5769 library(scrbook)
5770 data(wolverine)
5771 traps<-wolverine$wtraps
5772 y3d <-SCR23darray(wolverine$wcaps,wolverine$wtraps)
5773
5774 # this takes 341 seconds on a standard CPU circa 2011
5775 out<-wolvSCR0(y3d,traps,nb=1000,ni=2000,buffer=1,M=100,keepz=TRUE)
5776
5777 Sx<-out$sims.list$s[,1]
5778 Sy<-out$sims.list$s[,2]
5779 z<- out$sims.list$z
5780 obj<-list(Sx=Sx,Sy=Sy,z=z)
5781 tmp<-SCRdensity(obj,nx=10,ny=10,scalein=100,scaleout=100)

```

In these figures density is expressed in units of individuals per  $100 \text{ km}^2$ , while the area of the pixels is about  $103.7 \text{ km}^2$  and  $11.5 \text{ km}^2$ , respectively. That calculation is based on:

```

5782 > total.area<- (ylim[2]-ylim[1])*(xlim[2]-xlim[1])*100
5783 > total.area/(10*10)
5784 [1] 103.7427
5785 > total.area/(30*30)
5786 [1] 11.52697

```

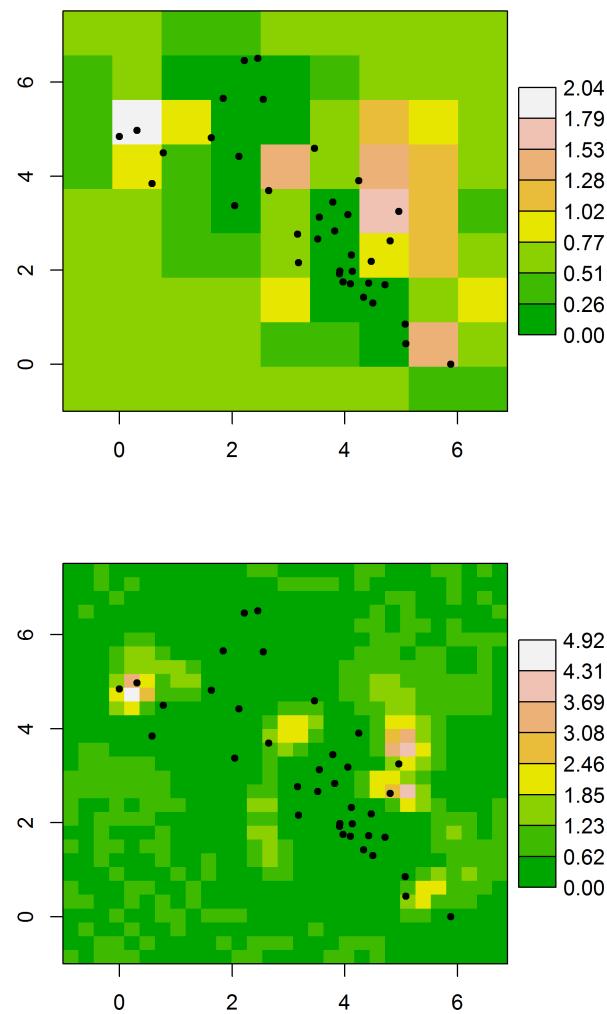
5789 A couple of things are worth noting: First is that as we move away from “where the  
 5790 data live” – away from the trap array – we see that the density approaches the mean  
 5791 density. This is a property of the estimator as long as the detection function decreases  
 5792 sufficiently rapidly as a function of distance. Relatedly, it is also a property of statistical  
 5793 smoothers such as splines, kernel smoothers, and regression smoothers – predictions tend  
 5794 toward the global mean as the influence of data diminishes. Another way to think of it is  
 5795 that it is a consequence of the prior – which imposes uniformity, and as you get far away  
 5796 from the data, the predictions tend to the expected constant density under the prior.  
 5797 The other thing to note about this map is that density is not 0 over water (although the  
 5798 coastline is not shown). This might be perplexing to some who are fairly certain that  
 5799 wolverines do not like water. However, there is nothing about the model that recognizes  
 5800 water from non-water and so the model predicts over water *as if* it were habitat similar to  
 5801 that within which the array is nested. But, all of this is OK as far as estimating density  
 5802 goes and, furthermore, we can compute valid estimates of  $N$  over any well-defined region  
 5803 which presumably wouldn’t include water if we so wished. Alternatively, areas covered by  
 5804 water could be masked out, which we discuss in the next section.

#### 5805 5.10.4 Predicting where an individual lives

5806 The density maps in the previous section show the expected number of individuals per  
 5807 unit area. A closely related problem is that of producing a map of the probable location  
 5808 of a specific individual’s activity center. For any observed encounter history, we can easily  
 5809 generate a posterior distribution of  $\mathbf{s}_i$  for individual  $i$ . In addition, for an individual that  
 5810 is *not* captured, we can use the MCMC output to produce a corresponding plot of where  
 5811 such an individual might live, say  $\mathbf{s}_{n+1}$ . Obviously, all such uncaptured individuals (for  
 5812  $i = n + 1, \dots, N$ ) should have the same posterior distribution. To illustrate, we show the  
 5813 posterior distribution of  $\mathbf{s}_1$ , the activity center for the individual labeled 1 in the data set,  
 5814 in Fig. 5.7. This individual was captured a single time at trap 30 which is circled in Fig.  
 5815 5.7. We see that the posterior distribution is affected by traps of capture *and* traps of  
 5816 non-capture in fairly intuitive ways. In particular, because there are other traps in close  
 5817 proximity to trap 30 which individual 1 was *not* captured, the model pushes its activity  
 5818 center away from the trap array. The help file for **SCRdensity** shows how to calculate Fig.  
 5819 5.7.

## 5.11 EFFECTIVE SAMPLE AREA

5820 One of the key issues in using ordinary capture recapture models which we’ve brought up  
 5821 over and over again is this issue that the area which is sampled by a trapping array is  
 5822 unknown – in other words, the  $N$  that is estimated by capture-recapture models does not  
 5823 have an explicit region of space associated with it. Classically this has been addressed in  
 5824 the ad hoc way of prescribing an area that contains the trap array usually by adding a  
 5825 buffer of some width, which is not estimated as part of the capture-recapture model. In  
 5826 SCR models we avoid the problem of not having an explicit linkage between  $N$  and “area”,  
 5827 by prescribing explicitly the area within which the underlying point process is defined –  
 5828 the state-space of the point process. This state-space is *not* the “effective sampled area”  
 5829 (ESA) – it is desirable that it be somewhat larger than the ESA, whatever that may be,



**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.

5830 in the sense that individuals at the edge of the state-space have no probability of being  
 5831 captured, but as part of the SCR model we don't need to try to estimate or otherwise  
 5832 characterize the ESA explicitly.

5833 However, it is possible to provide a characterization of effective sample area under  
 5834 any SCR model. This is directly analogous to the calculation of "effective strip width" in  
 5835 distance sampling (Buckland et al., 2001; Borchers et al., 2002). The conceptual definition  
 5836 of ESA follows from equating density to "apparent density" – ESA is the magic number  
 5837 that satisfies that equivalence:

$$D = N/A = n/\text{ESA}$$

5838 In other words, the ratio of  $N$  to the area of the state-space should be equal to the ratio of  
 5839 the observed sample size  $n$  to this magic number ESA. Both of these should equal density.  
 5840 So to compute ESA for a model we plug-in  $\mathbb{E}(n)$  into the equation to get:

$$\text{ESA} = \mathbb{E}(n)/D.$$

5841 Our following development assumes that  $D$  is constant, but these calculations can be  
 5842 generalized to allow for  $D$  to vary spatially. Imagine our habitat mask for the wolverine  
 5843 data, or the bins we just used to produce a density map, then we can write  $\mathbb{E}(n)$  according  
 5844 to

$$\mathbb{E}(n) = \sum_s \Pr(\text{encounter}|\mathbf{s}) \mathbb{E}(N(\mathbf{s}))$$

5845 where if we prefer to think of this more conceptually we could replace the summation with  
 5846 an integration (which, in practice, we would just replace with a summation, and so we  
 5847 just begin there). In this expression note that  $\mathbb{E}(N(\mathbf{s}))$  is the expected population size at  
 5848 pixel  $\mathbf{s}$  which is the density times the area of the pixel, i.e.,  $\mathbb{E}(N(\mathbf{s})) = D * a$ . Therefore

$$\mathbb{E}(n) = D * a * \sum_s \Pr(\text{encounter}|\mathbf{s})$$

5849 and (plugging this into the expression above for ESA)

$$\text{ESA} = \frac{D * a * \sum_s \Pr(\text{encounter}|\mathbf{s})}{D}$$

5850 We see that  $D$  cancels and we have  $\text{ESA} = a * \sum_s \Pr(\text{encounter}|\mathbf{s})$  So what you have to  
 5851 do here is substitute in  $\Pr(\text{encounter}|\mathbf{s})$  and just sum them up over all pixels. For the  
 5852 Bernoulli model of hair-snares

$$\Pr(\text{encounter}|\mathbf{s}) = 1 - (1 - p(\mathbf{s}))^K$$

5853 with slight modifications when encounter probability depends on covariates. Thus,

$$\text{ESA} = a \sum_s 1 - (1 - p(\mathbf{s}))^K \tag{5.11.1}$$

5854 Clearly the calculation of ESA is affected by the use of a habitat mask, because the  
 5855 summation in Eq. 5.11.1 only occurs over pixels that define the state-space.

5856 For the wolverine camera trapping data, we used the  $2 \times 2$  km habitat mask and the  
 5857 posterior means of  $p_0$  and  $\sigma$  (see Sec. 5.9.2) to compute the probability of encounter for

5858 each  $\mathbf{s}$  of the mask points. The result is shown graphically in Fig. 5.8. The ESA is the  
 5859 sum of the values plotted in that figure multiplied by 4, the area of each pixel. For the  
 5860 wolverine study, the result is 2507.152 km<sup>2</sup>. We note that the probability of encounter  
 5861 declines rapidly to 0 as we move away from the periphery of the camera traps, indicating  
 5862 the state-space constructed from a 40 km buffered trap array was indeed sufficient for the  
 5863 analysis of these data. An **R** script for producing this figure is in the **wolvESA** function of  
 5864 the **scrbook** package.

## 5.12 SUMMARY AND OUTLOOK

5865 We have emphasized in this chapter that our basic “model SCR0” which assumes that  
 5866 encounters of individuals in each trap are independent Bernoulli trials, is an ordinary  
 5867 capture-recapture model like model  $M_0$ , but augmented with a set of latent individual  
 5868 effects,  $\mathbf{s}_i$ , which relate encounter probability to some sense of individual location. SCR  
 5869 models are therefore a type of individual covariate model (as introduced in Chapt. 4) –  
 5870 but with imperfect information about the individual covariate. Such models are not too  
 5871 dissimilar from standard GLMMs used throughout statistics and, therefore, we find that  
 5872 they are easy to analyze using standard MCMC methods encased in black boxes such as  
 5873 **WinBUGS** or **JAGS**. We will also see that they are easy to analyze using likelihood  
 5874 methods, which we address in Chapt. 6.

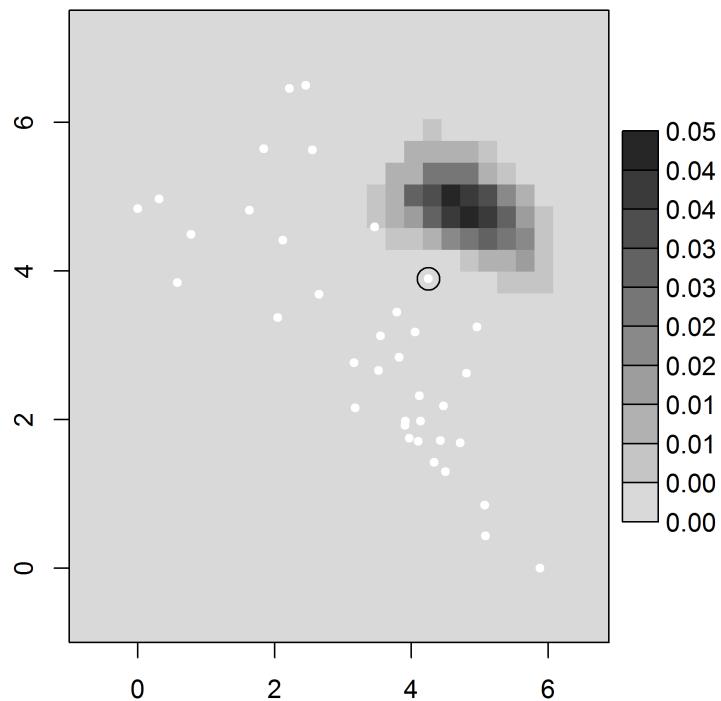
5875 Formal consideration of the collection of individual locations ( $\mathbf{s}_1, \dots, \mathbf{s}_N$ ) is funda-  
 5876 mental to all of the models considered in this book. In statistical terminology, we think  
 5877 of the collection of points  $\{\mathbf{s}_i\}$  as a realization of a point process. Because SCR models  
 5878 formally link individual encounter history data to an underlying point process, we can ob-  
 5879 tain formal inferences about the point process. For example, we showed how to produce  
 5880 a density map (Fig. 5.6). We can do traditional point process analyses such as testing for  
 5881 “complete spatial randomness” (CSR) (see Chapt. 7), calculation of Ripley’s K-functions,  
 5882 and related point process summaries (Illian et al., 2008).

5883 Part of the promise, and ongoing challenge, of SCR models is to develop models that  
 5884 reflect interesting biological processes, for example interactions among points or temporal  
 5885 dynamics in point locations. In this chapter we considered the simplest possible point  
 5886 process model in which points are independent and uniformly (“randomly”) distributed  
 5887 over space. Despite the simplicity of this model, it should suffice in many applications of  
 5888 SCR models, although we do address generalizations in later chapters. Moreover, even  
 5889 though the *prior* distribution on the point locations is uniform, the realized pattern may  
 5890 deviate markedly from uniformity as the observed encounter data provide information to  
 5891 impart deviations from uniformity. Thus, estimated density maps will typically appear  
 5892 distinctly non-uniform (as we saw in the wolverine example). In applications of the basic  
 5893 SCR model, we find that this simple *a priori* model can effectively reflect or adapt to  
 5894 complex realizations of the underlying point process. For example, if individuals are  
 5895 highly territorial then the data should indicate this in the form of individuals not being  
 5896 encountered in the same trap – the resulting posterior distribution of point locations should  
 5897 therefore reflect non-independence. Obviously the complexity of posterior estimates of the  
 5898 point pattern will depend on the quantity of data, both number of individuals and captures  
 5899 per individual. Because the point process is such an integral component of SCR models,  
 5900 the state-space of the point process plays an important role in developing SCR models.

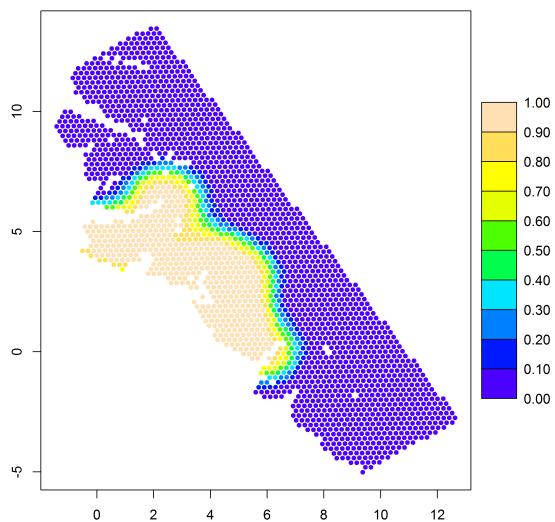
5901 As we emphasized in this chapter, the state-space is part of the model. It can have an  
5902 influence on parameter estimates and other inferences such as model selection (see chapter  
5903 7).

5904 One concept we introduced in this chapter, which has not previously been discussed  
5905 much in the literature on SCR models, is the manner in which the encounter probability  
5906 model relates to a model of space usage by individuals. The standard SCR models of  
5907 encounter probability can all be motivated as simplistic models of space usage and move-  
5908 ment, in which individuals make random use decisions from a probability distribution  
5909 proportional to the encounter probability model. This both clarifies the simplicity of the  
5910 underlying model of space usage and also suggests a direct extension to produce more  
5911 realistic models, which we discuss in Chapt. 11.

5912 In later chapters, we consider some important extensions of this basic SCR model.  
5913 For example, we consider models that include covariates that vary by individual, trap, or  
5914 over time (Chapt. 8), spatial covariates on density (Chapt. 13), open populations (Chapt.  
5915 15), and methods for model assessment and selection (Chapt. 7) among other topics. We  
5916 also consider technical details of Bayesian (Chapt. 14) and maximum likelihood (Chapt.  
5917 6) estimation so that the interested reader can develop or extend their own methods to  
5918 suit their 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.



**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.

# 6

---

5919

5920

5921

5922

## LIKELIHOOD ANALYSIS OF SPATIAL CAPTURE-RECAPTURE MODELS

5923 We have so far mainly focused on Bayesian analysis of spatial capture-recapture models.  
5924 And, in the previous chapters we learned how to fit some basic spatial capture-recapture  
5925 models using a Bayesian formulation of the models analyzed in BUGS engines including  
5926 **WinBUGS** and **JAGS**. Despite our focus on Bayesian analysis, it is instructive to develop  
5927 the basic concepts and ideas behind classical analysis based on likelihood methods and  
5928 frequentist inference for SCR models. This has been the approach taken by Borchers and  
5929 Efford (2008); Dawson and Efford (2009) and related papers. Therefore, in this chapter, we  
5930 provide some conceptual and technical foundation for likelihood-based analysis of spatial  
5931 capture-recapture models. We recognized earlier (Chapt. 5) that SCR models are versions  
5932 of binomial (or other) GLMs, but with random effects, i.e., GLMMs. These models are  
5933 routinely analyzed by likelihood methods. In particular, likelihood analysis is based on  
5934 the integrated likelihood in which the random effects are removed by integration from  
5935 the likelihood. In SCR models, the 2-dimensional coordinate,  $s$ , is a bivariate random  
5936 effect. Beyond that, there is little difference between likelihood analysis of SCR models  
5937 and ordinary GL(M)Ms.

5938 We will show here that it is straightforward to compute the maximum likelihood  
5939 estimates (MLE) for SCR models by integrated or marginal likelihood. We develop the  
5940 MLE framework using **R**, and we also provide a basic introduction to an **R** package **secr**  
5941 (Efford, 2011) which mostly does likelihood analysis of SCR models (see also the the  
5942 stand-alone package **DENSITY** (Efford et al., 2004)). To set the context for likelihood  
5943 analysis of SCR models, we first analyze the SCR model here when  $N$  is known because, in  
5944 that case, analysis is no different at all than a standard GLMM. We generalize the model  
5945 to allow for unknown  $N$  using both conventional ideas based on the “joint likelihood”  
5946 (e.g., Borchers et al., 2002) and also using a formulation based on data augmentation. We  
5947 obtain the MLEs for the SCR model from the wolverine camera trapping study (Magoun  
5948 et al., 2011) analyzed in 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 regression 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- $\mathbf{s}$  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- $\mathbf{s}$  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

$$[y|\boldsymbol{\alpha}] = \int_{\mathbf{s}} [y|\mathbf{s}, \boldsymbol{\alpha}][\mathbf{s}]d\mathbf{s}.$$

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- $\mathbf{s}$  likelihood. The observation model for each encounter observation  $y_{ij}$ , 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  $p_{ij}$  on  $\mathbf{s}$  and parameters  $\boldsymbol{\alpha}$  explicitly. For example,  $p_{ij}$  might be the Gaussian model given by

$$p_{ij} = \text{logit}^{-1}(\alpha_0) \exp(-\alpha_1 \|\mathbf{x}_j - \mathbf{s}_i\|)$$

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 (hence also *integrated likelihood*), from the conditional-on- $\mathbf{s}$  likelihood and regarding the *marginal* distribution of the data as the likelihood. That is, we compute:

$$[\mathbf{y}|\boldsymbol{\alpha}] = \int_{\mathcal{S}} [\mathbf{y}_i|\mathbf{s}_i, \boldsymbol{\alpha}][\mathbf{s}_i]d\mathbf{s}_i$$

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. 13 for alternative specifications of  $[\mathbf{s}]$ ).

5978     The joint likelihood for all  $N$  individuals, assuming independence of encounters among  
 5979     individuals, is the product of  $N$  such terms:

$$\mathcal{L}(\boldsymbol{\alpha}|\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N) = \prod_{i=1}^N [\mathbf{y}_i|\boldsymbol{\alpha}]$$

5980     We emphasize that two independence assumptions are explicit in this development: inde-  
 5981     pendence of trap-specific encounters within individuals and also independence among  
 5982     individuals. In particular, this would only be valid when individuals are not physically  
 5983     restrained or removed upon capture, and when traps do not “fill up” (i.e., this is model  
 5984     SCR0, from Chapt. 5).

5985     The key operation for computing the likelihood is solving a 2-dimensional integration  
 5986     problem. There are some general purpose **R** packages that implement a number of multi-  
 5987     dimensional integration routines including **adapt** (Genz et al., 2007) and **R2cuba** (Hahn  
 5988     et al., 2010). In practice, we won’t rely on these extraneous **R** packages (except see Chapt.  
 5989     13 for an application of **R2cuba**) but instead will use perhaps less efficient methods in which  
 5990     we replace the integral with a summation over an equal area mesh of points on the state-  
 5991     space  $\mathcal{S}$  and explicitly evaluate the integrand at each point. We invoke the rectangular  
 5992     rule for integration here<sup>1</sup> in which we evaluate the integrand on a regular grid of points  
 5993     of equal area and compute the average of the integrand over that grid of points. Let  
 5994      $u = 1, 2, \dots, nG$  index a grid of  $nG$  points,  $\mathbf{s}_u$ , where the area of grid cells is constant, say  
 5995      $A$ . In this case, the integrand, i.e., the marginal pmf of  $\mathbf{y}_i$ , is approximated by

$$[\mathbf{y}_i|\boldsymbol{\alpha}] = \frac{1}{nG} \sum_{u=1}^{nG} [\mathbf{y}_i|\mathbf{s}_u, \boldsymbol{\alpha}] \quad (6.1.2)$$

5996     This is a specific case of the general expression that could be used for approximating  
 5997     the integral for any arbitrary distribution  $[\mathbf{s}]$ . The general case is

$$[\mathbf{y}|\boldsymbol{\alpha}] = \frac{A(\mathcal{S})}{nG} \sum_{u=1}^{nG} [y|\mathbf{s}_u, \boldsymbol{\alpha}] [\mathbf{s}_u]$$

5998     Under the uniformity assumption,  $[\mathbf{s}] = 1/A(\mathcal{S})$  and thus the grid-cell area cancels in the  
 5999     above expression to yield eq. 6.1.2. The rectangular rule for integration can be seen as an  
 6000     application of the Law of Total Probability for a discrete random variable  $\mathbf{s}$ , having  $nG$   
 6001     unique values with equal probabilities  $1/nG$ .

### 6002     6.1.1 Implementation (simulated data)

6003     Here we will illustrate how to carryout this integration and optimization based on the  
 6004     integrated likelihood using simulated data (i.e., see Sec. 5.5). Using **simSCR0** we simulate  
 6005     data for 100 individuals and a 25 trap array laid out in a  $5 \times 5$  grid of traps having unit  
 6006     spacing. The specific encounter model is the Gaussian model. The 100 activity centers  
 6007     were simulated on a state-space defined by a  $8 \times 8$  square within which the trap array was  
 6008     centered (thus the trap array is buffered by 2 units). Therefore, the density of individuals  
 6009     in this system is fixed at  $100/64$ . In the following set of **R** commands we generate the  
 6010     data and then harvest the required data objects:

---

<sup>1</sup>e.g., [http://en.wikipedia.org/wiki/Rectangle\\_method](http://en.wikipedia.org/wiki/Rectangle_method)

```

6011 ## simulate a complete data set (perfect detection)
6012 data<-simSCR0(discard0=FALSE, rnd=2013)
6013 ## extract the objects that we need for analysis
6014 y<-data$Y
6015 traplocs<-data$traplocs
6016 nind<-nrow(y) ## in this case nind=N
6017 J<-nrow(traplocs)
6018 K<-data$K
6019 xlim<-data$xlim
6020 ylim<-data$ylim

```

6021 Now we need to define the integration grid, say **G**, which we do with the following set of  
 6022 **R** commands (here, **delta** is the grid spacing):

```

6023 delta<- .2
6024 xg<-seq(xlim[1]+delta/2,xlim[2]-delta/2,by=delta)
6025 yg<-seq(ylim[1]+delta/2,ylim[2]-delta/2,by=delta)
6026 npix<-length(xg) # valid for square state-space only
6027 G<-cbind(rep(xg,npix),sort(rep(yg,npix)))
6028 nG<-nrow(G)

```

6029 In this case, the integration grid is set up as a grid with spacing  $\delta = 0.2$  which produces,  
 6030 for our example, a  $40 \times 40$  grid of points for evaluating the integrand if the state-space  
 6031 buffer is set at 2. We note that the integration grid is set-up here to correspond exactly  
 6032 to the state-space used in simulating the data. However, in practice, we wouldn't know  
 6033 this, and our estimate of  $N$  (for the unknown case, see below) would be sensitive to choice  
 6034 of the extent of the integration grid. As we've discussed previously, density, which is  $N$   
 6035 standardized by the area of the state-space, will not be so sensitive in most cases.

6036 We are now ready to compute the conditional-on-s likelihood and carry-out the marginalization  
 6037 described by Eq. 6.1.2. We need to do this by defining an **R** function that computes  
 6038 the likelihood for the integration grid, as a function of the data objects **y** and **traplocs**  
 6039 which were created above. However, it is a bit untidy to store the grid information in  
 6040 your workspace, and define the likelihood function in a way that depends on these things  
 6041 that exist in your workspace. Therefore, we build the **R** function so that it computes the  
 6042 integration grid *within* the function, thereby avoiding potential problems if our trapping  
 6043 grid locations change, or if we want to modify the state-space buffer easily. We therefore  
 6044 define the function, called **intlik1**, to which we pass the data objects and other information  
 6045 necessary to compute the marginal likelihood. This function is available in the  
 6046 **scrbook** package (use **?intlik1** at the **R** prompt). The code is reproduced here:

```

6047 intlik1<-function(parm,y=y,delta=.2,X=traplocs,ssbuffer=2){
6048
6049 Xl<-min(X[,1]) - ssbuffer ## these lines of code are setting up the
6050 Xu<-max(X[,1]) + ssbuffer ## support for the integration which is
6051 Yu<-max(X[,2]) + ssbuffer ## the same as the state-space of "s"
6052 Yl<-min(X[,2]) - ssbuffer
6053 xg<-seq(Xl+delta/2,Xu-delta/2,,length=npix)

```

```

6054 yg<-seq(Yl+delta/2,Yu-delta/2,,length=npix)
6055 npix<-length(xg)
6056
6057 G<-cbind(rep(xg,npix),sort(rep(yg,npix)))
6058 nG<-nrow(G)
6059 D<- e2dist(X,G)
6060
6061 alpha0<-parm[1]
6062 alpha1<-exp(parm[2]) # alpha1 restricted to be positive here
6063 # for convenience (it is negated below)
6064 probcap<- plogis(alpha0)*exp(-alpha1*D*D)
6065 Pm<-matrix(NA,nrow=nrow(probcap),ncol=ncol(probcap))
6066 # all zero encounter histories
6067 n0<-sum(apply(y,1,sum)==0)
6068 # encounter histories with at least 1 detection
6069 ymat<-y[apply(y,1,sum)>0,]
6070 ymat<-rbind(ymat,rep(0,ncol(ymat)))
6071 lik.marg<-rep(NA,nrow(ymat))
6072 for(i in 1:nrow(ymat)){
6073   ## next line: log conditional likelihood for ALL possible values of s
6074   Pm[1:length(Pm)]<- (dbinom(rep(ymat[i,],nG),K,probcap[1:length(Pm)],log=TRUE))
6075   ## next line: sum the log conditional likelihoods, exp() result
6076   ## same as taking the product
6077   lik.cond<- exp(colSums(Pm))
6078   ## take the average value == computing marginal
6079   lik.marg[i]<- sum(lik.cond*(1/nG))
6080 }
6081 ## n0 = number of all-0 encounter histories
6082 nv<-c(rep(1,length(lik.marg)-1),n0)
6083 -1*(sum(nv*log(lik.marg)) )
6084 }

```

6085 The function `intlik1` accepts as input the encounter history matrix,  $y$ , the trap locations,  $X$ , and the state-space buffer. This allows us to vary the state-space buffer and easily evaluate the sensitivity of the MLE to the size of the state-space. Note that we have a peculiar handling of the encounter history matrix  $y$ . In particular, we remove the all-zero encounter histories from the matrix and tack-on a single all-zero encounter history as the last row which then gets weighted by the number of such encounter histories ( $n0$ ). This is a bit long-winded and strictly unnecessary when  $N$  is known, but we did it this way because the extension to the unknown- $N$  case is now transparent (as we demonstrate in the following section). The matrix  $Pm$  holds the log-likelihood contributions of each encounter frequency for each possible state-space location of the individual. The log contributions are summed up and the result exponentiated on the next line, producing `lik.cond`, the conditional-on-s likelihood (Eq. 6.1.1 above). The marginal likelihood (`lik.marg`) sums up the conditional elements weighted by the probabilities [s] (Eq. 6.1.2 above). This is a fairly primitive function which doesn't allow much flexibility in the data structure. For example, it assumes that  $K$ , the number of replicates, is constant for each trap. Further,

6100 it assumes that the state-space is a square. We generalize this to some extent later in this  
 6101 chapter.

6102 Here is the **R** command for maximizing the likelihood using **nlm** (the function **optim**  
 6103 could also be used) and saving the results into an object called **frog**. The output is a list  
 6104 of the following structure and these specific estimates are produced using the simulated  
 6105 data set:

```
6106 # should take 15-30 seconds
6107
6108 starts<-c(-2,2)
6109 frog<-nlm(intlik1,starts,y=y,delta=.1,X=traplocs,ssbuffer=2,hessian=TRUE)
6110 frog
6111
6112 $minimum
6113 [1] 297.1896
6114
6115 $estimate
6116 [1] -2.504824 2.373343
6117
6118 $gradient
6119 [1] -2.069654e-05 1.968754e-05
6120
6121 $hessian
6122 [,1]      [,2]
6123 [1,] 48.67898 -19.25750
6124 [2,] -19.25750 13.34114
6125
6126 $code
6127 [1] 1
6128
6129 $iterations
6130 [1] 11
```

6131 Details about this output can be found on the help page for **nlm**. We note briefly  
 6132 that **frog\$minimum** is the negative log-likelihood value at the MLEs, which are stored in  
 6133 the **frog\$estimate** component of the list. The Hessian is the observed Fisher information  
 6134 matrix, which can be inverted to obtain the variance-covariance matrix using the  
 6135 command:

```
6136 > solve(frog$hessian)
```

6137 It is worth drawing attention to the fact that the estimates are slightly different than  
 6138 the Bayesian estimates reported previously in Sec. 5.6. There are several reasons for this.  
 6139 First Bayesian inference is based on the posterior distribution and it is not generally the  
 6140 case that the MLE should correspond to any particular value of the posterior distribution.  
 6141 If the prior distributions in a Bayesian analysis are uniform, then the (multivariate) mode  
 6142 of the posterior is the MLE, but note that Bayesians almost always report posterior  
 6143 *means* and so there will typically be a discrepancy there. Secondly, we have implemented

6144 an approximation to the integral here and there might be a slight bit of error induced  
 6145 by that. We will evaluate that shortly. Third, the Bayesian analysis by MCMC is itself  
 6146 subject to some amount of Monte Carlo error which the analyst should always be aware of  
 6147 in practical situations. All of these different explanations are likely responsible for some  
 6148 of the discrepancy. Accounting for these, we see general consistency between the two  
 6149 estimates.

6150 In summary, for the basic SCR model, integrated likelihood is a really easy calculation  
 6151 when  $N$  is known. Even for  $N$  unknown it is not too difficult, and we will do that shortly.  
 6152 However, if you can solve the known- $N$  problem then you should be able to do a real  
 6153 analysis, for example by considering different values of  $N$  and computing the results for  
 6154 each value and then making a plot of the log-likelihood or AIC and choosing the value  
 6155 of  $N$  that produces the best log-likelihood or AIC. As a homework problem we suggest  
 6156 that the reader take the code given above and try to estimate  $N$  without modifying the  
 6157 code by just repeatedly applying it for different values of  $N$  in attempt to deduce the best  
 6158 value. We will formalize the unknown- $N$  problem next.

## 6.2 MLE WHEN N IS UNKNOWN

6159 Here we build on the previous introduction to integrated likelihood but we consider now  
 6160 the case in which  $N$  is unknown. We will see that adapting the analysis based on the  
 6161 known- $N$  model is straightforward for the more general problem. The main distinction is  
 6162 that we don't observe the all-zero encounter history so we have to make sure we compute  
 6163 the probability for that encounter history which we do by tacking a row of zeros onto the  
 6164 encounter history matrix. In addition, we include the number of such all-zero encounter  
 6165 histories (that is, the number of individuals *not* encountered) as an unknown parameter  
 6166 of the model. Call that unknown quantity  $n_0$ . Then,  $N = n_0 + n$ . We will usually  
 6167 parameterize the likelihood in terms of  $n_0$  because optimization over a parameter space  
 6168 in which  $\log(n_0)$  is unconstrained is preferred to a parameter space in which  $N$  must be  
 6169 constrained  $N \geq n$ . With  $n_0$  unknown, we have to be sure to include a combinatorial term  
 6170 to account for the fact that of the  $n$  observed individuals there are  $\binom{N}{n}$  ways to realize  
 6171 a sample of size  $n$ . The combinatorial term involves the unknown  $n_0$  and thus it must  
 6172 be included in the likelihood. In evaluating the *log*-likelihood, we have to compute terms  
 6173 such as the log-factorial  $\log(N!) = \log((n_0 + n)!)$ . We do this in **R** by making use of the  
 6174 log-gamma function (`lgamma`) and the identity

$$\log(N!) = \text{lgamma}(N + 1).$$

6175 Therefore, to compute the likelihood, we require the following 3 components: (1) The  
 6176 marginal probability of each  $\mathbf{y}_i$  as before,

$$[\mathbf{y}_i | \boldsymbol{\alpha}] = \int_{\mathcal{S}} [\mathbf{y}_i | \mathbf{s}_i, \boldsymbol{\alpha}] [\mathbf{s}_i] d\mathbf{s}_i.$$

6177 (2) We compute the probability of an all-0 encounter history:

$$\pi_0 = [\mathbf{y} = \mathbf{0} | \boldsymbol{\alpha}] = \int_{\mathcal{S}} \text{Binomial}(\mathbf{0} | \mathbf{s}_i, \boldsymbol{\alpha}) [\mathbf{s}_i] d\mathbf{s}_i$$

6178 (3) The combinatorial term:  $\binom{N}{n}$ . Then, the marginal likelihood has this form:

$$\mathcal{L}(\boldsymbol{\alpha}, n_0 | \mathbf{y}) = \frac{N!}{n! n_0!} \left\{ \prod_{i=1}^n [\mathbf{y}_i | \boldsymbol{\alpha}] \right\} \pi_0^{n_0}. \quad (6.2.1)$$

6179 This is discussed in Borchers and Efford (2008, p. 379) as the conditional-on- $N$  form of  
6180 the likelihood – we might also call it the “binomial form” because of its appearance.

6181 Operationally, things proceed much as before: We compute the marginal probability  
6182 of each observed  $\mathbf{y}_i$ , i.e., by removing the latent  $\mathbf{s}_i$  by integration. In addition, we com-  
6183 pute the marginal probability of the “all-zero” encounter history  $\mathbf{y}_{n+1}$ , and make sure to  
6184 weight it  $n_0$  times. We accomplish this by “padding” the data set with a single encounter  
6185 history having  $y_{n+1,j} = 0$  for all traps  $j = 1, 2, \dots, J$ . Then we be sure to include the  
6186 combinatorial term in the likelihood or log-likelihood computation. We demonstrate this  
6187 shortly. To analyze a specific case, we’ll read in our fake data set (simulated using the  
6188 parameters given above). To set some things up in our workspace we do this:

```
6189 data<-simSCR0(discard0=TRUE,rnd=2013) # obtain a simulated data set
6190     ## extract the items we need for analysis
6191 y<-data$Y
6192 nind<-nrow(y)
6193 traplocs<-data$traplocs
6194 J<-nrow(traplocs)
6195 K<-data$K
```

6196 Recall that these data were generated with  $N = 100$ , on an  $8 \times 8$  unit state-space repre-  
6197 senting the trap locations buffered by 2 units.

6198 As before, the likelihood is defined in the **R** workspace as an **R** function, `intlik2`,  
6199 which takes an argument being the unknown parameters of the model and additional  
6200 arguments as prescribed. In particular, we provide the encounter history matrix **y**, the  
6201 trap locations `traplocs`, the spacing of the integration grid (argument `delta`) and the  
6202 state-space buffer. Here is the new likelihood function:

```
6203 intlik2<-function(parm,y=y,delta=.3,X=traplocs,ssbuffer=2){
6204
6205 Xl<-min(X[,1]) - ssbuffer
6206 Xu<-max(X[,1]) + ssbuffer
6207 Yu<-max(X[,2]) + ssbuffer
6208 Yl<-min(X[,2]) - ssbuffer
6209
6210 xg<-seq(Xl+delta/2,Xu-delta/2,delta)
6211 yg<-seq(Yl+delta/2,Yu-delta/2,delta)
6212 npix.x<-length(xg)
6213 npix.y<-length(yg)
6214 area<- (Xu-Xl)*(Yu-Yl)/((npix.x)*(npix.y))
6215 G<-cbind(rep(xg,npix.y),sort(rep(yg,npix.x)))
6216 nG<-nrow(G)
6217 D<- e2dist(X,G)
6218     # extract the parameters from the input vector
```

```

6219 alpha0<-parm[1]
6220 alpha1<-exp(parm[2])
6221 n0<-exp(parm[3]) # note parm[3] lives on the real line
6222 probcap<- plogis(alpha0)*exp(-alpha1*D*D)
6223 Pm<-matrix(NA,nrow=nrow(probcap),ncol=ncol(probcap))
6224 ymat<-rbind(y,rep(0,ncol(y)))
6225
6226 lik.marg<-rep(NA,nrow(ymat))
6227 for(i in 1:nrow(ymat)){
6228 Pm[1:length(Pm)]<- (dbinom(rep(ymat[i,],nG),K,probcap[1:length(Pm)],log=TRUE))
6229 lik.cond<- exp(colSums(Pm))
6230 lik.marg[i]<- sum(lik.cond*(1/nG) )
6231 }
6232 nv<-c(rep(1,length(lik.marg)-1),n0)
6233 ## part1 here is the combinatorial term.
6234 ## math: log(factorial(N)) = lgamma(N+1)
6235 part1<- lgamma(nrow(y)+n0+1) - lgamma(n0+1)
6236 part2<- sum(nv*log(lik.marg))
6237 -1*(part1+ part2)
6238 }

```

6239 To execute this function for the data that we created with `simSCR0`, we execute the  
6240 following command (saving the result in our friend `frog`). This results in the usual output,  
6241 including the parameter estimates, the gradient, and the numerical Hessian which is useful  
6242 for obtaining asymptotic standard errors (see below):

```

6243 > starts<-c(-2.5,0,4)
6244 > frog<-nlm(intlik2,starts,hessian=TRUE,y=y,X=traplocs,delta=.2,ssbuffer=2)
6245
6246 Warning message:
6247 In nlm(intlik2, starts, hessian = TRUE, y = y, X = traplocs, delta = 0.2, :
6248 NA/Inf replaced by maximum positive value
6249
6250 > frog
6251 $minimum
6252 [1] 113.5004
6253
6254 $estimate
6255 [1] -2.538333 0.902807 4.232810
6256
6257 [... additional output deleted ...]

```

6258 While usually produces one or more **R** warnings due to numerical calculations happening  
6259 on extremely small or large numbers (calculation of  $p$  near the edge of the state-space),  
6260 and they also happen if a poor parameterization is used which produces evaluations of  
6261 the objective function beyond the boundary of the parameter space (e.g.,  $n_0 < 0$ ). Such  
6262 numerical warnings can often be minimized or avoided altogether by picking judicious  
6263 starting values of parameters or properly transforming or scaling the parameters. You will

6264 see from the `nlm` output that can be reproduced that the algorithm performed satisfactory  
 6265 in minimizing the objective function. The estimate of population size for the state-space  
 6266 (using the default state-space buffer) is

```
6267 nrow(y)+exp(4.2328)
6268 [1] 110.9099
```

6269 Which differs from the data-generating value ( $N = 100$ ) as we might expect for a single  
 6270 realization. We usually will present an estimate of uncertainty associated with this MLE  
 6271 which we can obtain by inverting the Hessian. Note that  $\text{Var}(\hat{N}) = n + \text{Var}(\hat{n}_0)$ . Since  
 6272 we have parameterized the model in terms of  $\log(n_0)$  we use the delta method<sup>2</sup> described  
 6273 in Williams et al. (2002, Appendix F4) (see also Ver Hoef, 2012) to obtain the variance  
 6274 on the scale of  $n_0$  as follows:

```
6275 (exp(4.2328)^2)*solve(frog$hessian)[3,3]
6276 [1] 260.2033
6277 > sqrt(260)
6278 [1] 16.12452
```

6279 Therefore, the asymptotic “Wald-type” confidence interval for  $N$  is  $110.91 \pm 1.96 \times 16.125 =$   
 6280  $(79.305, 142.515)$ . To report this in terms of density, we scale appropriately by the area of  
 6281 the prescribed state-space which is 64 units of area (i.e., an  $8 \times 8$  square).

### 6282 6.2.1 Integrated Likelihood under data augmentation

6283 The likelihood analysis developed in the previous sections is based on the likelihood in  
 6284 which  $N$  (or  $n_0$ ) is an explicit parameter. This is usually called the “full likelihood” or  
 6285 sometimes “unconditional likelihood” (Borchers et al., 2002) because it is the likelihood  
 6286 for all individuals in the population, not just those which have been captured, i.e., not that  
 6287 which is *conditional on capture*. It is also possible to express an alternative unconditional  
 6288 likelihood using data augmentation, replacing the parameter  $N$  with  $\psi$  (e.g., see Sec. 7.1.6  
 6289 Royle and Dorazio, 2008, for an example). We don’t go into detail here, but we note that  
 6290 the likelihood under data augmentation is a zero-inflated binomial mixture – precisely  
 6291 an occupancy type model (Royle, 2006). Thus, while it is possible to carryout likelihood  
 6292 analysis of models under data augmentation, we primarily advocate data augmentation  
 6293 for Bayesian analysis.

### 6294 6.2.2 Extensions

6295 We have only considered basic SCR models with no additional covariates. However, in  
 6296 practice, we are interested in other types of covariate effects including “behavioral re-  
 6297 sponse”, sex-specificity of parameters, and potentially other effects. Some of these can  
 6298 be added directly to the likelihood if the covariate is fixed and known for all individuals  
 6299 captured or not. An example is a behavioral response, which amounts to having a co-  
 6300 variate  $x_{ik} = 1$  if individual  $i$  was captured prior to occasion  $k$  and  $x_{ik} = 0$  otherwise.

<sup>2</sup> We found a good set of notes on the delta approximation on Dr. David Patterson’s ST549 notes: <http://www.math.umt.edu/patterson/549/Delta.pdf>

6301 For uncaptured individuals,  $x_{ik} = 0$  for all  $k$ . Royle et al. (2011b) called this a global  
6302 behavioral response because the covariate is defined for all traps, no matter the trap in  
6303 which an individual was captured. We could also define a *local* behavioral response which  
6304 occurs at the level of the trap, i.e.,  $x_{ijk} = 1$  if individual  $i$  was captured in trap  $j$  prior  
6305 to occasion  $k$ , etc.. Trap-specific covariates such as trap type or status, or time-specific  
6306 covariates such as date, are easily accommodated as well. As an example, Kéry et al.  
6307 (2010) develop a model for the European wildcat *Felis silvestris* in which traps are either  
6308 baited or not (a trap-specific covariate with only 2 values), and also encounter probability  
6309 varies over time in the form of a quadratic seasonal response. We consider models with  
6310 behavioral response or fixed covariates in Chapt. 8. The integrated likelihood routines we  
6311 provided above can be modified directly for such cases, which we leave to the interested  
6312 reader to investigate.

6313 Sex-specificity is more difficult to deal with since sex is not known for uncaptured  
6314 individuals (and sometimes not even for all captured individuals). To analyze such models,  
6315 we do Bayesian analysis of the joint likelihood using data augmentation (Gardner et al.,  
6316 2010b; Russell et al., 2012), discussed further in Chapt. 8. For such covariates (i.e.,  
6317 that are not fixed and known for all individuals), it is somewhat more challenging to  
6318 do MLE for these based on the joint likelihood as we have developed above. Instead it  
6319 is more conventional to use what is colloquially referred to as the “Huggins-Alho” type  
6320 model which is one of the approaches taken in the software package **secr** (Efford, 2011)  
6321 which we describe in Sec. 6.5 below. This idea is motivated by thinking about unequal  
6322 probability sampling methods known as Horvitz-Thompson sampling (e.g., see Overton  
6323 and Stehman, 1995).

### 6.3 CLASSICAL MODEL SELECTION AND ASSESSMENT

6324 In most analyses, one is interested in choosing from among various potential models, or  
6325 ranking models, or something else to do with assessing the relative merits of a set of  
6326 models. A good thing about classical analysis based on likelihood is we can apply AIC  
6327 methods (Burnham and Anderson, 2002) without difficulty. AIC is convenient for assessing  
6328 the relative merits of these different models although if there are only a few models it is  
6329 not objectionable to use hypothesis tests or confidence intervals to determine importance  
6330 of effects. The second model selection context has to do with choosing among various  
6331 detection functions although, as a general rule, we don’t recommend this application of  
6332 model selection. This is because there is hardly ever (if at all) a rational subject-matter  
6333 based reason motivating specific distance functions. As a result, we believe that doing  
6334 too much model selection will invariably lead to over-fitting and thus over-statement of  
6335 precision. This is the main reason that we haven’t loaded you down with a basket of  
6336 models for detection probability so far, although we discuss many possibilities in Chapt.  
6337 8.

6338 **Goodness-of-fit or model-checking** – For many standard capture-recapture mod-  
6339 els, it is possible to identify goodness-of-fit statistics (**XXXX REF XXXXX**) based  
6340 on the multinomial likelihood and evaluate model adequacy using formal statistical tests.  
6341 Similar strategies can be applied to SCR models using expected cell-frequencies based on  
6342 the marginal distribution of the observations. Also, because computing MLEs is somewhat  
6343 more efficient in many cases compared to Bayesian analysis, it is also sometimes feasible

6344 to use bootstrap methods. At the present time, we don't know of any applications of  
 6345 goodness-of-fit testing for SCR models based on likelihood inference, although we discuss  
 6346 the use of Bayesian p-values for assessing model fit in Chapt. 7. An important practical  
 6347 problem in trying to evaluate goodness-of-fit is that, in realistic sample sizes, fit tests often  
 6348 lack the power to detect departures from the model under consideration and so they may  
 6349 not be generally useful in practice.

## 6.4 LIKELIHOOD ANALYSIS OF THE WOLVERINE CAMERA TRAPPING DATA

6350 Here we compute the MLEs for the wolverine data using an expanded version of the  
 6351 function we developed in the previous section. To accommodate that each trap might  
 6352 be operational a variable number of nights, we provided an additional argument to the  
 6353 likelihood function (allowing for a vector  $\mathbf{K} = (K_1, \dots, K_J)$ ), which requires also a modifi-  
 6354 cation to the construction of the likelihood. In addition, we accommodate the state-space  
 6355 is a general rectangle, and we included a line in the code to compute the state-space  
 6356 area which we apply below for computing density. The more general function (`intlik3`)  
 6357 is given in the **R** package `scrbook`. Incidentally, this function also returns the area of  
 6358 the state-space for a given set of parameter values, as an attribute to the function value,  
 6359 which will be used in converting  $\hat{N}$  to  $\hat{D}$ . To use this function to obtain the MLEs for the  
 6360 wolverine camera trap study, we execute the following commands (note: these are in the  
 6361 help file and will execute if you type `example(intlik3)`):

```
6362 library(scrbook)
6363 data(wolverine)
6364
6365 traps<-wolverine$wtraps
6366 traplocs<-traps[,2:3]/10000
6367 K.wolv<-apply(traps[,4:ncol(traps)],1,sum)
6368
6369 y3d<-SCR23darray(wolverine$wcaps,traps)
6370 y2d<-apply(y3d,c(1,3),sum)
6371
6372 starts<-c(-1.5,0,3)
6373
6374 frog<-nlm(intlik3,starts,hessian=TRUE,y=y2d,K=K.wolv,X=traplocs,delta=.2,ssbuffer=2)
6375
6376 frog
6377 $minimum
6378 [1] 220.4313
6379
6380 $estimate
6381 [1] -2.8176120  0.2269395  3.5836875
6382
6383 [.... output deleted ....]
```

6384 Of course we're interested in obtaining an estimate of population size for the prescribed  
 6385 state-space, or density, and associated measures of uncertainty which we do using the delta

6386 method (Williams et al., 2002, Appendix F4). To do all of that we need to manipulate the  
 6387 output of `nlm` since we have our estimate in terms of `log(n0)`. We execute the following  
 6388 commands:

```
6389 frog<-nlm(intlik3,starts,hessian=TRUE,y=y2d,K=K.wolv,X=traplocs,delta=.2,ssbuffer=2)
6390 Nhat<-nrow(y2d)+exp(frog$estimate[3])
6391 area<-attr(intlik3(starts,y=y2d,K=K.wolv,X=traplocs,delta=.2,ssbuffer=2),"SSarea")
6392 Dhat<- Nhat/area
6393
6394 Dhat
6395 [1] 0.5494947
6396
6397 SE<- (1/area)*exp(frog$estimate[3])*sqrt(solve(frog$hessian)[3,3])
6398
6399 SE
6400 [1] 0.1087073
```

6401 So our estimate of density is 0.55 individuals per “standardized unit” which is 100  
 6402  $km^2$ , because we divided UTM coordinates by 10000. So this is about 5.5 individuals per  
 6403 1000  $km^2$ , with a SE of around 1.09 individuals. This compares closely with 5.77 reported  
 6404 in Sec. 5.8 based on Bayesian analysis of the model.

6405 To evaluate the effect of the integration grid density, we obtained the MLEs for a state-  
 6406 space buffer of 2 (standardized units) and for integration grid with spacing  $\delta = .3, .2, .1, .05$ .  
 6407 The MLEs for these 4 cases including the relative runtime are given in Table 6.1. We see  
 6408 the results change only slightly as the fineness of the integration grid increases. Conversely,  
 6409 the runtime on the platform of the day for the 4 cases increases rapidly. These runtimes  
 6410 could be regarded in relative terms, across platforms, for gaging the decrease in speed  
 6411 as the fineness of the integration grid increases. The effect of this is that we anticipate  
 6412 some numerical error in approximating the integral on a mesh of points, and that error  
 6413 increases as the coarseness of the mesh increases.

**Table 6.1.** Run time and MLEs for different integration grid resolutions for the wolverine camera trapping data.

$\delta$	Estimates			
	runtime	$\hat{\alpha}_0$	$\hat{\alpha}_1$	$\log(\hat{n}_0)$
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

6414 We studied the effect of the state-space buffer on the MLEs, using a fixed  $\delta = .2$  for  
 6415 all analyses. The results are show in Table 6.2. We used state-space buffers of 1 to 4  
 6416 units stepped by .5. As we can see in Table 6.2, the estimates of  $D$  stabilize rapidly and  
 6417 the incremental difference is within the numerical error associated with approximating the  
 6418 integral.

**Table 6.2.** Results of the effect of the state-space buffer on the MLE. Given here are the state-space buffer (buff), area of the state-space (area), the MLE of  $N$  ( $\hat{N}$ ) for the prescribed state-space and the corresponding MLE of density ( $\hat{D}$ ).

buff	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	0.556646

#### 6419    6.4.1 Using a habitat mask (Restricted state-space)

6420    In Sec. 5.9 we used a discrete representation of the state-space in order to have control  
 6421    over its extent and shape. This makes it easy to do things like clip out non-habitat, or  
 6422    create a *habitat mask* which defines suitable habitat. Clearly that formulation of the model  
 6423    is relevant to the calculation of the marginal likelihood in the sense that the discrete state-  
 6424    space is equivalent to the integration grid. Thus, for example, we could easily compute the  
 6425    MLE of parameters under some model with a restricted state-space merely by creating the  
 6426    required state-space at whatever grid resolution is desired, and then inputting that state-  
 6427    space into the likelihood function above, instead of computing it in the function itself. We  
 6428    can easily create an explicit state-space grid for integration from arbitrary polygons or  
 6429    GIS shapefiles which we demonstrate here. Our approach is to create the integration grid  
 6430    (or state-space grid) outside of the likelihood evaluation, and then determine which points  
 6431    of the grid lie in the polygon defined by the shapefile using functions in the **R** packages **sp**  
 6432    and **maptools**. For each point in the state-space grid (object **G** in the code below which is  
 6433    assumed to exist), we determine whether it is inside the polygon<sup>3</sup>, identifying such points  
 6434    with a value of **mask=1** and **mask=0** for points that are *not* in the polygon. We load the  
 6435    shapefile which originates by an application of the **readShapeSpatial** function. We have  
 6436    saved the result into an **R** data object called **SSp** which is in the **scrbook** package. Here  
 6437    are the **R** commands for doing this (see the helpfile **?intlik4**):

```
6438 library(maptools)
6439 library(sp)
6440 library(scrbook)
6441
6442 data("fakeshapefile")
```

<sup>3</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 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. 14.5 in Chapt. 14). 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.

```

6443 ##### replaces this:
6444 #####SSp<-readShapeSpatial('Sim_Polygon.shp')
6445 Pcoord<-SpatialPoints(G)
6446 PinPoly<-over(Pcoord,SSp) ### determine if each point is in polygon
6447 mask<-as.numeric(!is.na(PinPoly[,1])) ## convert to binary 0/1
6448 G<-G[mask==1,]

```

We created the function `intlik4` which accepts the integration grid as an explicit argument, and this function is also available in the package `scrbook`.

We apply this modification to the wolverine camera trapping study. Royle et al. (2011b) created 2, 4 and 8 km state-space grids so as to remove “non-habitat” (mostly ocean, bayes, and large lakes). We previously analyzed the model using **JAGS** and **WinBUGS** in Chapt. 5. To set up the wolverine data and fit the model we execute the following commands

```

6456 library(scrbook)
6457 data(wolverine)

6458
6459 traps<-wolverine$wtraps
6460 traplocs<-traps[,2:3]/10000
6461 K.wolv<-apply(traps[,4:ncol(traps)],1,sum)

6462
6463 y3d<-SCR23darray(wolverine$wcaps,traps)
6464 y2d<-apply(y3d,c(1,3),sum)
6465 G<-wolverine$grid2/10000

6466
6467 starts<-c(-1.5,0,3)
6468 frog<-nlm(intlik4,starts,hessian=TRUE,y=y2d,K=K.wolv,X=traplocs,G=G)

6469
6470 frog
6471 $minimum
6472 [1] 225.8355

6473
6474 $estimate
6475 [1] -2.9955424  0.2350885  4.1104757
6476
6477 [... some output deleted ...]

```

Next we convert the parameter estimates to estimates of total population size for the prescribed state-space, and then obtain an estimate of density (per 1000 km<sup>2</sup>) using the area computed as the number of pixels in the state-space grid `G` multiplied by the area per grid cell. In the present case (the calculation above) we used a state-space grid with 2km × 2km pixels. Finally, we compute a standard errors using the delta approximation:

```

6483 area<- nrow(G)*4
6484 Nhat<- 21+exp(frog$estimate[3])
6485 SE<- exp(frog$estimate[3])*sqrt(solve(frog$hessian)[3,3])
6486 D<- (Nhat/(nrow(G)*area))*1000
6487 SE.D<- (SE/(nrow(G)*area))*1000

```

**Table 6.3.** MLEs 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

6488 We did this for each the 2 km, 4 km and 8 km state-space grids which produced the  
 6489 estimates summarized in Table 6.3. These estimates compare with the 8.6 (2 km grid)  
 6490 and 8.2 (8 km grid) reported in Royle et al. (2011b) based on a clipped state-space as  
 6491 described in Sec. 5.9.

## 6.5 DENSITY AND THE R PACKAGE SECR

6492 **DENSITY** is a software program developed by Efford (2004) for fitting spatial capture-  
 6493 recapture models based mostly on classical maximum likelihood estimation and related  
 6494 inference methods. Efford (2011) has also released an **R** package called **secr**, that contains  
 6495 much of the functionality of **DENSITY** but also incorporates new models and features.  
 6496 Here, we briefly introduce the **secr** package which we prefer to use instead of **DENSITY**  
 6497 because it allows us to remain in the **R** environment for data processing and summarization.  
 6498 We provide a brief introduction to **secr** and some of its capabilities here, and we  
 6499 also use it for doing some analysis in other parts of this book.

6500 To install and run models in **secr**, you must download the package and load it in **R**.

```
6501 install.packages("secr")
6502 library(secr)
```

6503 **secr** allows the user to simulate data and fit a suite of models with various detection func-  
 6504 tions and covariate responses. It also contains a number of helpful constructor functions  
 6505 for creating objects of the proper class that are recognized by other **secr** functions. We  
 6506 provide a brief overview of the capabilities here, but the **secr** help manual can be accessed  
 6507 with the command:

```
6508 RShowDoc("secr-manual", package = "secr")
```

6509 We note that **secr** has many capabilities that we will not cover or do so only sparingly.  
 6510 We encourage you to read through the manual to get a better understanding of what the  
 6511 package is capable of.

6512 The main model-fitting function in **secr** is called **secr.fit**, which makes use of the  
 6513 standard **R** model specification framework with tildes. As an example, the equivalent of  
 6514 the basic model SCR0 is fitted as follows:

```
6515 secr.fit(capturedata, model = list(D ~ 1, g0 ~ 1, sigma ~ 1), buffer = 20000)
```

6516 where **capturedata** is the object created by **secr** containing the encounter history  
 6517 data and the trap information, and the model expression **g0~1** indicates the intercept-  
 6518 only (i.e., constant) model. Possible predictors for detection probability include both

6519 pre-defined variables (e.g., `t` and `b` corresponding to “time” and “behavior”), and user-  
 6520 defined covariates of several kinds. For example, to include a global behavioral response,  
 6521 this would be written as `g0~b`. The discussion of this (global versus local trap-specific  
 6522 behavioral response) and other covariates is developed more in Chapt. 8.

6523 Before we can fit the models, the data must first be packaged properly for `secr`.  
 6524 We require data files that contain two types of information: trap layout (location and  
 6525 identification information for each trap), which is equivalent to our trap deployment file  
 6526 (TDF) described in sec. 5.8 and the capture data file containing sampling *session*, ani-  
 6527 mal identification, trap day, and trap location, equivalent in information content to our  
 6528 encounter data file (EDF). Sample session can be thought of as primary period identifier  
 6529 in a robust design like framework – it could represent a yearly sample or multiple sample  
 6530 periods within a year, each of them producing data on a closed population. We discuss  
 6531 “multi-session” models in more detail below, in sec. 6.5.4.

6532 There are three important constructor functions that help package-up your data for  
 6533 use in `secr`: `read.traps`, `make.capthist` and `read.mask`. We provide a brief description  
 6534 of each here, but apply them to our wolverine camera trapping data in the next section:

6535 (1) `read.traps`: This function points to an external file *or* **R** data object containing the  
 6536 trap coordinates, and other information, and also requires specification of the type of  
 6537 encounter devices (described in the next section). A typical application of this function  
 6538 looks like the following, invoking the `data=` option when there is an existing **R** object  
 6539 containing the trap information:

```
6540 trapfile<-read.traps(data=traps,detector="proximity")
```

6541 (2) `make.capthist`: This function takes the EDF and combines it with trap information,  
 6542 and the number of sampling occasions. A typical application looks like this:

```
6543 capturedata<-make.capthist(enc.data,trapfile,fmt="trapID",noccasions=165)
```

6544 See `?make.capthist` for definition of distinct file formats. Specifying `fmt = trapID` is  
 6545 equivalent to our EDF format.

6546 (3) `read.mask`: If there is a habitat mask available (as described in sec. 6.4.1), then this  
 6547 function will organize it so that `secr.fit` knows what to do with it. The function  
 6548 accepts either an external file name (see `?read.mask` for details of the structure) or a  
 6549  $nG \times 2$  **R** object, say `mask.coords`, containing the coordinates of the mask. A typical  
 6550 application looks like the following:

```
6551 grid<-read.mask(data=mask.coords)
```

6552 These constructor functions produce output that can then be used in the fitting of models  
 6553 using `secr.fit`.

### 6554 6.5.1 Encounter device types and detection models

6555 The `secr` package requires that you specify the type of encounter device. Instead of  
 6556 describing models by their statistical distribution (Bernoulli, Poisson, etc..), `secr` uses  
 6557 certain operational classifications of detector types including ‘proximity’, ‘multi’, ‘single’,  
 6558 ‘polygon’ and ‘signal’. For camera trapping/hair snares we might consider ‘proximity’  
 6559 detectors or ‘count’ detectors. The ‘proximity’ detector type allows, at most, one detection

of each individual at a particular detector on any occasion (i.e., it is equivalent to the Bernoulli or binomial encounter process model, or model SCR0). The ‘count’ detector designation allows repeat encounters of each individual at a particular detector on any occasion. There are other detector types that one can select such as: ‘polygon’ detector type which allows for a trap to be a sampled polygon (Royle and Young, 2008) which we discuss further in Chapt. 17, and ‘signal’ detector which allows for traps that have a strength indicator, e.g., acoustic arrays (Dawson and Efford, 2009). The detector types ‘single’ and ‘multi’ refer to traps that retain individuals, thus precluding the ability for animals to be captured in other traps during the sampling occasion. The ‘single’ type indicates trap that can only catch one animal at a time (single-catch traps), while ‘multi’ indicates traps that may catch more than one animal at a time (multi-catch). These are both variations of the multinomial encounter models described in Chapt. 9.

As with all SCR models, **secr** fits a detection function relating the probability of detection to the distance of a detector from an individual activity center. **secr** allows the user to specify one of a variety of detection functions including the commonly used half-normal, hazard rate, and exponential. There are 12 different functions as of version 2.3.1 (see Table 8.1 in Chapt. 8), but some are only available for simulating data. The different detection functions are defined in the **secr** manual and can be found by calling the help function for the detection function:

?detectfn

It is useful to note that **secr** requires the buffer distance to be defined in meters and density will be returned as number of animals per hectare. Thus to make comparisons between **secr** and other models, we will often have to convert the density to the same units.

Most of the detection functions available in **secr** contain some kind of a scale parameter which is usually labeled  $\sigma$ . The units of this parameter default to meters in the **secr** output. We caution that the meaning of this parameter depends on the specific model being used and it should not be directly compared as a measure of home-range size across models. Instead, as we noted in sec. 5.3 every encounter probability model implies a model of space-usage and fitted encounter models should be converted to a common currency such as “area used.”

### 6.5.2 Analysis using the **secr** package

To demonstrate the use of the **secr** package, we will show how to do the same analysis on the wolverine study as shown in sec. 5.8. To use the **secr** package, the data need to be formatted in a similar but slightly different manner than we use in **WinBUGS**.

For example, in sec. 5.8 we introduced a standard data format for the encounter data file (EDF) and trap deployment file (TDF). The EDF shares the same format as that used by the **secr** package with 1 row for every encounter observation and 4 columns representing trap session (‘Session’), individual identity (‘ID’), sample occasion (‘Occasion’), and trap identity (‘trapID’). For a standard closed population study that takes place during a single season, the ‘Session’ column in our case is all 1s, to indicate a single primary sampling occasion. In addition to providing the encounter data file (EDF), we must tell **secr** information about the traps, which is formated as a matrix with column labels ‘trapID’, ‘x’ and

6603 'y', the last two being the coordinates of each trap, with additional columns representing  
 6604 the operational state of each trap during each occasion (1=operational, 0=not).

6605 We demonstrate these differences now by walking through an analysis of the wolverine  
 6606 camera trapping data using **secr**. To read in the trap locations and other related information,  
 6607 we make use of the constructor function **read.traps** which also requires that we specify the detector type.  
 6608 The detector type is important because it will determine the likelihood that **secr** will use to fit the model.  
 6609 Here, we have selected "proximity" which corresponds to the Bernoulli encounter model in which individuals are captured at most  
 6610 once in each trap during each sampling occasion:

```
6612 library(secr)
6613 library(scrbook)
6614 data(wolverine)

6615
6616 traps<-as.matrix(wolverine$wtraps)
6617 dimnames(traps)<-list(NULL,c("trapID","x","y",paste("day",1:165,sep="")))
6618 traps1<-as.data.frame(traps[,1:3])
6619 trapfile1<-read.traps(data=traps1,detector="proximity")
```

6620 Here we note that trap coordinates are extracted from the wolverine data but we do not standardize them. This is because **secr** defaults to coordinate scaling of meters which is the extant scaling of the wolverine trap coordinates. Note that we add a 'trapID' column to the trap coordinates and provide appropriate column labels to the 'traps' matrix. An important aspect of the wolverine study is that while the camera traps were operated over a 165 day period, each trap was operational during only a portion of that period. We need to provide the trap operation information which is contained in the columns to the right of the trap coordinates in our standard trap deployment file (TDF). Unfortunately, this is less easy to do in **secr**, which requires an external file with a single long string of 1's and 0's indicating the days in which each trap was operational (1) or not (0). The **read.traps** function will not allow for this information on trap operation if the data exists as an **R** object – instead, we can create this external file and then read it back in with **read.traps** using these commands:

```
6633 hold<-rep(NA,nrow(traps))
6634 for(i in 1:nrow(traps)){
6635 hold[i]<-paste(traps[i,4:ncol(traps)],collapse="")
6636 }
6637 traps1<- cbind(traps[,1:3],"usage"=hold)
6638
6639 write.table(traps1, "traps.txt", row.names=FALSE, col.names=FALSE)
6640 trapfile2<-read.traps("traps.txt",detector="proximity")
```

6641 These operations can be accomplished using the function **scr2secr** which is provided in  
 6642 the **R** package **scrbook**.

6643 After reading in the trap data, we now need to create the encounter matrix or array  
 6644 using the **make.capthist** command, where we provide the capture histories in EDF format,  
 6645 which is the existing format of the data input file **wcaps**. In creating the capture history,  
 6646 we provide also the trapfile created previously, the format (e.g., here EDF format is

6647 `fmt= “trapID”), and finally, we provide the number of occasions. We also set up a  
 6648 habitat mask using the  $2 \times 2 \text{ km}$  grid which we used previously in the analysis of the  
 6649 wolverine data and then pass the relevant objects to secr.fit as follows:`

```
6650 #
6651 # grab the encounter data file and format it:
6652 #
6653 wolv.dat<-wolverine$wcaps
6654 dimnames(wolv.dat)<-list(NULL,c("Session","ID","Occasion","trapID"))
6655 wolv.dat<-as.data.frame(wolv.dat)
6656 wolvcapt2<-make.capthist(wolv.dat,trapfile2,fmt="trapID",noccasions=165)
6657
6658 # grab the habitat mask (2 x 2 km) and format it:
6659 #
6660 gr2<-(as.matrix(wolverine$grid2))
6661 dimnames(gr2)<-list(NULL,c("x","y"))
6662 gr2<-read.mask(data=gr2)
6663 # To fit the model we use secr.fit:
6664 #
6665 wolv.secr2<-secr.fit(wolvcapt2,model=list(D ~ 1, g0 ~ 1, sigma ~ 1),
6666           buffer=20000,mask=gr2)
```

6667 We are using the basic “proximity detector” model (SCR0), so we do not need to make  
 6668 any specifications in the command line because we have specified the detector type using  
 6669 the constructor function `read.traps`, except to provide the buffer size (in  $m$ ). To specify  
 6670 different models, you can change the default `D~1`, `g0~1`, `sigma~1`, which the interested  
 6671 reader can do with very little difficulty. We provide all of these commands and additional  
 6672 analyses in the `scrbook` package with the function called `secr_wolverine`. Printing the  
 6673 output object produces the following (slightly edited):

```
6674 wolv.secr2
6675
6676 secr 2.3.1, 15:52:45 29 Aug 2012
6677
6678 Detector type      proximity
6679 Detector number     37
6680 Average spacing    4415.693 m
6681 x-range            593498 652294 m
6682 y-range            6296796 6361803 m
6683 N animals          : 21
6684 N detections       : 115
6685 N occasions        : 165
6686 Mask area          : 987828.1 ha
6687
6688 Model              : D ~ 1 g0 ~ 1 sigma ~ 1
6689 Fixed (real)        : none
6690 Detection fn        : halfnormal
```

```

6691 Distribution      : poisson
6692 N parameters     : 3
6693 Log likelihood   : -602.9207
6694 AIC              : 1211.841
6695 AICc             : 1213.253
6696
6697 Beta parameters (coefficients)
6698          beta    SE.beta      lcl      ucl
6699 D      -9.390124 0.22636698 -9.833795 -8.946452
6700 g0     -2.995611 0.16891982 -3.326688 -2.664535
6701 sigma  8.745547 0.07664648  8.595323  8.895772
6702
6703 Variance-covariance matrix of beta parameters
6704          D        g0        sigma
6705 D      0.0512420110 -0.0004113326 -0.003945371
6706 g0     -0.0004113326  0.0285339045 -0.006269477
6707 sigma -0.0039453711 -0.0062694767  0.005874683
6708
6709 Fitted (real) parameters evaluated at base levels of covariates
6710          link   estimate  SE.estimate      lcl      ucl
6711 D      log 8.354513e-05 1.915674e-05 5.360894e-05 1.301982e-04
6712 g0     logit 4.762453e-02 7.661601e-03 3.466689e-02 6.509881e-02
6713 sigma  log 6.282651e+03 4.822512e+02 5.406315e+03 7.301037e+03

```

6714 The object returned by `secr.fit` provides extensive default output when printed.  
6715 Much of this is basic descriptive information about the model, the traps, or the encounter  
6716 data. We focus here on the parameter estimates. Under the fitted (real) parameters, we  
6717 find  $D$ , the density, given in units of individuals/hectare (1 hectare = 10000  $m^2$ ). To  
6718 convert this into individuals/1000 km $^2$ , we multiply by 100000, thus our density estimate  
6719 is 8.35 individuals/1000 km $^2$ . The parameter  $\sigma$  is given in units of meters, and so this  
6720 corresponds to 6.283 km. Both of these estimates are very similar to those obtained in  
6721 our likelihood analysis summarized in Table 6.3 which, for the 2  $\times$  2 km grid, we obtained  
6722  $\hat{D} = 8.31$  with a SE of  $100000 \times 1.915674e - 05 = 1.9156$  and, accounting for the scale  
6723 difference (1 unit = 10000 m in the previous analysis),  $\hat{\sigma} = \sqrt{1/(2\hat{\alpha}_1)} * 10000 = 6.289$   
6724 km. The difference in the MLE between Table 6.3 and those produced by `secr` are likely  
6725 due to subtle differences in internal tuning of optimization algorithms, starting values or  
6726 other numerical settings. In addition, see the next section. On the other hand, the SE is  
6727 slightly larger based on `secr` which is due to a subtle difference in the interpretation of  $D$   
6728 under the `secr` model (See below).

### 6729 6.5.3 Likelihood Analysis in the `secr` Package

6730 The `secr` package does likelihood analysis of SCR models for most classes of models as  
6731 developed by Borchers and Efford (2008). Their formulation deviates slightly from the  
6732 binomial form we presented in sec. 6.2 above (though Borchers and Efford (2008) mention  
6733 the binomial form). Specifically, the likelihood that `secr` implements is that based on  
6734 removing  $N$  from the likelihood by integrating the binomial likelihood (Eq. 6.2.1 above)

6735 over a Poisson prior for  $N$  – what we will call the *Poisson-integrated likelihood* as opposed  
 6736 to the conditional-on- $N$  (*binomial-form*) considered previously.

6737 To develop the Poisson-integrated likelihood we compute the marginal probability of  
 6738 each  $\mathbf{y}_i$  and the probability of an all-0 encounter history,  $\pi_0$ , as before, to arrive at the  
 6739 marginal likelihood in the binomial-form:

$$\mathcal{L}(\boldsymbol{\alpha}, n_0 | \mathbf{y}) = \frac{N!}{n! n_0!} \left\{ \prod_i [\mathbf{y}_i | \boldsymbol{\alpha}] \right\} \pi_0^{n_0}$$

6740 Now, what Borchers and Efford (2008) do is assume that  $N \sim \text{Poisson}(\Lambda)$  and they do a  
 6741 further level of marginalization over this prior distribution:

$$\sum_{n_0=0}^{\infty} \frac{N!}{n! n_0!} \left\{ \prod_i [\mathbf{y}_i | \boldsymbol{\alpha}] \right\} \pi_0^{n_0} \frac{\exp(-\Lambda)\Lambda^{n_0}}{N!}$$

6742 Carrying-out the summation above produces exactly this marginal likelihood:

$$\mathcal{L}_2(\boldsymbol{\alpha}, \Lambda | \mathbf{y}) = \left\{ \prod_i [\mathbf{y}_i | \boldsymbol{\alpha}] \right\} \Lambda^n \exp(-\Lambda(1 - \pi_0))$$

6743 which is Eq. 2 of Borchers and Efford (2008) except for notational differences. It also  
 6744 resembles the binomial-form of the likelihood in Eq. 6.2.1 except with  $\Lambda^n \exp(-\Lambda\pi_0)$  re-  
 6745 placing the combinatorial term and the  $\pi_0^{n_0}$  term. We emphasize there are two marginal-  
 6746izations going on here: (1) the integration to remove the latent variables  $\mathbf{s}$ ; and, (2)  
 6747 summation to remove the parameter  $N$ . We provide a function for computing this in  
 6748 the `scrbook` package called `intlik3Poisson`. The help file for that function shows how  
 6749 to conduct a small simulation study to compare the MLE under the Poisson-integrated  
 6750 likelihood with that from the binomial form.

6751 The essential distinction between our MLE and Borchers and Efford as implemented in  
 6752 `secr` is whether you keep  $N$  in the model or remove it by integration over a Poisson prior.  
 6753 If you have prescribed a state-space explicitly with a sufficiently larger buffer, then we  
 6754 imagine there should be hardly any difference at all between the MLEs obtained by either  
 6755 the Poisson-integrated likelihood or the binomial-form of the likelihood which retains  $N$ .  
 6756 There is a subtle distinction in the sense that under the binomial form, we estimate the  
 6757 realized population size  $N$  for the state-space whereas, for the Poisson-integrated form we  
 6758 estimate the *prior* expected value which would apply to a hypothetical new study of a  
 6759 similar population.

6760 Both models (likelihoods) assume  $\mathbf{s}$  is uniformly distributed over space, but for the  
 6761 binomial model we make no additional assumption about  $N$  whereas we assume  $N$  is  
 6762 Poisson using the formulation in `secr` from (Borchers and Efford, 2008). Using data  
 6763 augmentation we could do a similar kind of integration but integrate  $N$  over a binomial  
 6764 ( $M, \psi$ ) prior – which we referred to as the binomial-integrated likelihood in sec. 4.2.4.  
 6765 So obviously the two approaches (data augmentation and Poisson-integrated likelihood)  
 6766 are approximately the same as  $M$  gets large. However, doing a Bayesian analysis by  
 6767 MCMC, we obtain an estimate of both  $N$ , the *realized population size*, and the parameter  
 6768 controlling its expected value  $\psi$  which are, in fact, both identifiable from the data even  
 6769 using likelihood analysis (Royle et al., 2007). That said we can integrate  $N$  out completely

6770 and just estimate  $\psi$  as we noted in sec. 6.2.1 above. And we could make a prediction for a  
 6771 new study which would be based on the posterior distribution of  $M\psi$  which, we imagine,  
 6772 should have slightly larger uncertainty associated with it.

6773 **6.5.4 Multi-Session Models in secr**

6774 In practice we will often deal with SCR data that have some meaningful stratification or  
 6775 group structure. For example, we might conduct mist-netting of birds on  $K$  consecutive  
 6776 days, repeated, say,  $T$  times during a year, or perhaps over  $T$  years. Or we might collect  
 6777 data from  $R$  distinct trapping grids. In these cases, we have  $T$  or  $R$  groups which we might  
 6778 reasonably regard as being samples of independent populations. While the groups might be  
 6779 distinct sites, year, or periods within years, they could also be other biological groups  
 6780 such as sex or age. Conveniently, **secr** fits a specific model for stratified populations –  
 6781 referred to as *multi-session* models. These models build on the Poisson assumption which  
 6782 underlies the integrated likelihood used in **secr** (as described in the previous section). To  
 6783 understand the technical framework, let  $N_g$  be the population size of group  $g$  and *assume*

$$N_g \sim \text{Poisson}(\Lambda_g).$$

6784 Naturally, we model group-specific covariates on  $\Lambda_g$ :

$$\log(\Lambda_g) = \beta_0 + \beta_1 z_g$$

6785 where  $z_g$  is some group-specific covariate such as a categorical index to the group, or a  
 6786 trend variable, or a spatial covariate, such as treatment effect or habitat structure, if the  
 6787 groups represent spatial units. Under this model, we can marginalize *all*  $N_g$  parameters  
 6788 out of the likelihood to concentrate the likelihood on the parameters  $\beta_0$  and  $\beta_1$  precisely  
 6789 as discussed in the previous section. This Poisson hierarchical model is the basis of the  
 6790 multi-session models in **secr**.

6791 To implement a multi-session model (or stratified population model) in **secr**, we pro-  
 6792 vide the relevant stratification information in the 'Session' variable of the input encounter  
 6793 data file (EDF). If 'Session' has multiple values then a "multi-session" object is created  
 6794 by default and session-specific variables can be described in the model. For example, if  
 6795 the session has 2 values for males and females then we have sex-specific densities , and  
 6796 baseline encounter probability  $p_0$  (named  $g_0$  in **secr**) by just doing this (see Chapt. 7 for  
 6797 the **R** code to set this up):

```
6798 out<-secr.fit(capdata,model=list(D ~ session, g0 ~ session, sigma ~ 1), buffer=20000)
```

6799 More detailed analysis is given in Sec. 7.1.2 where we fit a number of different models and  
 6800 apply methods of model selection to obtain model-averaged estimates of density.

6801 We can also easily implement stratified population models in the various **BUGS** en-  
 6802 gines using data augmentation (Converse and Royle, 2012; Royle and Converse, 2012)  
 6803 which we discuss, with examples, in Chapt. 16.

6804 **6.5.5 Some additional capabilities of secr**

6805 The **secr** package has capabilities to do complete analysis of SCR data sets, including  
 6806 model fitting, selection, and many summary analyses. In the previous sections, we've

given a basic overview, and we do more in later chapters of this book. Here we mention a few of these other capabilities that you should know about as you use **secr**. Of course, you might as well skim through the associated documentation (**?secr**) to see more of what's available.

### Alternative observation models

**secr** fits a wide range of alternative observation models besides the Bernoulli encounter model, including multinomial encounter models for “multi-catch” and “single catch” traps, models for sound attenuation from acoustic detection devices, and many others. We discuss many of these other methods in Chapt. 9 and elsewhere in the book.

### Summary Statistics

**secr** provides a useful default summary of the data, but it also has summary statistics about animal movement including mean-maximum distance moved. For example, see the help page **?MMDM** which lists a number of other summary functions which take a **capthist** object:

```
6821 dbar(capthist)
6822 RPSV(capthist)
6823 MMDM(capthist, min.recapt = 1, full = FALSE)
6824 ARL(capthist, min.recapt = 1, plt = FALSE, full = FALSE)
6825 moves(capthist)
```

### State-space buffer

**secr** will produce a warning if the state-space buffer is chosen too small. For example, in fitting the wolverine data as in Sec. 6.5.2 but with a 1000 m buffer, and we see the following warning message:

```
6830 Warning message:
6831 In secr.fit(wolvcapt2, model = list(D ~ 1, g0 ~ 1, sigma ~ 1), buffer = 1000) :
  predicted relative bias exceeds 0.01 with buffer = 1000
```

This should cause you to contemplate modifying the state-space buffer if that is a reasonable thing to do in the specific application.

### Model selection and averaging

**secr** does likelihood ratio tests to compare nested models using the function **LR.test**. You can create model selection tables based on AIC or AICc, and obtain model-averaged parameter estimates using **model.average** (See Chapt. 7 for examples).

### Population closure test

**secr** has a population closure test with the function **closure.test** which implements the tests of ? or Otis et al. (1978). The function is used like this: **closure.test(object, SB = FALSE)**. Here **object** is a **capthist** object and **SB** is a logical variable that, if TRUE, produces the ? test.

### Density mapping and effective sample area

**secr** produces likelihood versions of the various summaries of posterior density and effective sample area that we discussed in Chapt. 5. For example, while **secr** reports estimates of density directly in the summary output from fitting a model, you can use the function

6848 `region.N` to produce estimates of  $N$  for any given region. In addition, `secr` has functions  
 6849 for creating maps of detection contours for individuals traps, or for the entire trap array.  
 6850 See the function `pdot.contour`, and also `fxi.contour` for computing the 2-dimensional  
 6851 pdf of the locations of one or more individual activity centers (as in Sec. 5.10.4). In the  
 6852 context of likelihood analysis, estimation of a random effect  $s$  is based on kind of a plug-in  
 6853 application of Bayes' Rule. When  $s$  has a uniform distribution, and we use a discrete  
 6854 evaluation of the integral, can be computed simply by renormalizing the likelihood:

$$[s|y, \theta] = \frac{[y|s, \theta]}{\sum_s [y|s, \theta]}.$$

6855 Any of the `intlik` functions given previously in this chapter can be easily modified to  
 6856 return the posterior distribution of  $s$  for any, or all, individuals, or an individual that is  
 6857 not encountered.

6858 `secr` can calculate the effective sampled area (see Sec. 5.11) using `esa` (and see also  
 6859 `esa.plot`).

### 6860 Covariate models

6861 `secr` has many capabilities for modeling explicit covariates. It has a number of built-in  
 6862 models that allow certain covariates on encounter probability, which we covered to a large  
 6863 extent in Chapt. 8, and also see Chapt. 7 for more examples. But what if you have  
 6864 a covariate of some sort that isn't one of the pre-defined models? Multi-session models,  
 6865 great way to handle discrete covariates such as sex or whatever (see Chapt. 8) and you  
 6866 can also specify any arbitrary covariate in a model, using the standard model specification  
 6867 syntax. (**XXX Note: I'm not sure how to do this**)

6868 `secr` also allows covariates to be built into the density model. (**XXX Would be nice**  
 6869 **to refer to Richard's chapter here XXXX** Chapt. 13). It has some built in response  
 6870 surface models, allowing for the fitting of linear or quadratic response surfaces. This is  
 6871 done by modifying the density model in `secr.fit`. For example,  $D \sim 1$  is a constant  
 6872 density surface, and  $D \sim x + y$  fits a linear response surface, etc... See the manual on  
 6873 `secr-densitysurfaces.pdf` for the details.

6874 To make your own covariates, there seems to be many ways to do this. One is to  
 6875 use the `addCovariates` function and supply it a `mask` or `traps` object along with some  
 6876 "spatialdata" and what happens depends on the nature of "spatialdata" e.g., it can be a  
 6877 polygon shapefile. Or if you have covariates at each trap location then it will extrapolate  
 6878 to all points on the habitat mask. There's also a method by which the user can create a  
 6879 function, `userDfn`, which seems to provide additional flexibility although we haven't used  
 6880 this method.

6881 There is a handy function `predictDsurface` for producing density maps under the  
 6882 specified model for density.

## 6.6 SUMMARY AND OUTLOOK

6883 In this chapter, we discussed basic concepts related to classical analysis of SCR models  
 6884 based on likelihood methods. Analysis is based on the so-called integrated or marginal  
 6885 likelihood in which the individual activity centers (random effects) are removed from the  
 6886 conditional-on- $s$  likelihood by integration. We showed how to construct the integrated

6887 likelihood and fit some simple models in the **R** programming language. In addition,  
6888 likelihood analysis for some broad classes of SCR models can be accomplished using the  
6889 **R** library **secr** (Efford, 2011) which we provided a brief introduction to. In later chapters  
6890 we provide more detailed analyses of SCR data using likelihood methods and the **secr**  
6891 package.

6892 Why or why not use likelihood inference exclusively? For certain specific models, it  
6893 is probably more computationally efficient to produce MLEs (for an example see Chapt.  
6894 12). However, **BUGS** is extremely flexible in terms of describing models, although it  
6895 sometimes can be quite inefficient. We can devise models in the **BUGS** language easily  
6896 that we cannot fit in **secr**. E.g., random individual effects of various types (Chapt. 8), we  
6897 can handle missing covariates in complete generality and seamlessly, and impose arbitrary  
6898 distributions on random variables. Moreover, models can easily be adapted to include  
6899 auxiliary data types. For example, we might have camera trapping and genetic data and  
6900 we can describe the models directly in **BUGS** and fit a joint model (Gopalswamy et al.,  
6901 2012b). For the MLE we have to write a custom new piece of code for each model or hope  
6902 someone has done it for us, although you should be able to do this with the tools we have  
6903 provided here. Later we consider open population models which are straightforward to  
6904 develop in **BUGS** but, so far, there is no available platform for doing MLE of such models,  
6905 although we imagine one could develop this. On the other hand, likelihood analysis makes  
6906 it easy to do model-selection by AIC and in some cases compute standard errors or carry-  
6907 out goodness-of-fit evaluations.

## MODEL SELECTION AND ASSESSMENT

6911 Our purpose in life is to analyze models. By that, we mean one or more of the following  
6912 basic 4 tasks: estimate parameters, make predictions of unobserved random variables,  
6913 evaluate the relative merits of different models or choosing a best model (model selection),  
6914 and we check whether a specific model appears to provide a reasonable description of our  
6915 data or not (model checking, assessment, or “goodness-of-fit”). In previous chapters we  
6916 addressed the problem of estimation, and also making predictions of latent variables, either  
6917 *s* or derived functions of these variables such as density or population size. In this chapter,  
6918 we focus on the last two of these basic inference tasks: model selection (which model or  
6919 models should be favored), and model assessment (do the data appear to be consistent  
6920 with a particular model?).

6921 In this chapter we develop basic strategies of model selection and model assessment  
6922 or checking using both likelihood methods (as implemented in the **secr** package) and also  
6923 Bayesian analysis. Specifically, for model selection, we discuss the use of AIC and DIC  
6924 for model selection, and also the “indicator model selection” approach of Kuo and Mallick  
6925 (1998). To check model adequacy, or whether a specific model provides a satisfactory  
6926 description of our data set (i.e., “goodness-of-fit”), we rely exclusively on the Bayesian  
6927 p-value framework (Gelman et al., 1996). For checking adequacy of SCR models, part  
6928 of the challenge is coming up with good summaries of model fit, and there does not appear  
6929 much definitive guidance on this in the literature. Following Royle et al. (2011a), we break  
6930 the problem up into 2 components which we attack separately: (1) Conditional  
6931 on the underlying point process, does the encounter model fit? (2) Do the uniformity  
6932 and independence assumptions appear adequate for the point process model of activity  
6933 centers? The latter component of model fit has a huge amount of precedent in the ecological  
6934 literature as it is analogous to the classical problem of testing “complete spatial  
6935 randomness.”

6936 We apply some of these methods to the wolverine camera trapping data first introduced  
6937 in Chapt. 5 to investigate sex specificity of model parameters and whether there is a  
6938 behavioral response to encounter. We note that individuals are drawn to the camera trap  
6939 devices by food bait and therefore it stands to reason that once an individual discovers a

6940 trap, it might be more likely to return subsequently – the response of trap happiness.

## 7.1 STRATEGIES FOR MODEL SELECTION

6941 We review a number of standard methods of model selection that apply to “variable  
6942 selection” problems. That is, when our set of models consists of distinct covariate effects  
6943 and they represent constraints of some larger model achieved by setting some of the  
6944 parameter values to 0. For classical analysis based on likelihood, model selection by AIC  
6945 is the standard approach (Burnham and Anderson, 2002). For Bayesian analysis we rely  
6946 on a number of different methods. We demonstrate the use of the deviance information  
6947 criterion (DIC) (Spiegelhalter et al., 2002) for variable selection problems although we  
6948 recommend against its general use (see below). We use the Kuo and Mallick indicator  
6949 variable selection approach (Kuo and Mallick, 1998) which produces direct statements of  
6950 posterior model probabilities which we think are the most useful, and leads directly to  
6951 model-averaged estimates of density. There is a good review paper recently by O’Hara  
6952 and Sillanpää (2009) that hits on these and many more related ideas for variable selection.  
6953 In addition to O’Hara and Sillanpää (2009) we also recommend Link and Barker (2010,  
6954 Chapt. 7) for general information on model selection and assessment.

### 6955 7.1.1 Scope of the model selection problem

6956 There are two distinct classes of problems that we encounter in SCR models which might  
6957 require some type of model selection or ranking effort: (1) Choosing among models that  
6958 represent distinct, meaningful biological hypotheses; and, (2) choosing among different  
6959 parametric encounter probability models. We believe that the importance of model selec-  
6960 tion depends on which type of problem we have.

6961 **Choosing among biological models:** SCR models that represent extensions of the  
6962 basic null model by including specific covariates or other effects often represent explicit  
6963 biological hypotheses. Examples include models with a behavioral response, or seasonal  
6964 variation in encounter probability, or sex-specificity of model parameters. We anticipate  
6965 that such basic biological factors could be important, and therefore it can be useful to  
6966 choose among (or rank) a set of models that represent these hypotheses.

6967 **Choosing among models for encounter probability:** In Sec. 5.3 we introduced the  
6968 notion that encounter probability models imply specific models of space usage, an idea we  
6969 expand on and generalize in Chapt. 11. Because of this linkage between the model for  
6970 encounter probability and space usage, it is tempting to want to choose among the models  
6971 believing them to be biological models. Our feeling is that the encounter probability  
6972 models are not biological constructs (not motived by biological considerations) but, rather,  
6973 purely phenomenological descriptions of home range. Moreover, as the standard models  
6974 are all stationary and isotropic they are simply unrealistic models. Therefore, it seems to  
6975 us that choosing among a dozen or more arbitrary parametric forms that have no biological  
6976 motivation should tend to lead to an over-fitting. So we will apply ideas of model selection  
6977 to some problems below (and elsewhere in this book) but we avoid the problem of choosing  
6978 among detections functions and we discourage people from doing that.

---

**6979 7.1.2 Model selection by AIC**

6980 Using classical analysis based on likelihood, model selection is easily accomplished using  
 6981 AIC (Burnham and Anderson, 2002) which we demonstrate below. The AIC of a model is  
 6982 simply twice the negative log-likelihood evaluated at the MLE, penalized by the number  
 6983 of parameters ( $np$ ) in the model:

$$AIC = -2\log L(\mathbf{y}|\hat{\theta}) + 2np$$

6984 Models with small values of AIC are preferred. It is common to use a modified AIC  
 6985 referred to as  $AIC_c$  for small sample sizes which is

$$AIC_c = -2\log L(\mathbf{y}|\hat{\theta}) + \frac{2np(np+1)}{n-np-1}$$

6986 where  $n$  is the sample size. Two important problems with the use of  $AIC$  and  $AIC_c$  are  
 6987 that they don't apply directly to hierarchical models that contain random effects, unless  
 6988 they are computed directly from the marginal likelihood (for SCR models we can do this,  
 6989 see Chapt. 6). Moreover, it is not clear what should be the effective sample size  $n$  in  
 6990 calculation of  $AIC_c$ , as there can be covariates that affect individuals, that vary over  
 6991 time, or in space. We do not offer strict guidelines as to when to use a small sample size  
 6992 adjustment.

6993 The R package **secr** computes and outputs AIC automatically for each model fitted  
 6994 and it provides some capabilities for producing a model selection table (function **AIC**) and  
 6995 also doing model-averaging (function **model.average**), which we recommend for obtaining  
 6996 estimates of density from multiple models.

**6997 7.1.3 AIC analysis of the wolverine data**

6998 We provide an example of model selection using **secr** with the wolverine camera trapping  
 6999 data. We consider 4 distinct models to accommodate various types of sex specificity (see  
 7000 Sec. 8.5.4):

7001 Model 1: model SCR0 with constant parameter values for both male and female wolverines but with a parameter  $\psi_{sex}$  the population proportion of males;  
 7002 Model 2: sex-specific intercept  $p_0$  but constant  $\sigma$ ;  
 7003 Model 3: sex-specific  $\sigma$  but constant  $p_0$   
 7004 Model 4: sex-specific  $p_0$  and  $\sigma$ .

7006 The default in **secr** is to model covariates on  $p_0$  on the logit-scale, and covariates on  $\sigma$  on  
 7007 the log-scale which we adopt in our fitting of these models in **BUGS**. Thus, we express  
 7008 models allowing for sex-specificity as follows:

$$\text{logit}(p_{0,i}) = \alpha_0 + \alpha_{sex}\mathbf{sex}_i$$

7009 and

$$\log(\sigma_i) = \log(\sigma_0) + \beta_{sex}\mathbf{sex}_i$$

7010 To model sex-specific abundance (density), we use the multi-session models provided by  
 7011 **secr** (introduced in Sec. 6.5.4), which allows one to model session-specific effects on den-  
 7012 sity, baseline encounter probability,  $p_0$  (labeled  $g_0$  in **secr**), and also the scale parameter

7013  $\sigma$  of the encounter probability model. Using this formulation, we define the “Session”  
 7014 variable to be a sex code and thus session-specific parameters represent sex-specific pa-  
 7015 rameters. For example, if we model session-specific density,  $D$ , then this corresponds to  
 7016 Model 1 in our list above. We note that the use of multi-session models like this suggests  
 7017 one additional model which we haven’t itemized above, that being the model with *no* sex  
 7018 effect on any parameter, which is equivalent to fixing  $\psi_{sex} = 0.5$  instead of estimating it.  
 7019 We will label this last model “Model 0” and include it in our analysis below.

7020 Here are the **R** commands for loading the wolverine data and doing a slight bit of  
 7021 formatting to prepare the data objects for analysis by **secr**. The key difference from our  
 7022 analysis in Chapt. 6 is, here, we grab the wolverine sex information (**wolverine\$wsex**)  
 7023 which is a 0/1 indicator (1=male). We add 1 to that and then use it to define the “Session”  
 7024 variable based on sex. The **R** commands are as follows: **XXXXXX Kimmy test this**  
 7025 **code XXXXXX**

```
7026 ##  

7027 # I KNOW: This needs some comments  

7028 # ANDY: Please add more comments. ha ha  

7029 ##  

7030 ##  

7031 library(secr)  

7032 library(scrbook)  

7033 data(wolverine)  

7034 traps<-as.matrix(wolverine$wtraps)  

7035 dimnames(traps)<-list(NULL,c("trapID","x","y",paste("day",1:165,sep="")))  

7036  

7037 trapfile2<-scr2secr(scrtraps=traps,type="proximity")  

7038  

7039 gr<-as.matrix(wolverine$grid2)  

7040 dimnames(gr)<-list(NULL,c("x","y"))  

7041 gr2<-read.mask(data=gr)  

7042  

7043 wolv.dat<-wolverine$wcaps  

7044 dimnames(wolv.dat)<-list(NULL,c("Session","ID","Occasion","trapID"))  

7045 wolv.dat[,1]<-wolverine$wsex[wolv.dat[,2]]+1  

7046 wolv.dat<-as.data.frame(wolv.dat)  

7047 wolvcapt3<-make.caarthist(wolv.dat,trapfile2,fmt="trapID",noccasions=165)
```

7048 Once the data have been prepared in this way we use the **secr** model fitting function  
 7049 **secr.fit** to fit a number of different models and then the function **AIC** to package the  
 7050 models together and summarize them in the form of an AIC table, with rows of the table  
 7051 ordered from best to worst. The function **model.average** performs AIC-based model-  
 7052 averaging of the parameters specified by the **realnames** variable (below this is demon-  
 7053 strated for the parameter density,  $D$ ). Because this function defaults to averaging by  
 7054 AICc, we slightly modified this function (called **model.average2**) to do model averaging  
 7055 by either AIC or AICc as specified by the user. Together these commands and resulting  
 7056 output (abbreviated to fit on page) look like this:

7057 **XXX**

```

7058 XXX note to reader: I will delete some horizontal output and
7059 XXX put this into a Table somehow XXXXX
7060
7061 model0<-secr.fit(wolvapt3,model=list(D~1, g0~1, sigma~1), buffer=20000)
7062 model1<-secr.fit(wolvapt3,model=list(D~session, g0~1, sigma~1), buffer=20000)
7063 model2<-secr.fit(wolvapt3,model=list(D~session, g0~session, sigma~1), buffer=20000)
7064 model3<-secr.fit(wolvapt3,model=list(D~session, g0~1, sigma~session), buffer=20000)
7065 model4<-secr.fit(wolvapt3,model=list(D~session, g0~session, sigma~session), buffer=20000)
7066
7067 > AIC (model0,model1,model2,model3,model4)
7068          model   detectfn npar    logLik      AIC     AICc dAICc  AICwt
7069 model0           D~1 g0~1 sigma~1 halfnormal   3 -627.2603 1260.521 1261.932 0.000 0.5831
7070 model2           D~session g0~session sigma~1 halfnormal   5 -624.9051 1259.810 1263.810 1.878 0.2280
7071 model1           D~session g0~1 sigma~1 halfnormal   4 -627.2365 1262.473 1264.973 3.041 0.1275
7072 model4 D~session g0~session sigma~session halfnormal   6 -624.6632 1261.326 1267.326 5.394 0.0393
7073 model3           D~session g0~1 sigma~session halfnormal   5 -627.2358 1264.472 1268.472 6.540 0.0222
7074
7075 > model.average (model0,model1,model2,model3,model4,realnames="D")
7076          estimate  SE.estimate       lcl        ucl
7077 session=1 2.707190e-05 7.913577e-06 1.544474e-05 4.745224e-05
7078 session=2 2.927423e-05 8.270402e-06 1.700631e-05 5.039193e-05

```

7079 As usual, estimates and standard errors of the individual model parameters can be  
7080 obtained from the **secr.fit** summary output of any of the **modelX** objects shown above.  
7081 The default output of estimated density is in individuals per ha, so we have to scale this  
7082 up to something more reasonable. To get into units of per 1000 km<sup>2</sup> we need to multiply  
7083 by 100 to get to units of km<sup>2</sup> and then 1000. This produces an estimated density of about  
7084 2.71 for **session=1** (females) and 2.93 for **session=2** (males). We can use the generic **R**  
7085 function **predict** applied to the **secr.fit** output to obtain specific information about the  
7086 MLEs on the natural scale.

7087 We don't necessarily agree with the use of AIC<sub>c</sub> here and think its better to use AIC,  
7088 in general. This is because, as noted previously, it is not clear what the effective sample  
7089 size is for most capture-recapture problems. While we have 21 individuals in the data set,  
7090 most of the model structure has to do with encounter probability samples and for that  
7091 there are 100s of observations. We do note that the AIC and AIC<sub>c</sub> results are not entirely  
7092 consistent. By looking at the best model by AIC, we find that the model with two groups  
7093 (density varying by sex) and sex specific  $g_0$  is preferred (Model 2). This is just slightly  
7094 better than the model with a fixed sex ratio of  $\psi_{sex} = 0.50$ , which is the implied model  
7095 when there is no session effect on density.

7096 We fit the same models but now using a modified state-space which excludes the ocean  
7097 (this is a habitat mask in **secr** terminology). Results are shown in Tab. 7.1 along with the  
7098 previous models without a mask. We see AIC values are smaller for the model without  
7099 the mask. It is probably acceptable to compare these different fits (with and without  
7100 habitat mask) by AIC because we recognize the mask as having the effect of modifying  
7101 the random effects distribution (i.e., of the activity centers, **s**) and the results should be  
7102 sensitive to choice of the distribution for **s**. That said, we may not like the non-mask  
7103 model because it makes sense to exclude the water area from the state-space of **s**. For

**Table 7.1.** AIC Results wolverine data with/without habitat mask. fitted in secr. half-normal encounter probability model. Models ordered by AIC within each class of models (no mask or with mask). Density, D, reported in units of individuals per  $1000\ km^2$ .

NO HABITAT MASK				
model	npar	AIC	AICc	D
D~1 g0~1 sigma~1	3	1260.521	1261.932	
D~session g0~session sigma~1	5	1259.810	1263.810	
D~session g0~1 sigma~1	4	1262.473	1264.973	
D~session g0~session sigma~session	6	1261.326	1267.326	
D~session g0~1 sigma~session	5	1264.472	1268.472	
WITH HABITAT MASK				
model	npar	AIC	AICc	D
D~session g0~session sigma~1	5	1268.096	1272.096	
D~session g0~session sigma~session	6	1268.698	1274.698	
D~1 g0~1 sigma~1	3	1271.163	1272.574	
D~session g0~1 sigma~1	4	1273.115	1275.615	
D~session g0~1 sigma~session	5	1275.089	1279.089	

7104 females the model-averaged density is 3.88 individuals per  $1000\ km^2$  and for males the  
 7105 model-averaged density estimate is 4.46 individuals per  $1000\ km^2$  as we see here:

```
7106 > model.average (model0b,model1b,model2b,model3b,model4b,realnames="D")
7107
7108     estimate   SE.estimate      lcl      ucl
7109 session=1 3.876615e-05 1.189102e-05 2.153795e-05 6.977518e-05
7110 session=2 4.459658e-05 1.323696e-05 2.523280e-05 7.882022e-05
```

7111 This is quite a bit higher than that based on the rectangular state-space (i.e., not  
 7112 specifying a habitat mask). This is not surprising given that **the state-space is part of**  
 7113 **the model** and the specific state-space modification we made here should be extremely  
 7114 important from a biological standpoint.

7115 **XXXXX Kimmy can you tableize this?: XXXXX**

7116 more compact model notation with results.

7117

7118 **XXXXX**

7119 Somehow the estimates need combined with the AIC results (above) into  
 7120 a single table

7121 **XXXXXX**

7122

7123 without mask

model	female?	male?
	D p0 sigma	D p0 sigma
D,g0,\sigma	2.83 0.06 6298.66	2.83 0.06 6298.66
D(sex),g0,\sigma	2.69 0.06 6298.69	2.96 0.06 6298.69
D(sex),g0(sex),\sigma	2.45 0.08 6435.51	3.16 0.04 6435.51
D(sex),g0, \sigma(sex)	2.70 0.06 6280.49	2.95 0.06 6319.03

```

7130 D(sex),g0(sex),\sigma(sex) 2.59 0.08 6080.70 2.99 0.04 6833.16
7131
7132      with mask
7133          female?           male?
7134      model             D   p0   sigma   D   p0   sigma
7135 D, g0, \sigma        4.18 0.05 6282.62 4.18 0.05 6282.62
7136 D(sex),g0,\sigma    3.98 0.05 6282.65 4.38 0.05 6282.65
7137 D(sex),g0(sex),\sigma 3.64 0.07 6382.88 4.73 0.03 6382.88
7138 D(sex),g0,\sigma(sex) 3.93 0.05 6357.26 4.41 0.05 6220.22
7139 D(sex),g0(sex)\sigma(sex) 3.87 0.07 5859.40 4.41 0.03 7039.09

```

## 7.2 BAYESIAN MODEL SELECTION

7140 Model selection is somewhat less straightforward as a Bayesian and there is no canned  
 7141 all-purpose method like AIC. As such we recommend a pragmatic approach, in general,  
 7142 for all problems, based on a number of basic considerations:

- 7143 (1) For a small number of fixed effects we think it is reasonable to adopt a conventional  
 7144 “hypothesis testing” approach – i.e., if the posterior for a parameter overlaps zero  
 7145 substantially, then it is probably reasonable to discard that effect from the model.
- 7146 (2) Calculation of posterior model probabilities: In some cases we can implement methods  
 7147 which allow calculation of posterior model probabilities. One such idea is the indicator  
 7148 variable selection idea from Kuo and Mallick (1998). The idea is introduce a latent  
 7149 variable  $I \sim \text{Bern}(.5)$  and expand the model to include the variable  $I$  as follows:

$$\text{logit}(p_{ijk}) = \alpha_0 + I * \alpha_1 * C_{ijk}.$$

7150 The importance of the covariate  $C$  is then measured by the posterior probability that  
 7151  $I = 1$ .

- 7152 (3) DIC – the Deviance Information Criterion: Bayesian model selection is now routinely  
 7153 carried-out using the Deviance Information Criterion (DIC; Spiegelhalter et al. (2002))  
 7154 although its effectiveness in hierarchical models depends very much on the manner in  
 7155 which it is constructed (Millar, 2009). We recommend using it if it leads to sensible  
 7156 results but we think it should be calibrated to the extent possible for specific classes of  
 7157 models. This has not yet been done in the literature for SCR models, to our knowledge.
- 7158 (4) Logical argument: For something like sex-specificity of certain parameters, it seems  
 7159 to make sense to leave an extra parameter in the model no matter what because, bio-  
 7160 logically, we might expect a difference. (e.g., home range size). In some cases failure to  
 7161 apply logical argument leads to meaningless tests of gratuitious hypotheses (Johnson,  
 7162 1999).

7163 In all modeling activities, as in life itself, the use of logical argument should not be under-  
 7164 utilized.

### 7.2.1 Deviance Information Criterion (DIC)

7165 The availability of AIC makes the use of likelihood methods convenient for problems  
 7166 where likelihood estimation is achievable. For Bayesian analysis, the deviance information

7168 criterion (DIC) seemed like a general-purpose equivalent, at least for a brief period of time  
 7169 after its invention. However, there seems to be many variations of DIC, and a consistent  
 7170 version is not always reported across computing platforms. Even statisticians don't have  
 7171 general agreement on practical issues related to the use of DIC (Millar, 2009). Despite  
 7172 this, it is still widely reported. We think DIC is probably reasonable for certain classes  
 7173 of models that contain only fixed effects, or for which the latent variable structure is the  
 7174 same across models so that only the fixed effects are varied (this covers many SCR model  
 7175 selection problems). However, it would be useful to see some calibration of DIC for some  
 7176 standardized model selection problems.

7177 Model deviance is defined as negative twice the log-likelihood; i.e., for a given model  
 7178 with parameters  $\theta$ :  $\text{Dev}(\theta) = -2 * \log L(\mathbf{y}|\theta)$ . The DIC is defined as the posterior mean  
 7179 of the deviance,  $\text{Dev}(\bar{\theta})$ , plus a measure of model complexity,  $p_D$ :

$$\text{DIC} = \overline{\text{Dev}}(\theta) + p_D$$

7180 The standard definition of  $p_D$  is

$$p_D = \overline{\text{Dev}}(\theta) - \text{Dev}(\bar{\theta})$$

7181 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.  
 7182 Gelman et al. (2004) suggest a different version of  $p_D$  based on one-half the posterior  
 7183 variance of the deviance:

$$p_V = \text{Var}(\text{Dev}(\theta)|\mathbf{y})/2.$$

7184 This is what is produced from **WinBUGS** and **JAGS** if they are run from **R2WinBUGS** or  
 7185 **R2jags**, respectively. It is less easy to get DIC summaries from **rjags**, so we have used  
 7186 **R2jags** in our analyses below.

### 7188 7.2.2 DIC analysis of the wolverine data

7189 We repeated the analysis of the wolverine models with sex-specificity, but this time doing  
 7190 a Bayesian analysis paralleling the likelihood analysis we above in **secr**. We used the  
 7191  $\text{logit}(p_0), \log(\sigma)$  parameterization of the models:

$$\text{logit}(p_{0,i}) = \alpha_0 + \alpha_{\text{sex}} \text{sex}_i$$

7192 and

$$\log(\sigma_i) = \log(\sigma_0) + \beta_{\text{sex}} \text{sex}_i$$

7193 Unlike the multi-session model in **secr**, we carry-out the analysis of the sex-specific  
 7194 model here by putting all of the data into a single data set, and explicitly accounting for  
 7195 the covariate 'sex' in the model by assigning it a Bernoulli prior distribution with  $\psi_{\text{sex}}$   
 7196 being the proportion of males in the population. In this case, we produce "model 0"  
 7197 above, the model with no sex effect on density, by fixing  $\psi_{\text{sex}} = 0.5$ . This parallels our  
 7198 treatment of the ovenbird data in Sec. 9.3 (see also Sec. 8.5.4). As usual, handling of  
 7199 missing values of the sex variable is done seamlessly which might be a practical advantage  
 7200 of Bayesian analysis in situations where sex is difficult to record in the field which may  
 7201 lead to individuals of unknown sex (i.e., missing values). The **BUGS** model specification

for the most complex model, model 4, is shown in Panel 7.1. This model has sex-specific intercept and scale parameter,  $\sigma$ . We provide an **R** script named `wolvSCR0ms` in the `scrbook` package which will fit each model. The function uses **JAGS** by default for the fitting, using the `R2jags` package. The kernel of this function is the model specification in Panel 7.1, which gets modified depending on the model we wish to fit using a command line option `model`. For example, `model = 1` fits the model with constant parameter values for males and females, but sex-specific population sizes (`model = 0` constrains the data augmentation parameter to be constant). The **R** function fits each of the 5 models using a binary indicator variable to turn ‘on’ or ‘off’ each effect. Here is how we obtain the MCMC output for each of the 5 models:

```
7212 toad0<-wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=0)
7213 toad1<-wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=1)
7214 toad2<-wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=2)
7215 toad3<-wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=3)
7216 toad4<-wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=4)
```

We fitted the 5 models to the wolverine data and summarize the DIC computation results in Tab. XXXX (below). Note: I did 2 runs of 21k iters to see how different DIC was, I will average these or do a longer run. The model order (best to worst) was, for run 1: 1, 2, 3, 5, 4 and, for run 2: 1, 2, 4, 5, 3 with the 3, 4 models flipping, but just barely. Conversely with AIC we have 2, 4, 1, 5, 3 and slightly different with AICc (1, 3, 2, 5, 4). What can you believe? Interestingly, if you just look at meanDEV then it matches the AIC result in terms of order. Not sure what to say about all of this. The posterior mean and SD of model parameters under the 5 models are given in Table 7.2.

```
7225 AIC results (from above)
7226 rank      AIC
7227 2 D$\sim\$1 g0$\sim\$1 sigma$\sim\$1          & 3& 1260.521& 1261.932 & \\
7228 4 D$\sim\$session g0$\sim\$1 sigma$\sim\$1          & 4& 1262.473& 1264.973 & \\
7229 1 D$\sim\$session g0$\sim\$session sigma$\sim\$1    & 5& 1259.810& 1263.810 & \\
7230 5 D$\sim\$session g0$\sim\$1 sigma$\sim\$session     & 5& 1264.472& 1268.472 & \\
7231 3 D$\sim\$session g0$\sim\$session sigma$\sim\$session& 6& 1261.326& 1267.326 & \\
7232
7233
7234 NEW DIC RESULTS to PUT IN TABLE BELOW
7235 These are based on 21000 iterations, 3 chains, 1000 discarded, 60k
7236 XXX 12/11/2012 -- need to run longer chains XXXXXXXXXXXXXXX
7237
7238      meandev    pd      DIC
7239      [,1]      [,2]      [,3]   rank
7240 [1,] 440.7358 76.39288 517.1287   1
7241 [2,] 441.6863 77.56654 519.2528   2
7242 [3,] 440.3296 79.32346 519.6531   3
7243 [4,] 443.0906 80.32906 523.4197   5
7244 [5,] 441.1648 79.21601 520.3808   4
7245
```

---

```
alpha.sex ~ dunif(-3,3)
beta.sex ~ dunif(-3,3)
sigma0~dunif(0,50)
alpha0~dnorm(0,.1)
beta<- (1/(2*sigma*sigma) )
psi ~ dunif(0,1)
psi.sex ~ dunif(0,1)
for(i in 1:M){
  wsex[i] ~ dbern(psi.sex)
  w[i]~dbern(psi)
  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]
  beta.vec[i]<- 1/(2*sigma.vec[i]*sigma.vec[i])
  for(j in 1:ntraps){
    mu[i,j]<-w[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( - beta.vec[i]*dd[i,j] )
  }
}
```

---

Panel 7.1: Part of the **BUGS** specification for a complete sex-specificity of model parameters.

---

```

7246 Run 2: (average them?)
7247      [,1]     [,2]     [,3] [,4]
7248 [1,] 440.9120 75.63257 516.5446   1
7249 [2,] 441.6823 78.29509 519.9774   2
7250 [3,] 440.5675 79.58110 520.1486 4/3
7251 [4,] 443.0203 78.16366 521.1840   5
7252 [5,] 441.0890 78.98734 520.0763 3/4

```

**Table 7.2.** The following model outputs report the results of the  $\log(\sigma)$  parameterization (covariates modeled on  $\log(\sigma)$  where  $\sigma$  is the scale parameter of the Gaussian encounter probability model). Results based on 21k iters, 1k burn, 3 chains = 60k iters; all Rhat < 1.01.

parameter	model 0		model 1		model 2		model 3		model 4	
	mean	sd	mean	sd	mean	sd	mean	sd	mean	sd
D 5.82	1.14	5.79	1.14	5.69	1.13	5.76	1.14	5.65	1.14	
N 60.37	11.82	60.02	11.84	58.99	11.69	59.79	11.81	58.57	11.84	
alpha0 -2.81	0.18	-2.82	0.17	-2.44	0.25	-2.81	0.18	-2.42	0.25	
alpha.sex 0.00	1.74	0.00	1.74	-0.75	0.35	0.00	1.73	-0.81	0.35	
beta 1.26	0.21	1.25	0.21	1.18	0.21	1.24	0.29	1.30	0.32	
beta.sex 0.00	1.73	0.01	1.73	0.00	1.73	0.00	0.16	0.09	0.18	
sigma 0.64	0.05	0.64	0.05	0.66	0.06	0.65	0.08	0.64	0.08	
psi 0.30	0.07	0.30	0.07	0.30	0.07	0.30	0.07	0.29	0.07	
psi.sex 0.50	0.29	0.52	0.10	0.56	0.10	0.52	0.11	0.54	0.11	
deviance 440.74	12.36	441.69	12.46	440.33	12.60	443.09	12.68	441.16	12.59	
pD = 76.4		pD = 77.6		pD = 79.3		pD = 80.3		pD = 79.2		
DIC = 517.1		DIC = 519.3		DIC = 519.7		DIC = 523.4		DIC = 520.4		

### 7253 7.2.3 Bayesian model averaging with indicator variables

7254 A convenient way to deal with model selection and averaging problems in Bayesian analysis  
 7255 by MCMC is to use the method of model indicator variables (Kuo and Mallick,  
 7256 1998). Using this approach, we expand the model to include a set of prescribed models  
 7257 as specific reductions of a larger model. This has been demonstrated in some specific  
 7258 capture-recapture models in Royle and Dorazio (2008, sec. 3.4.3), and Royle (2009a) and  
 7259 in the context of SCR by Tobler et al. (2012). A useful aspect of this method is that  
 7260 model-averaged parameters are produced by default. We emphasize the need to be care-  
 7261 ful of reporting model-averaged parameters that don't have a common interpretation in  
 7262 the different models because they are meaningless (averaging apples and oranges....). For  
 7263 example, if a regression parameter is in a specific model then the posterior is informed by  
 7264 the data and a specific MCMC draw is from the appropriate posterior distribution. On  
 7265 the other hand, if the regression parameter is not in the model then the MCMC draw  
 7266 is obtained directly from the prior distribution, and so we need to think carefully about  
 7267 whether it makes sense to report an average of such a thing (in the vast majority of  
 7268 cases the answer is no). But some parameters like  $N$  or density,  $D$ , do have a consistent  
 7269 interpretation and we support producing model-averaged results of those parameters.

7270 To implement the Kuo and Mallick approach, we expand the model to include the  
 7271 latent indicator variables say  $I_m$ , for variable  $M$  in the model, such that

$$I_m = \begin{cases} 1 & \text{linear predictor includes covariate } m \\ 0 & \text{linear predictor does not include covariate } m \end{cases}$$

7272 We assume that the indicator variables  $I_m$  are mutually independent with:

$$I_m \sim \text{Bernoulli}(0.5).$$

7273 The expanded model has the linear predictor:

$$\text{logit}(p_{ijk}) = \alpha_0 + \alpha_1 I_1 C_{1,i} + \alpha_2 I_2 C_{2,ijk}$$

7274 where, let's suppose,  $C_{1,i}$  is an individual level covariate such as sex, and  $C_{2,ijk}$  is a behav-  
 7275 ior response covariate which is individual- trap- and occasion-specific. We can assume a  
 7276 parallel model specification on the parameter  $\sigma$  which is liable to vary by individual level  
 7277 covariates such as sex:

$$\log(\sigma_i) = \beta_0 + \beta_1 I_3 C_{1,i}$$

7278 Using this indicator variable formulation of the model selection problem we can char-  
 7279 acterize unique models by the sequence of  $I$  variables. In this case, each unique sequence  
 7280 ( $I_1, I_2, I_3$ ) represents a model, and we can tabulate the posterior frequencies of each model  
 7281 by post-processing the MCMC histories of ( $I_1, I_2, I_3$ ) (we demonstrate this shortly).

7282 Conceptually, analysis of this expanded model within the data augmentation frame-  
 7283 work does not pose any additional difficulty. One broader, technical consideration is  
 7284 that posterior model probabilities are well known to be sensitive to priors on parameters  
 7285 (Aitkin, 1991; Link and Barker, 2006) and see also Royle and Dorazio (2008, Sec. 3.4.3)  
 7286 and Link and Barker (2010, Sec. 7.2.5). What might normally be viewed as vague or non-  
 7287 informative priors, are not usually innocuous or uninformative when evaluating posterior  
 7288 model probabilities. The use of AIC seems to avoid this problem largely by imposing a  
 7289 specific and perhaps undesirable prior that is a function of the sample size (Kadane and  
 7290 Lazar, 2004). One solution is to compute posterior model probabilities under a model in  
 7291 which the prior for parameters is fixed at the posterior distribution under the full model  
 7292 (Aitkin, 1991). At a minimum, one should evaluate the sensitivity of posterior model  
 7293 probabilities to different prior specifications.

#### 7294 **Analysis of the wolverine data**

7295 Our R script `wolvSCR0ms` in `scrbook` provides the model indicator variable implementation  
 7296 for the fully sex-specific SCR0 model. It is run by setting `model=5` in the function call.  
 7297 We note that it is not very useful to report most parameter estimates from this model  
 7298 because their marginal posterior is a mixture of draws from the prior and draws informed  
 7299 by the data (i.e., from the posterior). On the other hand, the parameters  $N$  and density  
 7300  $D$  should be reported and they represent marginal posteriors over all models in the model  
 7301 set. In effect, model averaging is done as part of the MCMC sampling. The variable 'mod'  
 7302 contains the two binary indicator variables ( $I$  above) which pre-multiply the 'sex' term in  
 7303 each of the  $p_0$  and  $\sigma$  model components, like this:

$$\text{logit}(p_{0,i}) = \alpha_0 + \text{mod}[1]\alpha_{\text{sex}} \text{sex}_i$$

7304 and

$$\log(\sigma_i) = \log(\sigma_0) + \text{mod}[2]\beta_{\text{sex}}\text{sex}_i$$

7305 The third element of `mod` determines whether the  $\psi_{\text{sex}}$  parameter is estimated or fixed at  
 7306  $\psi_{\text{sex}} = 0.5$  which is accomplished with the line of **BUGS** code as follows: `sex.ratio<- psi.sex*mod[3] + .5*(1-mod[3])`  
 7307 The MCMC output for ‘mod’ was post-processed to obtain the model-weights using the  
 7308 following **R** commands:

```
7309
7310 > mod<-toad5$BUGSoutput$sims.list$mod
7311 > mod<-paste(mod[,1],mod[,2],mod[,3],sep="")
7312 >
7313 > table(mod)
7314 mod
7315   000   001   010   011   100   101   110   111
7316 17181  4935  1057   296 25211   8337  2275    708
7317
7318 > round( table(mod)/length(mod) , 3)
7319 mod
7320   000   001   010   011   100   101   110   111
7321 0.286  0.082  0.018  0.005  0.420  0.139  0.038  0.012
7322
```

7323 We see that the best model is that with sex-specific baseline encounter probability  $p_0$ , and  
 7324 with  $\psi_{\text{sex}} = 0.5$ , which has posterior model weight of 0.42, the model with no sex effect  
 7325 has posterior probability 0.286 and the remaining posterior mass is distributed over the  
 7326 other six models. We could arrive at a qualitatively similar conclusion using a more ad  
 7327 hoc approach based on looking at the posterior mass for each parameter under each of  
 7328 the full model (model 4; see Tab. 7.2, in part). Considering the sex-specific intercept,  
 7329 it appears to be very important as its posterior mass is mostly away from 0. On the  
 7330 other hand, the coefficient on log-sigma is concentrated around 0, and the estimated  
 7331  $\psi_{\text{sex}}$  (probability that an individual is a male) is 0.54 with a large posterior standard  
 7332 deviation. We might therefore be inclined to discard the sex effect on  $\log(\sigma)$  based on  
 7333 classical thinking-like-a-hypothesis-testing-guy and settle for the model with a sex-specific  
 7334 intercept. This is consistent with our indicator variable approach which found that model  
 7335 (1,0,0) has posterior probability of 0.420. So we’re not really misled too much in looking  
 7336 at the posteriors for each parameter. We can obtain model-averaged estimates from the  
 7337 indicator variable approach, which produces direct model-averaged estimates of  $N$  and  $D$ :

```
7338   mu.vect sd.vect   2.5%    25%    50%    75%   97.5% Rhat n.eff
7339 D     5.695  1.133  3.759  4.916  5.591  6.362   8.193 1.002  3600
7340 N    59.077 11.758 39.000 51.000 58.000 66.000  85.000 1.002  3600
```

7341 We obtain a model-averaged estimate (posterior mean) for density of  $D = 5.695$  which  
 7342 is hardly any different from our model specific estimates (Tab. 7.2) and, in particular,  
 7343 from model 2 which has only a sex-specific intercept.

**7.2.4 Choosing among detection functions**

7344 XXXX Note: not sure this is working so well right now – having trouble  
 7345 getting consistent answers between WinBUGS and JAGS XXXX

7347 Another approach to implementing model indicator variables is to introduce a categorical  
 7348 model identity variable which, i.e., so that “model identity” is itself a parameter  
 7349 of the model and each distinct model is associated with a unique covariate combination or  
 7350 other set of model features. This is convenient especially when we cannot specify the linear  
 7351 predictor as some general model that reduces to various alternative sub-models simply by  
 7352 switching binary variables on or off. In the context of SCR models, choosing among different  
 7353 encounter probability models would be an example. For this case we do something like this  
 7354 `mod dcat(probs[])` where `probs` is a vector with elements  $1/(\#models)$ , and  
 7355 the encounter probability matrix is filled-in depending on the value of `mod`. In particular,  
 7356 instead of a 2-dimensional array `p[i,j]`, we build `p[i,j,l]` for each of  $l = 1, 2, \dots, L$   
 7357 models. An example with 3 distinct models is:

```
7358 ##  

7359 ## Fill-in p[,] for each model:  

7360 ##  

7361 p[i,j,1] <- p0[i,1]*exp( - beta.vec[i,1]*dist2[i,j] )  

7362 p[i,j,2] <- 1-exp(-p0[i,2]*exp( - beta.vec[i,2]*dist2[i,j] ) )  

7363 logit(p[i,j,3])<- p0[i,3] - beta.vec[i,3]*dist2[i,j]  

7364  

7365 mu[i,j]<-w[i]*p[i,j,mod]  

7366 y[i,j]~ dbin(mu[i,j],K[j])
```

7367 As before the posterior probabilities can be highly sensitive to priors on the different  
 7368 model parameters and sometimes mixing is really poor. We provide an **R/JAGS** script  
 7369 (`wolvSCR0ms2`) in the `scrbook` package which has an example. In this script, there are  
 7370 3 different encounter probability models: The Gaussian encounter probability, Gaussian  
 7371 hazard, and logistic model with the square of distance. The key things to note are that  
 7372 there are 3 intercepts and 3 different ‘beta’ parameters (the coefficient on distance). The  
 7373 parameters should not be regarded as equivalent across the models, so it is important to  
 7374 have them separately defined (and estimated) for each model. In our analysis we used a  
 7375 vague normal prior (precision = 0.1) for the “intercept” parameter (either log or logit-scale  
 7376 of baseline encounter probability  $p_0$ ) and a `Uniform(0,5)` prior for one-half the inverse of  
 7377 the coefficient on distance-squared. In the BUGS model specification the priors look like  
 7378 this:

```
7379 for(i in 1:3){  

7380   alpha0[i] ~ dnorm(0,.1)  

7381   sigma[i] ~ dunif(0,5)  

7382   beta[i]<- 1/(2*sigma[i]*sigma[i])  

7383 }
```

7384 Then, we create a probability of encounter for each individual, trap *and* model so that  
 7385 the holder object ‘`p`’ in the model description is a 3-dimensional array. (sometimes this  
 7386 would have to be a 4 or 5-d array in more complex models with time effects, etc..), so that  
 7387 construction of the encounter probability models look like this:

```

7388 p[i,j,1]      <- p0[i,1]*exp( - beta[1]*dist2[i,j] )
7389 p[i,j,2]      <- 1-exp(-p0[i,2]*exp( - beta[2]*dist2[i,j] ) )
7390 logit(p[i,j,3])<- p0[i,3] - beta[3]*dist2[i,j]

```

7391 where

```

7392 logit(p0[i,1])<- alpha0[1]
7393 log(p0[i,2])<- alpha0[2]
7394 p0[i,3]<- alpha0[3]

```

7395 Results of fitting the multiple-encounter probability model to the wolverine camera trapping data are summarized as follows:

7397 **XXXXX CHECKING RESULTS NOW (12/17/2012) XXXXXXXXXX**

7398 WinBUGS results:

7399 10 chains , 1000 burn then 1000 post burn-in

7400

7401 > table(toad\$sims.list\$catmod)

7402

	1	2	3
4144	3484	2372	

7405 > table(toad\$sims.list\$catmod)/10000

7406

	1	2	3
0.4144	0.3484	0.2372	

7409

7410 Hardly a difference in density

7411 > tapply(toad\$sims.list\$D,toad\$sims.list\$catmod,mean)

7412       1         2         3

7413 5.858606 6.032470 5.770849

7414       = 488.1

7416

7417 % XXXX We need more practical advice like this

7418 The model averaged density: Convergence is not pretty, but fuck it

7419 > print(toad,digits=3)

7420 Inference for Bugs model at "modelfile5.txt", fit using WinBUGS,

7421 10 chains, each with 2000 iterations (first 1000 discarded)

7422 n.sims = 10000 iterations saved

	mean	sd	2.5%	25%	50%	75%	97.5%	Rhat	n.eff
7424 catmod	1.823	0.788	1.000	1.000	2.000	2.000	3.000	1.052	120
7425 beta.sex	-0.016	1.715	-2.842	-1.496	-0.029	1.460	2.846	1.001	10000
7426 alpha.sex	0.030	1.721	-2.851	-1.452	0.063	1.522	2.849	1.001	10000
7427 psi	0.506	0.107	0.314	0.430	0.499	0.576	0.734	1.009	650
7428 psi.sex	0.523	0.103	0.321	0.451	0.524	0.595	0.720	1.001	6700
7429 beta[1]	1.123	0.487	0.088	0.840	1.213	1.450	1.921	1.015	550
7430 beta[2]	1.111	0.506	0.083	0.760	1.206	1.476	1.919	1.019	330
7431 beta[3]	1.067	0.529	0.075	0.649	1.152	1.465	1.939	1.006	1000

---

```

7432 alpha0[1] -1.318 2.551 -5.169 -2.875 -2.566 0.126 4.837 1.057 130
7433 alpha0[2] -0.966 2.905 -5.477 -2.883 -2.269 0.984 5.794 1.053 120
7434 alpha0[3] -0.602 3.076 -5.933 -2.846 -1.117 1.468 6.110 1.021 270
7435 N 61.191 12.012 40.000 53.000 60.000 69.000 88.000 1.010 590
7436 D 5.898 1.158 3.856 5.109 5.784 6.651 8.483 1.010 590
7437 beta[1] 1.123 0.487 0.088 0.840 1.213 1.450 1.921 1.015 550
7438 beta[2] 1.111 0.506 0.083 0.760 1.206 1.476 1.919 1.019 330
7439 beta[3] 1.067 0.529 0.075 0.649 1.152 1.465 1.939 1.006 1000
7440 mod[1] 0.496 0.500 0.000 0.000 0.000 1.000 1.000 1.001 7700
7441 mod[2] 0.507 0.500 0.000 0.000 1.000 1.000 1.000 1.001 10000
7442 deviance 440.864 12.476 418.800 432.200 440.000 448.800 467.600 1.008 760
7443
7444 For each parameter, n.eff is a crude measure of effective sample size,
7445 and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
7446
7447 DIC info (using the rule, pD = var(deviance)/2)
7448 pD = 0.1 and DIC = 1.2
7449 DIC is an estimate of expected predictive error (lower deviance is better).

```

### 7.3 EVALUATING GOODNESS-OF-FIT

7450 In practical settings, we estimate parameters of a desirable model, or maybe fit a bunch of  
 7451 models and report estimates from all of them or a model-averaged summary of density. An  
 7452 important question is: Is our model worth a shit? In other words, does the model appear  
 7453 to be an adequate description of our data? Formal assessment of model adequacy or  
 7454 goodness-of-fit is a challenging problem and there are not all-purpose algorithms for doing  
 7455 this in either frequentist or Bayesian paradigms. Moreover, there are some philosophical  
 7456 challenges to evaluating model fit, such as, if we do model averaging then should all of  
 7457 the models have to fit? Or should the averaged model have to fit? What if none of the  
 7458 models fit? We don't know the answers to these questions and we won't try to answer  
 7459 them. Instead, we will provide what guidance we can on taking the first steps to evaluating  
 7460 model fit, of a single model, as if it were a cherished family heirloom of great importance.  
 7461 We suggest that if you have a model that you really like, a single model, then it is a  
 7462 sensible thing to check that the model is a good fit to your data. If it is not, we do not  
 7463 imagine that the model is useless but just that some thought should be put into why the  
 7464 model doesn't fit so that, perhaps, some remediation might happen as future data are  
 7465 collected. After all, you may have spent 2, 3 or many more years of your life collecting  
 7466 that data set, perhaps thousands of hours, and therefore it seems a reasonable proposition  
 7467 to expect to do some estimation and analysis of the model regardless of model fit. You  
 7468 can still learn something from a model that does not pass some technical litmus test of  
 7469 model fit.

7470 Conceptually, we can think of evaluation model fit as follows: if we simulate data under  
 7471 the model in question, do the simulated realizations resemble the data set that we actually  
 7472 have? For either Bayesian or classical inference, the basic strategy to assessing model fit is  
 7473 to come up with a fit statistic that depends on the parameters and the data set, which we  
 7474 denote by  $T(\mathbf{y}, \theta)$ , and then we compute this for the observed data set, and then compare

7475 its value to that computed for perfect data sets simulated under the correct model. In  
7476 the case of classical inference, we will often rely on the standard practice of parametric  
7477 bootstrapping (Dixon, 2002), where we simulate data sets conditional on the MLE  $\hat{\theta}$  and  
7478 compare realizations with what we've observed. The R package **unmarked** (Fiske and  
7479 Chandler, 2011) contains generic bootstrapping methods for all of the hierarchical models  
7480 fitted, including distance sampling (e.g., see Sillett et al., 2012, for an application). In  
7481 simple cases, using classical inference methods, it is sometimes possible to identify a test  
7482 statistic of theoretical merit, perhaps with a known asymptotic distribution. Examples  
7483 from the closed capture-recapture setting includes **XXXXXX REF XXXXXXXX**.  
7484 For Bayesian analysis we use a similar idea referred to as the Bayesian p-value (Gelman  
7485 et al., 1996). Using this approach, data sets are simulated for posterior samples of  $\theta$  (we  
7486 introduced the Bayesian p-value in sec. 3.9.1) and some fit statistic for the simulated data  
7487 sets, usually based on the discrepancy from the observed data from its expected values, is  
7488 compared to that for the actual data. In most cases, whether Bayesian or frequentist, the  
7489 main idea for assessing model fit is the same: We compare data sets from the model we're  
7490 interested in with the data set we have in hand. If they appear to be consistent with one  
7491 another, then our faith in the model increases, at least to some extent, and we say "the  
7492 model fits."

7493 To date, we are unaware of any goodness-of-fit applications based on likelihood anal-  
7494 ysis of SCR models (**XXX WE NEED TO DO SOME RESEARCH ON THIS**  
7495 **XXXXXX**) although the approach is standard in distance sampling **XXX REF HERE**  
7496 **??? XXXX**. Similarly, for Bayesian analysis of SCR models, there has not been a defini-  
7497 tive or general proposal for a fit statistic or even a class of fit statistics, although a few  
7498 specialized implementations of Bayesian p-values have been provided (Royle, 2009a; Royle  
7499 et al., 2011a; Gopalaswamy et al., 2012b,a; Russell et al., 2012). While we universally  
7500 adopt the Bayesian p-value approach, and suggest some fit statistics in the following text,  
7501 we caution that there is not general expectation to support how well they should do. As  
7502 such, one might consider doing some kind of custom evaluation or calibration when using  
7503 such methods, if the power of the test (ability to reject under specific departures from the  
7504 model) is of paramount interest. We note that this uncertain power or performance of  
7505 the Bayesian p-value is not a weakness of the Bayesian approach because the same issue  
7506 applies in using bootstrap approaches applied to classical analysis of models, if we were  
7507 to devise such methods.

## 7.4 THE TWO COMPONENTS OF MODEL FIT

7508 For most SCR models, there are at least two distinct components of model fit, and we  
7509 propose to evaluate these two distinct components individually. First, we can ask, does  
7510 the model explain the *observation* process, conditional on the underlying point process?  
7511 We can evaluate this based on the encounter frequencies of individuals *conditional* on  
7512 (posterior samples of) the underlying point process  $s_1, \dots, s_N$ . We discuss some potential  
7513 fit statistics for addressing this in the next section. Second, we can evaluate whether the  
7514 data appear consistent with the "uniformity" assumption about the point process. For  
7515 the simple model of independence and uniformity, this is similar to the assumption of  
7516 *complete spatial randomness* (CSR) which we consider in sec. 7.4.1 below. Actually, this  
7517 is not strictly the assumption of CSR because the binomial assumption on  $N$  under

7518 data augmentation, so we instead use the term *spatial randomness*.

#### 7519 7.4.1 Testing Uniformity or Spatial Randomness

7520 Historically, especially in ecology, there has been an extraordinary amount of interest in  
 7521 whether a realization of a point process indicates “complete spatial randomness,” i.e.,  
 7522 that the points are distributed uniformly and independently in space. A good reference  
 7523 for such things is Cressie (1991, Ch. 8) and Illian et al. (2008)<sup>1</sup> In the context of animal  
 7524 capture-recapture studies, the spatial randomness hypothesis is manifestly false, purely  
 7525 on biological grounds. Typically individuals will be clustered or more uniform (for ter-  
 7526 ritorial species) than expected under spatial randomness and heterogeneous habitat will  
 7527 generate the appearance of clustering even if individuals are distributed independently of  
 7528 one another. While we recommend modeling spatial structure explicitly when possible  
 7529 (Chaps. 13, 12, 11), the uniformity assumption may be a reasonable approximation to  
 7530 truth in some situations.

7531 The basic technical framework for evaluating the spatial randomness hypothesis is  
 7532 based on counts of activity centers in cells or bins. For that we use any standard goodness-  
 7533 of-fit test statistic, based on gridding (binning) the state-space of the point process into  
 7534  $g = 1, 2, \dots, G$  cells or bins, and we tabulate  $N_g \equiv N(\mathbf{x}_g)$  the number of activity centers  
 7535 in bin  $g$ , centered at coordinate  $\mathbf{x}_g$ . Specifically, let  $B(\mathbf{x})$  indicate a bin centered at  
 7536 coordinate  $\mathbf{x}$ , then  $N(\mathbf{x}) = \sum_{i=1}^N I(\mathbf{s}_i \in B(\mathbf{x}))$  is the population size of bin  $B(\mathbf{x})$ . In sec.  
 7537 5.10.2, we used the summaries  $N(\mathbf{x})$  for producing density maps from MCMC output.  
 7538 Here, we use them for constructing a fit statistic. We prefer to use the Freeman-Tukey  
 7539 statistic of this form:

$$T(\mathbf{N}, \theta) = \sum_g (\sqrt{N_g} - \sqrt{\mathbb{E}(N_g)})^2$$

7540 where  $\mathbb{E}(N_g)$  is estimated by the mean bin count. An alternative conventional assessment  
 7541 of fit is based on the following statistic: Conditional on  $N$ , the total number of activity  
 7542 centers in the state-space  $\mathcal{S}$ , the bin counts  $N_g$  should have a binomial distribution. It will  
 7543 usually suffice to approximate the binomial cell counts by Poisson cell counts, in which  
 7544 case we can use the classical “index-of-dispersion” test (Illian et al., 2008, p. 87):

$$I = (\#cells - 1) * s^2 / \bar{N}$$

7545 where  $s^2$  is the sample variance of the bin counts and  $\bar{N}$  is the sample mean. When the  
 7546 point process realization is *observed*, as in classical point pattern modeling (but not in  
 7547 SCR), this statistic has approximately a Chi-square distribution on ( $\#cells - 1$ ) degrees-  
 7548 of-freedom under the spatial randomness hypothesis. If  $s^2/\bar{N} > 1$ , clustering is suggested  
 7549 whereas,  $s^2/\bar{N} < 1$  suggests the point process is too regular.

7550 Whatever statistic we choose as our basis for assessing spatial randomness, the im-  
 7551 portant technical issue is that we don’t observe the point process and so the standard  
 7552 statistics for evaluating spatial randomness cannot be computed directly. However, using  
 7553 Bayesian analysis, we do have a posterior sample of the underlying point process and

---

<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). XXXX  
 Kimmy look this up make sure still active put access date here XXXXX.

so we suggest computing the posterior distribution of any statistic in a Bayesian p-value framework. For a given posterior draw of all model parameters,  $N$  is known, based on the value of the data augmentation variables  $z_i$ , and so we can obtain a posterior sample of  $N(\mathbf{x})$  by taking all of the output for MCMC iterations  $m = 1, 2, \dots$ , and doing this:

$$N(\mathbf{x})^{(m)} = \sum_{z_i^{(m)}=1} I(\mathbf{s}_i^{(m)} \in B(\mathbf{x}))$$

Thus,  $N(\mathbf{x})^{(1)}, N(\mathbf{x})^{(2)}, \dots$  is the Markov chain for the derived parameter  $N(\mathbf{x})$ .

In addition to computing the bin counts for each iteration of the MCMC algorithm, at the same time we generate a realization of the activity centers  $\mathbf{s}_i$  under the spatial randomness model, and we obtain bin counts for these “new” data,  $\tilde{N}(\mathbf{x})$ . For each of the posterior samples – that of the real data, and that of the posterior simulated data, we compute the fit-statistic. Our fit statistic based on the actual data:

$$T(\mathbf{N}, \theta) = \sum_x (\sqrt{N(\mathbf{x})} - \sqrt{\bar{N}(\mathbf{x})})^2$$

a fit statistic based on a simulated realization of points under the spatial randomness hypothesis:

$$T(\tilde{\mathbf{N}}, \theta) = \sum_x (\sqrt{\tilde{N}(\mathbf{x})} - \sqrt{\bar{N}(\mathbf{x})})^2$$

And we compute the Bayesian p-value by talling up the proportion of times that  $T(\tilde{\mathbf{N}}, \theta)$  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 **SCRgof** in our package **scrbook** will do this, given the output from **JAGS** (see below).

### Sensitivity to bin size

Evaluating fit based on bin counts in point process models are sensitive to the number of bins (Illian et al., 2008, p. 87-88). This is related to the classical problem of fit testing for binary regression because in a point process model, as the number of grid cells gets small, the grid cell counts go to 0 or 1 and standard fit statistics (e.g., based on deviance or Pearson residuals) are known not to be very useful. There is some good discussion of this in McCullagh and Nelder (1989, sec. 4.4.5)<sup>2</sup>. What it boils down to is, using the example of the Pearson residual statistic considered by McCullagh and Nelder (1989), the fit statistic is exactly a deterministic function of the sample size only, which clearly should not be regarded as useful for model fit. This is why, in order to do a fit check, one must always aggregate the data in some fashion. In the context of testing spatial randomness, the test statistic we described above has us bin the region  $\mathcal{S}$  into a bunch of bins and tally up  $N_g$  the frequency of activity centers in bin  $g$ . Suppose that we choose the bin size to be extremely small such that  $\mathbb{E}(N_g)$  tends to  $N/G$  ( $N$  being the number of activity centers). Further,  $N_g$  tends to a binary outcome. Therefore the fit statistic has  $N$  components that represent the  $N_g = 1$  values and it has  $G - N$  components that represent the  $N_g = 0$  values. So the fit statistic resembles this:

$$T(\mathbf{N}, \theta) = \sum_{g \ni N_g=1}^N (1 - \sqrt{N/G})^2 + \sum_{g \ni N_g=1}^{G-N} (N/G)^2 = N(1 + (G - N)/G)$$

---

<sup>2</sup>Thanks to M. Kéry for pointing this out to us.

(note  $\exists$  is used here to mean “such that”). If  $G$  is huge relative to  $N$ , then we see that this tends to about  $2 * N$ , which does not provide any meaningful assessment of model fit. So if you look at this in the limit in which the bin counts become binary, the fit statistic loses all its variability to the specific model used and is just a deterministic function of  $N$ . As a practical matter, it probably makes sense to restrict the number of bins to *fewer* than the number of observed individuals in the sample size. In SCR applications this will therefore result, usually, in very large (and few) bins.

There are some extensions that help resolve the issue of sensitivity to bin size. We can construct fit statistics based not just on quadrat counts but also the neighboring quadrat counts – this is the Greig-Smith method (Greig-Smith, 1964). In addition, there are a myriad of “distance methods” for evaluating point process models, and we believe that many of these can (and will) be adapted to SCR models. Again the main feature is that the point process for which inference is focused on is completely latent in SCR models – so this makes the fit assessment slightly different than in classical point processes. That said, the methods should be adaptable e.g., in a Bayesian p-value kind of way.

#### 7601 **Sensitivity to state-space extent**

An issue that we have not investigated is that any model assessment that applies to a *latent* point process is probably sensitive to the size of the state-space. As the size of the state-space increases then the cell counts (far away from the data) *are* independent binomial counts with constant density, and so we can overwhelm the fit statistic with extraneous “data” simulated from the posterior, which is equal to the prior as we move away from the data, and therefore uninformed by the actual data in the vicinity of the trap array. Therefore we recommend computing these fit statistics in the vicinity of the trap array only. For example, if typical trap spacing is, say, 10 km, then the bins used to obtain the observed and predicted activity centers should not extend any further from the traps than 5 km.

#### 7612 **7.4.2 Assessing Fit of the Observation Model**

It is less clear how to approach goodness-of-fit evaluation of the observation model. For most SCR problems, we have a 3-dimensional data array of *binary* observations,  $y_{ijk}$ , and, as discussed in the previous section, we need to construct fit statistics based on observed and expected frequencies that are aggregated in some fashion. In practice, the data will be way too sparse to have much power if the data are not highly aggregated. We recommend focusing on summary statistics that represent aggregated versions of  $y_{ijk}$  over 1 or 2 of the dimensions. We describe 3 such fit statistics below. We recognize that, depending on the model, some information about model fit will be lost by summarizing the data in this way. For example if there is a behavioral response and you aggregate over time to focus on the individual and trap level summaries then some information about lack of fit due to temporal structure in the data is lost.

**Fit statistic 1: individual x trap frequencies** We summarize the data by individual and trap-specific counts  $y_{ij}$ , aggregated over all sample occasions. Conditional on  $s_i$ , the expected value under any encounter model is:

$$\mathbb{E}(y_{ij}) = p_{ij} K$$

(or  $K_j$  if the traps are operational for variable periods). If there is time-varying structure to the model, then expected values would have to be computed according to  $\mathbb{E}(y_{ij}) = \sum_k p_{ijk}$ . Then we can define a fit statistic from the Freeman-Tukey residuals according to:

$$T_1(\mathbf{y}, \theta) = \sum_i \sum_j (\sqrt{y_{ij}} - \sqrt{\mathbb{E}(y_{ij})})^2$$

where we use  $\theta$  here to represent the collection of all parameters in the model. This is conditional on  $\mathbf{s}$  as well as on the data augmentation variables  $\mathbf{z}$ . We compute this statistic for *each* iteration of the MCMC algorithm for the observed data set and also for a new data set simulated from the posterior distribution, say  $\tilde{\mathbf{y}}$ .

We could also use a similar fit statistic derived from summarizing over traps to obtain an  $n_{\text{ind}} \times K$  matrix of count statistics. We imagine that either summary of the data will probably be too disaggregated in most practical settings.

**Fit statistic 2: Individual encounter frequencies.** SCR models represent a type of model for heterogeneous encounter probability, like model  $M_h$ , but with an explicit factor (space) that explains the heterogeneity. For model  $M_h$ , the individual encounter frequencies are the sufficient statistic for model parameters, and so it makes intuitive sense to provide some kind of omnibus fit assessment of the core heuristic that SCR model is adequately explaining the heterogeneity using a model  $M_h$  like statistic based on individual encounter frequencies. So, we build a fit statistic based on the individual total encounters (Russell et al., 2012),  $y_i = \sum_j \sum_k y_{ijk}$ . In addition, the expected value is a similar summary over traps and occasions such as this:  $\mathbb{E}(y_i) = \sum_j \sum_k p_{ijk}$

$$T_2(\mathbf{y}, \theta) = \sum_i (\sqrt{y_i} - \sqrt{\mathbb{E}(y_i)})^2$$

We imagine this test statistic should provide an omnibus test of extra-binomial variation and should therefore capture some effect of variable exposure to encounter of individuals, although we have not carried-out any evaluations of power under specific alternatives. Obviously, in using this statistic, we lose information on departures from the model that might only be trap or time-specific.

**Fit Statistic 3: Trap frequencies.** We construct an analogous statistic based on aggregating over individuals and replicates to form trap encounter frequencies:  $y_j = \sum_i \sum_k y_{ijk}$  (Gopalaswamy et al., 2012b) and the expected value is a similar summary over individuals and occasions:  $\mathbb{E}(y_j) = \sum_i \sum_k p_{ijk}$ . Then

$$T_3(\mathbf{y}, \theta) = \sum_j (\sqrt{y_j} - \sqrt{\mathbb{E}(y_j)})^2$$

This seems like a sensible fit statistic because we can think of SCR models as spatial models for counts (Chandler and Royle, In press). Therefore, we should seek models that provide good predictions of the observable spatial data which are the trap totals. In this context, it might even make sense to pursue cross-validation based methods for model selection. Cross-validation is a standard method of evaluating models such as in kriging or spline smoothing, so we could as well develop such ideas based on the trap-specific frequencies.

---

### 7.4.3 Does the SCR model fit the wolverine data?

7662 We use the ideas described in the previous section to evaluate goodness-of-fit of the SCR  
 7663 model to the wolverine camera trapping data which we have analyzed in previous chapters.

7664 We consider first whether the simple model of spatial randomness of the activity centers  
 7665 is adequate. We think that the encounter model shouldn't have a large effect on whether  
 7666 the spatial randomness assumption is adequate or not, so we fit "model 0" (parameters  
 7667 are *not* sex specific) using an **R** script provided in the function `wolvSCR0gof` which will  
 7668 default to fitting the model in **JAGS**. This is the same script as `wolvSCR0ms` except that  
 7669 it saves the MCMC output for the activity centers `s` and the data augmentation variables  
 7670 `z`, which are required in order to compute the test of spatial randomness.

7671 The MCMC output from a **BUGS** run is processed with an **R** function `SCRgof` which  
 7672 computes the test of spatial randomness based on quadrat counts and the Bayesian p-value  
 7673 calculation. The function `SCRgof` requires a few things as inputs: (1) the output from a  
 7674 **BUGS** run (in particular, the activity center coordinates and the data augmentation  
 7675 variables); (2) the number of bins to create for computing spatial frequencies of activity  
 7676 centers; (3) the buffer around the trap array to use. This buffer could be the buffer used  
 7677 for the state-space in the model fitting but we think it should be relatively tighter to the  
 7678 trap array. So we're using 10km grid cells (1 unit = 10 km) and so we used the buffer =  
 7679 0.4 for computing the GoF even though buffer = 2 was used in the fitting; (4) the trap  
 7680 locations. The R code to fit the model and obtain the goodness-of-fit result is as follows:

```
7682 toad1<-wolvSCR0gof(nb=1000,ni=6000,buffer=2,M=200,model=0)
7683 bugsout<- toad1$BUGSoutput$sims.list
7684
7685 traplocs<-wolverine$wtraps[,2:3]
7686 traplocs[,1]<-traplocs[,1] -min(traplocs[,1])
7687 traplocs[,2]<-traplocs[,2]- min(traplocs[,2])
7688 traplocs<-traplocs/10000
7689
7690 set.seed(2013) # set seed so Bayesian p-value is the same each time
7691
7692 SCRgof(bugsout,5,5,traplocs=traplocs,buffer=.4)
7693
7694 Cluster index observed: 1.099822
7695 Cluster index simulated: 1.000453
7696 P-value index of dispersion: 0.408
7697 P-value2 freeman-tukey: 0.6842667
```

7700 The output produced is the ratio of the variance to the mean cluster index (see above),  
 7701 which is computed as the posterior mean index of dispersion for the latent point process,  
 7702 and also the average value for simulated data. If this value is > 1 then clustering is  
 7703 suggested, which we see a minor amount of evidence for here. Two Bayesian p-values are  
 7704 produced: the first is based on the cluster index, and the 2nd is based on the Freeman-  
 7705 Tukey statistic calculated as described in Sec. 7.4.1. Because our p-values aren't too  
 7706 close to 0 or 1, we judge that the model of spatial randomness provides an adequate fit to

7707 the data. You can verify that a similar result is obtained if we use the model with fully  
 7708 sex-specific parameters (model 4).

7709 Next, we did a Bayesian p-value analysis of the observation component of the model,  
 7710 using the 3 fit statistics described in Sec. 7.4.2. These statistics can be calculated as  
 7711 part of the **BUGS** model specification or by post-processing the MCMC output returned  
 7712 from a **BUGS** run. The **R** script `wolvSCR0gof` contains the relevant calculations. For  
 7713 example, to compute fit statistic 1, we have to add some commands to the **BUGS** model  
 7714 specification such as this (note: this is only a fraction of the model specification):

```

7715 .....
7716 for(j in 1:ntraps){
7717   mu[i,j]<-w[i]*p[i,j]
7718
7719   y[i,j]~ dbin(mu[i,j],K[j])
7720   ynew[i,j] ~ dbin(mu[i,j],K[j])
7721
7722   err[i,j]<- pow(pow(y[i,j],.5) - pow(K[j]*mu[i,j],.5),2)
7723   errnew[i,j]<- pow(pow(ynew[i,j],.5) - pow(K[j]*mu[i,j],.5),2)
7724 }
7725
7726 X1obs<-sum(err[,])
7727 X1new<-sum(errnew[,])
7728 .....

```

7729 Similar calculations are carried-out to obtain the posterior samples of test statistics 2  
 7730 (individual totals) and 3 (trap totals). For the wolverine data, the Bayesian p-value  
 7731 calculations produce:

```

7732 > mean(toad1$BUGSoutput$sims.list$X1new>toad1$BUGSoutput$sims.list$X1obs)
7733 [1] 0
7734
7735 > mean(toad1$BUGSoutput$sims.list$X2new>toad1$BUGSoutput$sims.list$X2obs)
7736 [1] 0.17
7737
7738 > mean(toad1$BUGSoutput$sims.list$X3new>toad1$BUGSoutput$sims.list$X3obs)
7739 [1] 0.02066667

```

7740 The results suggests a general lack of fit of the observation model although based  
 7741 just on statistic 2 we might conclude that the model is adequate for explaining individual  
 7742 heterogeneity. A similar result is obtained using the fully sex-specific model. The result  
 7743 suggests that the trap totals are too heterogeneous than suggested by the null model  
 7744 alone. We note that one individual was captured 8 times in one traps which is pretty  
 7745 extreme under a model which assumes independent Bernoulli trials. We summarize that  
 7746 the trap-counts simply are not well-explained by this model.

7747 In attempt to resolve this probelm, we extended the model to include a behavioral  
 7748 response (following Royle et al. (2011b)) which can be fitted using the sample **R** script  
 7749 `wolvSCRMb`. To fit a model using **WinBUGS**, and then compute the Bayesian p-values  
 7750 we do this:

---

```

7751 > wolv.Mb<-wolvSCRMb(nb=1000,ni=6000,buffer=2,M=200)
7752
7753 > mean(wolv.Mb$sims.list$X1new>wolv.Mb$sims.list$X1obs)
7754 [1] 0.9666667
7755
7756 > mean(wolv.Mb$sims.list$X2new>wolv.Mb$sims.list$X2obs)
7757 [1] 0.3644667
7758
7759 > mean(wolv.Mb$sims.list$X3new>wolv.Mb$sims.list$X3obs)
7760 [1] 0.4990667

```

7761 Given that this model seems to fit better, we might prefer reporting estimates under this  
 7762 model which we do in Tab. XXXXXX. (below. TO DO.). Estimated density is about 1  
 7763 individual higher per 1000 km<sup>2</sup> compared with the various models that lack a behavioral  
 7764 response. It might be useful to try these fit assessment exercises using the clipped state-  
 7765 space as described in Sec. 5.9. That takes a long long time to run in BUGS though,  
 7766 especially for the behavioral response model.

```

7767 NEW TABLE
7768 > print(wolv.Mb,digits=2)
7769 Inference for Bugs model at "modelfile.txt", fit using WinBUGS,
7770   3 chains, each with 6000 iterations (first 1000 discarded)
7771   n.sims = 15000 iterations saved
7772      mean     sd    2.5%    25%    50%    75%   97.5% Rhat n.eff
7773 psi       0.36  0.10   0.20   0.29   0.35   0.41   0.58 1.00  2600
7774 sigma     0.88  0.13   0.68   0.79   0.86   0.95   1.17 1.00   730
7775 p0        0.01  0.00   0.01   0.01   0.01   0.01   0.02 1.01   530
7776 N         71.32 19.07  42.00  58.00  69.00  82.00 114.02 1.00  2100
7777 D         6.87  1.84   4.05   5.59   6.65   7.90  10.99 1.00  2100
7778 beta      0.69  0.19   0.37   0.55   0.67   0.81   1.10 1.00   730
7779 alpha2    2.50  0.27   1.99   2.31   2.50   2.68   3.04 1.00   700
7780 X1obs     54.71 6.12   43.69  50.41  54.39  58.63  67.47 1.00  3900
7781 X1new     64.73 7.62   50.93  59.46  64.39  69.55  80.96 1.00  3900
7782 X2obs     13.93 4.07   7.25   11.02  13.53  16.36  23.04 1.00  5700
7783 X2new     12.65 3.35   6.93   10.27  12.36  14.69  20.07 1.00  2000
7784 X3obs     12.80 1.74   9.80   11.58  12.64  13.85  16.61 1.00  2400
7785 X3new     12.94 3.05   7.77   10.78  12.67  14.81  19.58 1.00 15000
7786 deviance  1128.02 15.38 1101.00 1117.00 1127.00 1138.00 1161.00 1.00   640
7787
7788 For each parameter, n.eff is a crude measure of effective sample size,
7789 and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
7790
7791 DIC info (using the rule, pD = var(deviance)/2)
7792 pD = 117.9 and DIC = 1245.9
7793 DIC is an estimate of expected predictive error (lower deviance is better).

```

## 7.5 SUMMARY AND OUTLOOK

7794 In this chapter, we offered some general strategies for model selection and model checking  
7795 – or assessment of model fit. We think the strategies we outlined for model selection are  
7796 fairly standard and probably will be reasonably effective as applied to SCR models. Some  
7797 technical issues of Bayesian analysis need to be addressed (in general) before Bayesian  
7798 methods are more generally useful and accessible. For one thing, Bayesian model selection  
7799 based on the indicator variable approach of Kuo and Mallick (1998) can be tediously slow  
7800 even for small data sets, and so improved computation will improve our ability to do  
7801 Bayesian model selection in practical situations. Also, and most importantly, sensitivity  
7802 to prior distributions is an important issue. Further research and practice might identify  
7803 preferred prior configurations for SCR that provide a good calibration in relevant model  
7804 selection problems.

7805 Our suggestions for Bayesian model assessment, or goodness-of-fit checking, should  
7806 be viewed as experimental, based on mostly ad hoc attempts in a number of published  
7807 applications (Royle et al., 2009a, 2011a; Gopalaswamy et al., 2012b; Russell et al., 2012,  
7808 e.g.). We offered up a framework based on independent testing of the spatial model of  
7809 independence and uniformity, and testing fit of the observation model conditional on the  
7810 underlying point process. While we think this general strategy should be fruitful, we know  
7811 of no studies on the power to detect various model departures. We have not discussed  
7812 assessment of model fit for SCR models using likelihood methods, although we imagine  
7813 that standard bootstrapping ideas should be effective, perhaps based on the fit statistics  
7814 (or similar ones) we suggested here for computing Bayesian p-values.

7815 Clearly there is much research to be done on assessment of model fit in SCR models.  
7816 For testing the spatial randomness hypothesis, we used a classical quadrat count approach,  
7817 in which point locations are put into spatial bins, although other approaches from spatial  
7818 point process modeling should be pursued including nearest-neighbor methods or distance-  
7819 based metrics. In addition, studies to evaluate the power to detect interesting departures  
7820 from the standard assumptions, and the robustness of inferences about  $N$  or density, need  
7821 to be conducted. It seems reasonable to expect that most of these Bayesian p-value tests  
7822 will have lower power in typical data sets consisting of a few to a few dozen individuals. As  
7823 such, failure to detect a lack of fit may not be that meaningful. But, on the other hand, it  
7824 may not make a difference in terms of density estimates either. We think inference about  
7825 density should be relatively insensitive to departures from spatial randomness, because we  
7826 get to observe direct information on some component of the population – so a component  
7827 of density is *observed* – and, for those activity centers, the assumed model of the point  
7828 process should exert little over placement of the activity centers. Conversely, as is the case  
7829 with classical closed population models (Dorazio and Royle, 2003; Link, 2003), inferences  
7830 should be somewhat more sensitive to bad-fitting models for the observation process.



7831  
7832

# 8

7833

---

## MODELING ENCOUNTER PROBABILITY

### 8.1 INTRODUCTION

7834 In previous chapters we showed how to fit basic spatial capture-recapture models using  
7835 Bayesian analysis (in **WinBUGS** or **JAGS**; Chapt. 5) or by classical likelihood methods  
7836 (Chapt. 6 or using **secr**). We covered a suite of possible encounter models (e.g., the Bi-  
7837 nomial, Poisson, and Multinomial) for dealing with different types of sampling. We have  
7838 not, however, considered different detection functions or covariates that can affect the  
7839 parameters of the detection function, including those that may arise from the individual  
7840 or the trap device. In practice, investigators are invariably concerned with explicit factors  
7841 or covariates that might influence variation in parameters. Such covariates include time  
7842 (e.g., day of year, or season), behavior (e.g., is there an effect of trapping on subsequent  
7843 capture probabilities), sex of the individual, and trap type (e.g., various camera types, or  
7844 different constructions for hair snares). Traditionally, in the non-spatial capture recapture  
7845 literature, such models were called “model  $M_t$ ”, “model  $M_h$ ”, or “model  $M_b$ ”, identifying  
7846 models that account for variation in detection probability as a function of time, “individ-  
7847 ual heterogeneity” or “behavior”, where behavior describes whether or not an individual  
7848 had been previously captured. In SCR models, more complex covariate models are possi-  
7849 ble because we might also have trap-specific covariates, or covariates that vary spatially  
7850 over the landscape, and because we generally have more than one parameter describing  
7851 the detection function: Most detection functions include a baseline encounter rate ( $\lambda_0$ ) or  
7852 probability ( $p_0$ ) parameter, and a shape parameter ( $\sigma$ ), which takes on different interpre-  
7853 tations depending on the specific encounter probability function under consideration.

7854 In this chapter, we generalize the basic SCR model to accommodate both alternative  
7855 detection functions as well as many different kinds of covariates. We focus on the binomial  
7856 encounter model used throughout Chaps. 5 and 6 and the Gaussian (“half-normal”)  
7857 detection function (although we do show you some alternatives to the half-normal), but  
7858 the extension to other encounter and detection models is straightforward. Specifically, we  
7859 consider three distinct types of covariates - those which are fixed, partially observed or  
7860 completely unobserved (latent). Fixed covariates are those that are fully observed; for

example, the date of all sampling occasions. Partially observed covariates are those which are not known for all observations; for example, the sex of an individual cannot always be determined from photos taken during camera trapping. Even if we are able to observe the sex of all individuals sampled, we cannot know it for those individuals never observed during the study. And finally, unobserved covariates are those which we cannot observe at all, for example, the home range size of individuals, or unstructured random “individual effects”.

We will see that models containing these different types of covariates are relatively easy to describe in **WinBUGS** or **JAGS**, and therefore to analyze using Bayesian analysis of the joint likelihood based on data augmentation thus providing a coherent and flexible framework for inference for all classes of SCR models. Throughout the chapter, we will continue to develop the analysis of the black bear study introduced in Chapt. 4, using the software **JAGS**. We also consider the likelihood analysis of many of these models; to do so, we will demonstrate the use of the **R** package **secr** and how to do model comparison with AIC (this section is at the end of the chapter 8.7). There are other types of covariates that we do *not* cover in this chapter; for example, covariates that vary across the landscape might affect density and we consider these covariates in Chapt. 13. Alternatively, these landscape covariates might affect the way individuals use space. There are probably very few circumstances under which animals use all space equally and we develop more realistic models of encounter probability in which covariates affect space usage in Chapt. 12.

## 8.2 DETECTION FUNCTIONS

In Chapt. 5, we developed a basic spatial capture recapture model using a standard distance function based on the kernel of a normal (Gaussian) probability distribution:

$$p_{ij} = p_0 \exp(-\alpha_1 * ||\mathbf{s}_i - \mathbf{x}_j||^2)$$

where  $||\mathbf{s}_i - \mathbf{x}_j||$  is the distance between  $\mathbf{s}_i$  and  $\mathbf{x}_j$  and

$$\alpha_1 = 1/(2 * \sigma^2).$$

We argued (sec. 5.3) that this model corresponds to an explicit model of space usage – namely, that individual locations are draws from a bivariate normal distribution. We also mentioned that other detection models are possible, including a logit model of the form:

$$\text{logit}(p_{ij}) = \alpha_0 + \alpha_1 ||\mathbf{s}_i - \mathbf{x}_j||. \quad (8.2.1)$$

However, there's nothing preventing us from constructing a myriad of other models for encounter probability. The most commonly used detection functions are also those used in the distance sampling literature: the half-normal (Gaussian), the hazard, and the negative exponential. The negative exponential model is:

$$p_{ij} = p_0 * \exp(-\alpha_1 * ||\mathbf{s}_i - \mathbf{x}_j||)$$

or we could use the general power model (Russell et al., 2012) which includes both the Gaussian and exponential models as special cases:

$$p_{ij} = p_0 * \exp(-\alpha_1 * ||\mathbf{s}_i - \mathbf{x}_j||^\theta)$$

7893 or the hazard rate model (Hayes and Buckland, 1983) of which the Gaussian hazard rate  
 7894 model which we have considered previously is a special case:

$$p_{ij} = 1 - \exp(-\lambda_0 * \exp(-\alpha_1 * ||\mathbf{s}_i - \mathbf{x}_j||^2)).$$

7895 The **R** package **secr** allows the user to access 12 different detection models, of which  
 7896 some are only used for simulating data (see 8.1). These detection functions can also be  
 7897 implemented in **R**, **WinBUGS**, **JAGS** etc.

7898 Insofar as all these detection functions are symmetric and stationary, they are pretty  
 7899 crude descriptions of space usage by real animals. But this is not to say they are inadequate  
 7900 descriptions of the data and, as we discuss in Chaps. 12 and 11 we can use them as the  
 7901 basis for producing more realistic models of space usage.

**Table 8.1.** 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** but this 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`). XXXXXXXXXXXX probably need to change greek letters beta to alpha to be consistent with our notation. Maybe some other changes too? Beth sez: I thought maybe it makes sense to leave these different because Murray implements them differently than we do in some cases and I don't think it's fair to say they are identical to our parameters under all situations. XXXXXXXXXXXX

Code	Name	Parameters	Function
0	half-normal	$g_0, \sigma$	$g(d) = g_0 * \exp\{-d^2/(2\sigma^2)\}$
1	hazard rate	$g_0, \sigma, z$	$g(d) = g_0 * (1 - \exp(-(d/\sigma)^2 - z))$
2	exponential	$g_0, \sigma$	$g(d) = g_0 * \exp(-d/\sigma)$
3	compound half-normal	$g_0, \sigma, z$	$g(d) = g_0 * [1 - \{1 - \exp(-d^2/(2\sigma^2))\}]^z$
4	uniform	$g_0, \sigma$	$g(d) = g_0, d \leq \sigma;$ $g(d) = 0, \text{ otherwise}$
5	w exponential	$g_0, \sigma, w$	$g(d) = g_0, d < w;$ $g(d) = g_0 \exp(-(d-w)/\sigma), \text{ otherwise}$
6	annular normal	$g_0, \sigma, w$	$g(d) = g_0 * \exp(-(d-w)^2/(2\sigma^2))$
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	$\beta_0, \beta_1, sdS$	$g(d) = 1 - F[\{c - (\beta_0 + \beta_1 * d)\}/sdS]$
11	signal strength spherical	$\beta_0, \beta_1, sdS$	$g(d) = 1 - F[\{c - (\beta_0 + \beta_1 * (d - 1) - 10 * \log10(d^2))\}/sdS]$

7902 By changing the detection function and the specification of  $\alpha_1$ , we can basically create  
 7903 any distance function for the data. It is important to note that  $\sigma$  is not comparable under  
 7904 these different distance functions for detection and should not be regarded as "home range  
 7905 radius" in general. While there is generally a relationship between  $\sigma$  and home range size,  
 7906 that relationship varies depending on the model under consideration. We demonstrate  
 7907 how to fit different distance functions in the Bayesian framework here, and then provide  
 7908 a section on the likelihood analysis (in **secr**) in a separate section below.

### 8.3 BAYESIAN ANALYSIS WITH BEAR.JAGS

7909 To demonstrate how to incorporate various types of covariates into models for encounter  
 7910 probability using **JAGS**, we return to the data from the Fort Drum bear study. This  
 7911 data set was first introduced in Chapt. 4, but, to refresh your memory, there were 38  
 7912 baited hair snares that were operated between June and July 2006. The snares were  
 7913 checked each week for a total for  $K = 8$  sample occasions and  $n = 47$  individual bears  
 7914 were encountered at least once. The data are provided in the **R** package **scrbook** and an  
 7915 **R** function called **bear.JAGS** allows the user to easily pick which model to analyze. The  
 7916 function **bear.JAGS** will set up the data, write the model, define the MCMC specifications  
 7917 (e.g., initial values, etc.) and, finally, run the selected model in **JAGS**. In addition to  
 7918 choosing which model to run, the user can also specify the number of chains, iterations and  
 7919 length of the burn-in phase. Calling the function will provide all the code to implement  
 7920 the models independently as well. In the following sections we will present the model code  
 7921 and output for the most commonly employed models; for all analyses we ran 3 chains with  
 7922 a burn-in of 500 iterations and 20000 saved iterations.

### 8.4 BAYESIAN ANALYSIS OF DETECTION FUNCTIONS

7923 In panel 8.1, we present the basic SCR model and show how to specify the negative  
 7924 exponential detection function. To call each of these from the function **bear.JAGS** set  
 7925 **model='SCR0'** or **model='SCRexp'** in the function call, respectively.

7926 Applying the SCR model with half-normal distance function provides an estimate  
 7927 of  $D = 0.167$  bears per  $km^2$  and with the negative exponential distance function the  
 7928 estimate is virtually the same  $D = 0.167$ . In distance sampling, the use of different  
 7929 distance functions often results in very different estimates of density (especially when  
 7930 using the negative exponential function). There are two main reasons why the different  
 7931 distance function may have less of an impact on the density estimates under the SCR  
 7932 models. First, we can estimate the baseline detection parameter ( $p_0$ ). In most distance  
 7933 sampling models, detection at distance 0 is set to 1. In Table 8.2, the posterior mean of  
 7934  $p_0$  is 0.11 under the half-normal distance model and 0.34 under the negative exponential  
 7935 model. The larger baseline detection under the negative exponential reduces the impact of  
 7936 the having “no shoulder”. Secondly, the distance function in SCR is related to ‘movement’  
 7937 of individuals and the data collected are on individuals, which is mo

7938 (which we have alot more information on than in distance sampling), not the whole  
 7939 detection process, so the lack of a shoulder doesn’t impact the density estimation as much.

7940 the shape of the distance function in this case, with  $\sigma = 1.99$  and 1.12 for the half-  
 7941 normal and negative exponential function, is very different between the two models.

7942 the distance function here is governing ‘movement’ of individuals (which we have alot  
 7943 more information on than in distance sampling), not the whole detection process, so the  
 7944 lack of a shoulder doesn’t impact the density estimation as much.

7945 We can also see that the 97.5% percentile for  $N$  is 628 (), thus not reaching our  
 7946  $M = 650$  value, but close enough that we may want to check that  $N$  is not truncated by  
 7947 this level of data augmentation. In addition to checking the percentiles, we can also see  
 7948 from ??, the histogram of  $N$ , that  $N$  shows no signs of truncation. We could also increase  
 7949  $M$  as another check that our data augmentation is sufficient.

---

```
model {
alpha0 ~ dnorm(0,.1)
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(Xl,Xu)
s[i,2] ~ dunif(Yl,Yu)
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])
# p[i,j]<- z[i]*p0*exp(- alpha1*d[i,j]) #exponential distance function
}
}
N<-sum(z[])
D<-N/area
}
```

---

Panel 8.1: **JAGS** model specification for a basic SCR model with Gaussian distance function and the alternative exponential distance function.

**Table 8.2.** Posterior summaries of parameter estimates from different SCR models for the Fort Drum black bear data.

Parameter	Mean	SD	2.5	97.5
$\text{SCR}_0$				
$D$	0.17	0.022	0.122	0.207
$N$	500.63	66.652	371	628
$p_0$	0.11	0.014	0.081	0.135
$\psi$	0.77	0.104	0.566	0.966
$\sigma$	1.99	0.131	1.762	2.275
$\text{SCR}_{exp}$				
$D$	0.17	0.023		
$N$	511.55	68.730		
$p_0$	0.34	0.055		
$\psi$	0.79	0.107		
$\sigma$	1.12	0.094		

7950 A very important consideration when using different distance functions is the interpretation of  $\sigma$ . The estimate of  $\sigma$  under the negative exponential model is 1.12, which  
 7951 is distinct from our estimate of  $\sigma$  under the half normal model,  $\sigma = 1.996$ . The interpretation of  $\sigma$  in the two models is really quite distinct. In the normal model it can be  
 7952 interpreted as the standard deviation of a bivariate normal movement model whereas the  
 7953 manner in which  $\sigma$  relates to “area used” for the negative exponential model has nothing  
 7954 to do with a bivariate normal model of movement. This highlights that it is important for  
 7955 the user to know what distance function is used and what the interpretation of  $\sigma$  might  
 7956 be in relation to the home range size. This relationship was discussed in 5  
 7957

7958 We leave the detection functions for now and move onto incorporating covariates into  
 7959 the model using the **JAGS** language. For this part, we will stick with the half-normal  
 7960 distance model shown in the panel 8.1 above.  
 7961

## 8.5 MODELING COVARIATE EFFECTS

7962 The basic strategy for modeling covariate effects is to include them on the baseline en-  
 7963 counter rate or probability parameter,  $p_0$  (or  $\lambda_0$ ), or the scale parameter of the encounter  
 7964 model,  $\sigma$  or, in some cases, both parameters.

7965 Broadly speaking, we recognize (here) 3 types of covariates. Fixed covariates which are  
 7966 fully observable and might vary by trap alone (e.g., type of trap, baited or not, disturbance  
 7967 regime, even habitat), sample occasion (e.g., day of season or weather conditions), or both  
 7968 (e.g., behavior, weather - if over a large region). Another class of covariates are those  
 7969 which vary at the level of the individual (and possibly also over time). As a technical  
 7970 matter, and as noted before, these are different from fixed covariates because we cannot  
 7971 see all of the individuals and the covariates are almost always incompletely observed (if  
 7972 at all). The lone exception is the behavioral response to capture which is known for all  
 7973 individuals, captured or not (an animal never captured/observed has never been captured  
 7974 before). We noted many times before that space itself (i.e., the activity centers) is a  
 7975 type of individual covariate and this notion actually helped us derive the fully spatial  
 7976 capture-recapture model from the traditional, non-spatial model (Chapt. 5). 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 alone there are a suite of models for capturing latent heterogeneity.

**Table 8.3.** Examples of different covariate classifications.

Covariate type	Examples
individual	sex, age, home range
trap	baited/not, habitat (see also Chapter 11)
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,

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * C_{ijk}$$

$$p_{ijk} = p_{0,ijk} \exp(-\alpha_1 * ||\mathbf{s}_i - \mathbf{x}_j||^2)$$

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 8.3 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 various types in the following sections.

### 8.5.1 Date and Time

Often, researchers are interested in the effect of date on the encounter probability. For example, in a long term hair snare study, we may expect that seasonal shedding (Wegan et al., 2012) will influence encounter probabilities directly. Or we may expect behaviors such as denning, mating, etc. to influence the encounter of certain species at certain times of year (Kéry et al., 2011). There are two common ways to incorporate date or time information into a model for encounter probability. For cases with a small number of sampling occasions we can fit a time-specific intercept (analogous to “model  $M_t$ ” in classical capture-recapture (Otis et al., 1978)). That is, there are  $K$  sampling occasion-specific parameters to reflect potential variation in sampling effort or other factors that might vary across samples. Alternatively, we can model parametric functions of date or time such as polynomial or sinusoidal functions.

8008 In the first case, we allow each sampling occasion,  $k$ , to have its own baseline encounter  
 8009 probability, e.g.,

$$\text{logit}(p_{0,k}) = \alpha_{0,k}$$

8010 so that

$$p_{ijk} = p_{0,k} \exp(-\alpha_1 * ||\mathbf{s}_i - \mathbf{x}_j||^2).$$

8011 This description of the model includes  $k$  occasion-specific baseline detection probabilities.  
 8012 Thus, if we had 4 sampling occasions, we would have 4 different baseline detection  
 8013 probabilities. We imagine that complete time-specificity of  $p_0$  (i.e., one distinct value for  
 8014 each sample occasion) would be most useful in situations where we have just a few sam-  
 8015 pling occasions (if we have many, this formulation will dramatically increase the number  
 8016 of parameters to be estimated) or we do not expect systematic patterns over time (e.g.,  
 8017 explainable by a polynomial function).

8018 To implement this in **JAGS**,  $\alpha_0$  has to be estimated for each time period  $k$  either  
 8019 using an index vector or dummy variables (as described in Chapt. 2 and sec. 4.3) and this  
 8020 can be done by only changing only a few lines in panel 8.1.

```
8021 alpha0[k] ~ dnorm(0,.1)
8022 logit(p0[k])<- alpha0[k]
8023 .....
8024 .....
8025 y[i,j,k] ~ dbin(p[i,j,k],K)
8026 p[i,j,k]<- z[i]*p0[k]*exp(- alpha1*d[i,j]*d[i,j])
```

8027 Since the model estimates a parameter effect for each time period, the encounter  
 8028 histories must be time-dependent. Thus, a 3-d data array (called **bearArray** in our code),  
 8029 with dimensions **nind** × **ntraps** × **nreps** is required. In addition to using the 3-d data array,  
 8030 the initial values must be updated so that there are  $K$  values generated for  $\alpha_0$ . And finally,  
 8031 this means that we have to put in another nested for loop in the code to account for the  $K$   
 8032 sample occasions. A side note: the computation time will increase quite a bit (this model  
 8033 for the bear data may take up to 15 hours or more on your machine to obtain a sufficient  
 8034 posterior sample).

8035 Running this model with the function **bear.JAGS** by setting **model=SCRt**, returns esti-  
 8036 mates of density similar to those from the model without covariates (see Table 8.4), but  
 8037 now we have a characterization of variation in encounter probability over time. Encounter  
 8038 probability seems to increase for the first few time periods before stabilizing around 0.14,  
 8039 dropping off again at the end of the study. The differences in encounter probability from  
 8040 the first time periods to the others might actually be due to something like a behavioral  
 8041 response (see below) or possibly seasonal differences in the efficiency of the sampling tech-  
 8042 nique. Researchers have found that hair snares are more effective at different times of the  
 8043 year (even within season) due to shedding (Wegan et al., 2012). In this particular exam-  
 8044 ple, our density estimates are similar to the base model, likely because the differences in  
 8045 detection between occasion were not that large. In a longer term study or in one with  
 8046 greater variation in the detection probability, the implication of such differences might  
 8047 have a bigger impact on the estimates of density and  $\sigma$ .

8048 Fitting an occasion specific time effects model might not be the most appropriate  
 8049 for all scenarios (and could require the estimation of many parameters if we had many

**Table 8.4.** Posterior summaries of parameter estimates from a SCR model with time-dependent baseline detection for the Ft. Drum black bear data set.

Parameter	Mean	SD	2.5	97.5
$D$	0.17	0.02	0.13	0.21
$N$	509.24	66.13	381	632
$p_0(t = 1)$	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
$\psi$	0.78	0.10	0.58	0.97
$\sigma$	1.96	0.12	1.73	2.22

8050 sampling occasions, take the wolverine example from Chapt. 5.8 where there were 165  
 8051 daily sampling occasions). In some cases, such as the wolverine study, we might sample  
 8052 over longer time frames and expect the date to have a strong impact on encounter for  
 8053 a variety of reasons. For example, if we have camera traps running for an entire year  
 8054 and we expect mating behavior or denning behavior to change the behavioral patterns of  
 8055 individuals. Instead of fitting a model with  $K$  baseline encounter probabilities, we can  
 8056 include date as a linear (or quadratic, ...) effect. An example can be found in Kéry et al.  
 8057 (2011) who incorporated a day of year covariate, both as a linear and a quadratic effect,  
 8058 into their SCR model of European wildcats; the data had been collected over a year long  
 8059 period and cat behavior was expected to vary seasonally thus influencing the probability  
 8060 of encounter. In these cases, we would specifically incorporate day of year (Date) as a  
 8061 continuous covariate as:

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * \text{Date}_k \\ p_{ijk} = p_{0,ijk} \exp(-\alpha_1 * ||\mathbf{s}_i - \mathbf{x}_j||^2)$$

8062 or a quadratic effect of day-of-year:

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * \text{Date}_k + \alpha_3 * \text{Date}_k^2 \\ p_{ijk} = p_{0,ijk} \exp(-\alpha_1 * ||\mathbf{s}_i - \mathbf{x}_j||^2)$$

8063 where the variable Date is an integer coding of day-of-year, indexed to some arbitrary  
 8064 start point in time.

### 8065 8.5.2 Trap-specific covariates

8066 In some studies it makes sense to model encounter probability as a function of local or  
 8067 trap-specific covariates. These can be one of two types: genuine trap covariates that  
 8068 describe the trap or encounter site, such as whether a trap is baited or not, or how many  
 8069 traps were set at a sampling location, or what kind of bait was used, etc.. Or they can  
 8070 be local covariates that describe the likelihood that an animal would use the habitat in

8071 the vicinity of the trap (see Chapt. 11 for more on this situation). We imagine that  
 8072 these covariates, of either type, should affect baseline encounter probability. For example,  
 8073 Sollmann et al. (2011) found a large difference in the encounter probability of jaguars due  
 8074 to traps being located on roads, which the animals were using to travel along, as opposed  
 8075 to traps placed off of roads. In this case, the trap type is a binary variable - on/off road,  
 8076 (another binary variable could be baited/non-baited). We can write this such that:

$$\text{logit}(p_{0,j}) = \alpha_{0,\text{type}_j}$$

$$p_{ijk} = p_{0,j} \exp(-\alpha_1 * ||\mathbf{s}_i - \mathbf{x}_j||^2).$$

8077 Here, we use an indicator variable, “type”, that will be a numeric value for the trap-  
 8078 specific covariate. Thus for our example of on/off road, we would have  $\text{type}_j = 1$  if trap  
 8079  $j$  is on a road and  $\text{type}_j = 2$  otherwise, and we would estimate two separate  $\alpha_{0,\cdot}$  - one  
 8080 for on-road and one for off-road cameras. This general set up also allows for more than  
 8081 2 categories, say if 3 or 4 different camera models were used in a study, we would just  
 8082 need more dummy variables to allow for estimation of the different encounter rates (i.e.,  
 8083 the intercept). Note that instead of modeling several intercepts, we could also specify the  
 8084 model in terms of effects of trap type; in this case, for the 2-category example above, we  
 8085 would specify our “type” vector as  $\text{type}_j = 0$  if trap  $j$  is on a road and  $\text{type}_j = 1$  otherwise,  
 8086 and write our model such that

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 \text{type}_j$$

8087 Now,  $\alpha_0$  is the baseline detection probability (on the logit scale) for traps with  $\text{type}_j = 0$   
 8088 and  $\alpha_2$  is the effect on baseline detection of a trap being of type = 1. While these models  
 8089 yield identical results, sometimes one parameterization might work better than the other  
 8090 in **WinBUGS** or **JAGS** (Kéry, 2010).

### 8091 8.5.3 Behavior or Trap Response by Individual

8092 One of the most basic of encounter models is that which accommodates a change in  
 8093 encounter probability as a result of initial encounter. This is colloquially referred to as  
 8094 “trap happiness” or “trap shyness”, or in other words, a behavioral response of individuals  
 8095 to being captured (Otis et al., 1978). If a trap is baited with a food source, an individual  
 8096 might come back for more. On the other hand, if being captured is traumatic then an  
 8097 individual might learn to avoid traps. Both of these types of responses can occur in  
 8098 most species depending on the type of encounter mechanisms being employed. Moreover,  
 8099 behavioral response can be either global (Gardner et al., 2010b) or local (Royle et al.,  
 8100 2011b). The local response is a trap-specific response while a global response suggests that  
 8101 initial capture provides a net increase or decrease in subsequent probabilities of capture  
 8102 (across all traps). A behavioral response must not be enduring (i.e., persist for the entire  
 8103 study after the individual has been captured/observed for the first time) but can also be  
 8104 ephemeral, if, for example, an animal only avoids a trap on the occasion immediately after  
 8105 it was captured (Royle, 2008; Yang and Chao, 2005). While we will focus the examples  
 8106 in this chapter on enduring behavioral effects, extending such a model to the case of an  
 8107 ephemeral response should not pose any difficulties.

8108 To describe these behavioral models we need to create a binary matrix that indicates  
 8109 if an individual has been captured previously. For the global behavioral response, define

8110 the  $n \times K$  matrix,  $\mathbf{C}$  where  $C_{ik} = 1$  if individual  $i$  was captured at least once prior to  
 8111 session  $k$ , otherwise  $C_{ik} = 0$ .

$$\text{logit}(p_{0,ik}) = \alpha_0 + \alpha_2 * C_{ik}$$

$$p_{ijk} = p_{0,ik} \exp(-\alpha_1 * \|\mathbf{s}_i - \mathbf{x}_j\|^2)$$

8112 For the local behavioral response, which is trap specific, we create an array,  $C_{ijk}$ , that  
 8113 indicates if an individual  $i$  has been previously captured in trap  $j$  at time  $k$ . We then  
 8114 include this in the model in the exact same form as above (with the sole difference that  
 8115 both  $C$  and  $p$  are now also indexed by  $k$ ):

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * C_{i,j,k}$$

$$p_{ijk} = p_{0,ijk} \exp(-\alpha_1 * \|\mathbf{s}_i - \mathbf{x}_j\|^2)$$

8116 Since the behavioral response is occasion specific, to implement either the local or  
 8117 global response model in **JAGS**, we will have to use the 3-d array of the capture histories  
 8118 (`nind×ntraps×nreps`) as we did for the time-varying encounter probability model above.  
 8119 The code must loop over each sampling occasion, but otherwise, the model varies only a  
 8120 little from the basic SCR model shown in panel 8.1. Here is the specification of the the  
 8121 occasion specific ( $k$ ) loop:

```
8122 for(k in 1:K){  

  8123   logit(p0[i,j,k])<- alpha0 + alpha2*C[i,j,k]  

  8124   y[i,j,k] ~ dbin(p[i,j,k],1)  

  8125   p[i,j,k]<- z[i]*p0[i,j,k]*exp(- alpha1*d[i,j]*d[i,j])  

  8126 }
```

8127 Despite the minor changes to the **BUGS** code, this model can require quite a bit of  
 8128 time and computational effort to carry out the behavior response models. Implementing  
 8129 the behavioral models with the function `bear.JAGS` by setting `model=SCRb` or `model=SCRB`  
 8130 returns the results, shown in Table 8.5. There is a strong global behavior response sug-  
 8131 gested by the posterior mean of  $\alpha_2 = 0.90$ . The estimate of  $N$  and subsequently  $D$  are  
 8132 larger than under the non behavioral models, here we estimate  $N = 577.56$  and in the  
 8133 SCR0 model, we estimated  $N = 500$ . This makes sense given the large estimate of  $\alpha_2$ ,  
 8134 which suggests that bears are trap happy. In situations where animals are trap happy, we  
 8135 tend to over estimate detection (i.e., the bears that are never observed have a lower detec-  
 8136 tion probability than those that have been captured in the study) and thereby reduce the  
 8137 estimate of  $N$ . We do not include the results here, but the estimates were similar under  
 8138 the local behavioral response model, although the estimate of  $N$  was smaller and there  
 8139 was a larger recapture effect at the local level.

#### 8140 8.5.4 Individual Covariates

8141 Individual covariates are those which are measured (or measurable) on individuals, so  
 8142 we get to observe them only for the captured individuals. Sex is a simple example of  
 8143 an individual covariate, but one of the most commonly used in capture-recapture studies.  
 8144 The sex of an individual can influence many aspects of its ecology and behavior, including,

**Table 8.5.** Posterior summaries of parameter estimates from the SCR model with a global behavioral response in detection for the Ft. Drum black bear data set.

Parameter	Mean	SD	2.5	97.5
$D$	0.19	0.02	0.15	0.21
$N$	577.56	54.30	452	648
$\alpha_0$	-2.81	0.24	-2.91	-2.36
$\alpha_2$	0.90	0.23	0.45	1.35
$\psi$	0.88	0.08	0.69	0.99
$\sigma$	2.00	0.13	1.77	2.28

for example, its home range size, frequency of movement, and seasonal behavior. This is common in studies of carnivores where females often have smaller home ranges than males (Gardner et al., 2010b; Sollmann et al., 2011). Additionally, we may find differences in the baseline detection between males and females because females may move around less frequently, or possibly because they are less likely to use landscape structures that researchers may target with sampling devices in order to increase sample size, such as roads (e.g. Salom-Pérez et al., 2007).

Thus, we can imagine that sex may impact both the baseline encounter probability  $\alpha_0$  and the typical home range size, so that  $\alpha_1$  might be sex-specific also. The fully sex-specific model is:

$$\text{logit}(p_{0,i}) = \alpha_{0,sex_i}$$

$$p_{ijk} = p_{0,i} \exp(-\alpha_{1,sex_i} * ||\mathbf{s}_i - \mathbf{x}_j||^2)$$

where  $sex_i$  is a binary vector indicating the sex of each individual (1 = male, 2 = female). While we might know the sex of all individuals observed in the study, we will never know the sex of individuals that are not observed, resulting in missing values (Gardner et al., 2010b). It is also possible that we may not be able to determine the sex of individuals that are observed during the study. For example photographic captures do not necessarily result in pictures that allow the sex to be absolutely determined, thus sometimes resulting in missing values of this covariate for animals captured in the study. We deal with this slightly differently based on the framework that we select (Bayesian or likelihood). Here we demonstrate the Bayesian implementation and we discuss the likelihood approach using `secr` in detail below in sec. 8.7.2. Before proceeding with that, we note that it would be possible also to model covariates directly on the parameter  $\sigma$  (or its logarithm), e.g.,  $\log(\sigma_i) = \theta_1 + \theta_2 \text{sex}_i$  (see sec. gof.sec.XXXXX). One or the other (or perhaps *some* other) parameterization may yield a better performing MCMC algorithm or provide a more natural or preferred interpretation. In the context of Bayesian analysis, given that priors are not invariant to transformation of the parameters, this may be a consideration in choosing the particular parameterization.

Specifying a fully sex-specific model for **JAGS** is similar to the time-specific model shown above, we need to use an index or dummy variable to let  $\alpha_0$  and/or  $\alpha_1$  be estimated separately for males and females. The main difference arise from the fact that we do not observe sex for the augmented individuals, which means we have missing observations of the covariate for those individuals. As a result, sex is regarded as a random variable so these values – essentially missing data – can be estimated as part of the model. In Bayesian

8177 inference, missing data are treated like any other unknown quantity or parameter, which  
 8178 means that we need to specify a prior distribution for them. With only two possible  
 8179 outcomes, sex can be regarded as a Bernoulli random variable where  $\text{sex}_i \sim \text{Bernoulli}(\pi)$ .  
 8180 The parameter  $\pi$  is the sex ratio of the population and, since it is something we want  
 8181 to estimate (a hyperparameter of the model), we again have to specify a prior for  $\pi$ . To  
 8182 express our lack of knowledge about the parameter and account for the fact that being  
 8183 a probability, it can only take values between 0 and 1, we choose  $\pi \sim \text{Uniform}(0, 1)$ .  
 8184 The model specification in Panel ?? demonstrates how to incorporate a partially observed  
 8185 covariate, here, specifically sex.

8186 In both **JAGS** or **BUGS** missing data are indicated by `NA` in the data objects passed  
 8187 to the program through `bugs` or `jags`. In terms of setting up the data that means we need  
 8188 to create a vector of length  $M$  with the first  $n$  elements being 0 if individual  $i$  is a female,  
 8189 or 1 if  $i$  is a male (for the Fort Drum black bear data the function `bear.JAGS` extracts this  
 8190 information automatically from the `beardata` object), and the subsequent  $M - n$  elements  
 8191 being `NA`. It is generally a good idea to provide starting values for the missing data, but we  
 8192 cannot provide starting values for observed data; in this case where one vector (or other  
 8193 object) contains both observed and missing data, initial values for the observed data have  
 8194 to be specified as `NA`. The code snippet below shows you how to set up the data including  
 8195 the `sex` vector and the `initials` function, as well as the **JAGS** model specifications  
 8196 (the remainder of the code is identical to what we've shown before).

```
8197 sex<-beardata$sex #we read in the sex data for captured individual
8198 SEX<-c(sex-1, rep(NA, nz)) # in the data sex is coded as 1 and 2, so we recode it to 0 and 1
8199 data<-list(y=y,SEX=SEX, M=M,K=K, J=ntraps, Xl=Xl, Yl=Yl, Xu=Xu, Yu=Yu, X=X, area=areaX)
8200 params<-c('psi','p0','N', 'D', 'sigma', 'pi')
8201 inits =  function() {list(z=c(rep(1,nind), rbinom(nz,1,0.5)),psi=runif(1), s=cbind(runif(M, Xl, Xu),
8202 SEX=c(rep(NA, nind), rbinom(nz,1,0.5)),pi=runif(1), sigma=runif(2,2,3),alpha0=runif(2)) }
8203
8204 cat("
8205 model {
8206
8207 psi~dunif(0,1)
8208 pi~dunif(0,1)
8209
8210 for(t in 1:2){
8211 alpha0[t]~dnorm(0,.1)
8212 logit(p0[t])<- alpha0[t]
8213 alpha1[t]<-1/(2*sigma[t]*sigma[t])
8214 sigma[t]~dunif(0, 15)
8215 }
8216
8217 for(i in 1:M){
8218 z[i] ~ dbern(psi)
8219 SEX[i]~dbern(pi)
8220 SEX2[i]<-SEX[i] + 1
8221 s[i,1]~dunif(Xl,Xu)
8222 s[i,2]~dunif(Yl,Yu)
```

---

```

8223
8224 for(j in 1:J){
8225   d[i,j]<- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
8226   y[i,j] ~ dbin(p[i,j],K)
8227   p[i,j]<- z[i]*p0[SEX2[i]]*exp(-alpha1[SEX2[i]]*d[i,j]*d[i,j])
8228 }
8229 }
8230 N<-sum(z[])
8231 D<-N/area
8232 }
8233 ",file = "SCRsex.txt")
```

8234 Our estimate of density under the fully sex-specific model is still very similar to the  
 8235 previous models (Tab. 8.6), and while the baseline detection was not very different between  
 8236 males and females, we can see that they had very different  $\sigma$  estimates (note that the  
 8237 BCIs do not overlap). As usual, you can reproduce this analysis by calling the function  
 8238 `bear.JAGS` and set `model='SCRsex'`.

**Table 8.6.** Posterior summaries of parameter estimates from sex specific SCR models for the Ft. Drum black bear data set.

Parameter	Mean	SD	2.5	97.5
$D$	0.168	0.022	0.12	0.21
$N$	509.982	66.355	376	631
$p_0(\text{female})$	0.136	0.025	0.09	0.19
$p_0(\text{male})$	0.092	0.017	0.06	0.13
$\pi$	0.310	0.068	0.19	0.45
$\psi$	0.784	0.103	0.58	0.97
$\sigma_{\text{female}}$	1.542	0.132	1.31	1.83
$\sigma_{\text{male}}$	2.682	0.389	2.09	3.62

## 8.6 INDIVIDUAL HETEROGENEITY.

8239 Here we consider SCR models with individual heterogeneity, an individual effect that is  
 8240 completely latent. Capture-recapture models with individual heterogeneity in detection  
 8241 probability, so-called model  $M_h$ , have a long history in classical capture recapture models  
 8242 and they have special relevance to SCR (sec. 4.4). We note that their use has been called  
 8243 into question by Link (2004) who noted that  $N$  may not be identifiable across arbitrary  
 8244 classes of mixture models. One possible way to get around this problem is to identify  
 8245 explicit sources of heterogeneity in detection probability and model those directly. For  
 8246 example, we can do this by using individual covariate models (e.g., sec. 4.5). Of course,  
 8247 spatial capture-recapture models are such a class of models which seek to explain hetero-  
 8248 geneity in detection by describing the underlying mechanism explicitly. In particular, that  
 8249 mechanism is the juxtaposition of individuals with traps and the resulting heterogeneity  
 8250 that is induced by heterogeneity in exposure to trapping.

8251 While the advent of SCR models may appear to have rendered the use of classical  
 8252 model  $M_h$  obsolete (because the heterogeneity is being accounted for explicitly) we may

still wish to consider heterogeneity models for other biological reasons. It is reasonable to expect in real populations that there exists heterogeneity in home range size and so we think that  $\alpha_1$  could exhibit heterogeneity among individuals. As we noted previously, it may be advantageous or desirable in some cases to model heterogeneity directly in terms of the scale parameter of the distance function  $\sigma$  or some other transformation of the “distance coefficient”, perhaps even 95% home range area.

Here we develop and evaluate a class of spatial capture-recapture models which allow for individual heterogeneity in encounter probability. In particular, one class of models we propose explicitly admits individual heterogeneity in home range *size*. In addition, we consider a standard representation for heterogeneity in which an additive individual-specific random effect is included in the linear predictor for baseline encounter probability.

### 8.6.1 Models of Heterogeneity

An obvious model extends the SCR model by including an additive individual effect, analogous to classical “model  $M_h$ ”. We’ll call this model “SCR+Mh”:

$$\begin{aligned}\text{logit}(p_{0,i}) &= \alpha_0 + \eta_i \\ p_{ijk} &= p_{0,i} \exp(-\alpha_1 * ||\mathbf{s}_i - \mathbf{x}_j||^2)\end{aligned}$$

where  $\eta_i$  is an individual random effect having distribution  $[\eta|\tau]$ . A popular class of models arises by assuming  $\eta_i \sim \text{Normal}(0, \tau^2)$  (Coull and Agresti, 1999; Dorazio and Royle, 2003). Many other random effects distributions are possible, a popular one being the finite-mixture of point masses (Norris III and Pollock, 1996; Pledger, 2000). We demonstrate how to fit finite-mixture models using `secr` in sec. 8.7.3. However, heterogeneity seems naturally continuous unless one expects the heterogeneity to be due to meaningful biological groupings in which case such information would normally be collected if possible. Even so the more likely scenario is that heterogeneity is due to a lot of different sources contributing independent components of variation, and so the normal model seems sensible in that regard.

**Heterogeneity Induced by Variation in Home Range Size** – An alternative heterogeneity model, one that has more of a direct biological motivation and interpretation, describes heterogeneity in home range size among individuals. This is manifest in the scale parameter of the detection function  $\sigma^2$  or its inverse  $\alpha_1 = 1/\sigma^2$ . We might thus assume a distribution for either  $\sigma^2$  or its inverse,  $\alpha_1$ . We thus propose “model SCR + Ah” (Ah for area-induced heterogeneity).

$$\begin{aligned}\text{logit}(p_0) &= \alpha_0 \\ p_{ijk} &= p_0 \exp(-\alpha_{1,i} * ||\mathbf{s}_i - \mathbf{x}_j||^2)\end{aligned}$$

This model is a model of heterogeneity in home range area. For example if we assume that  $\alpha_{1,i} \sim \text{Normal}(\beta_0, \tau^2)$  with  $\beta_0 = 2$  and  $\tau = 0.50$ . Then the population distribution of  $\sigma$  in this case is given in Figure 8.1. The motivating point of this model is that we expect such variability in natural populations. Thus we suggest this biologically sensible model of heterogeneity, which fills a methodological gap in the literature in the sense that SCR models have usually been homogeneous with respect to their explicit treatment of home range size and geometry.

In panel 8.2, we show how to implement the SCR + Mh model.

---

```

model {
  alpha0 ~ dnorm(0,.1)
  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)
    z[i] ~ dbern(psi)
    s[i,1] ~ dunif(Xl,Xu)
    s[i,2] ~ dunif(Yl,Yu)
    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)
      logit(p0[i,j])<-alpha0 + eta[i]
      p.eff[i,j]<- z[i]*p0[i,j]*exp(- alpha1*d[i,j]*d[i,j])
    }
  }
  N<-sum(z[])
  D<-N/area
}

```

---

Panel 8.2: **JAGS** model specification for the SCR + Mh model with half-normal distance function.

**Figure 8.1.** Population distribution of  $\sigma$  if  $(1/\sigma^2) \sim \text{Normal}(2, 0.50)$ .

## 8.7 LIKELIHOOD ANALYSIS IN SECR

Previously, in Chapt. 6, we introduced the **R** package **secr** and described the likelihood based to inference taken by that package (see sec. 6.5.3). Here we cover how to implement a series of covariate models in **secr** and show briefly an example of model selection using AIC. In general, **secr** allows the user to simulate data and fit a suite of models with various detection functions and covariate responses. As we saw in Chapt. 6, **secr** uses the standard **R**model specification framework, defining the dependent and independent variable relationship using tildes (e.g.,  $y \sim x$ ). Thus, in **secr** we might have  $g0 \sim \text{behavior}$  or  $\text{sigma} \sim \text{time}$ ; when left unspecified or set to 1 (e.g.,  $g0 \sim 1$ ), this will default to a model with no covariates (i.e., constant parameter values). Additionally, **secr** allows us to specify covariates on density (see Chapt. 13), which are set for example as  $D \sim \text{habitat}$ .

To demonstrate a suite of 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 8.3 ). 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.

### 8.7.1 Notes for fitting standard models

In the **secr** package, the detection functions are specified by changing the “**detectfn**” option (an integer code) within the **secr.fit** command. Table 8.1 shows the possible detection functions that **secr** will fit; the default is that based on the kernel of a bivariate normal (“half-normal”) model and the (negative) exponential is **detectfn = 2**. See model 2 in panel 8.3 for how to fit the exponential model to the bear dataset.

The **secr** package easily fits a range of SCR equivalents of standard capture-recapture models. The package has pre-defined versions of the classic  $M_t$  where each occasion has its own detection probability, as well as a linear trend in baseline detection over occasions (in a spatial modeling framework  $\sigma$  could also be an occasion specific parameter, but having detection 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 command call (see model 3 in panel 8.3); whereas, for a linear trend over occasions **secr** uses ‘T’.

It is similarly easy to incorporate a global trap response model (model  $M_B$ ), or a trap specific behavioral response (model  $M_b$ , as in Royle et al. (2011b)). Despite our notion here, **secr** uses “b” for the global response model and “bk” for the local trap response model (see models 4 and 5 in panel 8.3; note that to fit the trap specific behavioral response model you need version 2.3.1 or newer of **secr**).

---

```
1. null model with a bivariate normal detection function
bear_0=secr.fit (bear.cap, model=list(D ~ 1, g0 ~ 1, sigma ~ 1),buffer = 20000)

2. null model with an exponential detection function
bear_0exp=secr.fit (bear.cap, model=list(D ~ 1, g0 ~ 1, sigma ~ 1),buffer = 20000, detectfn=2)

3. model with fixed time effects
bear_t=secr.fit (bear.cap, model=list(D ~ 1, g0 ~ t, sigma ~ 1),buffer = 20000)

4. global behavioral model
bear_B=secr.fit (bear.cap, model=list(D ~ 1, g0 ~ b, sigma ~ 1),buffer = 20000)

5. trap specific behavioral response
bear_b=secr.fit (bear.cap, model=list(D ~ 1, g0 ~ bk, sigma ~ 1),buffer = 20000)

6. global behavior model with fixed time effects
bear_bt=secr.fit (bear.cap, model=list(D ~ 1, g0 ~ b+t, sigma ~ 1),buffer = 20000)

7. sex-specific model
bear_sex=secr.fit (bear.cap, model=list(D ~ session, g0 ~ session, sigma ~ session),buffer = 20000)

8. heterogeneity model
bear_h2=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ h2, sigma ~ h2),buffer = 20000)
```

---

Panel 8.3: Models called from `secr.bear` function.

---

**8328 8.7.2 Sex Effects**

8329 Specifying models with sex specific parameters is not as straight forward in **secr** as the  
 8330 other models. Incorporating sex effects into models with **secr** can be done a few different  
 8331 ways. In all cases, individuals that are of unknown sex must be removed from the dataset  
 8332 (recall that in a Bayesian framework we can keep these individuals in the data set and the  
 8333 model will estimate their sex based on the underlying distribution, e.g. **sex** ~ Bernoulli( $\pi$ )  
 8334 and  $\pi$  = probability of being a male). The most common way to include sex is to code it as  
 8335 a distinct “session” using the multi-session models of **secr** (see sec. 6.5.4 for a description  
 8336 of the multi-session models), providing two sessions that represent males and females (see  
 8337 model 7 in panel 8.3). This method provides two separate density estimates, which can  
 8338 then be combined into a total density.

**8339 8.7.3 Individual heterogeneity**

8340 To incorporate heterogeneity, **secr** fits the finite mixture models (Norris III and Pollock,  
 8341 1996; Pledger, 2000). These are expensive in terms of parameters and we think they have  
 8342 been widely adopted because they are easy to analyze using likelihood methods, as the  
 8343 marginal distribution of the data is just a sum of two or a small number of components.  
 8344 More recently, continuous mixtures have been adopted in many settings because they  
 8345 are natural extensions of standard GLMs. Using **secr**, individual heterogeneity can be  
 8346 incorporated into the detection parameters as either a 2-part or 3-part finite mixture  
 8347 model with the use of “**h2**” or “**h3**”, respectively, in the model call. The 2-part mixture is  
 8348 shown in model 8 of panel 8.3 and the 3-part mixture can easily be fit by substituting **h3**  
 8349 for **h2**. The finite-mixture model can be fit in **JAGS**or **BUGS**, but we only showed the  
 8350 SCR + Mh continuous mixture in the version above (see sec. 8.6.1).

**8351 8.7.4 Model selection in **secr** with AIC**

8352 One practical advantage to using the **secr** package, or likelihood inference in general, is  
 8353 the convenience of automatic model selection using AIC (Burnham and Anderson, 2002).  
 8354 Running the function **secr.bear**, which calls all of the models we have described, will  
 8355 return, in addition to all model results, an AIC table with all of the summarized results  
 8356 including the AIC values, delta AIC, and model weights (see Table 8.7 or reproduce results  
 8357 in R using `out<- secr.bear(); out$ AIC.tab`).

8358 It is important to note here that because of the way we specified the sex-specific model  
 8359 as “sessions”, we coded the dataset as such when first loading it to **secr**. Thus, all model  
 8360 results will list separate parameter estimates for each session, even the null model with  
 8361 no covariates (but estimates are the same for both “sessions” in all but the sex specific  
 8362 model).

8363 The results from this AIC analysis are straightforward to interpret; the model with  
 8364 a local trap response on detection, “**bk**”, has a model weight of 1 and thus, according to  
 8365 AIC, 100% support. The 2-part finite mixture model for  $g_0$  and  $\sigma$  has the second lowest  
 8366 AIC, but considering the large dAICc compared to the local trap response model we would  
 8367 probably not consider it any further.

**Table 8.7.** 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

## 8.8 SUMMARY AND OUTLOOK

8368 There are endless covariates and detection functions that can be defined and our goal in  
 8369 this chapter was to introduce some basic concepts for how to construct and implement such  
 8370 models. Essentially, SCR's are GLMMS and therefore it is easy to incorporate covariates  
 8371 into the model for encounter probability. Just a few examples of different covariates are  
 8372 given in Table 8.3. We can also consider covariates by their

8373 XXX Just decided to add this table. More on it tomorrow. XX

**Table 8.8.** Examples of different covariate classifications.

Covariate class	Examples
Fixed	baited, weather, habitat
Partially observed	sex, age,
Unobserved	home range size,

8374 The move to spatially explicit models in capture-recapture studies has not rendered  
 8375 the basic CR models (Otis et al., 1978) obsolete, instead, we continue to use those models  
 8376 and incorporate space into them. Now, we can include not only these standard CR models,  
 8377 e.g.,  $M_0$ ,  $M_t$ ,  $M_b$ , and  $M_h$ , but also new models that allow for trap-specific information  
 8378 such as "baited/not-baited" or "on/off road". In Chaps. 12, 11 and 13, we explore  
 8379 additional models for explaining variation in encounter probability and density based on  
 8380 spatial covariates that describe variation in landscape or habitat conditions.

8381 Researchers are often concerned with describing the factors or covariates that influence  
 8382 variation in detection or encounter probability, particularly as this can directly influence  
 8383 other parameters in the model such as density. These covariates have various levels -  
 8384 specific to individual, trap, sampling occasion and can be fully or partially observed as  
 8385 well as completely latent. In SCR models, these more complex covariate models are fairly  
 8386 easy to fit, though one should take caution not to over parameterize the models particularly  
 8387 when a study yields a sparse data set.

8388  
8389

---

8390

# 9

8391

## ALTERNATIVE MODELS FOR THE ENCOUNTER PROCESS

8392 In the previous chapter we considered a very specific observation model consisting of two  
8393 elements: First a description of the encounter process by which individuals are detected in  
8394 traps and, secondly, a parametric model that relates encounter probability to an individ-  
8395 ual's activity center. We consider alternative parametric forms of the encounter probability  
8396 model in Chapt. 8 and elsewhere. Here, we focus on developing additional models for  
8397 the encounter process. The encounter process could be thought of as being determined  
8398 by the type of device – or the type of “detector” using the terminology of **secr** (Efford,  
8399 2011). In the previous chapter we considered the Bernoulli encounter process model (the  
8400 *proximity detector* in **secr**) which assumed that individual and trap-specific encounters  
8401 were independent Bernoulli trials.

8402 In this chapter, we consider alternative models of the encounter process that do not  
8403 require independence, and accommodate observed data that are not only binary. In par-  
8404 ticular, we consider models for encounter *frequencies*, and encounter process models based  
8405 on the multinomial distribution. For example, if sampling devices can detect an indi-  
8406 vidual some arbitrary number of times during an interval, then it is natural to consider  
8407 observation models for encounter frequencies, such as the Poisson model. Another type  
8408 of encounter device is the “multi-catch” device (Efford et al., 2009a) which is a physical  
8409 device that can capture and hold an arbitrary number of individuals. A typical example  
8410 is a mist-net for birds (Borchers and Efford, 2008). It is natural to regard observations  
8411 from these kinds of studies as independent multinomial observations. A related type of  
8412 device that produces *dependent* multinomial observations are the so-called *single-catch*  
8413 traps (Efford, 2004; Efford et al., 2009a). The canonical example are small-mammal live  
8414 traps (Converse et al., 2006b) which catch and hold a single individual. Competition  
8415 among individuals for traps induces a complex dependence structure among individual  
8416 encounters. To date, no formal inference framework has been devised for this method  
8417 although it stands to reason that the independent multinomial model should be a good  
8418 approximation in some situations (Efford et al., 2009a).

8419 We analyze a number of examples of these different observation models using **JAGS**  
 8420 and also the **R** package **secr** (Efford, 2011).

## 9.1 POISSON OBSERVATION MODEL

8421 The models we analyze in Chapt. 5 assumed binary observations – i.e., standard encounter  
 8422 history data – so that individuals are captured at most one time in a trap on any given  
 8423 sample occasion. This makes sense for many types of DNA sampling (e.g., based on hair  
 8424 snares) because distinct visits to sampled locations or devices cannot be differentiated.  
 8425 However, for some encounter devices, or methods, the potential sample size is *not* fixed,  
 8426 and so it is possible to encounter an individual some arbitrary number of times during any  
 8427 particular sampling episode. That is, we might observe encounter frequencies  $y_{ijk} > 1$   
 8428 for individual  $i$ , trap  $j$  and sampling interval  $k$ . As an example, if a camera device is  
 8429 functioning properly it may be programmed to take photos every few seconds if triggered.  
 8430 For a second example, suppose we are searching a quadrat or length of trail for scat, we may  
 8431 find multiple samples from the same individual. Therefore, we seek observation models  
 8432 that accommodate such encounter frequency data. In general, any discrete probability  
 8433 mass function could be used for this purpose, including the standard models for count  
 8434 data used throughout ecology, the Poisson and negative binomial. Here we focus on using  
 8435 the Poisson model only although other count frequency models are possible for SCR models  
 8436 (Efford et al., 2009b).

8437 Let  $y_{ijk}$  be the frequency of encounter for individual  $i$ , in trap  $j$ , during occasion  $k$ ,  
 8438 then assume:

$$y_{ijk} \sim \text{Poisson}(\lambda_{ij})$$

8439 where the expected encounter frequency  $\lambda_{ij}$  depends on both individual and trap. As we  
 8440 did in the binary model of Chapt. 5, we now seek to model the expected value of the  
 8441 observation (which was  $p_{ij}$  in Chapt 5) as a function of the individual activity center  $\mathbf{s}_i$ .  
 8442 We propose

$$\lambda_{ij} = \lambda_0 g(\mathbf{x}_j, \mathbf{s}_i)$$

8443 Where  $g(\mathbf{x}, \mathbf{s})$  is any positive valued function, such as the negative exponential or the  
 8444 bivariate Gaussian kernel, and  $\lambda_0$  is the baseline encounter rate – the expected number  
 8445 of encounters if a trap is placed precisely at an individuals home range center (note: in  
 8446 **secr** the notation for this is  $g_0$ ). Then,  $\lambda_0 g(\mathbf{x}_j, \mathbf{s}_i)$  is the expected encounter rate in trap  
 8447  $\mathbf{x}_j$  for an individual having activity center  $\mathbf{s}_i$ . Note that

$$\log(\lambda_{ij}) = \log(\lambda_0) + \log(g(\mathbf{x}_j, \mathbf{s}_i)).$$

8448 Equating  $\alpha_0 \equiv \log(\lambda_0)$ , and, if  $g(\mathbf{x}, \mathbf{s}) \equiv \exp(-d(\mathbf{x}, \mathbf{s})^2/(2\sigma^2))$  (i.e., the Gaussian model),  
 8449 then:

$$\log(\lambda_{ij}) = \alpha_0 + \alpha_1 d(\mathbf{x}_j, \mathbf{s}_i)^2 \quad (9.1.1)$$

8450 where  $\alpha_1 = -1/(2\sigma^2)$ , which is the same linear predictor as we have seen for the Bernoulli  
 8451 model in Chapt. 5. This Poisson SCR model is therefore a type of Poisson generalized  
 8452 linear mixed model (GLMM).

8453 We can accommodate covariates at the level of individual-, trap- or sample occasion  
 8454 by including them on the baseline encounter rate parameter  $\lambda_0$ . For example, if  $C_j$  is

8455 some covariate that depends on trap only, then we express the relationship between  $\lambda_0$   
 8456 and  $C_j$  as:

$$\log(\lambda_{0,ijk}) = \alpha_0 + \alpha_2 C_j$$

8457 and therefore covariates on the logarithm of baseline encounter probability appear also as  
 8458 linear effects on  $\lambda_{ij}$ . The parameter  $\alpha_1$  might also vary in response to explicit covariates  
 8459 (e.g., sex of individual). We don't get into too much discussion of general covariate models  
 8460 here, but see Chapt. 8 for a more comprehensive development of covariate models along  
 8461 with examples.

8462 For models in which we do not have covariates that vary across the sample occasions  
 8463  $k$ , we can aggregate the observed data by the property of compound additivity of the  
 8464 Poisson distribution (if  $x$  and  $y$  are *iid* Poisson with mean  $\lambda$  then  $x + y$  is Poisson with  
 8465 mean  $2\lambda$ ). Therefore,

$$y_{ij} = (\sum_{k=1}^K y_{ijk}) = \text{Poisson}(K\lambda_0 g(\mathbf{x}_j, \mathbf{s}_i))$$

8466 We see that  $K$  and  $\lambda_0$  serve the same role as affecting the base encounter rate. Since the  
 8467 observation model is the same, probabilistically speaking, for all values of  $K$ , evidently we  
 8468 need only  $K = 1$  "survey" from which to estimate model parameters (Efford et al., 2009b).  
 8469 We know this intuitively as sampling by multiple traps serves as replication in SCR models.  
 8470 This has great practical relevance to the conduct of capture-recapture studies and the use  
 8471 of SCR models. For example, if individuality is obtained by genetic information from scat  
 8472 sampling, one should only have to carry-out a single spatial sampling of the study area.  
 8473 However, one must be certain that sufficient spatial recaptures will be obtained so that  
 8474 effective estimation is possible. -

### 8475 9.1.1 Poisson model of space usage

8476 It is natural to interpret the Poisson encounter model as a model of space usage resulting  
 8477 from movement of individuals about their home range (Sec. 5.3). Imagine we have perfect  
 8478 samplers in every pixel of the landscape so that whenever an individual moves from one  
 8479 pixel to another, we can record it. Let  $m_{ij}$  be the number of times individual  $i$  was  
 8480 recorded in pixel  $j$  (i.e., it selected or used pixel  $j$ ). Then, we might think of the Poisson  
 8481 model for the observed *use* frequencies:

$$m_{ij} \sim \text{Poisson}(\lambda_0 g(\mathbf{x}_j, \mathbf{s}_i))$$

8482 where  $\lambda_0$  is related to the baseline movement rate of the animal (how often it moves). This  
 8483 model of space usage gives rise to the standard resource selection function (RSF) models  
 8484 (see Chapt. 11). But now suppose our samplers are not perfect but, rather, record only  
 8485 a fraction of the resulting visits. A sensible model is

$$y_{ij}|m_{ij} \sim \text{Binomial}(m_{ij}, p).$$

8486 The marginal distribution of  $y_{ij}$  is:

$$y_{ij} \sim \text{Poisson}(p_0 g(\mathbf{x}_j, \mathbf{s}_i)).$$

8487 where  $p_0$  is a composite of the movement rate and conditional detection probability  $p$ .  
 8488 Therefore, we see that encounters accumulate in proportion to the frequency of outcomes  
 8489 of an individual using space (or “selecting resources”).

8490 We introduced an interpretation of SCR models in terms of movement and space usage  
 8491 in Sec. 5.3, and it is one of the main underlying concepts of SCR models that is not present  
 8492 in ordinary capture-recapture models. As we noted there, the underlying model of space  
 8493 usage is only as complex as the encounter probability model which has been, so far in this  
 8494 book, only symmetric and stationary (does not vary in space). We generalize this model  
 8495 of space usage substantially in Chapt. 11.

### 8496 9.1.2 Poisson relationship to the Bernoulli model

8497 There is a sense in which the Poisson and Bernoulli models can be viewed as consistent with  
 8498 one another. Note that under the Poisson model, the relationship between the expected  
 8499 count and the probability of counting “at least 1”, is given by

$$\Pr(y > 0) = 1 - \exp(-\lambda) \quad (9.1.2)$$

8500 where  $\mathbb{E}(y) = \lambda$ . Therefore, if we equate the event “encountered” with the event that the  
 8501 individual was captured at least 1 time under the Poisson model, i.e.,  $y > 0$ , then it would  
 8502 be natural to set  $p_{ij} = \Pr(y > 0)$  according to Eq. 9.1.2.

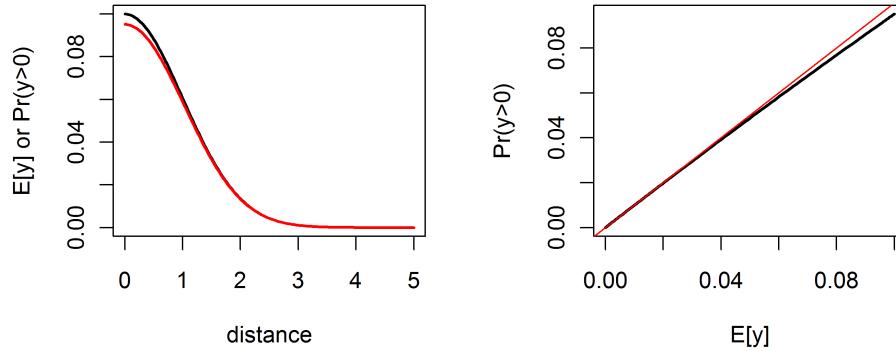
8503 In fact, as  $\lambda$  gets small, the Poisson model is a close approximation to the Bernoulli  
 8504 model in the sense that outcomes concentrate on  $\{0, 1\}$ , i.e.,  $\Pr(y \in \{0, 1\}) \rightarrow 1$  as  $\lambda \rightarrow 0$ .  
 8505 Indeed, under the Poisson model,  $\Pr(y > 0) \rightarrow \lambda$  for small values of  $\lambda$ . This phenomenon  
 8506 is shown in Fig. 9.1 where the left panel shows a plot of  $\lambda_{ij} = \lambda_0 g(\mathbf{x}_j, \mathbf{s}_i)$  vs. distance and  
 8507 superimposed on that is a plot of  $p_{ij} = 1 - \exp(-\lambda_{ij})$  vs. distance, for values  $\lambda_0 = 0.1$   
 8508 and  $\sigma = 1$ , and the right panel shows a plot of  $\Pr(y > 0)$  vs.  $\mathbb{E}(y)$ . We see that the two  
 8509 quantities are practically indistinguishable.

8510 This is convenient in some cases because the Poisson model might be more tractable to  
 8511 fit (or even vice versa). For an example, see the models described in Chapt. 18, and we also  
 8512 consider another case in Sec. 9.4 below. To evaluate the closeness of the approximation,  
 8513 you can use the following R commands which we used to produce Fig. 9.1:

```
8514 x<-seq(0.001,5,,200)
8515 lam0<- .1
8516 sigma<- 1
8517 lam<- lam0*exp(-x*x/(2*sigma*sigma))
8518
8519 par(mfrow=c(1,2))
8520 p1<- 1-exp(-lam)
8521 plot(x, lam, ylab="E[y] or Pr(y>0)", xlab="distance", type="l", lwd=2)
8522 lines(x, p1, lwd=2, col="red")
8523 plot(lam, p1, xlab="E[y]", ylab="Pr(y>0)", type="l", lwd=2)
8524 abline(0,1, col="red")
```

8525 To summarize, if  $y$  is Poisson then, as  $\lambda$  gets small,

$$\begin{aligned} \Pr(y > 0) &\approx \mathbb{E}(y) \\ 1 - \exp(-\lambda_0 g(\mathbf{x}, \mathbf{s})) &\approx \lambda_0 g(\mathbf{x}, \mathbf{s}) \end{aligned}$$



**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  $\lambda$ .

8526 What all of this suggests it that if we have very few observations  $> 1$  in our SCR data set,  
 8527 then we wont lose much information by using the Bernoulli model. On the other hand, the  
 8528 Poisson model may have some advantages in terms of analytic or numerical tractability  
 8529 in some cases. Further, this approximation explains the close correspondence we have  
 8530 found between these two versions of the Gaussian encounter probability model (Sec. 5.3).  
 8531 Namely, the Gaussian hazard model and the Gaussian encounter probability model are  
 8532 close approximations because  $1 - \exp(-\lambda) \approx \lambda$  if  $\lambda$  is small.

8533 Even in such cases where the Poisson and Bernoulli models are not quite equivalent, we  
 8534 might choose to truncate individual encounter frequencies to binary observations anyhow  
 8535 (transforming counts to 0/1 is called “quantizing”). We might do this intentionally in  
 8536 some cases, such as when the distinct encounter events are highly dependent as often  
 8537 happens in camera trap studies when the same individual moves back-and-forth in front  
 8538 of a camera during a short period of time. But sometimes, truncation is a feature of  
 8539 the sampling. For example, in the case of bear hair snares, the number of encounters  
 8540 might be well approximated by a Poisson distribution but we cannot determine unique  
 8541 visits and so only get to observe the binary event “ $y > 0$ ”. Similarly for scat sampling  
 8542 problems it will not generally be possible to diagnose distinct “independent” scat samples.  
 8543 Under this model the data are only binary encounters and we might therefore choose to  
 8544 model encounter probability using Eq. 9.1.3. Note that this is equivalent to using the  
 8545 complementary log-log link:

$$\text{cloglog}(p_{ij}) = \log(\lambda_0) + \log(g(\mathbf{x}, \mathbf{s}))$$

8546 where  $\text{cloglog}(u) = \log(-\log(1 - u))$ .

**9.1.3 A cautionary note on modeling encounter frequencies**

8548 Other models for counts might be appropriate. For example, ecologists are especially  
 8549 fond of negative binomial models for count data (Ver Hoef and Boveng, 2007; White  
 8550 and Bennetts, 1996; Kéry et al., 2005) but other models for excess-Poisson variation are  
 8551 possible. For example, we might add a normally distributed random effect to the linear  
 8552 predictor (Coull and Agresti, 1999).

8553 As a general rule we favor the Bernoulli observation model even if encounter frequencies  
 8554 are obtained by sampling. The main reason is that, with frequency data, we are forced  
 8555 to confront a model choice problem (i.e., Poisson, negative binomial, log-normal mixture)  
 8556 that is wholly unrelated to the fundamental space usage process that underlies the genesis  
 8557 of many types of SCR data. Repeated encounters over short time intervals are not likely  
 8558 to be the result of independent encounter events. E.g., an individual moving back and  
 8559 forth in front of a camera yields a cluster of observations that is not informative about the  
 8560 underlying spatial structure of the population. Similarly in scat surveys dogs are used to  
 8561 locate scats which are processed in the lab for individuality (Kohn et al., 1999; MacKay  
 8562 et al., 2008; Thompson et al., 2012). The process of local scat deposition is not strictly  
 8563 the outcome of movement or space usage but rather the outcome of complex behavioral  
 8564 considerations as well as dependence in detection of scat by dogs. For example, dogs find  
 8565 (or smell) one scat and then are more likely to find one or more nearby ones, if present, or  
 8566 they get into a den or latrine area and find many scats. The additional assumption required  
 8567 to model variation in observed frequencies (i.e., conditional on location) provides relatively  
 8568 no information about space usage and density, and we feel that the model selection issue  
 8569 should therefore be avoided.

8570 To elaborate on this, we suppose that an individual with activity center  $\mathbf{s}$  visits a  
 8571 particular pixel  $\mathbf{x}$  with some probability  $p(\mathbf{x}, \mathbf{s})$ , and then, once there, deposits a number of  
 8572 scat, or visits a camera some number of times with frequency  $y(\mathbf{x}, \mathbf{s}) \geq 0$ . We describe the  
 8573 outcome of this movement/usage process with a two-level hierarchical model of the form:  
 8574  $[y|z][z|p(\mathbf{x}, \mathbf{s})]$  where  $z(\mathbf{x}, \mathbf{s})$  is a binary variable that indicates whether the individual with  
 8575 activity center  $\mathbf{s}$  used pixel  $\mathbf{x}$  during some interval, and let  $z(\mathbf{x}, \mathbf{s}) \sim \text{Bernoulli}(p(\mathbf{x}, \mathbf{s}))$ .  
 8576 If we suppose encounter frequency  $y$  is independent of  $\mathbf{x}$  and  $\mathbf{s}$  conditional on the use  
 8577 variable  $z$ , then we see that the model for  $y$  (amount of use) does not depend on  $\mathbf{s}$ .

**9.1.4 Analysis of the Poisson SCR model in BUGS**

8578 We consider the simplest possible model here in which we have no covariates that vary  
 8579 over sample occasions  $k = 1, 2, \dots, K$  so that we work with the aggregated individual-  
 8580 and trap-specific encounters:

$$y_{ij} = \left( \sum_{k=1}^K y_{ijk} \right) = \text{Poisson}(K\lambda_0 g(\mathbf{x}_j, \mathbf{s}_i))$$

8582 and we consider the bivariate normal form of  $g(\mathbf{x}, \mathbf{s})$ :

$$g(\mathbf{x}, \mathbf{s}) = \exp(-d(\mathbf{x}, \mathbf{s})^2/(2\sigma^2))$$

8583 so that

$$\log(\lambda_{ij}) = \alpha_0 + \alpha_1 d(\mathbf{x}_j, \mathbf{s}_i)^2$$

8584 where  $\alpha_0 = \log(\lambda_0)$  and  $\alpha_1 = -1/(2\sigma^2)$ .

8585 As usual, we approach Bayesian analysis of these models using data augmentation  
 8586 (Sec. 4.2). Under data augmentation, we introduce a collection of all-zero encounter  
 8587 histories to bring the total size of the data set up to  $M$ , and a corresponding set of data  
 8588 augmentation variables  $z_i \sim \text{Bern}(\psi)$ . Then the observation model is specified conditional  
 8589 on  $z$  according to:

$$y_{ij} \sim \text{Poisson}(z_i K \lambda_{ij})$$

8590 which evaluates to a point mass at  $y = 0$  if  $z = 0$ . In other words, the observation model  
 8591 under data augmentation is a zero-inflated Poisson model which is easily analyzed by  
 8592 Bayesian methods, e.g., in one of the **BUGS** dialects or, alternatively, using likelihood  
 8593 methods, which we neglect here although the same principles as in Chapt. 6 apply.

### 8594 9.1.5 Simulating Data and Fitting the Model

8595 Simulating a sample SCR data set under the Poisson model requires only a couple minor  
 8596 modifications to the procedure we used in Chapt. 5 (see the function `simSCR0`). In  
 8597 particular, we modify the block of code which defines the model to be that of  $E(y)$  and  
 8598 not  $\Pr(y = 1)$ , and we change the random variable generator from `rbinom` to `rpois`:

```
8599 ##  
8600 ## S = activity centers and traplocs defined as in simSCR0()  
8601 ##  
8602 D<- e2dist(S,traplocs) # Distance between activity centers and traps  
8603  
8604 ## Define parameter values:  
8605 alpha0<- -2.5  
8606 sigma<- 0.5  
8607 alpha1<- 1/(2*sigma*sigma)  
8608  
8609 ## Encounter probability model:  
8610 muy <- exp(alpha0)*exp(-alpha1*D*D)  
8611  
8612 ## Now generate the encounters of every individual in every trap  
8613 Y<-matrix(NA,nrow=N,ncol=ntraps)  
8614 for(i in 1:nrow(Y)){  
8615   Y[i,]<-rpois(ntraps,K*muy[i,])  
8616 }
```

8617 We modified our simulation code from Chapt. 5 to simulate Poisson encounter fre-  
 8618 quencies for each trap and then we analyze an ideal data set using **BUGS**. This Poisson  
 8619 simulator function `simPoissonSCR` is available in the `scrbook` package (it can produce  
 8620 3-d encounter history data too, although we don't do that here). Here is an example  
 8621 of simulating a data set and harvesting the required data objects, and doing the data

```

8622 augmentation:
8623 data<-simPoissonSCR(discard0=TRUE,rnd=2013)
8624 y<-data$Y
8625 nind<-nrow(y)
8626 X<-data$traplocs
8627 K<-data$K
8628 J<-nrow(X)
8629 xlim<-data$xlim
8630 ylim<-data$ylim
8631
8632 ## Data augmentation
8633 M<-200
8634 y<-rbind(y,matrix(0,nrow=M-nind,ncol=ncol(y)))
8635 z<-c(rep(1,nind),rep(0,M-nind))

```

8636 The process for fitting the model in **WinBUGS** or **JAGS** is identical to what we've  
 8637 done previously in Chapt. 5. In particular, we set up some starting values, package  
 8638 the data and inits, identify the parameters to be monitored, and then send everything  
 8639 off to our MCMC engine. Here it all is for fitting the Poisson observation model (these  
 8640 commands are shown in the help file for **simPoissonSCR**):

```

8641 sst<-X[sample(1:J,M,replace=TRUE),] # starting values for s
8642 for(i in 1:nind){
8643 if(sum(y[i,])==0) next
8644 sst[i,1]<- mean( X[y[i,]>0,1] )
8645 sst[i,2]<- mean( X[y[i,]>0,2] )
8646 }
8647 sst<-sst + runif(nrow(sst)*2,0,1)/8
8648 data <- list (y=y,X=X,K=K,M=M,J=J,xlim=xlim,ylim=ylim)
8649 inits <- function(){
8650   list (alpha0=rnorm(1,-2,.4),alpha1=runif(1,1,2),s=sst,z=z,psi=.5)
8651 }
8652 parameters <- c("alpha0","alpha1","N","D")
8653
8654 cat(
8655 model {
8656 alpha0~dnorm(0,.1)
8657 alpha1~dnorm(0,.1)
8658 psi~dunif(0,1)
8659
8660 for(i in 1:M){
8661 z[i] ~ dbern(psi)
8662 s[i,1]~dunif(xlim[1],xlim[2])
8663 s[i,2]~dunif(ylim[1],ylim[2])
8664 for(j in 1:J){
8665 d[i,j]<- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)

```

```

8666 y[i,j] ~ dpois(lam[i,j])
8667 lam[i,j]<- z[i]*K*exp(alpha0)*exp(- alpha1*d[i,j]*d[i,j])
8668 }
8669 }
8670 N<-sum(z[])
8671 D<- N/64
8672 }
8673 ",file = "SCR-Poisson.txt")
8674
8675 library(R2WinBUGS)
8676 out1 <- bugs (data, inits, parameters, "SCR-Poisson.txt", n.thin=1,n.chains=3,
8677 n.burnin=1000,n.iter=2000,working.dir=getwd(),debug=TRUE)

```

8678 Or, using **JAGS** via **rjags** we would do something like this:

```

8679 library(rjags)
8680 jm <- jags.model("SCR-Poisson.txt", data=data, inits=inits, n.chains=3, n.adapt=1000)
8681 out2 <- coda.samples(jm, parameters, n.iter=1000, thin=1)

```

8682 Summarizing the output from the **WinBUGS** run produces the following:

```

8683 > print(out1,digits=2)
8684 Inference for Bugs model at "SCR-Poisson.txt", fit using WinBUGS,
8685 3 chains, each with 2000 iterations (first 1000 discarded)
8686 n.sims = 3000 iterations saved
8687      mean    sd   2.5%   25%   50%   75% 97.5% Rhat n.eff
8688 alpha0   -2.57  0.19  -2.95  -2.69  -2.57  -2.44  -2.19 1.00  2600
8689 alpha1    2.34  0.36   1.69   2.08   2.32   2.57   3.12 1.00  3000
8690 N       114.13 15.25  87.97 103.00 113.00 124.00 147.00 1.01  370
8691 D       1.78  0.24   1.37   1.61   1.77   1.94   2.30 1.01  370
8692 deviance 329.95 21.92 290.00 314.20 329.50 344.40 375.80 1.00  1700
8693 ...
8694 [...some output deleted...]
8695 ...

```

### 8696 9.1.6 Analysis of the Wolverine Study Data

8697 We reanalyzed the data from the wolverine camera trapping study that were first introduced in Sec. 5.8. 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 Tab. 9.1. The results are almost indistinguishable from the Bernoulli model fitted previously, where we had a posterior mean for  $N$  of 59.84,  $\sigma$  it was 0.64. The **R** script is provided in the **scrbook** package which you can use to modify the model or obtain more posterior samples.

**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
$\psi$	0.30	0.07	0.19	0.30	0.45	1	650
$\sigma$	0.64	0.06	0.54	0.64	0.76	1	730
$\lambda_0$	0.06	0.01	0.04	0.06	0.08	1	5000
$\log(p_0)$	-2.89	0.17	-3.22	-2.89	-2.57	1	5000
$N$	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

### 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:

```

8709 library(secr)
8710 library(scrbook)
8711 data(wolverine)

8712
8713 traps<-as.matrix(wolverine$wtraps)
8714 dimnames(traps)<-list(NULL,c("trapID","x","y",paste("day",1:165,sep="")))
8715 traps1<-as.data.frame(traps[,1:3])
8716 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” traps (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 types 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 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 type of model in which the 2nd assumption is violated

are the “single catch” trap systems which we address in Sec. 9.4 below. In general we could imagine non-independence being important in any multi-catch situation but to the best of our knowledge a general model that encompasses complete dependence (i.e., single-catch) and complete independence (multi-catch) of individuals has not been proposed. So we treat the cases individually and, in this section, we address the multi-catch situation wherein individuals behave independently.

In this case we assume the observation  $\mathbf{y}_{ik}$  for individual  $i$  during sample occasion  $k$  is a multinomial observation which consists of a sequence of 0’s and a single 1 indicating the trap of capture, or “not captured”. For the “not captured” event we define an additional outcome, by convention element  $J + 1$  of the vector. As an example, if we capture an individual in trap 2 during some occasion of a 6 sample period study then the multinomial observation has length  $J + 1 = 7$ , and the observation is  $\mathbf{y}_i = (0, 1, 0, 0, 0, 0, 0)$ . An individual not captured at all would have the observation vector  $(0, 0, 0, 0, 0, 0, 1)$ . If we sample for 5 occasions in all and the individual is also caught in trap 4 during occasion 3, but otherwise uncaptured, then the 5 encounter observations for that individual are as follows:

occassion	1	2	3	4	5	6	7	"not captured"
1	0	1	0	0	0	0	0	
2	0	0	0	0	0	0	1	
3	0	0	0	1	0	0	0	
4	0	0	0	0	0	0	1	
5	0	0	0	0	0	0	1	

Statistically we regard the *rows* of this data matrix as *independent* multinomial trials.

Analogous to our previous Bernoulli and Poisson models, we seek to construct the multinomial cell probabilities for each individual, as a function of *where* that individual lives, through its center of activity  $\mathbf{s}$ . Thus we suppose that

$$\mathbf{y}_{ik} | \mathbf{s}_i \sim \text{Multinom}(1, \boldsymbol{\pi}(\mathbf{s}_i)) \quad (9.2.1)$$

where  $\boldsymbol{\pi}(\mathbf{s}_i)$  is a vector of length  $J + 1$ , where  $\pi_{i,J+1}$ , the last cell, corresponds to the probability of the event “not captured”. Now we have to construct these cell probabilities in some meaningful way that depends on each individual’s  $\mathbf{s}$ . We use the standard multinomial logit with distance as a covariate:

$$\pi_{ij} = \frac{\exp(\alpha_0 + \alpha_1 d_{ij})}{1 + \sum_j \exp(\alpha_0 + \alpha_1 d_{ij})}$$

for  $j = 1, 2, \dots, J$  and, for  $J + 1$ , i.e., “not captured”,

$$\pi_{i,(J+1)} = \frac{\exp(0)}{1 + \sum_j \exp(\alpha_0 + \alpha_1 d_{ij})}$$

or, more commonly, we use  $d_{ij}^2$  to correspond to our Gaussian kernel model for encounter probability. Whatever function of distance we use in the construction of multinomial probabilities will have a direct correspondence to the standard encounter probability models we used in the Bernoulli or Poisson models as well (see Sec. 5.3).

8769 It is convenient to express these multinomial models short-hand as follows, e.g., for  
 8770 the Gaussian encounter probability model:

$$\text{mlogit}(\pi_{ij}) = \alpha_0 + \alpha_1 d_{ij}^2$$

8771 In this way we can refer to models with covariates in a more concise way. For example, a  
 8772 model with a trap-specific covariate, say  $C_j$ , is:

$$\text{mlogit}(\pi_{ij}) = \alpha_0 + \alpha_1 d_{ij}^2 + \alpha_2 C_j$$

8773 or we could include occasion-specific covariates too, such as behavioral response.

8774 A statistically equivalent distribution to the multinomial is the *categorical* distribution.  
 8775 If  $\mathbf{y}$  is a multinomial trial with probabilities  $\boldsymbol{\pi}$  than the *position* of the non-zero element of  
 8776  $\mathbf{y}$  is a categorical random variable with probabilities  $\boldsymbol{\pi}$ . We express this for SCR models  
 8777 as

$$\mathbf{y}|\mathbf{s} \sim \text{Categorical}(\boldsymbol{\pi}(\mathbf{s}))$$

8778 In the SCR context, the categorical version of the multinomial trial corresponds to the  
 8779 *trap of capture*. Using our example above with 6 traps then we could as well say  $y_{ik}$  is a  
 8780 categorical random variable with possible outcomes (1, 2, 3, 4, 5, 6, 7) where outcome  $y = 7$   
 8781 corresponds to “not captured.” Obviously, how this is organized or labeled is completely  
 8782 irrelevant, although it is convenient to use the integers 1 to  $(J + 1)$  where  $J + 1$  is the  
 8783 event not captured. Therefore, for our illustration in the previous table,  $y_{i1} = 2$ ,  $y_{i2} = 7$ ,  
 8784  $y_{i3} = 4$  and so on.

8785 For simulating and fitting data in the **BUGS** engines we will typically use the cat-  
 8786 egorical representation of the model because it is somewhat more convenient. We have  
 8787 found that fitting multinomial models in **WinBUGS** is less efficient than **JAGS** (Royle  
 8788 and Converse, 2012), which we use in the subsequent examples involving multinomial  
 8789 observation models.

### 8790 9.2.1 Multinomial Resource Selection Models

8791 The multinomial probabilities in Eq. 9.2.2 look similar to the multinomial resource selec-  
 8792 tion function (RSF) model for telemetry data (Manly et al., 2002; Lele and Keim, 2006).  
 8793 This suggests how we might model landscape or habitat covariates using such methods  
 8794 – i.e., by including them as explicit covariates in a larger multinomial model for “use” –  
 8795 which, if we take the product of use with encounter produces a model for the observable  
 8796 encounter data. This leads naturally to the development of models that integrate RSF  
 8797 data from telemetry studies with SCR data (Royle et al., 2012a), which is the topic of  
 8798 Chapt. 11.

---

**8799 9.2.2 Simulating data and analysis using JAGS**

8800 We're going to show the nugget of a simulation function which is used in the function  
 8801 `simMnSCR` found in the **R** package `scrbook`. The first lines of the following **R** code make  
 8802 use of some things that you need to define, but we omit them here (e.g., `xlim`, `ylim` are  
 8803 the boundaries of the state-space, `N` is the population size, etc.):

```

8804 ##
8805 ## Simulate random activity centers:
8806 ## (first define N, xlim, ylim, etc..)
8807 ##
8808 S<-cbind(runif(N,xlim[1],xlim[2]),runif(N,ylim[1],ylim[2]))
8809
8810 ## Distance from each individual to each trap
8811 D<- e2dist(S,traplocs)
8812
8813 ## Set parameter values
8814 sigma<- 0.5
8815 alpha0<- -1
8816 alpha1<- -1/(2*sigma*sigma)
8817
8818 ## make an empty data matrix and fill it up with data
8819 Ycat<-matrix(NA,nrow=N,ncol=K)
8820 for(i in 1:N){
8821   for(k in 1:K){
8822     lp<- alpha0 + alpha1*D[i,]*D[i,]
8823     cp<- exp(c(lp,0))
8824     cp<- cp/sum(cp)
8825     Ycat[i,k]<- sample(1:(ntraps+1),1,prob=cp)
8826   }
8827 }
```

8828 The resulting data matrix in this case has the maximal dimension  $N$  and so, for  
 8829 analysis, to mimic a real situation, we would have to discard the uncaptured individuals.  
 8830 The function `simMnSCR` in the package `scrbook` will also simulate data that includes a  
 8831 behavioral response which will be the typical situation in small-mammal trapping problems  
 8832 (see Converse and Royle, 2012, for details).

8833 Here we use our function `simMnSCR` to simulate a data set with  $K = 7$  occasions.  
 8834 We'll run the model using JAGS which we have found is much more effective for this class  
 8835 of models. We get the data set-up for analysis by augmenting the size of the data set  
 8836 to  $M = 200$ . In addition we choose starting values for  $\mathbf{s}$  and the data augmentation  
 8837 variables  $z$ . For starting values of  $\mathbf{s}$  we cheat a little bit here and use the true values for  
 8838 the observed individuals and then augment the  $M \times 2$  matrix  $\mathbf{S}$  with  $M - n$  randomly  
 8839 selected activity centers. Our function `spiderplot` returns the mean observed location of  
 8840 individuals for use as starting values. The parameters input to `simMnSCR` are the intercept  
 8841  $\alpha_0$ ,  $\sigma = \sqrt{1/(2\alpha_1)}$  for the Gaussian encounter probability model, and  $\alpha_2$  is the behavioral  
 8842 response parameter. The data simulation and set-up proceeds as follows:

```
8843 set.seed(2013)
8844 parms<-list(N=100,alpha0= -.40, sigma=0.5, alpha2= 0)
8845 data<-simMnSCR(parms,K=7,ssbuff=2)
8846 nind<-nrow(data$Ycat)

8847 M<-200
8848 Ycat<-rbind(data$Ycat,matrix(nrow(data$X)+1,nrow=(M-nind),ncol=data$K))
8849 Sst <-rbind(data$S,cbind(runif(M-nind,data$xlim[1],data$xlim[2]),
8850 runif(M-nind,data$ylim[1],data$ylim[2])))
8851 zst<-c(rep(1,160),rep(0,40))
```

8853 The model specification is not much more complicated than the binomial or Poisson  
 8854 models given previously. The main consideration is that we define the cell probabilities for  
 8855 each trap  $j = 1, 2, \dots, J$  and then define the last cell probability,  $J + 1$ , for “not captured”,  
 8856 to be the complement of the sum of the others. The code is shown in Panel 9.1. In the  
 8857 last lines of code here we specify  $N$  and density,  $D$ , as derived parameters.

```

cat("
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
}
",file="model.txt")

```

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`.

8858 To fit the model, we need to package everything up (inits, parameters, data) and send  
 8859 it off to **JAGS** to build an MCMC simulator for us (these commands are executed in  
 8860 the help file for **simMnSCR**). In addition to the usual data objects, we also pass the limits  
 8861 of the assumed rectangular state-space (**ylim**, **xlim**, both  $1 \times 2$  vectors) and the scale of  
 8862 the standardized units, called **trap.space** here because we typically will define the trap  
 8863 coordinates to be an integer grid. If the trap spacing is 10 m and we want units of density  
 8864 computed in terms of individuals per meter, then we input **trap.space=10**. The analysis  
 8865 is carried out as follows:

```
8866 inits <- function(){ list (z=zst,sigma=runif(1,.5,1) ,S=Sst) }
8867
8868 # parameters to monitor
8869 parameters <- c("psi","alpha0","alpha1","sigma","N","D")
8870
8871 # bundle the data. Note this reuses "data"
8872 data <- list (X=data$X,K=data$K, trap.space=1,Ycat=Ycat,M=M,
8873 ntraps=nrow(data$X),ylim=data$ylim,xlim=data$xlim)
8874
8875 library(R2jags)
8876 out <- jags (data, inits, parameters, "model.txt", n.thin=1,
8877 n.chains=3, n.burnin=1000, n.iter=2000)
```

8878 The posterior summaries are provided in the following **R** output (recall that  $N = 100$ ,  
 8879  $\alpha_0 = -0.40$ , and  $\sigma = 0.5$ ):

```
8880 Inference for Bugs model at "model.txt", fit using jags,
8881 3 chains, each with 2000 iterations (first 1000 discarded)
8882 n.sims = 3000 iterations saved
8883      mu.vect sd.vect   2.5%    25%    50%    75%   97.5% Rhat n.eff
8884 D       1.873  0.189  1.531  1.750  1.859  2.000  2.250 1.006 1300
8885 N      119.867 12.107 98.000 112.000 119.000 128.000 144.000 1.006 1300
8886 alpha0   -0.435  0.151 -0.738 -0.535 -0.439 -0.331 -0.146 1.004  580
8887 alpha1   -2.195  0.286 -2.785 -2.372 -2.180 -2.004 -1.658 1.003 2800
8888 psi      0.599  0.069  0.465  0.552  0.599  0.645  0.739 1.006 1400
8889 sigma    0.480  0.032  0.424  0.459  0.479  0.500  0.549 1.003 2400
8890 deviance 892.164 21.988 850.922 877.417 891.561 906.246 937.728 1.003  950
8891
8892 [... output deleted ....]
```

### 8893 9.2.3 Multinomial Relationship to Poisson

8894 The multinomial is related to the Poisson encounter rate model by a conditioning argument.  
 8895 Let  $y_{ij}$  be the number of encounters for individual  $i$  in trap  $j$ . If  $y_{ij} \sim \text{Poisson}(\lambda_{ij})$ ,  
 8896 then, conditional on the total number of captures (i.e., across all traps),  $y_i = \sum_j y_{ij}$ , the  
 8897 trap encounter frequencies are multinomial with probabilities

$$\pi_{ij} = \frac{\lambda_{ij}}{\sum_j \lambda_{ij}}$$

8898 for  $j = 1, 2, \dots, J$ . Or equivalently the *trap of capture* is categorical with probabilities  $\pi_{ij}$   
 8899 as given above. Under the Gaussian kernel model, these probabilities are:

$$\pi_{ij} = \frac{\exp(\alpha_1 d(\mathbf{x}_i, \mathbf{s}_j)^2)}{\sum_j \exp(\alpha_1 d(\mathbf{x}_i, \mathbf{s}_j)^2)} \quad (9.2.2)$$

8900 where, we note, the intercept  $\alpha_0$  has canceled from both the numerator and denominator.  
 8901 This makes sense because, here, these probabilities describe the trap-specific capture prob-  
 8902 abilities *conditional on capture*. Therefore, the model is not completely specified, absent  
 8903 a model for the “overall” probability of encounter or the expected frequency of captures,  
 8904 say  $\phi_i$ . Depending on how we specify a model for this quantity  $\phi_i$ , we can reconcile it  
 8905 directly with the Poisson model. Let  $y_{ij}$  be the total number of encounters for individual  
 8906  $i$  and suppose  $y_{ij}$  has a Poisson distribution with mean  $\phi_i$ . Then, marginalizing Eq. 9.2.1  
 8907 over the Poisson distribution for  $y_{ij}$  produces the original set of *iid* Poisson frequencies  
 8908 with probabilities:

$$\lambda_{ij} = \phi_i \pi_{ij}$$

8909 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  
 8910 the marginal distribution of  $y_{ij}$  is Poisson with mean  $\exp(\alpha_0 + \alpha_1 d(\mathbf{x}, \mathbf{s})^2)$ , equivalent to  
 8911 Eq. 9.1.1.

8912 In summary, the Poisson and multinomial models are equivalent in how they model  
 8913 the distribution of captures among traps. It stands to reason that, if the encounter  
 8914 rate of individuals is low, we could use the Poisson and multinomial models interchange-  
 8915 ably. In fact, based on our discussion in Sec. 9.1.2 above we could use any of the bino-  
 8916 mial/Poisson/multinomial models with little ill-effect when encounter rate is low.

### 9.3 AVIAN MIST-NETTING EXAMPLE

8917 We analyze data from a mist-netting study of ovenbirds, conducted at the Patuxent  
 8918 Wildlife Research Center, Laurel MD, by DK Dawson and MG Efford. The data from  
 8919 this study are available in the **secr** package, and have been analyzed previously by Efford  
 8920 et al. (2004), see also Borchers and Efford (2008). Forty-four mist nets spaced 30 m apart  
 8921 on the perimeter of a 600-m x 100-m rectangle were operated on 9 or 10 non-consecutive  
 8922 days in late May and June for 5 years from 2005-2009. From the **secr** documentation of  
 8923 this data set we take this:

8924 *Netting was passive (i.e. song playback was not used to lure birds into the nets).*  
 8925 *Birds received individually numbered bands, and both newly banded and previously*  
 8926 *banded birds were released at the net where captured. Sex was determined in the*  
 8927 *hand from the presence of a brood patch (females) or cloacal protuberance (males).*

8928 The ovenbird data can be loaded as follows:

```
8929 library(secr)
8930 data(ovenbird)
```

8931 The dataset consists of adult ovenbirds caught during sampling in each of 5 years, 2005-  
 8932 2009. (one ovenbird was killed in 2009, indicated by a negative net number in the encounter  
 8933 data file). As with most mist-netting studies, nets are checked multiple times during a  
 8934 day (e.g., every hour during a morning session). However, for this data set, the within-day

recaptures are not included so each bird has at most a single capture per day. Therefore the multinomial model (detector type ‘multi’ in **secr**) is appropriate. Although several individuals were captured in more than one year, this information is not used in the models presently offered in **secr**, but we do make use of it in the development of open models in Chapt. 15.

### 9.3.1 Multiple Sample Sessions

Up to this point we have only dealt with a basic closed population sampling situation consisting of repeated sample occasions of a single population of individuals using a single array of traps. In practice, many studies produce repeated samples over time, or at different locations – we adopt the **secr** terminology of *session* for such replication by groups of time or space, and the models are *multi-session* models, although we think of such models as being relevant to any stratified population (see Chapt. 16). We introduced **secr**’s multi-session models in Sec. 6.5.4. In the case of the ovenbird data, sampling was carried out in multiple years, with a number of sample occasions within each year (9 or 10), a type of data structure commonly referred to as “the robust design” (Pollock, 1982). In this context, it stands to reason that there is recruitment and mortality happening across years. In Chapt. 15 we model these processes explicitly but, here, we provide an analysis of the data that does not require explicit models for recruitment and survival, regarding the yearly populations as independent strata, and fitting a multi-session model.

What exactly is the multi-session model that **secr** fits? The model can be thought of as a type of open population model, either with explicit time periods that represent sessions, or by equating spatial units to sessions. A special case of open models arises when we assume  $N_t$  (time-specific population sizes) are independent from one time period or session to the next – this can be thought of as a “random temporary emigration” model of the Kendall et al. (1997) variety, and this is essentially the multi-session model implemented in **secr**. In particular, by assuming that  $N_t$  is Poisson with mean  $\Lambda_t$  one can model variation in abundance among sessions based on the Poisson-integrated likelihood in which parameters of  $\Lambda_t$  appear directly in the likelihood as we noted in Sec. 6.5.4. We provide an analysis (below) of the ovenbird data here using the multi-session models in **secr**. We formalize the multi-session model approach from a Bayesian perspective using data augmentation in Chapt. 16, based on Converse and Royle (2012) and Royle and Converse (2012).

A 3rd way to develop multi-session models which is convenient for **BUGS** is to regard the data from each session as an independent data set with its own  $N_t$  parameter, and do  $T$  distinct data augmentations. Because each  $N_t$  is regarded as a free parameter, independent of the other parameters, we’ll call this the nonparametric multi-session model. We can analyze this model in the normal context of data augmentation by augmenting each year separately in the same **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  $N_t$ .

We demonstrate all of the 3 approaches with the ovenbird data: In the following section we provide the nonparametric model with unconstrained  $N_t$ , and we demonstrate the model-based multi-session models from **secr** both here (following section) and in

8979 Chapt. 16 from a Bayesian standpoint. We address the fully open “Spatial Jolly-Seber”  
 8980 model in Chapt. 15,

### 8981 9.3.2 Analysis in JAGS

8982 The ovenbird data are provided as a multi-session `capthist` object `ovenCH` which, by  
 8983 regarding years as independent strata, allows for the fitting of the multi-session model.  
 8984 For doing a Bayesian analysis in one of the **BUGS** engines (we use **JAGS** here) there are  
 8985 a number of ways to structure the data and describe the model. We can analyze either  
 8986 a 2-d data set with all years (data augmented) “stacked” into a data set of dimension  
 8987  $(5 * M) \times 10$  ( $5$  years,  $M$  = size of the augmented data set,  $K = 10$  replicate sample  
 8988 occasions). Or, we could produce a 3-d array ( $M \times J \times K$ ). We adopted the former  
 8989 approach, analyzing the data as a 2-d array and creating an additional indicator variable  
 8990 for “year” to indicate which stratum (year) each record goes with. There was a single  
 8991 loss-on-capture which we accounted for by fixing  $p = 0$  for all subsequent encounters of  
 8992 that individual (indicated by the binary variable `dead`, as shown in Panel 9.2). We have  
 8993 an **R** script in `scrbook` package called `SCRovenbird`, so you can see how to set-up the data  
 8994 and run the model. Executing the script `SCRovenbird` produces the posterior summaries  
 8995 given in Table 9.2. Here, density is in units of birds per ha.

8996 Data on individual sex is included with `secr`, but we provide an analysis of a single  
 8997 model for all adults, constant  $\sigma$  across years, constant  $p_0$ , and year-specific values of  $N_t$   
 8998 (and hence  $D_t$ ). There is a habitat mask provided with the data but the mask appears  
 8999 to just be a modified rectangle around the net locations, clipped to have rounded corners,  
 9000 and so we don’t use it here.

### 9001 9.3.3 Analysis in secr

9002 Included with the ovenbird data are a number of models fitted as examples. Those include:

```
9003 ovenbird.model.1    fitted secr model -- null
9004 ovenbird.model.1b   fitted secr model -- g0 net shyness
9005 ovenbird.model.1T   fitted secr model -- g0 time trend within years
9006 ovenbird.model.h2   fitted secr model -- g0 finite mixture
9007 ovenbird.model.D   fitted secr model -- trend in density across years
```

9008 The model fits provided in `secr` use a habitat mask which is provided in that `secr`  
 9009 package. However, we refit all of the models here because we chose not to use the mask  
 9010 in our **JAGS** analysis of the previous section. The re-analysis proceeds as follows:

```
9011 ## fit constant-density model
9012 ovenbird.model.1 <- secr.fit(ovenCH)
9013 ## fit net avoidance model
9014 ovenbird.model.1b <- secr.fit(ovenCH, model = list(g0~b))
9015 ## fit model with time trend in detection
9016 ovenbird.model.1T <- secr.fit(ovenCH, model = list(g0 ~ T))
9017 ## fit model with 2-class mixture for g0
9018 ovenbird.model.h2 <- secr.fit(ovenCH, model = list(g0~h2))
```

---

```

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)
  }
  # bigM = total size of jointly augmented data set
  for(i in 1:bigM){
    z[i] ~ dbern(psi[year[i]])
    S[i,1] ~ dunif(xlim[1],xlim[2])
    S[i,2] ~ dunif(ylim[1],ylim[2])
  }
  # X = trap locations, S = activity centers
  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){
      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.** Inference for the ovenbird mist-netting data using the independent multinomial observation model. MCMC was done using jags with 3 chains, each with 5000 iterations, discarding the first 1000, for a total of 12000 posterior samples.

parameter	mu.vect	sd.vect	2.5%	50%	97.5%	Rhat	n.eff
D[1]	1.056	0.177	0.778	1.037	1.469	1.001	12000
D[2]	1.109	0.174	0.835	1.095	1.498	1.002	1600
D[3]	1.302	0.187	0.979	1.296	1.728	1.002	2700
D[4]	0.959	0.159	0.691	0.951	1.296	1.001	8000
D[5]	0.811	0.146	0.576	0.807	1.152	1.002	2000
N[1]	36.666	6.132	27.000	36.000	51.000	1.001	12000
N[2]	38.513	6.056	29.000	38.000	52.000	1.002	1600
N[3]	45.197	6.501	34.000	45.000	60.000	1.002	2700
N[4]	33.279	5.511	24.000	33.000	45.000	1.001	8000
N[5]	28.145	5.071	20.000	28.000	40.000	1.002	2000
alpha0	-3.528	0.164	-3.853	-3.530	-3.208	1.002	1700
alpha1	0.000	0.000	0.000	0.000	0.000	1.002	2400
psi[1]	0.369	0.077	0.235	0.363	0.535	1.001	12000
psi[2]	0.388	0.076	0.252	0.382	0.549	1.002	2400
psi[3]	0.453	0.080	0.307	0.448	0.622	1.001	3300
psi[4]	0.336	0.071	0.210	0.332	0.488	1.001	12000
psi[5]	0.286	0.066	0.170	0.281	0.429	1.002	1600
sigma	75.781	6.016	65.443	75.359	88.993	1.002	2400

```
9019 ## fit a model with session (year)-specific Density
9020 ovenbird.model.DT <- secr.fit(ovenCH, model = list(D~session))
```

9021 All of these can be fitted easily in **JAGS** but we only fitted the first one in the previous  
 9022 section, so we will compare our results from **JAGS** to the model `ovenbird.model.1`, the  
 9023 null model with no covariates. The `secr` output is extensive and so we do not reproduce it  
 9024 completely here. By default, it summarizes the trap information for each year, encounter  
 9025 information, and then output for each year. Here is an abbreviated version for the null  
 9026 model:

```
9027 > ovenbird.model.1
9028
9029 secr.fit( capthist = ovenCH )
9030 secr 2.3.1, 21:58:26 20 Sep 2012
9031
9032 $`2005`
9033 Object class      traps
9034 Detector type    multi
9035 Detector number  44
9036 Average spacing  30.27273 m
9037 x-range          -50 49 m
```

```

9038 y-range           -285 285 m
9039
9040 [... deleted ....]
9041
9042
9043          2005 2006 2007 2008 2009
9044 Occasions      9   10   10   10   10
9045 Detections    35   42   52   30   33
9046 Animals       20   22   26   19   16
9047 Detectors     44   44   44   44   44
9048
9049 [... deleted ....]
9050
9051 session = 2005
9052      link   estimate SE.estimate      lcl      ucl
9053 D      log  1.25448819 0.137674225  1.01234988  1.5545422
9054 g0    logit 0.02384623 0.003647758  0.01765135  0.0321441
9055 sigma  log 76.58675001 5.911524450 65.84900819 89.0754536
9056
9057 [... deleted ...]

```

9058 To do model selection we use the handy helper-function **AIC** as follows (output edited  
9059 to fit on the page):

```

9060 AIC (ovenbird.model.1, ovenbird.model.1b, ovenbird.model.1T,
9061 ovenbird.model.h2, ovenbird.model.D)
9062
9063      model detectfn npar logLik      AIC      AICc      dAICc
9064 ovenbird.model.1T [edited output]  4 -1113.429 2234.858 2235.266  0.000
9065 ovenbird.model.1b [edited output]  4 -1120.161 2248.323 2248.731 13.465
9066 ovenbird.model.h2 [edited output]  5 -1119.337 2248.674 2249.293 14.027
9067 ovenbird.model.D  [edited output]  4 -1120.747 2249.495 2249.903 14.637
9068 ovenbird.model.1  [edited output]  3 -1122.650 2251.299 2251.542 16.276

```

9069 We see that the null model is way down at the bottom of the list, and the model  
9070 with year-specific density is just ahead of that. Instead, the model with a time-trend  
9071 (within-season) in detection probability is preferred, followed by a behavioral response.  
9072 We encourage you to adapt the **JAGS** model specification for such models which is easily  
9073 done (see Chapt. 8 for many examples). We provide the summary results for the model  
9074 having **D ~ session** as follows:

```

9075 Results for "D[session]" model:
9076
9077 session = 2005, Session = 0
9078      link   estimate SE.estimate      lcl      ucl
9079 D      log  1.03219052 0.199449228  0.70923143  1.50221386
9080 g0    logit 0.02760077 0.004042557  0.02069122  0.03673114
9081 sigma  log 78.57955355 6.387162285 67.02478446 92.12631245

```

```

9082
9083 session = 2006, Session = 1
9084     link estimate SE.estimate      lcl      ucl
9085 D      log  0.96834386 0.187112197  0.66536156  1.40929367
9086 g0    logit 0.02760077 0.004042557  0.02069122  0.03673114
9087 sigma  log 78.57955355 6.387162285 67.02478446 92.12631245
9088
9089 session = 2007, Session = 2
9090     link estimate SE.estimate      lcl      ucl
9091 D      log  0.90844647 0.175538278  0.62420529  1.32212111
9092 g0    logit 0.02760077 0.004042557  0.02069122  0.03673114
9093 sigma  log 78.57955355 6.387162285 67.02478446 92.12631245
9094
9095 session = 2008, Session = 3
9096     link estimate SE.estimate      lcl      ucl
9097 D      log  0.85225406 0.164680271  0.58559476  1.24034065
9098 g0    logit 0.02760077 0.004042557  0.02069122  0.03673114
9099 sigma  log 78.57955355 6.387162285 67.02478446 92.12631245
9100
9101 session = 2009, Session = 4
9102     link estimate SE.estimate      lcl      ucl
9103 D      log  0.79953746 0.154493890  0.54937251  1.16361876
9104 g0    logit 0.02760077 0.004042557  0.02069122  0.03673114
9105 sigma  log 78.57955355 6.387162285 67.02478446 92.12631245

```

9106 The point estimates (MLEs) of density are uniformly lower than the Bayesian estimates  
9107 (posterior means). We expect some difference in this direction due to small-sample skew  
9108 of the posterior, although for year 2 and 3 the difference is even in the opposite relative  
9109 direction which is strange. The estimated  $\sigma$  is very similar between the **JAGS** analysis and  
9110 **secr**. Finally, you will find that these estimates from **secr** are a bit different from those  
9111 obtained by using the habitat mask. This is surprising because the mask is a rectangle  
9112 with the corners rounded which we would not imagine should influence the results<sup>1</sup>.

## 9.4 SINGLE-CATCH TRAPS

9113 The classical animal trapping experiment is based on a physical trap which captures a  
9114 single animal and holds that individual until subsequent molestation by a biologist. This  
9115 type of observation model – the “single-catch” trap – was the original situation considered  
9116 in the context of spatial capture-recapture by Efford (2004). Nowadays, capture-recapture  
9117 data are more often obtained by other methods (DNA from hair snares, or scat sampling,  
9118 camera traps etc...) but nevertheless the single-catch traps are still widely used in small  
9119 mammal studies (Converse et al., 2006b; Converse and Royle, 2012) and other situations.

9120 The single-catch model is basically a multinomial model but one in which the number  
9121 of available traps is reduced as each individual is captured. As such, the constraints on the  
9122 joint likelihood for the sample of  $n$  encounter histories are very complicated. As a result,

---

<sup>1</sup> Possibly the habitat mask is not sufficiently extensive.

at the time of this writing, there has not been a formal development of either likelihood or Bayesian analysis of this model and applications of SCR models to single-catch systems have used the independent multinomial model as an approximation (see below).

Nevertheless, we can make some progress to describing the basic observation model formally. In particular, if we imagine that all of the individuals captured queued up at the beginning of the capture session to draw a number indicating their order of capture, then there is a nice conditional structure resulting from a “removal process” operating on the traps. The first individual captured has the multinomial observation model:

$$\mathbf{y}_1 \sim \text{Multinom}(\boldsymbol{\pi}_1)$$

whereas the 2nd guy captured also has a multinomial encounter probability model but with the trap which captured the first individual removed. We might express this as:

$$\mathbf{y}_2 \sim \text{Multinom}(\boldsymbol{\pi}_2)$$

where

$$\pi_{2j} = \frac{(1 - y_{1j}) * \exp(\alpha_0 + \alpha_1 d_{ij}^2)}{\sum_j (1 - y_{1j}) * \exp(\alpha_0 + \alpha_1 d_{ij}^2)}$$

and so on for  $i = 3, 4, \dots, n$ . In a certain way, this model is a type of local behavioral response model but where the response is to other individuals being captured. Evidently, the **order of capture** is relevant to the construction of these multinomial cell probabilities. More generally, the *time* of capture of an individual in any trapping interval will affect the encounter probability of subsequently captured individuals, but we think that order of capture might lead to a practical approximation to the single-catch process (this is how we simulate the data in our function `simScSCR`). In the simulation of single catch data, we randomly ordered the population of individuals for each sample occasion, and then cycled through them, turning off each trap if an individual was captured in it.

#### 9.4.1 Inference for single-catch systems

For the single-catch model, we argued that the observations have a multinomial type of observation model, but the multinomial observations have a unique conditional dependence structure among them owing to the “removal” of traps as they fill-up with individuals. Thus, competition for single-catch traps renders the independence assumptions for the independent multinomial model invalid. However, as Efford et al. (2009a) noted, we expect “bias to be small when trap saturation (the proportion of traps occupied) is low. Trap saturation will be higher when population density is high...” relative to trap density, or when net encounter probability is high. Efford et al. (2009a) did a limited simulation study and found essentially no effective bias and concluded that estimators of density from the misspecified independent multinomial model are robust to the mild dependence induced when trap saturation is low. Naturally then, we expect that the Poisson model could also be an effective approximation under the same set of circumstances.

In the **R** package `scrbook` we provide a function for simulating data from a single-catch system (function `simScSCR`) and fitting the misspecified model (`example(simScSCR)`) in **JAGS** so that you can evaluate the effectiveness of this misspecified model for situations that interest you.

---

**9160 9.4.2 Analysis of Efford's possum trapping data**

9161 We provide an analysis here of data from a study of brushtail possums in New Zealand.  
 9162 The data are available with the **R** package **secr** (Efford et al., 2009a); see the help file  
 9163 **?possum** after loading the **secr** package. Originally the data were analyzed by Efford et al.  
 9164 (2005), and a detailed description of the data set is available in the help file, from which  
 9165 we summarize:

9166     *Brushtail possums (*Trichosurus vulpecula*) are an unwanted invasive species in New*  
 9167     *Zealand. Although most abundant in forests, where they occasionally exceed densities*  
 9168     *of 15/ha, possums live wherever there are palatable food plants and shelter.*

9169 To load the possum data, execute the following commands:

9170 **library(secr)**  
 9171 **data(possum)**

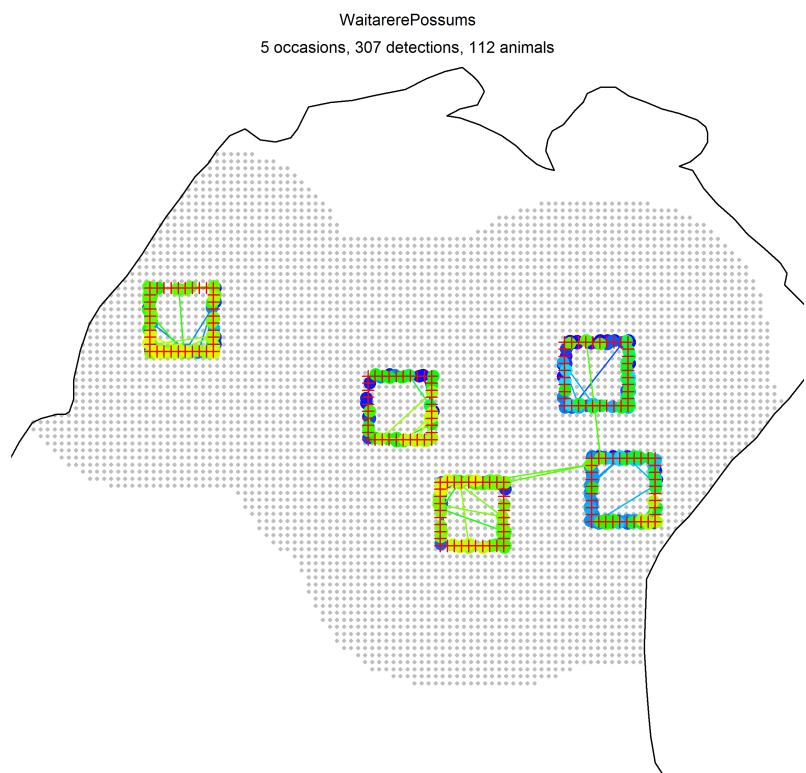
9172 The study area encompasses approximately 300 ha, and 180 live traps were organized  
 9173 in 5 distinct grids, shown in Fig. 9.2. Each square arrangement of traps consisted of 36  
 9174 traps with a spacing of 20 m. Thus the squares are 180 m on a side. Individuals were  
 9175 captured, tagged, and released over 5 days during April, 2002. A noteworthy aspect of this  
 9176 study is that it involves replicated grids selected in some fashion from within a prescribed  
 9177 region. From an analysis standpoint, we could adopt the use of the multi-session models  
 9178 which we used previously to analyze the ovenbird data. This would be useful if we had  
 9179 covariates at the trapping grid level that we wanted to model. Alternatively, we could  
 9180 pool the data from all of the grids and analyze them jointly as if they were based on a  
 9181 single trapping grid which is clearly a reasonable view in this case. In doing this sort  
 9182 of pooling, there is an implicit assumption that  $N_t$  is Poisson distributed, with constant  
 9183 mean (Royle, 2004b; Royle et al., 2012b) which we also address in Chapt. 16.

9184 The data file **possumCH** contains 112 encounter histories, and we analyze those here  
 9185 although the last 8 of those are recaptures treated as new individuals<sup>2</sup>. The encounter  
 9186 process is not strictly a single-catch multinomial process because, as noted in the **possum**  
 9187 help file “One female possum was twice captured at two sites on one day, having entered  
 9188 a second trap after being released; one record in each pair was selected arbitrarily and  
 9189 discarded.” which is a similar situation to what might happen in bird mist net studies,  
 9190 as a bird might fly into a net upon release from another. If this was a significant problem  
 9191 then it might be worth describing a model for  $n_{ik}$  = the number of captures of individual  $i$   
 9192 during sample occasion  $k$  to make use of all captures. By discarding the two extra-capture  
 9193 events, we can satisfactorily view these data as single-catch data, for which **secr** uses the  
 9194 independent multinomial likelihood (M. Efford, pers. comm.).

9195 For our Bayesian analysis here, we used a rectangular state-space which doesn't ac-  
 9196 count for any geographic boundaries of the survey region, but we note that a habitat mask  
 9197 is included in **secr** and it could be used in a Bayesian analysis. Whether or not we use the  
 9198 mask is probably immaterial as long as we understand the predictions of  $N$  or  $D$  over the  
 9199 water don't mean anything biological and we probably wouldn't report such predictions.  
 9200 Shortly we will show how to make a density map that is suitably restricted. The **JAGS**  
 9201 model specification is based on that of the ovenbird analysis given previously, and so we  
 9202 don't reproduce the model here. The **R/JAGS** script called **SCRpossum** which is in our  
 9203 **scrbook** package. The results are summarized in Tab. 9.3.

---

<sup>2</sup>M. Efford, personal communication



**Figure 9.2.** Trapping grids used in possum study from Efford et al. (2005), data are contained in the R package `secr` (Efford, 2011), refer to the help file `?possum` for additional details of this study.

**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. 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
D	0.000	0.000	0.000	0.000	0.000	1.009	340
Dha	1.549	0.115	1.343	1.547	1.777	1.009	340
N	235.407	17.435	204.000	235.000	270.000	1.009	340
alpha0	-0.935	0.167	-1.270	-0.934	-0.605	1.007	870
alpha1	0.000	0.000	0.000	0.000	0.000	1.001	2800
psi	0.783	0.062	0.666	0.782	0.903	1.008	340
sigma	52.020	2.675	47.067	51.933	57.585	1.001	2800

9204        The estimated density (posterior mean) is about 1.53 possums/ha. To obtain the **secr**  
 9205        results for the equivalent null model, we execute the following command

9206        `secr.fit( capthist = possumCH, trace = F )`

9207        which produces (edited) summary output:

9208        [... some output deleted ...]

9209

9210        Fitted (real) parameters evaluated at base levels of covariates

9211        link estimate SE.estimate        lcl        ucl

9212        D        log    1.6988930    0.17352645    1.3913904    2.0743547

9213        g0        logit    0.1968542    0.02256272    0.1563319    0.2448321

9214        sigma    log    51.4689114    2.59981905    46.6204139    56.8216500

9215

9216        [... some output deleted ...]

9217        As we’ve discussed previously, there are many reasons for why there might be differences  
 9218        between Bayesian and likelihood estimates. But even among likelihood estimates – any  
 9219        time you run a model there is some numerical integration going on which requires some  
 9220        specific choices of how to do the integration (see Chapt. 6). For now we just observe that  
 9221        the estimated density is certainly in the ballpark and so too is the estimated  $\sigma$ .

## 9.5 ACOUSTIC SAMPLING

9222        The last decade has seen an explosion of technology that benefits the study of animal  
 9223        populations. This includes DNA sampling methods that allow for identification from hair  
 9224        or scat, camera trapping and identification software that allow efficient sampling of many  
 9225        carnivores, and the resulting statistical technology that helps us to make sense of such data  
 9226        (Borchers and Efford, 2008; Royle and Young, 2008; Efford et al., 2009b; Gopalaswamy  
 9227        et al., 2012b; Sollmann et al., 2012b; Chandler and Royle, In press). One other extremely  
 9228        promising technology area is that of acoustic sampling using microphones or recording

9229 devices. That is, instead of having cameras record data, or humans pick up scat, we can  
 9230 establish an array of (usually) electronic recording devices which, instead of establishing  
 9231 a visual identity of individuals, they record a vocal expression of each individual. In this  
 9232 context, Efford et al. (2009b) referred to audio recorders as “signal strength proximity  
 9233 detectors” to distinguish them from other types of proximity detections, including camera  
 9234 traps, which are *visual* proximity detector. Using audio records, the spatial pattern of  
 9235 the *signal strength* at the different microphones can be used for inference about density  
 9236 (Dawson and Efford, 2009; Efford et al., 2009b) in the same way as the spatial pattern  
 9237 of detections is used in the types of SCR models we have discussed so far. The basic  
 9238 technical formulation of these models comes from Efford et al. (2009b), and it was applied  
 9239 to field study of birds by Dawson and Efford (2009). In that study, recording devices were  
 9240 organized in groups of 4 (in a cross pattern), with an array of  $5 \times 15$  such cross-patterns,  
 9241 separated by 100 m (300 total recorder locations). This data set is provided with the **secr**  
 9242 package along with some sample analyses and help files. See Efford and Dawson (2010),  
 9243 a version of the document **secr-sound.pdf** that comes with the **secr** package, which you  
 9244 can access directly from the main help file (**?secr**).

9245 Our development here mostly follows Efford et al. (2009b), but we change some nota-  
 9246 tion to be consistent with our previous material. Let  $S(\mathbf{x}, \mathbf{u})$  be the strength of a signal  
 9247 emanating from location  $\mathbf{u}$ , as recorded by a device at location  $\mathbf{x}$ . Just as ordinary SCR  
 9248 models represent a model of *encounter frequency* as a function of distance, in acoustic  
 9249 models, the acoustic SCR model is a model of sound attenuation as a function of distance.  
 9250 In particular, the acoustic models assumes that  $S$  (or a suitable transformation) declines  
 9251 with distance  $d$  from the origin of the sound, to the recording device. In the context of  
 9252 spatial sampling of animals, the origin is the actual location of some individual animal,  
 9253 and the recording device is something we nailed to a tree, or put on a stick somewhere, or  
 9254 whatever. For example, a model of sound attenuation used by Dawson and Efford (2009)  
 9255 is the following:

$$S(\mathbf{x}, \mathbf{u}) = \alpha_0 + \alpha_1 d(\mathbf{x}, \mathbf{u}) + \epsilon \quad (9.5.1)$$

9256 where  $\epsilon \sim \text{Normal}(0, \sigma_s^2)$ . In many standard situations,  $S$  will be measured in decibels,  
 9257 which can be any value on the real line. In the conduct of acoustic sampling and the  
 9258 development of custom models for your own situation, it would probably be helpful to  
 9259 know something about sound dynamics and signal processing, although we don’t know  
 9260 much. In this model, the parameters  $\alpha_0$ ,  $\alpha_1$  and  $\sigma_s^2$  are to be estimated. We abbreviate  
 9261 the set of parameters by  $\boldsymbol{\theta}$  for short.

9262 We assert that an individual is detected if  $S$  exceeds a threshold,  $c$ . The reason  
 9263 for introducing this threshold  $c$  is that sound recorders will always record some sound,  
 9264 and so effective use of the acoustic SCR models requires specification of the threshold of  
 9265 measured signal below which the record is censored (non-detection occurs) because the  
 9266 recorded sound is assumed to be background noise. So we assert that an individual is  
 9267 detected if  $S > c$  which occurs with probability  $\Pr(S > c)$ , the encounter probability. To  
 9268 expand on and formalize this, let  $S_{ij}$  be the observed value of  $S$  for animal  $i$  at detector  $j$ .  
 9269 The encounter probability is  $\Pr(S_{ij} > c)$  which is  $\Pr(S_{ij} > c) = 1 - \Pr(S_{ij} < c)$ , so that,  
 9270 if we standardize the variate we have

$$1 - \Pr\left(\frac{(S_{ij} - \mathbb{E}(S))}{\sigma_s} < \frac{(c - \mathbb{E}(S))}{\sigma_s}\right)$$

9271 This probability calculation is the CDF of a standard normal variate say  $\eta = (S_{ij} - \mathbb{E}(S))/\sigma_s$  is less than the quantity  $\gamma(\boldsymbol{\theta}) = (c - \mathbb{E}(S))/\sigma_s$  which is a function of all the  
 9272 parameters  $\alpha_0$ ,  $\alpha_1$ ,  $\sigma_s^2$  and also the individual location  $\mathbf{u}$  and trap location  $\mathbf{x}$ . We'll  
 9273 identify it by  $\gamma(\boldsymbol{\theta}, \mathbf{x}, \mathbf{u})$  when we need to be explicit about those things. We can compute  
 9274  $\Pr(S_{ij} > c) = 1 - \Pr(\eta < \gamma(\boldsymbol{\theta}, \mathbf{x}, \mathbf{u}))$  easily using any software package including **R** which  
 9275 has a standard function, **pnorm**, for computing the normal cdf. To be more precise, we'll  
 9276 use the  $\Phi()$  to represent the normal cdf. Therefore, an individual is encountered whenever  
 9277  $S_{ij} > c$  which happens with probability  $\Pr(S_{ij} > c) = 1 - \Phi(\gamma(\boldsymbol{\theta}, \mathbf{x}, \mathbf{u}))$ .

9278 Naturally this quantity should depend on *where* an individual is located at the time  
 9279 of recording – what we call its instantaneous location, say  $\mathbf{u}$ , to distinguish it from its  
 9280 home-range center  $\mathbf{s}_i$  (but we outline a model below that contains both  $\mathbf{u}$  and  $\mathbf{s}$ ), and  
 9281 also the trap  $\mathbf{x}$ , so we index the quantity  $\gamma$  by those two quantities, in addition to the  
 9282 parameters  $\alpha_0$ ,  $\alpha_1$  and  $\sigma_s$ . The probability of detection is therefore

$$p_{ij} = p(\alpha_0, \alpha_1, \sigma_s | \mathbf{x}_j, \mathbf{u}_i) = 1 - \Phi(\gamma(\cdot))$$

9283 where  $\mathbf{u}_i$  is the instantaneous location of individual  $i$  and  $\mathbf{x}_j$  is the location of trap  $j$ .  
 9284 We'll suppose here that the random variables  $\mathbf{u}_i$  have state-space  $\mathcal{S}$ .

9285 How do we interpret this probability? Well, two things have to happen for an individual  
 9286 to be encountered by a trap: (1) it has to vocalize; (2) the microphone has to record a  
 9287 signal  $> c$ . These two things together are a product of biological and environmental factors  
 9288 which could include time of day, wind direction and speed, or maybe rain, humidity and  
 9289 other things. The bottom line is a lot of factors are balled up in whether or not the  
 9290 microphone records a sound greater than the threshold. In addition, we have to suppose  
 9291 that the vocalizations are reconcilable and reconciled successfully by the investigator.

9292 The observations from an acoustic survey are the signal strength measurements, and  
 9293 the likelihood of the observed signal strength from individual  $i$  at detection device  $j$  can  
 9294 be specified by noting that the likelihood is the normal pdf for the observed signal *if* the  
 9295 signal strength is  $> c$  and, otherwise, the contribution to the likelihood is  $\Phi(\gamma(\cdot))$  (see Eq.  
 9296 8 from Efford et al. (2009b)):

$$\Pr(S_{ij} | \mathbf{u}_i) = \Phi(\gamma(\cdot))^{1 - 1(S_{ij} > c)} \text{Normal}(S_{ij}; \alpha_0, \alpha_1, \sigma_s, \mathbf{x}_j, \mathbf{u}_i)^{1(S_{ij} > c)}$$

9297 We can use this as the basis for constructing the binomial-form of the likelihood as  
 9298 we did in Chapt. 6, which involves the number of individuals not captured,  $n_0$ . The  
 9299 probability that an individual is *not* captured is equal to the probability that its signal  
 9300 strength doesn't exceed  $c$  at any microphone. The probability of not being captured at a  
 9301 microphone  $\mathbf{x}_j$  is:

$$1 - p_{\mathbf{u}, j} = \Phi(\gamma(\cdot))$$

9302 and therefore the probability of not being captured at any microphone is:

$$\Pr(\text{all } S < c | \mathbf{u}) = \prod_{j=1}^J (1 - p_{\mathbf{u}, j}) = \prod_{j=1}^J \Phi(\gamma(\cdot, \mathbf{x}_j, \mathbf{u}))$$

9303 and therefore the marginal probability of not being captured is

$$\pi_0 = [\text{all } S < c | \boldsymbol{\alpha}] = \int_{\mathcal{S}} \left\{ \prod_{j=1}^J \Phi(\gamma(\boldsymbol{\theta}, \mathbf{x}_j, \mathbf{u})) \right\} d\mathbf{u}$$

9305 which can be used to construct the binomial form of the likelihood as we did in Chapt. 6  
 9306 (see Eq. 6.2.1).

9307 **9.5.1 Implementation in secr**

9308 Fitting acoustic encounter models in **secr** is no more difficult than other SCR models.  
 9309 There is a handy manual (**secr-sound.pdf**) with examples (Efford and Dawson, 2010)  
 9310 which comes with the **secr** package. The basic process is that **make.capthist** will make a  
 9311 **capthist** object from a 3-d encounter array – which is a binary array indicating whether  
 9312 each individual was detected or not at each recorder/microphone. In the case of acoustic  
 9313 data, # occasions = 1 because the recorders obtain data for a single interval. The “signal”  
 9314 attribute of the **capthist** object contains the signal strength in decibels. The best way to  
 9315 include the signal attribute is to use **make.capthist** in the usual way, providing it with  
 9316 the encounter data and trap data and, in addition, the variable “cutval” (which is *c* in  
 9317 our notation above) and then provide the signal strength data as an extra column of the  
 9318 **capthist** object. See **?make.capthist** for details.

9319 **9.5.2 Implementation in BUGS**

9320 We don’t know of any Bayesian applications of acoustic SCR models, although we imagine  
 9321 that implementation of such models in the **BUGS** engines should be achievable. It seems  
 9322 easy enough to write down a general hierarchical model that would accomodate sampling  
 9323 on repeated occassions. Let  $\mathbf{s}_i$  be the home range center, and let  $\mathbf{u}_{ik}$  the instantaneous  
 9324 location of individual *i* during sample occasion *k* (see Chapt. 17 for similar models). The  
 9325 model for  $\mathbf{u}_{ik}$  can be specified conditional on  $\mathbf{s}_i$ . For example, we could assume that  $\mathbf{u}_{ik}$   
 9326 are bivariate normal draws with mean  $\mathbf{s}_i$  and some variance  $\sigma_u^2$ . Then, conditional on  $\mathbf{u}_{ik}$   
 9327 an individual produces a signal according to the signal attenuation model (Eq. 9.5.1), or  
 9328 perhaps some other model. Then we generate the binary encounter data by truncating the  
 9329 observed signal at *c*. This general model then is an example of an SCR model in which  
 9330 parameters of a movement model are identifiable (see Sec. 2.6) because there is direct  
 9331 information about movement outcomes from the sampling method, unlike other types of  
 9332 encounter methods (e.g., camera traps) for which animal locations are restricted to a set of  
 9333 fixed, pre-determined points where traps are located. Other types of SCR methods allow  
 9334 for movement information too, including some of the search-encounter models (Chapt.  
 9335 17).

9336 Instead of developing a Bayesian version of this model here, we leave it to the reader  
 9337 to explore simulating data and devising a Bayesian implementation of the acoustic model  
 9338 in one of the **BUGS** engines. Note that for a single occasion, you can simulate the data  
 9339 using the two stage model (having both **s** and **u**) or you can simulate **u** uniformly at the  
 9340 start.

9341 **9.5.3 Other types of acoustic data**

9342 Efford and Dawson (2010) noted that various other types of acoustic data might arise

for which SCR-like models would be useful<sup>3</sup>. For example, we could have devices that measure the *time of arrival* of a queue of some sort. Another example is that where we measure *direction* to a queue from multiple devices and do, effectively, a type of statistical triangulation to the multiple but unknown number of sources. This has direct relevance to types of double or multiple-observer sampling that people do in field studies of birds. Normally 2 observers stand in close proximity and record birds, reconciling their detections after data collection. An SCR-based formulation of the double-observer method has two observers (or more) standing some distance apart, e.g., 50 or 100 meters, and marking individual birds on a map (or at least a direction) and a time of detection. The SCR/double-observer method could be applied to such data.

## 9.6 SUMMARY AND OUTLOOK

In this chapter we extended SCR models to accommodate alternative observation models, including Poisson and multinomial models. Along with the binomial model described in Chapt. 5, this sequence of models will accommodate a substantial majority of contemporary spatial capture-recapture problems, including the 4 main types of encounter data: binary encounters, multinomial trials from “multi-catch” and “single-catch” (Efford, 2004, 2011; Royle and Gardner, 2011) trap systems, and Poisson encounter frequency data from devices that can record multiple encounters of the same individual at a device. We summarize the standard observation models and the corresponding `secr` terminology in Table 9.4. What we refer to as search-encounter (or area-search) models (see Chapt. 17) are distinct from most of the other classes in that the observation location can also be random (in contrast to traps, where the location is fixed by design). This auxiliary data is informative about an intermediate process related to movement (Royle and Young, 2008).

**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. 9.4	<code>single-catch</code>
Acoustic	Sec. 9.5	<code>signal</code>
Search-encounter	Chapt. 17	<code>polygon</code> (in part)

There is a need for other types of encounter models that arise in practice. We identify a few of them here, although we neglect a detailed development of them at the present time or, in some cases, put that off until later chapters: (1) Removal systems – Sometimes traps kill individuals and SCR models can handle that. This can be viewed as a kind of open model, with mortality only, and we handle such models (in part) in Chapt. 15; (2) There are models for which only specific summary statistics are observable (Chandler and Royle, In press; Sollmann et al., 2012b) which we cover in Chaps. 18- 19; (3) We can

<sup>3</sup>Some of the following is also related to material presented by D.L. Borchers at the ISEC 2012 conference in Norway.

9372 have multiple observation methods working together as in Gopalaswamy et al. (2012b)  
9373 which we summarize in Sec. XXXXX.

9374 There remains much research to be done to formalize models for certain observation  
9375 systems. For example, while we think one will usually be able to analyze single-catch  
9376 systems using the multi-catch model, or even the Bernoulli model if encounter probability  
9377 is sufficiently low, a formalization of the single-catch model would be a useful development  
9378 and, we believe, it should be achievable using one or another of the **BUGS** engines. In  
9379 addition, classical “trapping webs” (Anderson et al., 1983; Wilson and Anderson, 1985a;  
9380 Jett and Nichols, 1987; Parmenter and MacMahon, 1989; Link and Barker, 1994) have  
9381 been around for quite some time and it seems like they are amenable to formulation as  
9382 a type of SCR model although we have not pursued that development simply because  
9383 trapping webs are rarely used in practice.

9384  
9385

# 10

9386

## SAMPLING DESIGN

9387 Statistical design is recognized as an important component of animal population studies  
9388 (Morrison et al., 2008; Williams et al., 2002). There are probably few to no field biologists  
9389 who have never been in the situation where a problem with data could be traced back  
9390 to some flaw in study design. Commonly, design is thought of in terms of number of  
9391 samples to take, when to sample, methods of capture, desired sample size (of individuals),  
9392 power of tests, and related considerations. In the context of spatial sampling problems,  
9393 where populations of mobile animals are sampled by an array of traps or devices, there  
9394 are a number of critical design elements. Two of the most important ones are the spacing  
9395 and configuration of traps (or sampling devices) within the array. While conceptual and  
9396 heuristic design considerations have been addressed by a number of authors (e.g., Nichols  
9397 and Karanth, 2002, Chapt. 11), little formal analysis focused on spatial design of arrays  
9398 has been carried out. Bondrup-Nielsen (1983) investigated the effect of trapping grid size  
9399 (relative to animal home range area) on density estimates using a simulation study and  
9400 some authors have addressed trap spacing and configuration by sensitivity “re-analysis”  
9401 (deleting traps and reanalyzing; Wegge et al., 2004; Tobler et al., 2008). The scarcity of  
9402 simulation-based studies looking at study design issues is surprising, as it seems natural  
9403 to evaluate prescribed designs by Monte Carlo simulation in terms of their accuracy and  
9404 precision.

9405 Theoretically optimal (or extremely good) designs are useful to study in order to  
9406 understand how factors influence the design problem. However, often field studies face  
9407 logistic difficulties, especially when dealing with wide-ranging, rare and cryptic species  
9408 like large carnivores. The need to sample large areas with limited resources often forces  
9409 researchers to compromise between a study design that is optimized for data analysis and  
9410 a study design that is logically viable. In such studies, traps are often placed along  
9411 roads or rivers where a large number of traps can be accessed with relative ease. Unless  
9412 the study site is crossed by a network of roads or waterways, this will lead to a sampling  
9413 design that is much more ‘linear’ compared to an area-based design where traps are spread  
9414 out uniformly over the study site.

9415 In this chapter we recommend a general framework for evaluating specific design

9416 choices for SCR studies based on Monte Carlo simulation of specific design scenarios  
9417 based on trade-offs between available effort, funding, logistics and other practical consider-  
9418 ations – what we call *scenario analysis*. Many study design-related issues such as how  
9419 long to survey in order to obtain sufficient data, can be addressed with preliminary field  
9420 studies that will give you an idea of how much data you can expect to collect within a unit  
9421 of effort (a camera trap day or a point count survey, for example). But it is also always  
9422 useful to perform scenario analysis based on simulation before conducting the actual field  
9423 survey not only to evaluate the design in terms of its ability to generate useful estimates  
9424 of things, but also so that you have an expectation of what the data will look like as they  
9425 are being collected. This gives you the ability to recognize some pathologies and possi-  
9426 bly intervene to resolve issues before they render a whole study worthless. Suppose you  
9427 design a study to place 40 camera traps based on your expectations of parameter values  
9428 you obtained from a careful review of the literature, and simulation studies suggest that  
9429 you should get 3-5 captures of individuals per night of sampling. In the field you find  
9430 that you're realizing 0 or 1 captures per night and therefore you have the ability to sit  
9431 down and immediately question your initial assumptions and possibly take some remedial  
9432 action. Simulation evaluation of design *a priori* is therefore a critical element of any field  
9433 study.

9434 While we recommend scenario analysis as a general tool to understand your *expected*  
9435 *data* before carrying out a spatial capture-recapture study, it is possible to develop some  
9436 heuristics and even analytic results related to the broader problem of model-based spatial  
9437 design (Muller, 2007) using an explicit objective function based on the inference objective.

## 10.1 GENERAL CONSIDERATIONS

9438 Many biologists have experience with the design of natural resource surveys from a classical  
9439 perspective (Thompson, 2002; Cochran, 2007), a key feature of which involves sampling  
9440 space. That is, we identify a sample frame comprised of spatial units and we sample  
9441 randomly (or by some other method, such as generalized random tessellation stratified  
9442 (GRTS) sampling (Stevens Jr and Olsen, 2004)) those units and measure some attribute.  
9443 The resulting inference applies to the attribute of the sample frame. There are some  
9444 distinct aspects of the design of SCR studies which many people struggle with in their  
9445 attempts to reconcile SCR design with classical survey design problems.

### 9446 10.1.1 Model-based not design-based

9447 **XXX ANDY: I find this little subsection hard to understand; maybe it needs**  
9448 **some more elaborating. I really like the following one though. XXXXXX**

9449 Classical finite-population sampling is often “design-based” which means properties  
9450 of estimators (bias, variance) are evaluated over repeated realizations of the *sample*. The  
9451 sample is random, but the attribute being observed is not. However, in the SCR modeling  
9452 framework properties of our estimators are distinctly model-based. We evaluate estimators  
9453 (usually) or care only about a *fixed* sample, averaged over realizations of the underlying  
9454 process and data we might generate. This is a classical parametric frequentist idea which  
9455 we think makes as much sense as a Bayesian too.

**9456 10.1.2 Sampling space or sampling individuals?**

9457 A fundamental question in any sampling problem is what is the sample frame – or the  
9458 population we are hoping to extrapolate too? In the context of capture-recapture studies,  
9459 it is tempting to think of the sample frame as being spatial (the space within “the study  
9460 area”, tiled into quadrats perhaps). Clearly SCR models involve a type of spatial sampling  
9461 – we have to identify spatial locations for traps, or arrays of traps. However, unlike  
9462 conventional natural resource sampling the attribute we measure is *not* directly relevant  
9463 to the *sample location*, such as where we place a trap and, therefore, it may not be  
9464 sensible to think of the sample frame as being comprised of spatial units. On the other  
9465 hand, capture-recapture studies clearly obtain a sample of *individuals* and SCR models are  
9466 models of *individual* encounter and space use. Therefore, it is more natural to think of the  
9467 sample frame as a list of  $N$  individuals, determined by the definition of the state-space,  
9468 or a subset of the state-space, i.e., the study-area, but the number  $N$  is unknown.

9469 Spatial sampling in SCR studies is important, but only as a device for accumulating  
9470 individuals in the sample from which we can learn about their inclusion probability. That  
9471 is, we’re not interested in any sample unit attribute directly but, rather, we use spatial  
9472 units as a means for sampling individuals and obtaining individual level encounter histories  
9473 that indicate the different sample locations at which each individual is encountered. It  
9474 makes sense in this context that we should want to choose a set of spatial sample units  
9475 that provides an adequate sample size of individuals, perhaps as many as possible. The  
9476 key technical consideration as it relates to spatial sampling and SCR is that arbitrary  
9477 selection of sample units has a side-effect that it induces unequal probabilities of inclusion  
9478 into the sample and so we must also learn about these unequal probabilities of sample  
9479 inclusion as we obtain our sample.

9480 The fact that SCR sampling induces unequal probabilities of sampling is consis-  
9481 tent with the classical sampling idea of Horvitz-Thompson estimation (see [http://en.wikipedia.org/wiki/Horvitz-Thompson\\_estimator](http://en.wikipedia.org/wiki/Horvitz-Thompson_estimator)) which has motivated capture-  
9482 recapture models similar to SCR (Huggins, 1989; Alho, 1990). In the Horvitz-Thompson  
9483 framework, the sample inclusion probabilities are usually fixed and known. However, in  
9484 all real animal sampling problems they are unknown because we never know precisely  
9485 where each individual lives and therefore cannot characterize its encounter probability.  
9486 Therefore, we have to estimate the sample inclusion probabilities using a model. SCR  
9487 models achieve this effect formally, using a fully model based approach based on a model  
9488 that accounts for the organization of individual activity centers and trap locations. This  
9489 notion of Horvitz-Thompson estimation suggests that perhaps we should consider designing  
9490 SCR studies based on the H-T variance estimator as a design criterion. We discuss  
9491 this a little bit later in this chapter.

**9493 10.1.3 Scope of inference vs. state-space**

9494 In SCR models we make a distinction between the scope of inference – the population we  
9495 care about – and the state-space of the point process, which we are required to prescribe  
9496 in order to fit models. These are not the same thing. The geographic scope of inference  
9497 is the region within which animals live that you care about in your study – let’s call this  
9498 “the study area”. This is often prescribed for political reasons or legal reasons (e.g. a  
9499 National Park). To initiate a study, or perhaps motivating the study, you have to draw

9500 a line on a map to delineate a study area, although often it is difficult to draw this line,  
9501 and where you draw it is not so much a statistical/SCR issue. On the other hand, you  
9502 need to prescribe the state-space to define and fit an SCR model. This is the region that  
9503 contains individuals that you *might* capture. This is different from the study area in most  
9504 cases.

9505 It is helpful to think about this distinction operationally. We define our study area *a*  
9506 *priori*. As a conceptual device, we might think of this as the area that, given an infinite  
9507 amount of resources, we might wall-off so that we can study a real closed population.  
9508 This 'study area' should exist independent of any model or estimator of some population  
9509 quantity. i.e., the subject-matter context should determine what the study area is. Given  
9510 a well-defined study area, we use some method to arrange data collecting devices within  
9511 this study area. The method of arrangement can be completely arbitrary but, naturally,  
9512 we want to choose arrangements of traps that are better in terms of obtaining statistical  
9513 information from the data we wind up collecting.

9514 Lets face it – Its quite a nuisance that animals move around and this makes the idea of  
9515 a spatial study area kind of meaningless in terms of management in most cases. Wherever  
9516 you draw a line on a map, there will be animals who live mostly beyond that line that  
9517 will sometimes be subjected to your study. One of the benefits of SCR models is they  
9518 formalize the exposure and contribution of these individuals to your study. That is a good  
9519 thing. Thus, you can probably be a bit sloppy or practical in your definition of "the study  
9520 area" and not worry too much.

## 10.2 STUDY DESIGN FOR (SPATIAL) CAPTURE-RECAPTURE

9521 The importance of adequate trap spacing and overall configuration of the trapping array  
9522 has long been discussed in the capture-recapture literature. A heuristic based on recog-  
9523 nizing the importance of typical home range sizes (Dice, 1938, 1941) and thus being able  
9524 to obtain information about home range size is that traps should be spaced such that the  
9525 array of available traps exposes as many individuals as possible but, at the same time,  
9526 individuals should be captureable in multiple traps. Thus, good designs should generate  
9527 a high sample size  $n$  and a large number of spatial recaptures. These two considerations  
9528 trade-off in building designs. On one hand, having a lot of traps very close together should  
9529 produce the most spatial recaptures but produce very few unique individuals captured (as-  
9530 suming that studies are limited in the total number of sampling devices they can deploy).  
9531 On the other hand, spreading the traps out as much as possible, in a nearly systematic or  
9532 regular design, should yield the most unique individuals. We will formalize this trade-off  
9533 later, when we consider formal model-based design of SCR studies.

9534 Traditional CR models require that all individuals in the study area have a probability  
9535  $> 0$  of being captured, which means that the trap array must not contain holes large  
9536 enough to contain an animal's entire home range (Otis et al., 1978). As a consequence,  
9537 trap spacing should be on the same order as the radius of a typical home range (e.g.,  
9538 Dillon and Kelly, 2007). For example, imagine a camera trap study implemented in South  
9539 America with the objective to survey populations of both jaguars and the much smaller  
9540 ocelots. Ocelots also have much smaller home ranges and therefore should require closer  
9541 trap spacing than the large wide-ranging jaguars. Where approaches such as MMDM are  
9542 used in combination with traditional CR models to obtain density estimates (see Chapt.

9543 4), trap spacing also has a major effect on movement estimates, since it determines the  
9544 resolution of the information on individual movement (Parmenter et al., 2003; Wilson and  
9545 Anderson, 1985a). If trap spacing is too wide, there is little to no information on animal  
9546 movement because most animals will only be captured at one trap (Dillon and Kelly,  
9547 2007). In addition, only a trapping grid that is large relative to individual movement can  
9548 capture the full extent of such movements, and researchers have suggested that the grid  
9549 size should be at least four times that of individual home ranges to avoid positive bias in  
9550 estimates of density (Bondrup-Nielsen, 1983). This recommendation originated in small  
9551 mammal trapping, and it should be relatively easy to follow when dealing with species  
9552 covering home ranges < 1ha. However, translated to large mammal research, this can  
9553 entail having to cover several thousands of square kilometers – a logistical and financial  
9554 challenge probably few projects could realistically tackle.

9555 Holes in the study area are of no concern in SCR studies. As a practical matter, some  
9556 animals within the study area might have vanishingly small probability of being included  
9557 in the sample, i.e.,  $p \approx 0$ , the nice thing about SCR models is that  $N$  is explicitly tied to  
9558 the state-space, and not the traps which expose them to encounter. Within an SCR model,  
9559 extending inference from the sample to individuals that live in these holes represents an  
9560 extrapolation (prediction of the model outside the range of the data), but one that the  
9561 model is capable of producing because we have explicit declarations, in the model, that it  
9562 applies to any area within the state-space (the state-space is a part of the model!), even  
9563 to areas where we can't capture individuals because we happened to not put a trap near  
9564 them. Conversely, classical capture-recapture models only apply to individuals that have  
9565 encounter probability that is consistent with the model being considered. Presumably,  
9566 the existence of a hole in the trap array would introduce individuals with  $p = 0$ , which is  
9567 not accommodated in those models.

9568 Whereas traditional CR studies are concerned with the number of individuals and  
9569 recaptures and with satisfying the model assumption of all individuals having some prob-  
9570 ability of being captured, in spatial capture-recapture we are looking at an additional  
9571 level of information: We need spatially spread out captures and recaptures. That means,  
9572 it is not enough to recapture an individual, but we need to recapture at least some in-  
9573 dividuals at several traps. Therefore, in general, design of SCR studies boils down to  
9574 obtaining three bits of information: total captures of unique individuals, gross recaptures  
9575 informative about baseline encounter rate, and spatial recaptures, informative about  $\sigma$ .  
9576 Most SCR design choices wind up trading these three things against each other to achieve  
9577 some optimal (or good) mix. So for example if we sample a very small number of sites  
9578 a huge number of times then we can get a lot of recaptures but only very few spatial  
9579 ones, and few unique individuals etc. This need for spatial recaptures may appear as an  
9580 additional constraint on study design, but actually, SCR studies are much less restricted  
9581 than traditional CR studies, because of the way animal movement is incorporated into the  
9582 model:  $\sigma$  is estimated as a specified function of the ancillary spatial information collected  
9583 in the survey and the capture frequencies at those locations and this function is able to  
9584 make a prediction across distances even when these are latent, including distances larger  
9585 than the extent of the trap array. When there is enough data across at least some range  
9586 of distances, the model will do well at making predictions at unobserved distances. The  
9587 key here is that there needs to be 'enough data across some range of distances', which  
9588 induces some constraint on how large our overall trap array must be to provide this range

9589 of distances (e.g., Marques et al., 2011). We will review the flexibility of SCR models in  
9590 terms of trap spacing and trapping grid size in the following section.

### 10.3 TRAP SPACING AND ARRAY SIZE RELATIVE TO ANIMAL MOVEMENT

9591 **XXXX Does Efford have anything out there with discussions of trap spacing?**  
9592 **XXXXX**

9593 Using a simulation study, Sollmann et al. (2012a) investigated how trap spacing and  
9594 array size relative to animal movement influence SCR parameter estimates and we will  
9595 summarize this study here. They simulated encounter histories on an  $8 \times 8$  trap array  
9596 with regular spacing of 2 units, using a Binomial encounter model with Gaussian hazard  
9597 encounter model (complementary log-log link), across a range of values for the movement  
9598 parameter  $\sigma^*$ . We refer to the movement parameter as  $\sigma^*$  here, because Sollmann et al.  
9599 (2012a) use a slightly different parametrization of SCR models, in which  $\sigma^*$  corresponds  
9600 to  $\sigma \times \sqrt{2}$ .

9601 XXXX ANDY: I am totally blanking here: We formulated the model so that the  
9602 distance function was  $d2/\text{sig2}$ . That means  $\sigma^* = \sigma \times \sqrt{2}$ , right? It was different be-  
9603 fore but I think now it's right. If I got it wrong, let me know and I'll fix it. Sorry!!  
9604 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX In Sec. 5.3 we pointed out that  
9605 under the bivariate normal (or half-normal) detection model  $\sigma$  can be converted into an  
9606 estimate of the 95 % home range or "use area" around  $s_i$ . Based on this transformation,  
9607 values for  $\sigma^*$  were chosen so that there was a scenario where the trap array was smaller  
9608 than a single individual's home range, i.e. trap spacing was narrow relative to individual  
9609 movements ( $\sigma^* = 5$ ), a scenario where spaces between traps were large enough to con-  
9610 tain entire home ranges ( $\sigma^* = 0.5$ ), and two intermediate scenarios and where sigma was  
9611 smaller ( $\sigma^* = 1$  unit) and larger ( $\sigma^* = 2.5$  units) than the trap spacing, respectively.  $N$   
9612 was 100 and the baseline trap encounter rate  $\lambda_0$  was 0.5 for all four scenarios, and trap  
9613 encounters were generated over 4 occasions. Table 10.1 shows the results as the average  
9614 over 100 simulations.

9615 All model parameters were identifiable and estimated with relatively low bias (< 10  
9616 %) and high to moderate precision (rrmse < 25 %) for all scenarios of  $\sigma^*$ , except  $\sigma^* =$   
9617 0.5 units (therefore excluded from Table 10.1). Data for the latter case mostly differed  
9618 from the other scenarios in that fewer animals were captured and very few of the captured  
9619 animals were recorded at more than 1 trap (Table 10.2). For  $\sigma^* = 0.5$ , abundance ( $N$ )  
9620 was not identifiable in 88 % of the simulations, and when identifiable, was underestimated  
9621 by approximately 50 %. This shows that a trap spacing that is considerably too large may  
9622 be problematic in SCR studies.

9623 Estimates of  $N$  were least biased and most precise under the  $\sigma^* = 2.5$  scenario, and in  
9624 general, all parameters were estimated best under the  $\sigma^* = 2.5$  or the  $\sigma^* = 5$  scenario. All  
9625 estimates had the highest relative bias and the lowest precision under the  $\sigma^* = 1$  scenario.  
9626 These results clearly demonstrate that SCR models can successfully handle a range of trap  
9627 spacing to animal movement ratios, and even when using a trapping array smaller than  
9628 an average home range: at  $\sigma^* = 5$ , the home range of an individual was approximately  
9629 235  $\text{units}^2$ , while the trapping grid only covered 196  $\text{units}^2$ . Still, the model performed  
9630 very well.

**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 95BCI coverage (BCI) 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
<b><math>\sigma^* = 1</math> (<math>\sigma = 0.71</math>)</b>							
$N$	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
<b><math>\sigma^* = 2.5</math> (<math>\sigma = 1.77</math>)</b>							
$N$	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
<b><math>\sigma^* = 5</math> (<math>\sigma = 3.54</math>)</b>							
$N$	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)

An important consideration in this simulation study is that all but the  $\sigma^* = 0.5$  units scenarios provided reasonably large amounts of data, including 20 + individuals being captured on the trapping grid. When dealing with real-life animals that are often territorial and may have lower trap encounter rates, a very small grid compared to an individual's home range may result in the capture of few to no individuals. In that case, the sparse data will limit the ability of the model to estimate parameters (Marques et al. 2011), which is true of most models.

In summary, SCR models performed best when  $\sigma^*$  was slightly larger than trap spacing (or in other words, when  $\sigma$  was slightly smaller) and did well as long as  $\sigma^*$  was at least 0.5 times the average distance between traps (which corresponds to  $\sigma$  being 0.35 times the average distance between traps). Although at this trap spacing to movement ratio, most individuals are captured at one trap only (see Tab. 10.2), parameter estimates exhibited low bias and remained relatively precise (see simulation results for  $\sigma^* = 1$  in Tab. 10.1). Below this trap spacing to movement ratio the spatial information in the simulated data apparently was not sufficient to inform SCR model parameters.

---

**10.3.1 Example: 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. (2012a) 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 allowed for differences in the baseline trap encounter rate and  $\sigma^*$  between males and females, and  $\lambda_0$  varied across occasions. This was motivated by a) the lower average number of detections for male bears and 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, the full data set and data subsets were analyzed with SCR models. 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.

**Table 10.3.** Posterior summaries of SCR model parameters for black bears.

	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

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 female black bear home range - approximately  $260 \text{ km}^2$  and  $100 \text{ km}^2$ , respectively, when converting estimates of  $\sigma^*$  to home range size. Table 10.3 shows that this did not greatly influence model results, compared to the full data set. The observed smaller differences in parameter estimates may be due to individual differences in detection and movement that manifest themselves when only a smaller portion of the overall population is sampled. By reducing the number of traps we effectively reduced the size of the overall data set estimates were based on (both in terms of individuals captured and recaptures). This was reflected in overall higher SE and wider confidence intervals. In spite of these differences, density estimates - the main objective of applying SCR models - remained largely constant. Removing 80 % of the traps and thereby reducing the area of the trap array to  $64 \text{ km}^2$  - well below the average black bear home range - had a great effect on sample size (only 25 of the original 83 individuals sampled) and parameter estimates. Particularly, male black bear movement was overestimated and imprecise. The combination of the low baseline trap encounter rate of males and the considerable reduction in sample size led to a low

level of information on male movement: 5 of the 12 males were captured at one trap only. Although they moved over smaller areas, owing to their higher trap encounter rate females were, on average, captured at more traps (3.4 traps per individual compared to 2.6 for males) so that their movement estimate remained relatively accurate. Overestimated male movements and female trap encounter rates resulted in an underestimate of density of almost 40 %. This effect is contrary to what we would expect to see in non-spatial CR models, where too small an area leads to underestimated movement and overestimated density (Bondrup-Nielsen, 1983; Dillon and Kelly, 2007; Maffei and Noss, 2008). While this example again demonstrates the ability of SCR models to deal with a range of trapping grid sizes, it also clearly shows that your study design needs to consider the amount of data you can expect to collect.

### 10.3.2 Final musings: SCR models, trap spacing and array size

When designing a capture-recapture study for a single species, trap spacing and the size of the array can (and should) be tailored to the spatial behavior of that species to ensure adequate data collection. However, some trapping devices like camera traps may collect data on more than one species and researchers may want to analyze these data, too. Independent of the trapping device used, study design will in most cases face a limit in terms of the number of traps available or logically manageable. As a consequence, researchers need to find the best compromise between trap spacing and the overall grid area and the following section will develop an approach to derive such an optimal study design.

Particularly for large mammal research SCR models have much more realistic requirements in terms of area coverage than non-spatial CR models, under which density estimates can be largely inflated with small trapping grids relative to individual movement (Maffei and Noss, 2008). How large the spatial survey effort needs to be does not only depend on the extent of movement of the target species, but also on the temporal effort, density and detection probability (Marques et al., 2011) – in summary, the amount of data that can be collected with any given trap array. For low-density species, like the black bears in the above example, small trapping grids bear the risk of not collecting enough data for parameter estimation. Simulation studies can help you assess how effective a certain study design is given a set of parameters. Alternatively, Efford et al. (2009b) provide a mathematical procedure to determine the expected number of individuals captured and recaptures for a given detector array and set of model parameters.

Overall, while there are limits to the flexibility in spatial trap array design for SCR modeling, the method is fairly robust to changes in trap array size and spacing relative to animal movement. Trapping grids with an extent of approximately a home range diameter can in theory - adequately estimated density and home range size. However, these results should not encourage researchers to design non-invasive trap arrays based on minimum area and spacing requirements. Study design should still strive to expose as many individuals as possible to sampling and obtain adequate data on individual movement. Large amounts of data do not only improve precision of parameter estimates (the density estimate for the full black bear data set has narrower confidence intervals than estimates from the reduced data sets), they also allow including potentially important covariates (such as gender or time effects in the black bear example) into SCR models to obtain

9721 density estimates that reflect the actual state of the studied population.

#### 10.4 SPACING OF TRAPS WITH TELEMETERED INDIVIDUALS

9722 In Chapt. 11 we discussed SCR models that integrate auxiliary information on resource se-  
9723 lection obtained by telemetry. Telemetry data are directly informative about the distance  
9724 parameter ( $\sigma$  or  $\alpha_1$  as the case may be). It stands to reason that, when telemetry data are  
9725 available, it should affect considerations related to trap spacing. Conceivably even, one  
9726 might be able to build SCR designs that don't yield any formal spatial recaptures because  
9727 all of the information about  $\sigma$  is provided by the telemetry data.

#### 10.5 SAMPLING OVER LARGE SCALES

9728 Trap spacing is an essential aspect of design of SCR studies. However, it is only the most  
9729 important aspect if one can uniformly cover a study area with traps. In many practical  
9730 situations where the study area is large relative to effort that can be expended, one has to  
9731 consider other strategies which deviate from a strict focus on trap spacing. There are two  
9732 general strategies that have been suggested which we think are useful in practice, either  
9733 by themselves or combined: Sampling based on *clusters* of traps and sampling based on  
9734 *rotating* groups of traps over the landscape.

9735 Karanth and Nichols talk about moving traps around in the green book.....

9736 Efford (unpublished) looked at clusters....

9737 In practice, employing both of these strategies might be necessary.

9738 Work on formalizing and generalizing these ideas is needed. We believe the model-  
9739 based spatial design approach, which we introduce below, is the way to do that.

#### 10.6 MODEL-BASED SPATIAL DESIGN

9740 A point we have stressed in previous chapters is that SCR models are basically glorified  
9741 versions of generalized linear models (GLMs) with a random effect that represents a latent  
9742 spatial attribute of individuals, the activity center or home range center. This formula-  
9743 tion makes analysis of the models readily accessible in freely available software and also  
9744 allows us to adapt and use concepts from this broad class of models to solve problems  
9745 in spatial capture recapture. In particular, we can exploit well-established model-based  
9746 design concepts (Kiefer 1959; Box and Draper 1959, 1987; Fedorov 1972; Sacks et al. 1989;  
9747 Hardin and Sloane 1993; Fedorov and Hackl 1997) to develop a framework for designing  
9748 spatial trapping arrays for capture-recapture studies. Müller (2007) provides a recent  
9749 book treatment of the subject that is very readable.

9750 In the following sections, we adapt these classical methods for constructing optimal  
9751 designs to obtain the configuration of traps (or sampling devices) in some region (the design  
9752 space,  $\mathcal{X}$ ), that minimizes some appropriate objective function based on a compromise  
9753 between the variance of estimating  $N$  for a prescribed state-space. We show that this  
9754 criterion – based on the variance of an estimator of  $N$  – represents a formal compromise  
9755 between minimizing the variance of the MLEs of the detection model parameters and  
9756 obtaining a *high* expected probability of capture. Intuitively, if our only objective was  
9757 to minimize the variance of parameter estimates than all of our traps should be in one or

9758 a small number of clusters where we can recapture a small number of individuals many  
 9759 times each. Conversely, if our objective was only to maximize the expected probability  
 9760 of encounter then the array should be highly uniform so as to maximize the number  
 9761 of individuals being exposed to capture. By seeking to minimize the variance of an  
 9762 estimator of  $N$ , our objective function is, formally, a compromise between these two  
 9763 objectives and the resulting designs are not always highly regular nor clustered.

#### 9764 10.6.1 Formalization of the Design Problem for SCR Studies

9765 Let  $\mathcal{X}$ , the *design space*, denote some region within which sampling could occur and let  
 9766  $\mathbf{X} = \mathbf{x}_1, \dots, \mathbf{x}_J$  denote the *design*, the set of sample locations (e.g., of camera traps)  
 9767 which henceforth will be referenced as “traps.” Operationally, we could equate  $\mathcal{X}$  to the  
 9768 study area itself (which is of management interest) but, in practical cases, there will be  
 9769 parts of the study area that we cannot sample. Those areas need to be excluded from  
 9770  $\mathcal{X}$ . The technical problem addressed in the following is how to choose the locations  $\mathbf{X}$  in  
 9771 a manner that is statistically efficient for estimating abundance or density. The design  
 9772 space,  $\mathcal{X}$ , which determines potential design points, will have to be prescribed. This could  
 9773 be some polygon describing a park or forest unit from which we may choose trap locations.  
 9774 Further, while  $\mathcal{X}$  maybe be continuous, in practice it will be sufficient to represent  $\mathcal{X}$  by  
 9775 a discrete collection of points. This is especially convenient when the geometry of  $\mathcal{X}$  is  
 9776 complicated and irregular (which would be in most practical applications).

9777 We regard the population of  $N$  such individual “activity centers” as the outcome of a  
 9778 point process. Denote the home range center of an individual by the coordinate  $\mathbf{s}$  which is  
 9779 regarded as the outcome of a random variable uniformly distributed over the state-space  
 9780  $\mathcal{S}$ , some 2-dimensional region. The importance of  $\mathcal{S}$  is obvious as it defines a population of  
 9781 individuals (i.e., activity centers) and, in practice, it is not usually the same as  $\mathcal{X}$  due to  
 9782 the fact that animals move freely over the landscape and the location of traps is typically  
 9783 restricted by policies, ownership and other considerations. That  $\mathcal{X}$  and  $\mathcal{S}$  are not the same  
 9784 is the basic problem of geographic non-closure of the population for which spatial capture-  
 9785 recapture models have been devised (Efford 2004; Borchers and Efford 2008; Royle and  
 9786 Young 2008; Royle and Gardner 2009).

9787 The basic strategy is: Given (1)  $\mathcal{X}$ , (2) a number of design points; (3) state-space  $\mathcal{S}$ ,  
 9788 and (4) an SCR model, and (5) a design criterion  $Q(\mathbf{X})$ , we want to choose *which* design  
 9789 points we should select in order to obtain the *optimal* design under the chosen model,  
 9790 where the optimality is with respect to  $Q(\mathbf{X})$ .

9791 To see how this goes in a simplified situation, suppose you know  $\mathbf{s}$  for an individual.  
 9792 In this case, its vector of counts of encounter in each trap  $\mathbf{y}$  are either binomial or Poisson  
 9793 counts, i.e., just a GLM, with

$$g(\mathbb{E}(\mathbf{y})) = \alpha_0 + \alpha_1 \|\mathbf{x} - \mathbf{s}\|$$

9794 Lets think about this in the context of a normal linear model then:

$$\mathbf{y} = \mathbf{M}(\mathbf{X}, \mathbf{s})' \boldsymbol{\alpha} + \text{error}$$

9795 We could analyze the design problem for the binomial or Poisson case but to establish  
 9796 basic ideas here lets just look at the normal model. The variance-covariance matrix of  $\hat{\boldsymbol{\alpha}}$

9797 is, supressing the dependence on  $\mathbf{X}$ , is:

$$\text{Var}(\boldsymbol{\alpha}) = (\mathbf{M}(\mathbf{s})' \mathbf{M}(\mathbf{s}))^{-1}$$

9798 Therefore if we know *all*  $N$  values of  $\mathbf{s}$  we could now easily find the design  $\mathbf{X}$  that optimizes  
 9799 some function of the variance-covariance matrix, whatever function we want. If we don't  
 9800 know  $\mathbf{s}$  then we might as well minimize the expected variance:

$$E_{\mathbf{s}} \{ \text{Var}(\boldsymbol{\alpha}) \} = \sum_{s \in S} (\mathbf{M}'(\mathbf{s}) \mathbf{M}(\mathbf{s}))^{-1}$$

9801 This can be done for any number of design points  $\mathbf{x}_1, \dots, \mathbf{x}_J$  using imperfect exchange  
 9802 algorithms (Sec. 10.6.3) which always improve the criterion but will not necessarily yield  
 9803 the optimal design. But it is usually good enough for practice.

9804 It is worth noting that asymptotic formulae for  $\text{Var}(\boldsymbol{\alpha})$  can be cooked up fro any type  
 9805 of GLM (e.g., see McCullagh and Nelder. .XXXXXX p. XXXXX) and we will see that  
 9806 design for SCR models is closely related to the basic GLMs such as binomial or poisson  
 9807 regression.

9808 Interestingly, if you minimize obvious functions of the variance of the encounter pa-  
 9809 rameter estimates, then this produces strongly clustered designs. For example if I pick a  
 9810 design of size 11 then it puts 2 or 3 points in each corner of the square and 1 or 2 points  
 9811 in the center. I think this makes a lot of sense if your objective function is simply to  
 9812 minimize the variance of your estimates of  $\boldsymbol{\alpha}$ . This suggests to us that maybe minimiz-  
 9813 ing this variance isn't really the right thing to do. In fact, it is not sufficient to make a  
 9814 design that is optimal for estimating regression parameters – we also want to produce a  
 9815 low variance for estimating  $N$ . since  $n \sim \text{Bin}(N, pbar)$  we want  $n$  to be as close to  $N$  as  
 9816 possible, generally speaking. This suggests that we should find a design that maximizes  
 9817 "pbar" – i.e., generates the highest expected sample size.

9818 If we just design networks to maximize  $\bar{p}$  then, as you expect, these designs are highly  
 9819 regular.

9820 Ok, so what we really would like to do is optimize some function of these three things.  
 9821 We want to minimize the variance of ( $\alpha_0$ ,  $\alpha_1$  and  $n_0$ ). The problem with this is  
 9822 that there aren't easy formuals for this but we devise an approximation in the following  
 9823 section and then we build designs for that criterion.

### 9824 10.6.2 An Optimal Design Criterion for SCR

9825 Can derive based on the H-A estimator. We tried that too hard.  
 9826 We take an approach based on standard conditional estimator....  
 9827 (how about data augmentation?)

9828 Consider a conditional estimator of  $N$  of the form

$$\tilde{N} = \frac{n}{\bar{p}}$$

9829 where  $\bar{p}$  is the probability that an individual appears in the sample of  $n$  unique individuals.  
 9830 In SCR models an individual with activity center  $\mathbf{s}_i$  is captured if it is captured in *any*

trap and therefore, under the Bernoulli model,

$$\bar{p}(\mathbf{s}_i, \mathbf{X}) = 1 - \prod_{j=1}^J (1 - p_{ij}(\mathbf{x}_j, \mathbf{s}_i))$$

and, under the Poisson model, we have:

$$\bar{p}(\mathbf{s}_i, \mathbf{X}) = 1 - \exp(-\lambda_0 \sum_j \exp(\beta * d(\mathbf{x}_j, \mathbf{s}_i)))$$

where here we emphasized that this is conditional on  $\mathbf{s}_i$  and also the design – the trap locations  $\mathbf{x}_j$ . The *marginal* probability of encounter, averaging over all possible locations of  $\mathbf{s}$  is:

$$\bar{p}(\mathbf{X}) = 1 - \int_{\mathbf{s}} \bar{p}(\mathbf{s}_i, \mathbf{X}) d\mathbf{s}.$$

It is important to note that this can be calculated directly *given* the design  $\mathbf{X}$ . This is handy because we see that it is used in the variance formulae given subsequently.

The approach we take here is we develop the variance of  $\tilde{N}$  conditional on knowing the locations of all  $N$  individuals and then we suggest to unconditon on the realized point process by taking a Monte Carlo average over realizations of  $\mathbf{s}$  under a suitable model for  $\mathbf{s}$ . The variance criterion we propose here is based on a delta approximation  $Var(n/\bar{p})$ :

$$Var(\tilde{N}(\alpha) | \{\mathbf{s}_i\}_{i=1}^N) = \frac{N^2 Var(\bar{p})}{\bar{p}^2} + N \frac{(1-\bar{p})}{\bar{p}} \quad (10.6.1)$$

It is important to note that this is the sum of two parts which are essentially those due to (1) estimation of  $\bar{p}$  from the sample and (2) the variance of  $n$ . We see that generally this criterion is improved (decreases) as we do a better job estimating  $\bar{p}$  and also as  $n$  approaches  $N$ , i.e., as  $\bar{p}$  increases to 1. Thus, good designs should generate information about detection probability *and* produce large samples of individuals.

In order to work with this experssion we will have to do some analysis of  $Var(\bar{p})$  which we take up now. We note that  $\bar{p}$  is itself a deterministic function of the parameters that we need to estimate,  $\boldsymbol{\alpha}$ . Therefore, we use a delta approximation to express  $Var(\bar{p})$  in terms of the variance of the MLE  $\hat{\boldsymbol{\alpha}}$ . This produces:

$$Var(\bar{p}) = \left( \frac{\delta \bar{p}}{\delta \alpha_0}, \frac{\delta \bar{p}}{\delta \alpha_1} \right) Var(\hat{\boldsymbol{\alpha}}) \begin{pmatrix} \frac{\delta \bar{p}}{\delta \alpha_0} \\ \frac{\delta \bar{p}}{\delta \alpha_1} \end{pmatrix} \quad (10.6.2)$$

We need to break this down into its constituent pieces:

**(1)  $Var(\hat{\boldsymbol{\alpha}})$ .** It is not actually so obvious what the form of this matrix is. Some calculus would have to be done on the conditional likelihood (e.g., from Borchers and Efford 2008) to figure out the asymptotic form of this matrix. For now, a good heuristic is to use the analogous result from a Poisson or Binomial GLM to approximate it, since we have formulas for those. If we knew the activity centers of all individuals then the resulting data  $y(x, s)$  are Poisson counts. The asymptotic variance-covariance matrix of  $\boldsymbol{\alpha}$  in that case is:

$$Var(\hat{\boldsymbol{\alpha}} | \mathbf{X}, \mathbf{s}) = (\mathbf{M}(\mathbf{s})' \mathbf{D}(\boldsymbol{\alpha}, \mathbf{s}) \mathbf{M}(\mathbf{s}))^{-1}. \quad (10.6.3)$$

9859 where  $\mathbf{M}$  is a matrix which has a column of 1's and a column of  $N \times J$  entries that are the  
 9860 distances between each individual and each trap and the matrix  $\mathbf{D}$  is a diagonal matrix  
 9861 having elements  $\text{Var}(y_j|\mathbf{s}) = \exp(\mathbf{m}'\boldsymbol{\alpha})$  for  $y_{ij}$  the frequency of encounter in trap  $j$ . Thus,  
 9862 the variance is a function of the design  $\mathbf{X}$  as well as  $\mathbf{s}$  both of which are balled-up in  $\mathbf{M}$   
 9863 – the regression design matrix and the matrix  $\mathbf{D}$ .

9864 (2) The derivative terms: multiple applications of the chain rule can be used (see Huggins  
 9865 (1989) and Alho (1990) for relevant examples). Under the Poisson model we have

$$\bar{p} = 1 - \sum_s \exp(-\lambda_0 \sum_j \exp(\beta * d(x_j, s)))$$

9866 where the summation over  $\mathbf{s}$  arises as a result of approximating the integral in Eq. XXXXX  
 9867 by a summation. If we differentiate this with respect to  $\lambda_0$  and  $\beta$  we have the following:

$$\frac{\delta \bar{p}}{\delta \lambda_0} = 1 - \sum_s \left\{ \left( - \sum_j \exp(\beta d_{ij}^2) \right) \exp(-\lambda_0 \sum_j \exp(\beta d_{ij}^2)) \right\}$$

9868 and

$$\frac{\delta \bar{p}}{\delta \beta} = \left\{ \sum_s -\lambda_0 \left( \sum_j \exp(\beta d_{ij}^2) \right) \right\} \left\{ 1 - \sum_s \left( -\lambda_0 \sum_j \exp(\beta d_{ij}^2) \right) \exp \left( -\lambda_0 \sum_j \exp(\beta d_{ij}^2) \right) \right\}$$

9869 Therefore we have a design criterion which is obtained by plugging  $\bar{p}$  from Eq. XXXX  
 9870 and the variance expression Eq. 10.6.2 into Eq. 10.6.1. This is a function of the design  
 9871  $\mathbf{X}$ . Furthermore, we emphasize that the above variance expression is *conditional* on the  
 9872 realization  $\mathbf{s}_1, \dots, \mathbf{s}_N$  which is, in the context of design, not observable. We will therefore  
 9873 develop design criteria which are unconditional on  $\{\mathbf{s}\}$ . The total variance expression is  
 9874 unconditional on  $\mathbf{s}$  is:

$$\text{Var}(\tilde{N}) = E_s \text{Var}(\tilde{N}|\mathbf{s}) + \text{Var}_s E(\tilde{N}|\mathbf{s})$$

9875 if we assert that sample sizes will be large enough so that our estimator is unbiased,  
 9876 then the 2nd term will be close to 0 and we can ignore it. Therefore to evaluate the  
 9877 unconditional variance we need to solve an  $N$ -fold integration to average over  $\mathbf{s}_1, \dots, \mathbf{s}_N$ ,  
 9878 or we can do this by taking a monte carlo average.

### 9879 10.6.3 Optimization of the criterion

9880 We need to come up with a ballpark guess of the model parameters. i.e., what is  $\boldsymbol{\alpha}$  and  
 9881  $N$ ? If we do that, and specify the state-space  $\mathcal{S}$  and the number of traps to place, then  
 9882 we can optimize the variance criterion.

9883 In formulating the optimization problem note that we have  $J$  sample locations corre-  
 9884 sponding to rows of  $\mathbf{X}$ . The problem is a  $2J$  dimensional optimization problem which,  
 9885 for  $J$  small, could be solved using standard numerical optimization algorithms as exist in  
 9886 almost every statistical computation environment. However,  $J$  will almost always be large  
 9887 enough so as to preclude effective use of such algorithms. This is a common problem in ex-  
 9888 perimental design, design for response surface estimation, computer experiments, spatial

9889 sampling designs and other disciplines for which sequential exchange or swapping algorithms can be used (e.g., Wynn 1970; Fedorov 1972; Mitchell 1974; Meyer and Nachtsheim 9890 1995). The basic idea is to pose the problem as a sequence of 1-dimensional optimization 9891 problems in which the objective function is optimized over 1 or several coordinates at a 9892 time.

9893 In the present case, we consider swapping out  $\mathbf{x}_j$  for some point in  $\mathcal{X}$  that is nearby 9894  $\mathbf{x}_j$  (e.g., a 1st order neighbor). The objective function is evaluated for all possible swaps 9895 (at most 4 in the case of 1st order neighbors) and whichever point yields the biggest 9896 improvement is swapped for the current value. The algorithm is iterated over all  $J$  design 9897 points and this continues until convergence is achieved. Such algorithms may yield local 9898 optima and optimization for a number of random initial designs can yield incremental 9899 improvements. We implemented this swapping algorithm in **R**, using the basic strategy 9900 employed elsewhere (e.g., Nychka et al. 1997; Royle and Nychka 1998). A version of 9901 a swapping algorithm used to optimize a space-filling criterion is implemented in the 9902 **R** package **fields** (Fields Development Team 2006). I developed an implementation that 9903 requires a discrete representation of  $\mathcal{S}$  (an arbitrary matrix of coordinates) and an indicator 9904 of which elements of  $\mathcal{S}$  are members of the design space  $\mathcal{X}$ . For each point in  $\mathbf{X}$ , only the 9905 nearest neighbors (the number is specified) are considered for swapping into the design 9906 during each iteration.

9907 While swapping algorithms are convenient to implement, and efficient at reducing 9908 the criterion in very high dimensional problems, they do not always yield the global 9909 optimum. In practice, as in the examples below, it is advisable to apply the algorithm to 9910 a large number of random starting designs. My experience is that essentially meaningless 9911 improvements are realized after searching through a few dozen random starts.

#### 9913 10.6.4 Illustration

9914 Consider designing a study for camera traps in a square region defined by the square 9915  $[10, 20] \times [10, 20]$  and with  $\mathcal{X} = \mathcal{S}$ . For this illustration I assumed  $\beta_0 = \log(\lambda_0) = -2.7$  9916 and  $\beta_1 = 1/(\sigma^2) = 1/4, 1/9$  and  $1/16$ , so  $\sigma = 2, 3, 4$ . (this was dumb - note that  $\sigma$  is really 9917 2 times the standard deviation of a normal distribution. Oh well!). Designs of size 9 and 9918 10 were computed for each value of  $\sigma$  using many random starting designs. The putative 9919 optimal designs (henceforth “best”) are shown<sup>1</sup> in Figure 16.2. For  $J=9$ ,  $\sigma = 2$ , the best 9920 design was produced in 180 out of 1000 random starts. For  $\sigma = 3$  (row 2, left panel) the 9921 best design was produced in about 88% of all optimizations from random starting values. 9922 For  $J = 10$ , and  $\sigma = 2$  (row 1, right panel), the best design was found about 24% of the 9923 time (from random starts). The  $\sigma = 3$  best design (row 2, right panel; 14% of random 9924 starts) clusters 2 points in the center. Finally, consider the  $\sigma = 4$  case (last row of Fig. 9925 16.2). We have two irregular looking designs and the design points cluster in various ways.

9926 I computed the best designs using the same settings but increasing the size of  $\mathcal{S}$  relative 9927 to  $\mathcal{X}$ . In particular, I nested  $\mathcal{X}$  into  $[9, 21] \times [9, 21]$  (Figure ??) and then  $[8, 22]^2$  (Figure 9928 ??). The obvious effect of this is that the best designs move points toward the edge of 9929 the design space  $\mathcal{X}$  so as to provide more exposure to points in  $\mathcal{S}$ . The effect is more

<sup>1</sup>My intention is to provide many of these results in an Appendix in order to reduce the length of the paper.

9930 pronounced, obviously, as you provide more area outside of  $\mathcal{X}$  that is allowed to influence  
9931 the design.

9932 As a final example, consider placing 20 camera traps in this region. Where do they  
9933 go? Look at the 3 buffers, 3 values of sigma, that's 9 total designs (use a single panel).  
9934 An interesting feature of the designs is that they are not regular. Traps occur in clusters  
9935 of several traps close together with the clusters more widely spaced.

## 10.7 COVARIATE MODELS

9936 if the objective is to estimate a covariate effect on density then you should build this into  
9937 the criterion..... In this case we can think of the captures in a trap being a Poisson r.v.  
9938 with mean  $\lambda(\text{trap}, s) * D(s)$  and the problem is slightly more complicated but it can  
9939 be done... the calculus to work out the var-cov matrix of density parameters needs to be  
9940 done, ....

9941 Intuitively, model-based approaches in this case should favor areas of higher density....

## 10.8 SUMMARY AND OUTLOOK

9942 Heuristics: recaptures vs sample size of individuals. The two objectives trade-off. We  
9943 need designs that are good for estimating  $\bar{p}$  and also designs that obtain a high sample  
9944 size of  $n$ . Designs that are only good for one or the other will produce bad SCR designs, or  
9945 designs in which  $N$  is not estimable. One exception is when telemetry is available. These  
9946 provide information on  $\sigma$  or other parameters of the detection model and this changes the  
9947 whole situation so that trap arrays should be more spread out.

9948 In general though, for basic situations, trap spacing should be about XXXX  $\sigma$ .  
9949 Clustering is important too.

9950 We should always do a simulation study. This allows us to learn what to expect as we  
9951 start collecting real data. Plus we can simulate for any complex situation that we desire.

9952 However formal model-based design of SCR models has great potential and we think  
9953 this is where things will be going. SCR models are amenable to some degree of analytic  
9954 study using classical spatial design ideas. We have just barely scratched the surface here,  
9955 showing how to formulate a criterion that is a function of the design, and then optimizing  
9956 the criterion over all possible designs. We believe this approach merits more attention.

9957

## Part III

9958

---

9959

# Spatial Processes in SCR



9960  
9961

# 11

---

9962  
9963  
9964

## INTEGRATING RESOURCE SELECTION WITH SPATIAL CAPTURE-RECAPTURE MODELS

9965 Up to this point we have developed many variations of SCR models to describe the obser-  
9966 vation process. These included models of the relationship between encounter probability  
9967 and distance, and different types of covariates such as behavioral responses that can affect  
9968 detection probability. Although these different observation models are immensely useful,  
9969 they are rather basic in the sense that they imply simplistic models of how individuals use  
9970 space (section 5.3) and how individuals are distributed in space. In the following several  
9971 chapters we generalize some of the core SCR assumptions to accommodate more realistic  
9972 notions of how animals use space.

9973 In Sec. 5.3 we briefly discussed the notion of how SCR encounter probability models  
9974 relate to models of space usage. When we use symmetric and stationary encounter prob-  
9975 ability models, SCR models imply that space usage is a decreasing function of distance  
9976 from an individuals home range center. In this chapter, we extend SCR models to in-  
9977 corporate models of resource selection (RS), such as when one or more explicit landscape  
9978 covariates are available which the investigator believes might affect how individual ani-  
9979 mals use space within their home range (this is what (Johnson, 1980) called *third-order*  
9980 selection). Our treatment follows Royle et al. (2012a) who integrated a standard family  
9981 of resource selection models based on auxilliary telemetry data into the capture-recapture  
9982 model for encounter probability. The approach is consistent with the manner in which  
9983 classical “resource selection function” (RSF) models (Manly et al., 2002) or utilization  
9984 distributions (Worton, 1989; Fieberg and Kochanny, 2005; Fieberg, 2007) are estimated  
9985 from animal telemetry data. Royle et al. (2012a) argued that SCR models and resource  
9986 selection models estimated from telemetry are based on the same basic underlying model  
9987 of space usage. The important distinction between SCR and RSF studies are that, in  
9988 SCR studies, encounter of individuals is imperfect (i.e., “ $p < 1$ ”) whereas, with RSF data  
9989 obtained by telemetry, encounter is perfect (or, rather, detection is not a *stochastic* out-

9990 come). We can think of the two as being exactly equivalent either if we have a dense  
 9991 array of trapping devices, or if our telemetry apparatus is imperfect such as only samples  
 9992 a small area of space (this would be consistent with telemetry stations for sampling fish  
 9993 which only measure passage).

9994 Telemetry studies are extremely common in animal ecology for studying movement and  
 9995 resource selection, and SCR studies frequently obtain such data on a subset of individuals.  
 9996 Thus, formal integration of capture-recapture with telemetry data for the purposes of  
 9997 modeling resource selection has a number of immediate benefits. For one, telemetry data  
 9998 provide direct information about  $\sigma$  (Sollmann et al., 2012b, in revision). As a result, this  
 9999 leads to improved estimates of model parameters, and also has design consequences (see  
 10000 Sec. 10.4). In addition, active resource selection by animals induces a type of heterogeneity  
 10001 in encounter probability, which is misspecified by standard SCR encounter probability  
 10002 models. As a result, estimates of population size or density under models that do not  
 10003 account for resource selection can be biased (Royle et al., 2012a). Finally, because the  
 10004 resource selection model translates directly to a model for encounter probability for spatial  
 10005 capture-recapture data, the implication of this is that it allows us to estimate resource  
 10006 selection model parameters directly from SCR data, i.e., *absent* telemetry data. This  
 10007 fact should broaden the practical relevance of spatial capture-recapture for studying or  
 10008 estimating not just density, but also for directly studyin movement and resource selection.

10009 Telemetry data has been widely used in conjunction with capture-recapture data.  
 10010 For example, White and Shenk (2001) and Ivan (2012) suggested using telemetry data  
 10011 to estimate the probability that an individual is exposed to sampling. However, their  
 10012 estimator requires that individuals are sampled in proportion to this unknown quantity,  
 10013 which seems impossible to acheive in many studies. In addition, they do not directly  
 10014 integrate the telemetry data with the capture-recapture model so that common parameters  
 10015 are jointly estimated. Sollmann et al. (in revision) and Sollmann et al. (2012b) used  
 10016 telemetry data to directly inform the parameter  $\sigma$  from the bivariate normal SCR model  
 10017 in order to improve estimates of density, although these models did not include an explicit  
 10018 resource selection component.

## 11.1 A SIMPLE MODEL OF SPACE USAGE

10019 We assume here that our landscape is defined in terms of a discrete raster of one or more  
 10020 covariates, having the same dimensions and extent. Let  $\mathbf{x}_1, \dots, \mathbf{x}_{nG}$  identify the center  
 10021 coordinates of  $nG$  pixels that define a landscape. We organize these coordinates into the  
 10022 matrix  $\mathbf{X}_{nG \times 2}$ . Let  $z(\mathbf{x})$  denote a covariate measured (or defined) for every pixel  $\mathbf{x}$ . For  
 10023 clarity, and without loss of generality, we develop the basic ideas here in terms of a single  
 10024 covariate. We suppose that a population of individuals wanders around space in some  
 10025 manner related to the covariate  $z(\mathbf{x})$ , and their locations accumulate in pixels by some  
 10026 omnipotent accounting mechanism. We will define “use of  $\mathbf{x}$ ” to be the event that an  
 10027 individual animal appeared in some pixel  $\mathbf{x}$  during some interval of time.

10028 As a biological matter, use is the outcome of individuals moving around their home  
 10029 range (Hooten et al., 2010), i.e., where an individual is at any point in time is the result  
 10030 of some movement process. However, to understand space usage, it is not necessary to  
 10031 entertain explicit models of movement, just to observe the outcomes, and so we don’t  
 10032 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 (Jonsen et al., 2005; Forester et al., 2007; Patterson et al., 2008; Hooten et al., 2010; McClintock et al., 2012). XXX Also cite Ovasakinen papers XXXXXX We consider explicit movement models in the context of SCR models later chapters of this book (Chapt. XXX and XXX).

If an individual moves from a pixel  $\mathbf{x}$  to another pixel  $\mathbf{x}'$  this is defined as a decision to “use” pixel  $\mathbf{x}'$ . This also induces a definition of “truth” – that is, over any prescribed time interval, the animals makes some number, say  $R$  of use decisions, and they are, conceivably, observable by our omnipotent accounting mechanism (e.g., continuous telemetry). In this case, let  $r_{ij}$  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  $\mathbf{r}_i = (r_{i1}, \dots, r_{iG})$  has a multinomial distribution:

$$\mathbf{r}_i \sim \text{Multinom}(R, \boldsymbol{\pi}_i)$$

where

$$\pi_{ij} = \frac{\exp(\alpha_2 z(\mathbf{x}_j))}{\sum_x \exp(\alpha_2 z(\mathbf{x}))}$$

This is the standard RSF model (Manly et al., 2002) used to model telemetry data. One thing about Manly et al 2002 is that they offer numerous ways of modeling resource selection. They offer three “protocols” (pg 5) describing how used and unused resources are sampled. What we are discussing is their protocol A where all available resources (pixels) are censused, and used pixels are sampled randomly for each individual. They also describe 3 designs that vary in whether or not individual level data is collected. I think it is just worth being aware of this stuff because everybody that talks about RSFs thinks in these terms. The parameter  $\alpha_2$  is the effect of the landscape covariate  $z(\mathbf{x})$  on the relative probability of use. Thus, if  $\alpha_2$  is positive, the relative probability of use increases as the covariate increases.

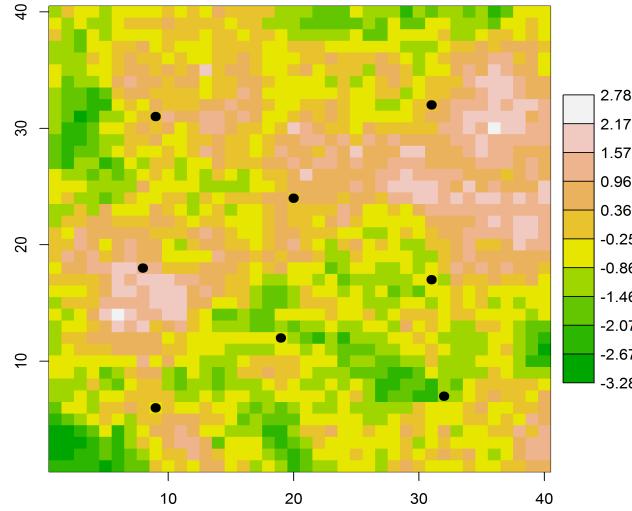
In practice, we don’t get to observe  $r_{ij}$  for all individuals but, instead, only for a small subset which we capture and install telemetry devices on. For the telemetered individuals, we assume their behavior is according to the same RSF model as the population as a whole.

We extend this model slightly to make it more realistic spatially. Let  $\mathbf{s}$  denote the centroid of an individuals 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 individuals home range centroid:

$$\pi_{ij} = \frac{\exp(-\alpha_1 D_{ij}^2 + \alpha_2 z(\mathbf{x}_j))}{\sum_x \exp(-\alpha_1 D_{ij}^2 + \alpha_2 z(\mathbf{x}_j))} \quad (11.1.1)$$

where  $\alpha_1 = 1/(2\sigma^2)$  describes the rate at which capture probability declines as a function of distance. This has some context w.r.t. Johnson et al. and Forester in terms of modeling “availability” as a function of distance. But it is not necessary to distinguish between use vs. availability – really this model is cleanly interpreted as individuals using space as a function of how far away  $\mathbf{x}$  is from the individuals home range center. Don’t see a need to call that “availability”.

Note that Eq. 11.1.1 resembles standard encounter models used in spatial capture-recapture but with an additional covariate  $z(\mathbf{x})$  (and see Chapt. 9). In particular, under this model for space usage or resource selection, if you have no covariates at all, or if  $\alpha_2 = 0$ , then the probabilities  $\pi_{ij}$  are directly proportional to the SCR model for encounter



**Figure 11.1.** 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.

10074 probability. For example, setting  $\alpha_2 = 0$ , then this implies probability of use for pixel  $j$   
 10075 is:

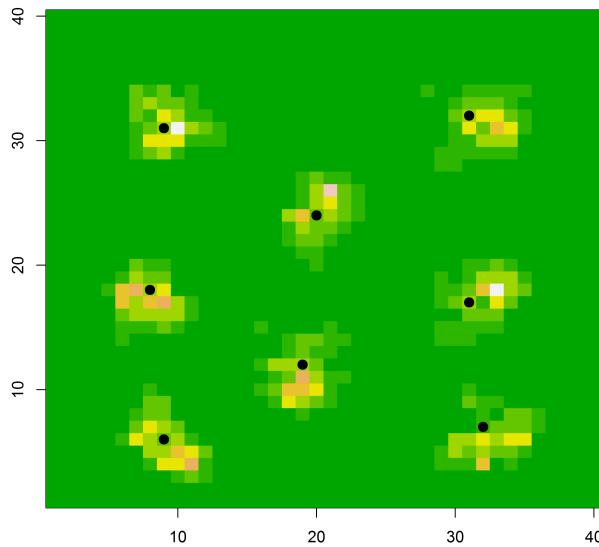
$$p_{ij} \propto \exp(-\alpha_1 D_{ij}^2)$$

10076 so whatever function of distance we use in our RSF implies an equivalent model of space  
 10077 usage (sec. 5.3) when used in SCR models. In particular, for whatever model we choose  
 10078 for  $p_{ij}$  in an ordinary SCR model, we can modify the distance component in the RSF  
 10079 function in Eq. 11.1.1 accordingly to be consistent with that model, by using whatever  
 10080 function  $p_{ij}$  we choose according to

$$\pi_{ij} \propto \exp(\log(p_{ij}) + \alpha_2 z(\mathbf{x}_j))$$

10081 One difference between this observation model and those that we have considered in  
 10082 previous chapters is that it includes the normalizing constant  $\sum_x \exp(-\alpha_1 D_{ij}^2 + \alpha_2 z(\mathbf{x}_j))$ ,  
 10083 which ensures that the use distribution is a proper probability density function. Thus we  
 10084 are able to characterize the probability of encounter in terms of both distance and space  
 10085 use.

10086 As an illustration of space usage patterns under this model, we simulated a covariate  
 10087 that represents variation in habitat structure (Fig. 11.1) such as might correspond to  
 10088 habitat quality. This was simulated by using a kriging interpolator with the following R  
 10089 commands:



**Figure 11.2.** Space usage patterns of 8 individuals under a space usage model that contains a single covariate (shown in Fig. 11.1). Plotted value is the multinomial probability  $\pi_{ij}$  for pixel  $j$  under the model in Eq. 11.1.1.

```

10090 set.seed(1234)
10091 gr<-expand.grid(1:40,1:40)
10092 Dmat<-as.matrix(dist(gr))
10093 V<-exp(-Dmat/5)
10094 z<-t(chol(V))%*%rnorm(1600)

```

10095 Space usage patterns for 8 individuals are shown in Fig. 11.2, simulated with  $\alpha_1 = 1/(2\sigma^2)$   
 10096 with  $\sigma = 2$  and the coefficient on  $z(\mathbf{x})$  set to  $\alpha_2 = 1$ . These space usage densities – “home  
 10097 ranges” – exhibit clear non-stationarity in response to the structure of the underlying  
 10098 covariate, and they are distinctly asymmetrical. We note that if  $\alpha_2$  were set to 0, the 8  
 10099 home ranges shown here would resemble bivariate normal kernels with  $\sigma = 2$ . Another  
 10100 interesting thing to note is that the activity centers are not typically located in the pixel  
 10101 of highest use or even the centroid of usage. That is, the observed “average” location is  
 10102 not an unbiased estimator of  $\mathbf{s}$  under the model in Eq. 11.1.1.

---

**11.1.1 Poisson use model**

10103 A natural way to motivate the multinomial model of space usage is to assume that in-  
 10104 individuals make a sequence of resource selection decisions so that the outcomes  $r_{ij}$  are  
 10105 marginally *independent*, having a Poisson distribution:

$$r_{ij} \sim \text{Poisson}(\lambda_{ij})$$

10107 where

$$\log(\lambda_{ij}) = a_0 - \alpha_1 D_{ij}^2 + \alpha_2 z(\mathbf{x})$$

10108 In this case, the number of visits to any particular cell is affected by the covariate  $z(\mathbf{x})$   
 10109 but has a baseline rate ( $\exp(a_0)$ ) related to the amount of movement occurring over some  
 10110 time interval. This is an equivalent model to the multinomial model given previously in  
 10111 the sense that, if we condition on the total sample size  $r_i = \sum_j r_{ij}$ , then the vector  $\mathbf{r}_i$   
 10112 has a multinomial distribution with probabilities

$$\pi_{ij} = \frac{\lambda_{ij}}{\sum_j \lambda_{ij}}$$

10113 which is the same as Eq. 11.1.1 (see also Chapt. 9) because  $a_0$  cancels from the numerator  
 10114 and denominator of the multinomial cell probabilities and thus this parameter is not  
 10115 relevant to understanding space usage.

10116 Also note that if use frequencies are summarized over individuals for each pixel, i.e.,  
 10117 create the totals  $r_j = \sum_i r_{ij}$ , then a standard Poisson regression model for the resulting  
 10118 “quadrat counts” is reasonable. This is “Design I” in Manly et al. (2002).

**11.1.2 Thinning**

10120 Suppose our sampling is imperfect so that we only observe a smaller number of telemetry  
 10121 fixes than true use,  $r_{ij}$ . As developed in sec. 5.3, we assume that the observed number of  
 10122 uses is

$$m_{ij} \sim \text{Bin}(r_{ij}, \phi_0).$$

10123 We can think of these counts as arising by thinning the underlying point process (here,  
 10124 aggregated into pixels) where  $\phi_0$  is the thinning rate of the point process. In this case,  
 10125 the marginal distribution of  $m_{ij}$  is also Poisson but with mean

$$\log(\lambda_{ij}) = \log(\phi_0) + a_0 - \alpha_1 D_{ij}^2 + \alpha_2 z(\mathbf{x}).$$

10126 Thus, the space-usage model (RSF) for the thinned counts  $m_{ij}$  is the same as the space-  
 10127 usage model for the original variables  $r_{ij}$ . This is because if we remove  $r_{ij}$  from the  
 10128 conditional model by summing over its possible values, then the vector of  $\mathbf{m}_i$  is *also*  
 10129 multinomial with cell probabilities

$$\pi_{ij} = \frac{\phi_0 \lambda_{ij}}{\sum_j \phi_0 \lambda_{ij}}$$

10130 and so the nuisance parameter  $\phi_0$  cancels from the numerator and denominator. Thus,  
 10131 the underlying RSF model applies to the true unobserved count frequencies  $\mathbf{r}_i$  and also  
 10132 those produced from thinning or sampling,  $\mathbf{m}_i$ .

10133 In summary, if we conduct a telemetry study we observe  $\mathbf{r}_i$ , the  $nG \times 1$  vector of pixel-  
 10134 counts for each individual  $i = 1, \dots, N_{tel}$ . We declare these data to be “resource-selection  
 10135 data” which are typical of the type used to estimate resource-selection functions (RSFs)  
 10136 (Manly et al., 2002). Sometimes in RSF modeling activities we might have continuous  
 10137 covariates and so the denominator in Eq. 11.1.1 involves an integration over a distribution  
 10138 for the covariate which is the conditional intensity of observed point locations in a point  
 10139 process model. However, in a discrete landscape, entertaining pdfs for the covariates isn’t  
 10140 necessary (Royle et al., 2012a) when we recognize that the denominator should be the  
 10141 expectation over *space* and not the pdf of some covariate.

### 10142 11.1.3 Capture-recapture Data

10143 XXXXXXXXXXXXXXXXXXXX RC says: After reconciling SCR and RSF, cite that pa-  
 10144 per by Boyce and McDonald where they try to do accomplish the same objective using  
 10145 ad-hoc methods. That is the only effort to do something similar that I am aware of.  
 10146 XXXXXXXXXXXXXXXXXXXXXXX

10147 The key to combining RSF data with SCR data is to work with this underlying resource  
 10148 utilization process and formulate SCR models in terms of that process. The idea in (Royle  
 10149 et al., 2012a) was to define the true use frequency for each pixel as the intermediate latent  
 10150 variable to which both telemetry data and SCR data are linked. Obviously we have to  
 10151 assume that both telemetered individuals and SCR individuals are using space according  
 10152 to the same resource selection model. The difference is that, for SCR data, we do not  
 10153 have sampling devices in all locations (pixels) in the landscape, and hence the data are  
 10154 only recorded at a subsample of them. XXXXX move the following XXXXX In other  
 10155 words, imagine that we have a sampling device, such as a camera trap, in *every* pixel. If  
 10156 the device operates continually then it is no different from a telemetry instrument. If it  
 10157 operates intermittently and does not expose the entire area of each pixel then a reasonable  
 10158 model for this imperfect observation is the “thinned” binomial model given above, where  
 10159  $\lambda_0 \equiv \exp(\phi_0)$  represents the sampling effectiveness of the device. So we imagine that the  
 10160 hypothetical perfect data from a camera trapping study are the thinned counts  $m_{ij}$  for  
 10161 every pixel  $j$ .

10162 Introducing the latent use frequencies  $m_{ij}$ , and considering the Bernoulli SCR model  
 10163 where  $y_{ij} = 1$  if the individual  $i$  visited the pixel containing a trap and was detected, then  
 10164 we imagine that  $y_{ij}$  is related to the latent variable  $m_{ij}$  being the event  $m_{ij} > 0$ , which  
 10165 occurs with probability

$$p_{ij} = 1 - \exp(-\lambda_{ij})$$

10166 where

$$\log(\lambda_{ij}) = \log(\phi_0) + a_0 - \alpha_1 D_{ij}^2 + \alpha_2 z(\mathbf{x}).$$

10167 We combine the constants so that  $\alpha_0 = \log(\phi_0) + a_0$  is the baseline encounter rate which  
 10168 includes the constant intensity of use by the individual and also the baseline rate of  
 10169 detection, conditional on use. The Bernoulli observation model implies that the observed  
 10170 encounter frequencies for individual  $i$  and trap  $j$ , from sampling over  $K$  encounter periods  
 10171 is:

$$y_{ij} | \mathbf{s}_i \sim \text{Bin}(K; p_{ij})$$

10172 We imagine that any of the standard SCR observation model could be implemented here  
 10173 with only minor modifications of the encounter probability model (following the develop-  
 10174 ments of Chapt. 9.

## 11.2 THE JOINT RSF/SCR LIKELIHOOD

10175 To construct the likelihood for SCR data when we have auxiliary covariates on space usage  
 10176 or direct information on space usage from telemetry data, we regard the two samples (SCR  
 10177 and RSF) as independent of one another. In practice, this might not always be the case  
 10178 but (1) often time the telemetry data come from a previous study; (2) the individuals are  
 10179 not the same at all; (3) or even if they are some of the same individuals being captured,  
 10180 we might not be able to match individuals captured by a sampling method such as hair-  
 10181 snares with the individuals wearing radio-collars; (4) In cases where we *can* match some  
 10182 individuals between the two samples, regarding them as independent should only entail a  
 10183 minor loss of efficiency because we are disregarding more precise information on a small  
 10184 number of activity centers. Moreover, we believe, it is unlikely in practice to expect the  
 10185 two samples to be completely reconcilable and that the independence formulation is the  
 10186 most generally realistic.

10187 Regarding the two data sets as being independent, our approach here is to form the  
 10188 likelihood for each set of observations as a function of the same underlying parameters and  
 10189 then combine them. In particular, let  $\mathcal{L}_{scr}(\alpha_0, \alpha_1, \alpha_2, N; \mathbf{y}_{scr})$  be the likelihood for the  
 10190 SCR data in terms of the basic encounter probability parameters and the total (unknown)  
 10191 population size  $N$ , and let  $\mathcal{L}_{rsf}(\alpha_1, \alpha_2; \mathbf{m}_{rsf})$  be the likelihood for the RSF data based  
 10192 on telemetry which, because the sample size of such individuals is fixed, does not depend  
 10193 on  $N$ . Assuming independence of the two datasets, the joint likelihood is the product of  
 10194 these two pieces:

$$\mathcal{L}_{rsf+scr}(\alpha_0, \alpha_1, \alpha_2, N; \mathbf{y}_{scr}, \mathbf{m}_{rsf}) = \mathcal{L}_{scr} \times \mathcal{L}_{rsf}$$

10195 In what follows, we provide a formulation of each likelihood component, along with an R  
 10196 function for obtaining the MLEs of model parameters using standard methods available  
 10197 in R.

10198 Where the  $L(scr)$  is the normal integrated likelihood (chapt XXX, equation XXXX)  
 10199 and the rsf likelihood is the multinomial telemetry likelihood from eq. XXXX above.  
 10200 That is, if we estimate  $s$  for the telemtered guys by the mean location then we could use  
 10201 that plug-in otherwise we also have to integrate the multinomial likelihood ..... For the  
 10202 RSF data from the sample of individuals with telemetry devices we adopt the same basic  
 10203 strategy of describing the conditional-on-s likelihood and then computing the marginal  
 10204 likelihood by averaging over possible values of  $s$ . We have  $\mathbf{m}_i$ , the vector of pixel counts  
 10205 for individual  $i$ , where these counts are derived from a telemetry study or similar. The  
 10206 conditional-on- $\mathbf{s}_i$  distribution of the telemetry data from individual  $i$  is:

$$[\mathbf{m}_i | \boldsymbol{\alpha}] = \prod_{g=1}^G \pi_{ij}(\mathbf{s}_i, z(\mathbf{x}_j))^{r_{ij}}$$

10207 where

$$\pi_{ij} = \frac{\exp(-\alpha_1 D_{ij}^2 + \alpha_2 z(\mathbf{x}_j))}{\sum_g \exp(-\alpha_1 D_{ij}^2 + \alpha_2 z(\mathbf{x}_j))}$$

10208 The marginal pmf is:

$$[\mathbf{m}_i | \boldsymbol{\alpha}] = \int_{\mathcal{S}} [\mathbf{m}_i | \mathbf{s}_i, \boldsymbol{\alpha}] g(\mathbf{s}_i) d\mathbf{s}_i$$

10209 and therefore the likelihood for the RSF data is

$$\mathcal{L}_{rsf}(\boldsymbol{\alpha} | \mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_{Ntel}) = \prod_{i=1}^{Ntel} [\mathbf{m}_i | \boldsymbol{\alpha}].$$

10210 An R script is given in `scrbook` which is the function XXXX from manuscript ADD  
 10211 TO REPO XXXXXX... will handle estimated s by sbar or whatever. Should be obvious  
 10212 that estimating s with sbar is not the greatest thing because there is a covariate affecting  
 10213 use, and so we expect the geographic centroid to be biased for s.

### 11.3 APPLICATION: NEW YORK BLACK BEAR STUDY

10214 Royle et al. (2012a) applied the integrated SCR+RSF model to data from a study of  
 10215 black bears in a region of approximately 4,600 km<sup>2</sup> in southwestern New York (Sun,  
 10216 in prep). We reproduce their results here. The data can be loaded from the `scrbook`  
 10217 library with the command `data(nybears)`. **XXXX do this on the repo XXXXX**  
 10218 A noninvasive, genetic, mark-recapture study was conducted to estimate density, and a  
 10219 concurrent telemetry study was conducted in the same study area to understand patterns  
 10220 of landscape connectivity and space usage. The study used DNA sampling obtained from  
 10221 103 hair snares (Fig. ??) set from 6 June - 9 July, 2011. Hair snares were baited and  
 10222 scented and checked weekly for hair. See Sun (in prep) for details of the genetic analysis.

10223 The study yielded captures of 33 individuals and a total of 14 recaptures (27 individuals  
 10224 captured 1 time only; 3 individuals captured twice; 1 individual each three and four times).  
 10225 Extra trap recaptures included 3 individuals captured in 2 traps, 1 individual in each of 3  
 10226 and 4 traps. We used data from 3 radio-telemetered individual bears (2M, 1F) from the  
 10227 same time period as the SCR data. Radio fixes were obtained approximately once per  
 10228 hour and a total of 1,948 fixes on the 3 individuals were obtained. We thinned these hourly  
 10229 fixes to once per 10 hours to approximate the data as independent movement outcomes,  
 10230 producing 195 telemetry locations used in the RSF component of the model. We used the  
 10231 covariate elevation in the model, derived from a one arc-second digital elevation model  
 10232 (USGS National Elevation Dataset, accessed June 2012). This is shown in Fig. ?? (on a  
 10233 standardized scale) which also shows the locations of each capture (multiple captures at  
 10234 a trap location are dithered by adding random noise).

10235 We fitted a sequence of models based on the Gaussian hazard model (eq. 17.2.1)  
 10236 including an ordinary SCR model with no covariates or telemetry data, the SCR model  
 10237 with elevation affecting either  $\lambda_0$  or density  $D(\mathbf{x})$ , and models that use telemetry data. We  
 10238 have not discussed modeling covariate effects on density, but such models are described  
 10239 by Borchers and Efford (2008) and we have not provided any novel treatment of that  
 10240 modeling aspect here. The full list of models (with labels) is as follows:

10241 Model 1: SCR – ordinary SCR model

10242 Model 2: SCR+p(z) – ordinary SCR model with elevation as a covariate on baseline  
 10243 encounter probability  $\lambda_0$ .

10244 Model 3: SCR+D(z) – ordinary SCR model with elevation as a covariate on density  
 10245 only.  
 10246 Model 4: SCR+p(z)+D(z) – ordinary SCR model with elevation as a covariate on both  
 10247 baseline encounter probability and density.  
 10248 Model 5: SCR+p(z)+RSF – SCR model including data from 3 telemetered individuals.  
 10249 Model 6: SCR+p(z)+RSF+D(z) – SCR model including telemetered individuals and  
 10250 with elevation as a covariate on density.

10251 The first 4 models can be viewed together for purposes of model-selection by AIC since  
 10252 they are nested models. The last two models can be viewed together but cannot be  
 10253 compared to the first 4 because they include telemetry data. The results of fitting these  
 10254 6 models – the parameter estimates and standard errors are shown in Table 11.1. We  
 10255 provide a full R script for fitting all of these models to simulated data in Appendix 1.

10256 Among models 1-4, those models *without* the telemetry data, we see that the two mod-  
 10257 els with elevation affecting density are preferred – and, there is a large positive response  
 10258 to elevation. This is consistent with the visual pattern apparent in Fig. ?? where we see  
 10259 individual captures favoring high elevation sites. We also see a negative effect of elevation  
 10260 on *space usage* (the parameter  $\alpha_2$ ). It is interesting that the sign of the estimate of  $\alpha_2$   
 10261 changes from positive to negative when we add elevation as a covariate on density. Thus,  
 10262 the effect of elevation on density appears to have masked its effect on space usage. The  
 10263 estimate of  $N$  for the 4600 km<sup>2</sup> state-space is about 103 bears ( $\exp(4.25) + 33$ ).

10264 In the two models that include the additional telemetry data, a couple points stand out:  
 10265 Clearly the elevation effect on density is important, reducing the negative log-likelihood  
 10266 by 5 units. The effect of elevation on density and space usage are roughly consistent with  
 10267 Model 4 which did not use telemetry data. Furthermore, the standard errors (SE) of  
 10268 those two parameter estimates are reduced considerably when the model uses telemetry  
 10269 data, as is the SE for estimating  $\log(\sigma)$ . The SE for estimating  $\log(n_0)$  is only improved  
 10270 incrementally compared to the models without telemetry data. We used the best model,  
 10271  $\text{SCR}+p(z)+\text{RSF}+D(z)$ , to produce a map of density (Fig. ??) which shows clearly the  
 10272 pattern induced by elevation. We also produced a map (Fig. ??) to illustrate the effect  
 10273 of elevation on space usage. This shows the relative probability of using a pixel  $x$  relative  
 10274 to one of mean elevation, and of the same distance from an individual's activity center.

10275 Resource selection can be described in hierarchical orders (Johnson, 1980), from selec-  
 10276 tion of a geographical area (first-order selection), selection of a home range within a study  
 10277 area (second-order), or selection of resources with that home range (third-order). Animals  
 10278 may select resources at different scales as a result of variability in the distribution of re-  
 10279 sources on the landscape (Mayor et al., 2009). Indeed, black bears make habitat selection  
 10280 decisions at multiple spatial scales, and decisions made at the second-order can differ from  
 10281 those at the third-order (Lyons et al., 2003; Sadeghpour and Ginnett, 2011). As a result  
 10282 of multi-scale resource selection, we can expect that the modeled covariates (elevation in  
 10283 our example) may affect density and space usage differently. We suggest that density is  
 10284 operating at the second-order and is largely related to the spacing of individuals and their  
 10285 associated home ranges across the landscape. On the other hand, our RSF was defined  
 10286 based on selection of resources within the home range (third-order). Because density and  
 10287 our third-order RSF were at different spatial scales, there is no expectation that the mod-  
 10288 eled covariate describing space usage (elevation) would influence each in a similar manner.  
 10289 Consistent with our positive relationship between elevation and density, the distribution

**Table 11.1.** Summary of model-fitting results for the black bear study. Parameter estimates are  $\alpha_0 = \log(\lambda_0)$  and  $\sigma$  is the scale parameter of the half-normal hazard rate encounter model. The SCR data are based on  $n = 33$  individuals, and the telemetry data are based on 3 individuals.

model	$\alpha_0$	$\log(\sigma)$	$\alpha_2$	$\log(n_0)$	$\beta$	-loglik
SCR+p(z)	-2.8600	-1.1170	0.1750	4.1400		122.7380
SE	0.3899	0.1390	0.2478	0.3657		
SCR	-2.7290	-1.1220	—	4.1100		122.9900
SE	0.3454	0.1404		0.3618		
SCR+D(z)	-2.7150	-1.1330	—	4.1140	1.2470	118.0070
SE	0.3526	0.1394		0.3575	0.4083	
SCR+p(z)+D(z)	-2.4840	-1.1570	-0.3840	4.2550	1.5710	117.0750
SE	0.3910	0.1421	0.2761	0.3768	0.4630	
SCR+RSF	-3.0680	-0.8140	-0.2810	3.8840		1271.7390
SE	0.2722	0.0364	0.1176	0.3626		
SCR+RSF+D(z)	-3.0700	-0.8100	-0.3710	4.0280	1.2730	1266.7000
SE	0.2720	0.0368	0.1239	0.3661	0.4110	

of a black bear population in the central Appalachian Mountains was positively associated with elevation (Frary et al., 2011). At the second-order, however, we observed a negative effect of elevation on space usage. Our study was conducted during summer, and seasonal shifts in elevation have been widely documented in black bears, often attributed to seasonal variation in food availability (Reynolds and Beecham, 1980; Gruber and White, 1983). The negative relationship between elevation and space usage during the summer could be attributable to either access to food resources at lower elevations, or access to river and stream corridors. Within their home ranges, black bears selected areas with high stream densities (Fecske et al., 2002), and in our study area, lower elevations were associated with river corridors which likely provided bears cooler conditions during the heat of summer.

## 11.4 SIMULATION STUDY

Royle et al. XXXXX carried-out a simulation study using the landscape shown in Fig. 11.1, and based on a population of  $N = 100$  and  $N = 200$  individuals with activity centers distributed uniformly over the landscape. This patchy covariate was simulated by generating a field of spatially correlated noise to emulate a typical patchy habitat covariate such as tree or understory density, or some other covariate relevant to habitat quality for a species. We subjected individuals to sampling over  $K = 10$  sampling periods, using a  $7 \times 7$  array of trapping devices located on the integer coordinates ( $u * 5, v * 5$ ) for  $u, v = 1, 2, 3, 4, 5, 6, 7$ . The model parameters were

$$\text{cloglog}(p_{ij}) = -2 - \frac{1}{2\sigma^2} D_{ij}^2 + 1 \times z(\mathbf{x}_j)$$

for  $\sigma = 2$ . In the absence of the covariate  $z$ , this corresponds to an individual having a bivariate normal home range with standard deviation 2. These settings yielded an average of about  $n = 61$  individuals captured for the  $N = 100$  case and about  $n = 123$  for the

10312     $N = 200$  case. The later case represents what we believe is an extremely large sample size  
 10313    based on our own experience and thus it should serve to gauge the large sample bias of  
 10314    the likelihood estimator (note: we expect little to no large sample bias).

10315    In addition to simulating data from this capture-recapture study, we simulated 2, 4,  
 10316    8, 12, 16 telemetered individuals to assess the improvement in precision as sample size  
 10317    increases. For all cases we observed 20 telemetry fixes *per* individual. The main things  
 10318    we're focused on with this simulation study were: (1) how much does the SE of estimated  
 10319     $N$  improve as we add or increase the number of telemetered individuals? (2) How well  
 10320    does the SCR model do at estimating the parameter of the RSF with *no* telemetry data?  
 10321    (3) How much does the precision of the RSF parameter improve if we add SCR data to  
 10322    the telemetry data?

	N=100, 300 iters each, mean SCR only N: 99.418						N=200, 500 iters. Mean SCR only N = 199.712						
	n=2	Nhat	RMSE	ahat	RMSE	sighat	RMSE	Nhat	RMSE	ahat	RMSE	sighat	RMSE
10323	SCR only:	99.73	9.97	0.99	0.14	2.00	0.124	198.85	14.24	0.99	0.10	2.00	0.091
10324	SCR/RSF:	99.94	9.54	0.99	0.12	2.00	0.097	199.37	12.80	0.99	0.09	2.00	0.078
10325	sbar	98.89	9.50	0.93	0.14	1.97	0.100	197.87	13.94	0.96	0.10	1.99	0.080
10326	RSF only	--	--	1.03	0.33	2.00	0.160	--	--	1.04	0.33	1.99	0.169
10327	n=4												
10328	SCR only	99.10	9.83	0.99	0.13	2.00	0.127	200.06	15.34	1.00	0.09	2.00	0.092
10329	SCR/RSF	99.17	9.47	0.99	0.11	2.00	0.086	200.25	14.36	1.00	0.08	2.01	0.073
10330	sbar	97.43	9.68	0.89	0.16	1.97	0.090	198.14	14.31	0.94	0.10	1.98	0.075
10331	RSF only	--	--	0.98	0.22	2.00	0.119	--	--	1.02	0.21	2.01	0.122
10332	n=8												
10333	SCR only	99.59	10.00	1.00	0.13	2.00	0.130	200.85	14.06	1.00	0.09	2.00	0.087
10334	SCR/RSF	98.90	10.02	0.99	0.10	2.00	0.071	200.29	13.98	1.00	0.08	2.00	0.061
10335	sbar	96.07	10.37	0.84	0.19	1.96	0.078	196.46	14.59	0.90	0.13	1.97	0.069
10336	RSF only	--	--	0.98	0.16	2.01	0.084	--	--	0.99	0.16	2.00	0.084
10337	n=12												
10338	SCR only	99.44	10.73	0.98	0.13	2.02	0.128	198.76	14.47	0.99	0.10	2.00	0.091
10339	SCR/RSF	99.96	10.26	1.00	0.09	2.00	0.059	198.72	14.14	1.00	0.08	2.00	0.054
10340	sbar	96.30	10.49	0.82	0.20	1.96	0.071	193.83	15.14	0.87	0.15	1.97	0.063
10341	RSF only	--	--	1.01	0.12	2.00	0.069	--	--	1.01	0.13	2.00	0.069
10342	n=16												
10343	SCR only	99.23	10.74	0.99	0.14	2.00	0.128	200.04	14.09	0.99	0.10	2.01	0.088
10344	SCR/RSF	99.20	9.79	1.00	0.09	1.99	0.057	200.25	13.40	1.00	0.07	2.00	0.047
10345	sbar	95.10	10.17	0.80	0.22	1.95	0.075	194.38	14.26	0.85	0.17	1.96	0.059
10346	RSF only	--	--	1.00	0.10	1.99	0.061	--	--	1.00	0.11	2.00	0.055
10347	To check misspecification with isotropic h/r model I refitted the N =												
10348	200 cases and fit the SCR only and SCR/RSF models IN ADDITION to the												
10349	SCR0 model with isotropic encounter model.												
10350	n=2												
10351	SCR only	199.11	14.28	0.99	0.09	2.00	0.090						
10352	SCR/RSF	199.11	13.80	0.99	0.09	2.00	0.079						
10353	sbar	161.48	39.98	--	--	1.84	0.180						
10354	RSF only	--	--	1.00	0.10	1.99	0.061	--	--	1.00	0.11	2.00	0.055

---

10358	n=4						
10359	SCR only	199.67	13.87	1.00	0.09	2.00	0.090
10360	SCR/RSF	199.65	13.59	1.00	0.09	2.00	0.072
10361	SCRO	161.32	40.00	--	--	1.83	0.191
10362							
10363	n=8						
10364	SCR only	199.24	15.49	0.99	0.10	2.01	0.093
10365	SCR/RSF	199.55	14.17	0.99	0.08	2.00	0.063
10366	SCRO	161.46	40.06	--	--	1.84	0.184
10367							
10368	n=12						
10369	SCR only	200.41	15.16	0.99	0.10	2.00	0.086
10370	SCR/RSF	200.95	13.04	1.00	0.08	2.00	0.051
10371	SCRO	162.40	38.95	--	--	1.84	0.185
10372							
10373	n=16						
10374	SCR only	199.16	15.62	1.00	0.09	2.00	0.095
10375	SCR/RSF	199.63	13.38	1.00	0.07	2.00	0.052
10376	SCRO	160.93	40.44	--	--	1.84	0.190

The replicate runs of the SCR-only situation give us an idea of the inherent MC error in these simulations, which is roughly about 0.25 and 0.89 on the  $N$  scale for the  $N = 100/N = 200$  cases. The mean  $N$  under “SCR only” across all 5 simulations for  $N = 100$  was  $\text{mean}(\hat{N}) = 99.418$ , an empirical bias of 0.6%. For  $N=200$ , the estimated  $N$  across all 5 simulations (5 levels of ntel) was  $\text{mean}(\hat{N}) = 199.712$ , an empirical bias of about 0.15%, within the MC error of the true value of  $N = 200$ . The results suggests a very small bias of < 1% in the MLE of  $N$  in general when estimation is based on the full marginal likelihood. However, there is apparent bias of as much as 2-4% when  $s$  is estimated by the average observed location. The bias is slightly diminished as we double the expected sample size by doubling  $N$  from 100 to 200. In practice, we expect a small amount of bias in MLEs as likelihood theory only guarantees asymptotic unbiasedness. Moreover, the landscape resolution is fairly coarse relative to  $\sigma$  in our study, having a 1 km resolution whereas  $\sigma = 2$ , which we expect to introduce a small amount of negative bias because it is an explicit under-statement of the true heterogeneity in  $p$  due to the spatial context of the problem. The apparent bias that arises as a result of estimating  $s$  is expected because the average location of an individual would be unbiased for  $s$  only if the individual is moving according to a stationary isotropic kernel. Under the model of space usage with covariate  $z(\mathbf{x})$ , then the average location is biased to favor good values of  $z(\mathbf{x})$  and so  $\bar{s}$  is really biased for  $s$ .

In terms of RMSE of the MLEs, generally there is about a 5% reduction in RMSE when we have at least 2 telemetered individuals, and, although there is a lot of MC error in the RMSE quantities, it might be as much as a 10% reduction (tops) as  $n$  increases under the higher  $N = 200$  setting. This makes sense because we nail down the parameters and still don't know where guys are, and get info about mean  $p$ , i.e.  $\alpha_0$ , only from the SCR data. Thus estimating  $N$  benefits only slightly from the addition of telemetry data.

Estimating the RSF parameter  $\alpha_2$  exhibits negligible or no bias except when  $s$  is estimated and, interestingly, it is well-estimated from SCR data alone and even better

than RSF data alone (in terms of RMSE) until we have more than 200 or so telemetry observations. The big improvement comes in estimating the home range parameter  $\sigma$  which is unbiased except when we estimate  $s$  in which case it exhibits only modest bias. However, there is huge improvement in RMSE of  $\hat{\sigma}$ , perhaps as much as 50-60% in some cases, but that really doesn't translate much into estimating  $N$ . Improvement due to adding RSF data from telemetry diminishes as the expected sample sizes increases, and so telemetry data does less to improve the precision of  $\hat{\sigma}$  and  $\hat{\alpha}_2$  for  $N = 200$  than for  $N = 100$ .

We simulated a low  $p$  situation in which  $\alpha_0 = -3$  producing  $E[n] = 37$  under the  $N = 100$  scenario. The effect is we have only incremental relative improvements in RMSE of  $N$  but relatively more improvement in RMSE for estimating  $\sigma$ . The MLE of  $N$  is positively biased. Interestingly this bias opposes slightly negative bias for the estimator based on estimating  $s$  and so that the wrong estimator actually does better. This is a complete chance occurrence and we should not get too excited by that.

N=100, low p, 500 iterations							
	n=2	Nhat	RMSE	ahat	RMSE	sighat	RMSE
10420	SCR only	103.85	22.88	1.00	0.19	2.02	0.261
10421	SCR/RSF	102.90	20.98	1.00	0.17	2.00	0.136
10422	sbar	101.55	20.91	0.90	0.19	1.96	0.136
10423	RSF only	--	--	1.02	0.30	1.99	0.163
10424	n=4						
10425	SCR only	105.65	26.52	1.01	0.20	2.01	0.258
10426	SCR/RSF	103.55	22.92	1.01	0.14	2.00	0.104
10427	sbar	100.86	22.57	0.86	0.20	1.95	0.113
10428	RSF only	--	--	1.01	0.21	1.99	0.114
10429	n=8						
10430	SCR only	107.41	45.05	0.99	0.19	2.01	0.254
10431	SCR/RSF	104.28	22.13	1.00	0.12	2.00	0.076
10432	sbar	99.82	21.55	0.80	0.23	1.95	0.091
10433	RSF only	--	--	1.01	0.15	1.99	0.081
10434	n=12						
10435	SCR only	106.35	27.32	0.99	0.19	2.00	0.255
10436	SCR/RSF	104.11	21.81	1.00	0.10	2.00	0.063
10437	sbar	99.21	20.86	0.77	0.24	1.95	0.077
10438	RSF only	--	--	1.01	0.12	2.00	0.065
10439	n=16						
10440	SCR only	104.05	31.41	0.99	0.19	2.02	0.252
10441	SCR/RSF	101.98	20.78	1.00	0.09	2.00	0.055
10442	sbar	96.78	20.25	0.76	0.26	1.95	0.070
10443	RSF only	--	--	1.00	0.10	2.00	0.056

## 11.5 SUMMARY AND OUTLOOK

How animals use space is a fundamental interest to ecologists, and important in the conservation and management of many species. Normally this is done by telemetry and models referred to as resource selection functions (Manly et al., 2002). Conversely, spatial

capture-recapture models have grown in popularity over the last several years (Efford, 2004; Borchers and Efford, 2008; Royle, 2008; Efford et al., 2009b; Royle et al., 2009a; Gardner et al., 2010a,b; Kéry et al., 2010; Sollmann et al., 2011; Mollet et al., 2012; Gopalaswamy et al., 2012b). These, and indeed, most, development and applications of SCR models have focused on density estimation, not understanding space usage. However, it is intuitive that space usage should affect encounter probability and thus it should be highly relevant to density estimation in SCR applications. Despite this, a description of the relationship between encounter probability and space usage has not been developed explicitly in the literature on spatial capture-recapture models. Here we developed an SCR model in terms of a basic underlying model of space or resource use, that is consistent with existing views of resource selection functions (RSFs) (Manly et al., 2002).

Basically everyone does telemetry with SCR even though no one knows what to do with this stuff.

Our new class of integrated SCR/RSF models allows investigators to model how the landscape and habitat influence movement and space usage of individuals around their home range, using non-invasively collected capture-recapture data or capture-recapture data augmented with telemetry data. This should improve our ability to understand, and study, aspects of space usage and it might, ultimately, aid in addressing conservation-related problems such as reserve or corridor design. And, it should greatly expand the relevance and utility of spatial capture-recapture beyond simply its use for density estimation.

Integration of RSF data from telemetry with SCR models achieves a number of useful extensions of both ordinary SCR and RSF models: (1) Integration of the two distinct data sources (capture-recapture and telemetry) leads to an improvement in our ability to estimate density, and also an improvement in our ability to estimate parameters of the RSF function. As many animal population studies have auxiliary telemetry information, the ability to incorporate such information into SCR studies has broad applicability to many studies. It seems possible even to estimate density now, with no spatial recaptures, provided telemetry data are available. (2) The integrated model allows for the estimation of RSF model parameters directly from SCR data *alone*. This establishes clearly that SCR models *are* explicit models of space usage. In our view, this greatly broadens the utility and importance of capture-recapture studies beyond their primary historical use of estimating density or population size. (3) It is also now clear that one of the important parameters of SCR models, that controlling “home range radius”, is also directly estimable from telemetry data alone, and certainly its estimation is greatly improved with even moderate amounts of telemetry data. We pursue this topic from a design standpoint in Chapt. 10. (4) Resource selection can be viewed as inducing a type of heterogeneous encounter probability in capture-recapture studies. We say (Royle et al., 2012a) that misspecification of a simple resource selection model with a symmetric encounter probability model produces extremely biased estimates of  $N$  when the population of individuals does exhibit resource selection. As such, it is important to account for space usage when important covariates are known to influence space usage patterns.

In our formulation of the joint likelihood for RSF and SCR data, we assumed the data from a capture-recapture and telemetry studies were independent of one another. This implies that whether or not an individual enters into one of the data sets has no effect on whether it enters into the other data set. We cannot foresee situations in which violation of

10493 this assumption should be problematic or invalidate the estimator under the independence  
10494 assumption. In some cases it might so happen that some individuals appear in *both* the  
10495 RSF and SCR data sets. In this case, ignoring that information should entail only an  
10496 incremental decrease in precision because a slight bit of information about an individual's  
10497 activity center is disregarded. Heuristically, an SCR observation (encounter in a trap)  
10498 is like one additional telemetry observation, and so the misspecification (independence)  
10499 regards the two pieces of information as having separate activity centers. Our model  
10500 pretends that we don't know anything about the telemetered individuals in terms of  
10501 their encounter history in traps. In principle it shouldn't be difficult to admit a formal  
10502 reconciliation of individuals between the two lists. In that case, we just combine the two  
10503 conditional likelihoods before we integrate  $\mathbf{s}$  from the conditional likelihood. This would  
10504 be almost trivial to do if *all* individuals were reconcilable (or none as in the case we have  
10505 covered here) but, in general, we think you will always have an intermediate case – i.e.,  
10506 either none will be or at most a subset of telemetered guys will be known. More likely  
10507 you have variations of “well, that guy looks telemetered but we don't know which guy it  
10508 is...hmmm” and that case, basically a type of marking uncertainty or misclassification, is  
10509 clearly more difficult to deal with.

10510 In our formulation of the combined likelihood for RSF and SCR data, we assumed  
10511 the data from capture-recapture and telemetry studies were independent of one another.  
10512 This implies that whether or not an individual enters into one of the data sets has no  
10513 effect on whether it enters into the other data set. We cannot foresee situations in which  
10514 violation of this assumption should be problematic or invalidate the estimator under the  
10515 independence assumption. In some cases it might so happen that some individuals appear  
10516 in *both* the RSF and SCR data sets. In this case, ignoring that information should entail  
10517 only an incremental decrease in precision because a slight bit of information about an  
10518 individual's activity center is disregarded.

10519 Discussion point: Note that we could relax the uniformity assumption by specifying  
10520 an inhomogeneous point process model (Borchers and Efford, 2008) as shown in Chapt.  
10521 XXXXX. This allows for modeling second-order habitat selection as defined by Johnson  
10522 (1980). Thus, SCR models provide insight into the hierarchical nature of habitat selection.  
10523 Simultaneously we model all types of habitat selection in a single unified model based on  
10524 capture-recapture data.

10525 Bayesian analysis might have an advantage in situations where the landscape is charac-  
10526 terized by a very fine covariate raster, or even continuous covariates, because the individual  
10527 activity centers can be updated in the MCMC algorithm by evaluating the likelihood con-  
10528 ditional on a single candidate value of  $\mathbf{s}$  for each individual. Conversely, evaluation of  
10529 the marginal likelihood becomes tedious and memory intensive as the size of the raster  
10530 increases, and so some effort has to be made to efficiently calculate the likelihood in such  
10531 cases (e.g., see Warton and Shepherd, 2010). Independent of its effect on integration,  
10532 raster size is itself an important practical concern. Whenever we have explicit spatial  
10533 covariates, it is possible that selection is occurring at a much finer resolution than is re-  
10534 quired to effectively integrate the likelihood over the state-space of  $\mathbf{s}$ . In this case, too  
10535 coarse of a raster will likely cause biased parameter estimates (having an effect analogous  
10536 to measurement error in regression, we suspect). Too fine, however, creates concomitant  
10537 effects on computing and memory requirements. Choice of raster size or spatial resolution  
10538 is thus both a fundamental scientific question, but also very much a practical computing

10539 issue.

10540 We developed the model in a discrete landscape which regarded potential trap locations  
10541 and the covariate  $z(\mathbf{x})$  as being defined on the same set of points. In practice, trap locations  
10542 may have been chosen independent of the definition of the raster and this does not pose  
10543 any challenge or novelty to the model as we developed it. In that case, the covariate(s)  
10544 need to be defined at each trap location. The model should be applicable also to covariates  
10545 that are naturally continuous (e.g., distance-based covariates) although, in practice, it will  
10546 usually be sufficient to work with a discrete representation of such covariates.

10547 We used an RSF model for telemetry data that is most suitable for independent obser-  
10548 vations of space usage. This would be reasonable if telemetry fixes are made reasonably  
10549 far apart in time, or if the telemetry data is thinned, as we did in our analysis of the black  
10550 bear data. However, use of the independence model for non-independent data is probably  
10551 only a minor problem for estimating density or other model parameters because we ex-  
10552 pect that the pixel use frequencies should remain unbiased in this case<sup>1</sup>. We imagine that  
10553 precision should be over-stated for the parameters of the RSF model because the sample  
10554 size is not reflecting the dependence of the observations. In general, however, it will be  
10555 desirable to incorporate more general (or explicit) models of movement into the framework  
10556 proposed here, so that SCR models can be used to improve inferences about animal move-  
10557 ment, and because more explicit models may improve inferences about density obtained by  
10558 capture-recapture studies. As we noted, our specific model of independence corresponds  
10559 to a limiting case of the Gaussian process movement model (Johnson et al., 2008), but  
10560 including the general RSF movement model for correlated data from Johnson et al. (2008)  
10561 should not pose any difficulty in terms of constructing the combined SCR+RSF likelihood  
10562 (but contain one additional parameter).

---

<sup>1</sup>As a technical matter, we think regular movement models should exhibit an ergodic property analogous to standard MCMC algorithms, time-series models and related dynamical systems.



10563  
10564

# 12

10565

## MODELING LANDSCAPE CONNECTIVITY

10566     XXXXx Kimmy: As you go through this chapter, we need to know which R  
10567     packages are used in the chapter (and other chapters too) XXXXX

10568     Every spatial capture-recapture model that we have considered so far has expressed  
10569     encounter probability as function of the Euclidean distance between individual activity  
10570     centers  $s$  and trap locations  $x$ . As a practical matter, models based on Euclidean distance  
10571     imply circular, symmetric, and stationary home ranges of individuals, and these are not  
10572     often biologically realistic. While these simple encounter probability models will often be  
10573     sufficient for practical purposes, especially in small data sets, sometimes developing more  
10574     complex models of the detection process as it relates to space usage of individuals will  
10575     be useful. Animals may not judge distance in terms of Euclidean distance but, rather,  
10576     according to quality of local habitat, landscape connectivity, perceived mortality risk, and  
10577     other considerations affecting movement behavior. Moreover, because encounter proba-  
10578     bility and the distance metric upon which it is based represent outcomes of individual  
10579     movements about their home range, ecologists might have explicit hypotheses about how  
10580     environmental variables affect the distance metric, and it is therefore desirable to incor-  
10581     porate these hypotheses directly into SCR models so that they may be formally evaluated  
10582     statistically.

10583     Assessing the impacts of habitat fragmentation and habitat loss on population density  
10584     and landscape connectivity are high priorities in applied ecological research. Landscape  
10585     connectivity is defined as the degree to which landscape structure impedes or facilitates  
10586     movement (Tischendorf and Fahrig, 2000) and is widely recognized to be an important  
10587     component of population viability (With and Crist, 1995). Although much theory has been  
10588     developed to predict the effects of decreasing connectivity, few empirical studies have been  
10589     conducted to test these predictions due to the paucity of formal methods for estimating  
10590     connectivity parameters (Cushman et al., 2010). Instead, ecologists often rely on expert  
10591     opinion or *ad hoc* methods of specifying connectivity values, even in important applied  
10592     settings (Adriaensen et al., 2003; Beier et al., 2008; Zeller et al., 2012). In addition, no  
10593     methods are available for simultaneously estimating population density and connectivity  
10594     parameters, in spite of theory predicting interacting effects of density and connectivity

on population viability (Tischendorf et al., 2005; Cushman et al., 2010). In this chapter, following Royle et al. (2012c), we provide a framework for modeling landscape connectivity using SCR models, by parameterizing models for encounter probability based on “ecological distance”. A 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. Together multiple environmental variables create a resistance surface, which forms the linchpin of all connectivity planning (Spear et al., 2010).

Recently Royle et al. (2012c) provided an SCR framework based on least-cost path for modeling landscape connectivity. They parameterized encounter probability *not* based on Euclidean distance but, rather, based 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. For these models based on least-cost path, it is convenient to use a likelihood-based inference framework which we follow here in this chapter. 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, which is commonly done in practice. 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. Royle et al. (2012c) showed that, under simple models of landscape connectivity (governed by a single covariate), a misspecified model based on Euclidean distance can produce substantial bias in estimates of  $N$  and hence density.

## 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

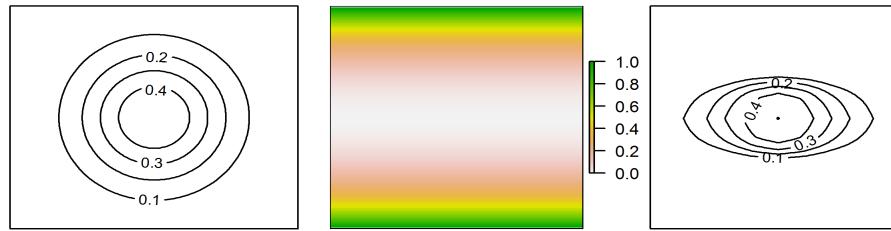
$$p_{ij} = p_0 \exp(-d_{ij}^2/(2\sigma^2))$$

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 here we mean in the formal sense of invariance to translation. That is, the properties of an individual home range centered at some point

10637  $\mathbf{s}$  are exactly the same as any other point say  $\mathbf{s}'$ . As an example, if the common detection  
 10638 model based on a bivariate normal probability distribution function is used, then the  
 10639 implied space usage by *all* individuals, no matter their location in space or local habitat  
 10640 conditions, is symmetric with circular contours of usage intensity.

10641 In the framework of Royle et al. (2012c), SCR models explicitly incorporate information  
 10642 about the landscape so that a unit of distance is variable depending on identified  
 10643 covariates, say  $z(\mathbf{x})$ . Thus, where an individual lives on the landscape, and the state of the  
 10644 surrounding landscape, will determine the character of its usage of space. In particular,  
 10645 they suggest distance metrics, based on least-cost path, that imply irregular, asymmetric  
 10646 and non-stationary home ranges of individuals. As an example, Fig. 12.1 shows a  
 10647 typical symmetric home range (left panel), and a compressed home range (right panel)  
 10648 resulting from the effect of an environmental variable (center panel) on an animal's movement  
 10649 behavior. We might think of the environmental variable as representing an elevation  
 10650 gradient of a valley and so, for a species that avoids high elevation, space usage will be  
 10651 concentrated in flatter terrain at lower elevations and therefore producing the elliptical  
 10652 home range shape. We reproduce the application from Royle et al. (2012c) later in this  
 10653 chapter, in addition to providing an alternative applied context that involves computing  
 10654 distances within odd-shaped landscape patches (sec. 12.7).



**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

10655 We adopt a cost-weighted distance metric here which defines the effective distance between  
 10656 points by accumulating pixel-specific costs determined using a cost function defined by the  
 10657 user. The idea of cost-weighted distance to characterize animal use of landscapes is widely  
 10658 used in landscape ecology for modeling connectivity, movement and gene flow (Beier et al.,  
 10659 2008). For reasons of computational tractability we consider a discrete landscape defined  
 10660 by a raster of some prescribed resolution. The distance between any two points  $\mathbf{x}$  and  
 10661  $\mathbf{x}'$  can be represented by a sequence of line segments connecting neighboring pixels, say  
 10662  $\mathbf{l}_1, \mathbf{l}_2, \dots, \mathbf{l}_m$ . Then the cost-weighted distance between  $\mathbf{x}$  and  $\mathbf{x}'$  is

$$d(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{m-1} cost(\mathbf{l}_i, \mathbf{l}_{i+1}) \|\mathbf{l}_i - \mathbf{l}_{i+1}\| \quad (12.2.1)$$

10663 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  
 10664  $\mathbf{l}_{i+1}$  in the sequence. Given the cost of each pixel, it is a simple matter to compute the  
 10665 cost-weighted distance between any two pixels, along *any* path, simply by accumulating  
 10666 the incremental costs weighted by distances. In the context of spatial capture-recapture  
 10667 models (and, more generally, landscape connectivity) we are concerned with the *minimum*  
 10668 cost-weighted distance, or the *least-cost path*, between any two points which we will denote  
 10669 by  $d_{lcp}$ , which is the sequence  $\mathcal{P} = (\mathbf{l}_1, \mathbf{l}_2, \dots, \mathbf{l}_m)$  that minimizes the objective function  
 10670 defined by Eq. 12.2.1. That is,

$$d_{lcp}(\mathbf{x}, \mathbf{x}') = \min_{\mathcal{P}} \sum_{i=1}^{m-1} \text{cost}(\mathbf{l}_i, \mathbf{l}_{i+1}) \|\mathbf{l}_i - \mathbf{l}_{i+1}\| \quad (12.2.2)$$

10671 The least-cost path distance can be calculated in many geographic information systems  
 10672 and other software packages, including the R package **gdistance** (van Etten, 2011) which  
 10673 we use below.

10674 The key ecological aspect of least-cost path modeling is the development of models for  
 10675 pixel-specific cost. In this paper we model cost as a function of one or more covariates  
 10676 defined on every pixel of the according raster. For example, using a single covariate  $z(\mathbf{x})$   
 10677 we define the cost of moving from some pixel  $\mathbf{x}$  to neighboring pixel  $\mathbf{x}'$  as

$$\log(\text{cost}(\mathbf{x}, \mathbf{x}')) = \alpha_2 \left( \frac{z(\mathbf{x}) + z(\mathbf{x}')}{2} \right) \quad (12.2.3)$$

10678 Thus, if  $\alpha_2 = 0$  then substituting  $\text{cost}(\mathbf{x}, \mathbf{x}') = \exp(0) = 1$  into Eq. 12.2.2 will pro-  
 10679 duce the ordinary Euclidean distance between points. Here we assume the covariate  $z$  is  
 10680 positive-valued and constrain  $\alpha_2 \geq 0$  so as to avoid negative costs. While not necessarily  
 10681 problematic from a mathematical standpoint, negative costs are unrealistic biologically.

10682 The use of least-cost path models to model landscape connectivity has been around  
 10683 for a long time. And, although  $\alpha_2$  is rarely known, conservation biologists design linkages  
 10684 that require this resistance value as input (see Beier et al., 2008, and articles cited therein).  
 10685 However, formal inference (e.g., estimation) of parameters is not often done. Instead, in  
 10686 many existing applications of least-cost path analysis, the parameter  $\alpha_2$  is fixed by the  
 10687 investigator, or based on expert opinion (Beier et al., 2008), although recently researchers  
 10688 have begun to define costs based on resource selection functions, animal movement (Tracy,  
 10689 2006; Fortin et al., 2005), or genetic distance data (e.g., Gerlach and Musolf (2000); Epps  
 10690 et al. (2007); Schwartz et al. (2009)). We address the integration of resource selection  
 10691 models based on telemetry data with SCR models in Chapt. 11.

10692 To formalize the use of cost-weighted distance in SCR models, we substitute Eq.  
 10693 12.2.2 in the expression for encounter probability (Eq. ??) and maximize the resulting  
 10694 likelihood which we address below. In doing so, we can directly estimate parameters of  
 10695 the least-cost path model, evaluate how landscape covariate influence connectivity, and  
 10696 test explicit hypotheses about these things using only individual level encounter history  
 10697 data from capture-recapture studies.

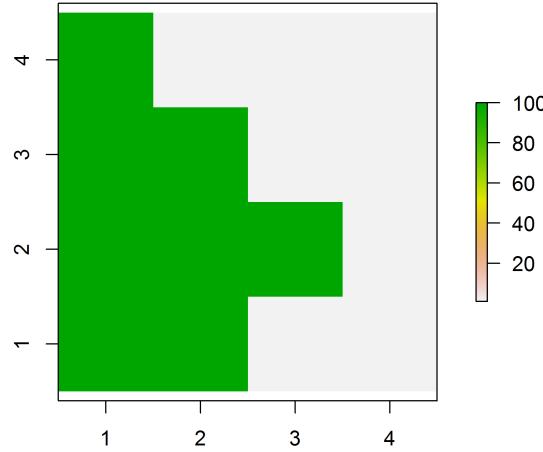
### 10698 12.2.1 Example of Computing Cost-weighted distance

10699 XXXX Kimmy: Reconcile the raster example here with what is in the Ap-  
 10700 pendix of the accepted Ecology paper. XXXXX

10701 As an example of the cost-weighted distance calculation consider the following land-  
 10702 scape comprised of 16 pixels with unit spacing identified as follows, along with the pixel-  
 10703 specific cost:

10704	pixel ID				Cost			
10705	1	5	9	13	100	1	1	1
10706	2	6	10	14	100	100	1	1
10707	3	7	11	15	100	100	100	1
10708	4	8	12	16	100	100	1	1

10709 This simple cost raster is shown in Fig. 12.2. We assume the scale is such that the  
 10710 distance between neighboring pixels in any cardinal direction is 1 unit, and the distance  
 10711 between neighbors on a diagonal is  $\sqrt{2}$  units. We assigned low cost of 1 to “good habitat”  
 10712 pixels (or pixels we think of as “highly connected” by virtue of being in good habitat)  
 10713 and, conversely, we assign high cost (100) to “bad habitat”. So the shortest cost-weighted  
 10714 distance between pixels 5 and 9 in this example is just 1 unit, the shortest cost-distance  
 10715 between pixels 5 and 10 is  $\sqrt{2}(1+1)/2 = 1.414214$  units, the shortest distance between  
 10716 pixels 4 and 8 is 100 units, while the shortest cost-distance between 4 and 12 is 150.5. A  
 10717 tough one is: what is the shortest distance between 7 and 16? An individual at pixel 7  
 10718 can move diagonal (which has distance  $\sqrt{2}$ ) and pay  $\text{sqrt}(2) * (100 + 1) / 2 + 1 = 72.41778$ .



**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.

10719 Once the cost raster is created, the least-cost path distances are computed with just a  
 10720 couple **R** commands, and those can be inserted directly into the likelihood construction for  
 10721 an ordinary spatial capture-recapture model. The **R** package **gdistance** calculates least-  
 10722 cost path using Dijkstra's algorithm (Dijkstra, 1959) (from the **igraph** package (Csardi  
 10723 and Nepusz, 2006)). Using **gdistance**, we define the incremental cost of moving from one  
 10724 pixel to another as the distance-weighted *average* of the 2 pixel costs. We demonstrate  
 10725 how to do this subsequently.

10726 **Kimmy: Make sure the structure of this example is copied directly from**  
**10727 the Ecology paper that was accepted. There are some things slightly different.**

10728 The **R** commands for computing the least-cost distance between all pairs of pixels are  
 10729 as follows:

```
10730 r<-raster(nrows=4,ncols=4)
10731 projection(r)<- "+proj=utm +zone=12 +datum=WGS84"
10732 extent(r)<-c(.5,4.5,.5,4.5)
10733 costs1<- c(100,100,100,100,1,100,100,100,1,1,100,1,1,1,1,1)
10734 values(r)<-matrix(costs1,4,4,byrow=FALSE)
10735 par(mfrow=c(1,1))
10736 plot(r)
```

10737 Then we use the functions **transition**, **geoCorrection XXX Kimmy do we need**  
**this function? XXXX** (which is only necessary if the data are not projected or if  
 10738 cells are considered to have more than 4 neighbors) and **costDistance** to compute the  
 10739 distance matrix. The transition function computes the cost of making a transition be-  
 10740 tween any two pixels, and it operates on the inverse-scale ("conductance") and so the  
 10741 **transitionFunction** argument is given as  $1/\text{mean}(x)$ . To compute the cost distance we  
 10742 prescribe a set of points, or we can compute it between two sets of points (which is handy  
 10743 when one of the sets is of trap locations, and the other is of individual activity centers).  
 10744 To compute the distances for pixels in a raster, we use the center points of each raster.  
 10745 The **R** commands altogether are as follows:

```
10747 tr1<-transition(r,transitionFunction=function(x) 1/mean(x),directions=8)
10748 tr1CorrC<-geoCorrection(tr1,type="c",multpl=FALSE,scl=FALSE)
10749 pts<-cbind( sort(rep(1:4,4)),rep(4:1,4))
10750 costs1<-costDistance(tr1CorrC,pts)
10751 outD<-as.matrix(costs1)
```

10752 Now we can look at the result and see if it makes sense to us. Here we produce the  
 10753 first 5 columns of this distance matrix to illustrate a couple of examples of calculating  
 10754 the minimum cost-weighted distance between points: **XXXX Kimmy: All of these**  
 10755 **numbers are probably different in the paper's appendix and we need to change**  
**them here XXXXX**

```
10757 > outD[1:5,1:5]
10758      1       2       3       4       5
10759 1  0.0000 100.00000 200.0000 205.2426 50.50000
10760 2 100.0000   0.00000 100.0000 200.0000 71.41778
10761 3 200.0000 100.00000   0.0000 100.0000 171.41778
```

10762 4 205.2426 200.00000 100.0000 0.0000 154.74264  
 10763 5 50.5000 71.41778 171.4178 154.7426 0.00000

10764 An interesting case is that between point 1 and 4. Note that simply taking the shortest  
 10765 Euclidean distance, weighted by cost, produces a cost-weighted distance of  $100 \times 1$  to  
 10766 move from pixel 1 to pixel 2, and similarly from 2 to 3 and 3 to 4, producing a total cost-  
 10767 weighted distance of 300. However, the actual *least-cost path* has cost-weighted distance  
 10768 205.2426 which has an individual moving from pixel 1 to 5, then 5 to 10, 10 to 15, 15 to  
 10769 12, 12 to 8 and 8 to 4, adding up to a cost weighted distance of 205.2426.

### 12.3 SIMULATING SCR DATA USING ECOLOGICAL DISTANCE

10770 Royle et al. (2012c) simulated data based on two hypothetical landscapes typical of how  
 10771 cost-weighted distance models might be used in real capture-recapture problems. They  
 10772 defined a  $20 \times 20$  pixel covariate raster with extent  $= [0.5, 4.5] \times [0.5, 4.5]$  which we imagine  
 10773 to be a coarse landscape covariate, with pixels having some arbitrary scaling. For example,  
 10774 think of each pixel as representing, say, a  $1 \times 1$  km grid cell in which case the raster defines  
 10775 a landscape of  $20 \times 20$  km. We suppose that 16 camera traps are established at the integer  
 10776 coordinates  $(1, 1), (1, 2), \dots, (4, 4)$ . We could think of this as a landscape within which  
 10777 we're studying a population of ocelots, lynx or some other cat.

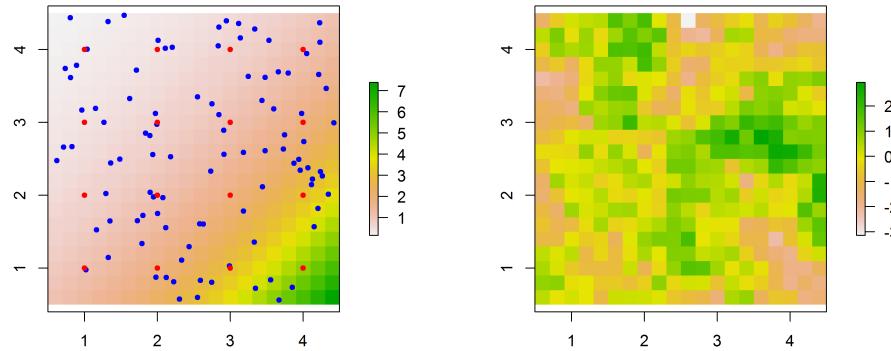
10778 For our analyses here, we characterize cost by one covariate, and we consider two  
 10779 specific cases. First is an increasing trend from the NW to the SE ("systematic covariate"),  
 10780 where  $z(\mathbf{x})$  is defined as  $z(\mathbf{x}) = r(\mathbf{x}) + c(\mathbf{x})$  and  $r(\mathbf{x})$  and  $c(\mathbf{x})$  are just the row and column,  
 10781 respectively, of the raster. This might mimic something related to distance from an urban  
 10782 area or a gradient in habitat quality due to land use, or environmental conditions such  
 10783 as temperature or precipitation gradients. In the second case we make up a covariate by  
 10784 generating a field of spatially correlated noise to emulate a typical patchy habitat covariate  
 10785 ("patchy covariate") such as tree or understory density. The two covariates are shown in  
 10786 Fig. 12.3, along with a sample realization of  $N = 100$  individuals (left panel only). For  
 10787 both covariates we use a cost function in which transitions from pixel  $\mathbf{x}$  to  $\mathbf{x}'$  is given by:

$$\log(\text{cost}(\mathbf{x}, \mathbf{x}')) = \alpha_2 \frac{z(\mathbf{x}) + z(\mathbf{x}')}{2}$$

10788 where  $\alpha_2 = 1$  for simulating the observed data. Remember that with  $\alpha_2 = 0$  the model  
 10789 reduces to one in which the cost of moving across each pixel is constant, and therefore  
 10790 Euclidean distance is operative.

10791 When distance is defined by the cost-weighted distance metric given by Eq. 12.2.2  
 10792 then individual space-usage varies spatially in response to the landscape covariate(s) used  
 10793 in the distance metric. As a consequence, home ranges contours are no longer circular, as  
 10794 in SCR models based on Euclidean distance. For example, using one of the covariates we  
 10795 use in our simulation study below (Fig. 12.3, right panel) with a Gaussian pdf detection  
 10796 function but having distance metric defined by Eq. 12.2.2, produces home ranges such as  
 10797 those shown in Fig. 12.4.

10798 To simulate data, we have to load the `scrbook` package and call the function `make.EDcovariates`  
 10799 to generate our raster covariates (see the help file for how that is done). We process the co-  
 10800 variate into a least-cost path distance matrix, and then simulate observed encounter data  
 10801 using standard methods which we have used many times previously in this book. The



**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.

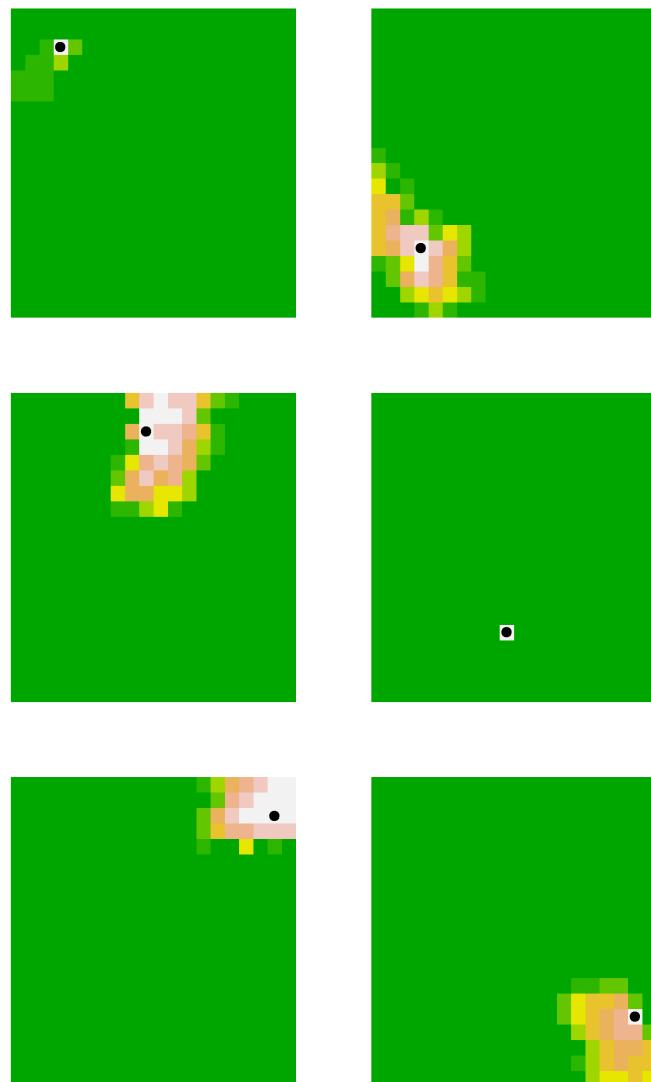
10802 complete set of R commands is: XXXX Kimmy: Test this code! XXXXX XXX  
 10803 Kimmy: Do we need the geocorrection() statement in this code? XXXXX

```

10804 #### Grab a covariate
10805 library("scrbook")
10806 out<-make.EDcovariates()
10807 covariate<-out$covariate.patchy
10808 set.seed(2013)

10809 #### prescribe some settings
10810 N<-200
10811 alpha0<- -2
10812 sigma<- .5
10813 alpha1<- 1/(2*sigma*sigma)
10814 alpha2<-1
10815 K<- 5
10816 S<-cbind(runif(N,.5,4.5),runif(N,.5,4.5))

10817 # make up some trap locations
10818 xg<-seq(1,4,1); yg<-4:1
10819 traplocs<-cbind( sort(rep(xg,4)),rep(yg,4))
10820 points(traplocs,pch=20,col="red")
10821 ntraps<-nrow(traplocs)
10822
  
```



**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.

```

10824
10825 ### make a raster and fill it up with the "cost"
10826 r<-raster(nrows=20,ncols=20)
10827 projection(r)<- "+proj=utm +zone=12 +datum=WGS84"
10828 extent(r)<-c(.5,4.5,.5,4.5)
10829 cost<- exp(alpha2*covariate)
10830
10831 ### compute least-cost path distance
10832 tr1<-transition(cost,transitionFunction=function(x) 1/mean(x),directions=8)
10833 tr1CorrC<-geoCorrection(tr1,type="c",multpl=FALSE,scl=FALSE)
10834 D<-costDistance(tr1CorrC,S,traplocs)
10835 probcap<-plogis(alpha0)*exp(-alpha1*D*D)
10836
10837 # now generate the encounters of every individual in every trap
10838 # discard uncaptured individuals
10839 Y<-matrix(NA,nrow=N,ncol=ntraps)
10840 for(i in 1:nrow(Y)){
10841   Y[i,]<-rbinom(ntraps,K,probcap[i,])
10842 }
10843 Y<-Y[apply(Y,1,sum)>0,]

```

## 12.4 LIKELIHOOD ANALYSIS OF ECOLOGICAL DISTANCE MODELS

10844 Throughout much of this book we rely on Bayesian analysis by MCMC mostly using  
 10845 **BUGS**, but sometimes (as in Chapt. 14) developing our own implementations. However,  
 10846 occasionally we prefer to use likelihood estimation, such as when we can compare a set  
 10847 of models directly by likelihood either to do a direct hypothesis test of a parameter, or  
 10848 to tabulate a bunch of AIC values. For the class of models that use least-cost path, we  
 10849 also prefer likelihood methods not because they have any conceptual or methodological  
 10850 benefit, but simply because they are more computationally efficient to implement (Royle  
 10851 et al., 2012c).

10852 There are no technical considerations in adapting our formulation of maximum likeli-  
 10853 hood estimation (Borchers and Efford, 2008) from Chapt. 6 for the class of models based  
 10854 on least-cost path (see the appendix in Royle et al. (2012c) for complete details). Likeli-  
 10855 hood analysis is really just a straightforward adaptation in which we replace the Euclidean  
 10856 distance with least-cost path. Consider the Bernoulli model in which the individual- and  
 10857 trap-specific observations have a binomial distribution conditional on the latent variable  
 10858  $\mathbf{s}_i$ :

$$y_{ij} | \mathbf{s}_i \sim \text{Binomial}(K, p_{\boldsymbol{\alpha}}(d_{lcp}(\mathbf{x}_j, \mathbf{s}_i; \boldsymbol{\alpha}_2); \boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1)) \quad (12.4.1)$$

10859 where we have indicated the dependence of  $p$  on the parameters  $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2)$ , and  
 10860 also  $d_{lcp}$  which itself depends on  $\boldsymbol{\alpha}_2$ , and the latent variable  $\mathbf{s}$ . We note that the only  
 10861 difference between likelihood analysis of this model and the standard Bernoulli model, is  
 10862 the use of  $d_{lcp}$  here.

10863 XXXXX Andy here XXXXXXXXXXXXXXXXXXXXXXX

10864 For the random effect we have  $\mathbf{s}_i \sim \text{Uniform}(\mathcal{S})$ , we can easily compute the integrated  
 10865 likelihood.

10866 We have an R script in the `scrbook` package? adapted from Royle et al. ....  
 10867 XXXXXXXX

10868 We provide an **R** function to evaluate the likelihood, and optimize it using the **R**  
 10869 function `nls`. The likelihood is given in the `scrbook` package as the function `intlik3ed`.  
 10870 The help file provides an example of its usage and for simulating data. To use this  
 10871 function the cost covariate  $z(\mathbf{x})$  has to be of class `RasterLayer` which requires packages  
 10872 `sp` and `raster` to manipulate.

#### 10873 12.4.1 Example of SCR with Least-Cost Path

10874 Now we use the **R** function `nls` along with our `intlik3ed` function to obtain the MLEs  
 10875 of the model parameters for the data simulated in sec. 12.3. We'll do that for both the  
 10876 standard Euclidean distance and then for the ecological distance based on the "patchy"  
 10877 covariate using the following commands:

```
10878 frog1<-nls(intlik3ed,c(alpha0,alpha1,3)),hessian=TRUE,y=Y,K=K,X=traplocs,  

  10879           distmet="euclid",covariate=covariate,alpha2=1)  

  10880  

  10881 frog2<-nls(intlik3ed,c(alpha0,alpha1,3,-.3),hessian=TRUE,y=Y,K=K,X=traplocs,  

  10882           distmet="ecol",covariate=covariate,alpha2=NA)
```

10883 The abbreviated output for the two model fits is shown in Table XXX. XXXXXX  
 10884 **KIMMY: Put this in a table like the one later in this chapter XXXXXX**

Distance	-LL	alpha0	alpha1	log(n0)	alpha2
true value	-2	2	[??]	1	
euclidean	133.4951	-1.885005	1.247305	3.549064	--
LCP (truth)	70.11916	-1.78029983	2.47083431	4.45867628	0.04560194

10885 The model based on least-cost path (the data generating model) appears to be much  
 10886 preferred in terms of negative log-likelihood. The output parameter order is  $(\alpha_0, \alpha_1, \log(n_0), \text{and} \log(\alpha_2))$   
 10887 (remember, we want to keep  $\alpha_2$  positive, so its logarithm is estimated). The data gener-  
 10888 ating parameter values were  $\alpha_0 = -2$ ,  $\alpha_1 = 2$  and  $\log(\alpha_2) = 0$ . The simulated sampling  
 10889 produced a sample of 96 individuals and so  $n_0 = 104$ , so  $\log(n_0) = 4.64$ . We see that the  
 10890 MLEs of the least-cost path model are pretty close whereas they are not so close under  
 10891 the misspecified model based on Euclidean distance.

## 12.5 BAYESIAN ANALYSIS

10892 While implementation of these ecological distance SCR models is reasonably straightfor-  
 10893 ward, we do not believe the model can be fitted in the **BUGS** engines because least-cost  
 10894 path distance cannot be computed. It would be possible to fit the models in **BUGS** if  
 10895 the parameter  $\alpha_2$  was fixed. In that case, one could compute the distance matrix ahead of  
 10896 time and reference the required elements for a given `s`. Alternatively, it would be possible  
 10897 to write a custom MCMC routine using the methods we present in Chapt. 14, although  
 10898 we have not yet developed our own MCMC implementation of SCR models with ecological  
 10899 distance metrics.

## 12.6 SIMULATION EVALUATION OF THE MLE

10904 Royle et al. (2012c) carried-out a limited simulation study to evaluate the general statisti-  
 10905 cal performance of the density estimator under this new model, the effect of mis-specifying  
 10906 the model with a normal Euclidean distance metric and evaluate the general bias and pre-  
 10907 cision properties of the MLE. We recapitulate their results here. For population sizes of  
 10908 100 and 200, individuals with activity centers randomly distributed on the  $20 \times 20$  land-  
 10909 scape, they subjected individuals to encounter by 16 traps arranged in a  $4 \times 4$  grid using  
 10910 a Gaussian encounter model with least-cost path distance metric:

$$\log(p_{ij}) = \alpha_0 + \alpha_1 d_{lcp}(\mathbf{x}_j, \mathbf{s}_i; \alpha_2)^2$$

10911 where  $\alpha_0 = -2$  and  $\alpha_1 = 2$ , the latter value corresponding to  $\sigma = 0.5$  of a stationary  
 10912 bivariate normal home range model. Different numbers of replicate samples were consid-  
 10913 ered,  $K = 3, 5, 10$  (e.g., nights in a camera trapping study), in order to produce varying  
 10914 sample sizes. For each of the “systematic” and “patchy” landscapes defined previously,  
 10915 200 data sets were simulated and, for each of those, two different models were fitted:  
 10916 the misspecified Euclidean distance model; and (ii) the true data-generating model but  
 10917 estimating the relative cost parameter by maximum likelihood.

### 10918 12.6.1 Simulation Results

10919 For both landscapes and all simulation conditions (levels of  $K$  and  $N$ ) the average sample  
 10920 sizes of individuals captured are given in Tab. 12.1. The simulation results for estimating

**Table 12.1.** Expected sample sizes of captured individuals under each configuration of  $N$  (population size for the prescribed state-space) and  $K$  (number of replicate samples).

	Systematic		Patchy	
	$N=100$	$N=200$	$N=100$	$N=200$
$K=3$	38.69	78.17	37.30	74.93
$K=5$	51.10	103.18	51.89	103.71
$K=10$	65.81	132.39	69.44	138.76

10921  $N$  for the prescribed state-space are presented in Tab. 12.6.1. For the “patchy” landscape  
 10922 we see extreme bias in estimates of  $N$  when the Euclidean distance is used. There is  
 10923 moderate small sample bias of 3-5% in the MLE of  $N$  using the least-cost distance which  
 10924 becomes negligible as  $K$  increases. For  $N = 200$  the bias is on the order of 2% for the  
 10925 lowest sample size case ( $K = 3$ ) but negligible otherwise. Interestingly, for the landscape  
 10926 exhibiting systematic structure, there is a persistent bias in the MLE of  $N$  of 1-3% even  
 10927 for the highest level of  $K$ . As noted by Royle et al. (2012c), this is due to the fact that the  
 10928 state-space is small relative to the extent of the trapping grid and sensitivity to a state-  
 10929 space that is too small is expected because the support of the integrand is truncated.  
 10930 In the particular case of the systematic landscape, we find that, in the NW corner of  
 10931 the raster where cost of movement is low, individuals use large areas of space, and the  
 10932 fitted model is under-stating the apparent heterogeneity in encounter probability for the  
 10933 prescribed raster. Royle et al. (2012c) found that the issue is resolved when the traps are  
 10934 moved away from the boundary (results shown in Tab. 12.6.1).

The performance of estimating the cost parameter  $\alpha_2$  mirrors the results for estimating  $N$  for the prescribed state space. In the patchy landscape where we don't expect a systematic gradient in space usage around the edge of the state-space, we see (Table 12.3) that  $\alpha_2$  is estimated with diminishing bias as the sample size increases, but with persistent bias due to truncation of the likelihood under the systematic landscape which, as with the MLE of  $N$ , is resolved by moving the traps away from the edge of the raster. Equivalently, in practice, this could be resolved by expanding the raster away from the trap locations so that all regions used by animals exposed to capture are included in the state-space.

**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 ( $K$ ) chosen to affect the observed sample size of individuals (Tab. 12.1). 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).

Systematic trend raster:										
	N=100					N=200				
	mean	SD	0.025	0.50	0.975	mean	SD	0.025	0.50	0.975
K=3										
euclid	63.65	12.62	44.77	61.17	90.98	126.68	17.05	98.93	124.49	168.26
lcp	101.93	21.68	67.95	101.56	156.21	201.58	28.14	154.96	200.15	263.20
K=5										
euclid	64.60	7.11	51.52	63.86	77.33	130.02	10.25	113.48	128.96	151.32
lcp	98.94	12.97	74.68	99.00	123.88	198.80	19.60	166.87	197.97	239.46
K=10										
euclid	69.24	4.83	59.37	69.47	79.18	139.83	7.62	125.65	139.65	154.82
lcp	97.53	8.18	82.02	97.62	113.16	195.19	13.28	171.63	194.58	217.96
Patchy "random" raster:										
	N=100					N=200				
	mean	SD	0.025	0.50	0.975	mean	SD	0.025	0.50	0.975
K=3										
euclid	78.68	18.12	49.40	76.34	125.47	154.34	33.74	107.00	146.34	221.43
lcp	110.96	28.65	69.55	106.98	181.84	208.77	49.29	141.68	197.89	325.77
K=5										
euclid	77.85	11.55	59.17	77.44	101.14	153.39	15.57	129.31	149.54	185.38
lcp	104.44	15.79	78.38	101.47	139.55	200.91	20.78	164.42	200.47	246.46
K=10										
euclid	78.01	5.26	68.00	77.96	87.81	156.27	8.51	142.17	156.05	174.55
lcp	100.42	7.56	86.72	100.34	115.47	198.45	11.44	180.06	198.04	219.52

**Table 12.3.** Mean of sampling distribution of the cost function parameter  $\alpha_2$  for the different simulation conditions.

	Patchy		Systematic	
	N=100	N=200	N=100	N=200
$K = 3$	1.05	1.03	1.17	1.14
$K = 5$	1.02	1.01	1.12	1.12
$K = 10$	1.01	1.00	1.10	1.08

**Table 12.4.** Simulation results for estimating population size  $N$  for a prescribed state-space with  $N = 100$  or  $N = 200$  and various levels of replication ( $K$ ) chosen to affect the observed sample size of individuals. These results correspond to those of the systematic landscape in Table XXXXXX except with the traps moved 0.5 units in from the boundary of the landscape. Each grouping of 2 rows (for a given value of  $K$ ) summarizes the performance of  $\hat{N}$  under models based on Euclidean distance ("euclid") and a model based on least-cost path, 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					N=200				
	mean	SD	0.025	0.50	0.975	mean	SD	0.025	0.50	0.975
K=3										
euclid	84.48	20.42	51.16	81.51	140.62	163.70	24.55	126.64	157.67	223.63
lcp	105.90	26.19	65.95	103.40	182.30	201.34	29.54	161.88	192.36	268.98
K=5										
euclid	81.21	11.33	61.35	79.20	98.86	163.27	13.06	140.21	162.97	185.94
lcp	100.84	13.15	79.96	99.51	119.08	200.25	16.53	168.88	199.29	227.39
K=10										
euclid	80.10	7.81	66.45	79.14	93.33	158.40	9.25	142.74	157.86	173.18
lcp	100.10	9.88	82.31	100.91	116.27	197.52	13.03	169.49	200.68	217.82

## 12.7 DISTANCE IN AN IRREGULAR PATCH

10943 We provide another illustration of how to employ ecological distance calculations in SCR  
 10944 models. This example is meant to mimic a situation where we have something like a hard  
 10945 habitat boundary such as a habitat corridor or park unit or some other block of relatively  
 10946 homogeneous good-quality habitat for some species. This particular system (shown in  
 10947 Fig. 12.5) could be habitat surrounded by a suburban wasteland of McDonalds and Wal-  
 10948 Marts, much less hospitable habitat for most species. For our purposes, we suppose that  
 10949 individuals live within the buffered “f-shaped” region, although we could also imagine the  
 10950 negative of the situation in which individuals live outside of the region, so that the polygon  
 10951 represents a barrier (a lake) or bad habitat (an urban area) or similar. We describe the  
 10952 steps for creating this landscape shortly, so that you can use a similar process to generate  
 10953 more relevant landscapes for your own problems.

10954 In this case we’re not going to estimate any parameters of the cost function (though  
 10955 you could adapt the analyses of the previous sections to do that) but instead we’re going  
 10956 to use ecological distance ideas only to constrain movement within (or to avoid) landscape  
 10957 features. Note that, normally, distance “as the crow flies” would not be suitable for  
 10958 irregular habitat patches such as that shown in Fig. 12.5.

### 10959 12.7.1 Basic Geographic Analysis in R

10960 In practical applications our landscape will contain one or more polygons which de-  
 10961 lineate good or bad habitat or other important characteristics of the landscape. These  
 10962 might exist as GIS shapefiles or merely as a text file with coordinates defining polygon  
 10963 boundaries. To work with polygons in the context of SCR models we need to create a  
 10964 raster, overlay the polygon and assign values to each pixel depending on whether pixels  
 10965 are in the polygon or not, or how far they are from polygon boundaries. These opera-  
 10966 tions are relatively easy to do within a GIS system but we need to be able to do them  
 10967 in **R** in order to compute the least-cost paths needed in the likelihood evaluation. Some  
 10968 additional geographic analyses have been discussed in secs. ?? and 14.5 where we talked  
 10969 about reading in the shapefile and doing SCR calculations on that.

10970 Often we will have GIS shapefiles that define polygons but, here, we create a set of  
 10971 polygons by buffering and joining some line segments. In the **R** library **scrbook**, we  
 10972 provide a function **make.seg** which allows you to make such line segments given a specific  
 10973 trap region. To involve **make.seg** we first create a plot region and then call **make.seg**  
 10974 which has a single argument being the number of points used to define the line segment.  
 10975 The user will click on the visual display until the required number of points has been  
 10976 obtained by **make.sec**. In the following set of commands we generate two line segments,  
 10977 11 consisting of 9 points and 12 consisting of 5 points, and these reside in a geographic  
 10978 region enclosed by  $[0, 10] \times [0, 10]$ :

```
10979 library("scrbook")
10980 library("sp")
10981 plot(NULL,xlim=c(0,10),ylim=c(0,10))
10982 l1<-make.seg(9)
10983 plot(l1)
10984 l2<-make.seg(5)
```

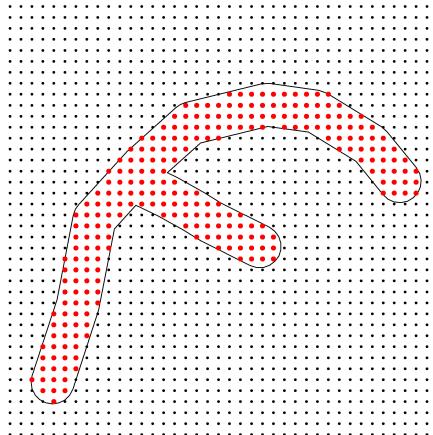
```
10985 plot(l1)
10986 lines(l2)
```

10987 We used this function as above to create a habitat corridor composed of line segments  
 10988 of class **SpatialLines** from the **R** package **sp**. The corridor can be loaded from **scrbook**  
 10989 by typing the command `data("fakecorridor")`. This data list has 2 line files in it (l1  
 10990 and l2) and a trap locations file (traps). We use some functions from the **R** packages  
 10991 **sp** and **rgeos** to join and buffer (by 0.5 units) the two segments. The commands are as  
 10992 follows and the result is shown in Fig. 12.5.

10993 **XXXX Kimmy: check all of this code! XXXXX**

```
10994 data("fakecorridor")
10995 library("sp")
10996 library("rgeos")
10997
10998 buffer<- 0.5
10999 par(mfrow=c(1,1))
11000 aa<-gUnion(l1,l2)
11001 plot(gBuffer(aa,width=buffer),xlim=c(0,10),ylim=c(0,10))
11002 pg<-gBuffer(aa,width=buffer)
11003 pg.coords<- pg@polygons[[1]]@Polygons[[1]]@coords
11004
11005 xg<-seq(0,10,,40)
11006 yg<-seq(10,0,,40)
11007
11008 delta<-mean(diff(xg))
11009 pts<- cbind(sort(rep(xg,40)),rep(yg,40))
11010 points(pts,pch=20,cex=.5)
11011
11012 in_pts<-point.in.polygon(pts[,1],pts[,2],pg.coords[,1],pg.coords[,2])
11013 points(pts[in_pts==1,],pch=20,col="red")
```

11014 In this example, we're not going to estimate parameters of the cost function. Instead, d  
 11015 the point is compute ordinary Euclidean distance but restricted by the boundaries of the  
 11016 corridor (or patch geometry in general) and thus not distance "as the crow flies." To do  
 11017 this, we imagine that animals will tend to severely avoid leaving the buffered habitat zone.  
 11018 Therefore, we assign `cost = 1` if a pixel is within the buffer, and `cost = 10000` if a pixel  
 11019 is outside of a buffer. Therefore the cost to move to a neighboring pixel outside of the  
 11020 buffered area is 5000.5 compared to the cost of 1 to move to a neighboring pixel inside the  
 11021 buffer. With this cost specification, we can compute the least-cost path distance matrix  
 11022 one time and modify our likelihood code to accept the distance matrix as input. We give  
 11023 that likelihood in the library **scrbook** as the function `intlik3edv2`. We note also that this  
 11024 function accepts a habitat mask in the form of a vector of 0's and 1's **XXXX Kimmy:**  
 11025 **check that this is true XXXXX** that define any potential state-space restrictions.  
 11026 i.e., 1 if the pixel is an element of the state-space and 0 if it is not, and so additional  
 11027 modifications to the geometry of the region could be made. However, in the analysis of  
 11028 this simulated data set, we define the state-space to be the buffered corridor system. Here



**Figure 12.5.** A made-up wildlife corridor or reserve. The boundary outlines a polygon of suitable habitat surrounded by suburban development.

11029 we simulate a population of  $N = 200$  individuals in the corridor system and so we restrict  
 11030 our state-space accordingly for purposes of fitting the model. However we encourage you  
 11031 to refit the model without the state-space restriction (for fitting the model only) and then  
 11032 compare the results. The code for doing all of this is in the help file for `intlik3edv2`,  
 11033 which contains the likelihood function and sample **R** script (`?intlik3edv2`).

```

11034 ### Define the cost structure
11035 cost<-rep(NA,nrow(pts))
11036 cost[in pts==1]<-1      # low cost to move among pixels but not 0
11037 cost[in pts!=1]<-10000  # high cost
11038
11039 ### Stuff costs into a raster
11040 library("raster")
11041 r<-raster(nrows=40,ncols=40)
11042 projection(r)<- "+proj=utm +zone=12 +datum=WGS84"
11043 extent(r)<-c(0-delta/2,10+delta/2,0-delta/2,10+delta/2)
11044 values(r)<-matrix(cost,40,40,byrow=FALSE)
11045
11046 # check what it looks like
```

```

11047 plot(r)
11048 points(pts,pch=20,cex=.4)
11049
11050 # compute ecological distances:
11051 library("gdistance")
11052 tr1<-transition(r,transitionFunction=function(x) 1/mean(x),directions=8)
11053 tr1CorrC<-geoCorrection(tr1,type="c",multpl=FALSE,scl=FALSE)
11054 costs1<-costDistance(tr1CorrC,pts)
11055 outD<-as.matrix(costs1)

11056 In the next block of code we simulate some data and then fit a model to the simulated
11057 data. KIMMY: I forget where “traps” came from. Is it in the “fakecorridor”
11058 object? Test this code XXXXXXXX XXX ANDY: change beta to alpha1 XXXX

11059 library('scrbook')
11060 traplocs<-traps$loc
11061 trap.id<-traps$locid
11062 ntraps<-nrow(traplocs)
11063
11064 set.seed(2013)
11065 N<-200
11066 S.possible<- (1:nrow(pts))[in pts==1]
11067 S.id<-sample(S.possible,N,replace=TRUE)
11068 S<- pts[S.id,]

11069 D<- outD[S.id,trap.id]
11070 eD<- e2dist(S,traplocs)
11071 Dtraps<-outD[trap.id,]

11072 alpha0<- -1.5
11073 sigma<- 1.5
11074 beta<- 1/(2*sigma*sigma)
11075 K<-10
11076
11077 probcap<-plogis(alpha0)*exp(-beta*D*D)
11078 Y<-matrix(NA,nrow=N,ncol=ntraps)
11079 for(i in 1:nrow(Y)){
11080   Y[i,]<-rbinom(ntraps,K,probcap[i,])
11081 }
11082 Y<-Y[apply(Y,1,sum)>0,]

11083 frog1<-nlm(intlik3edv2,c(-2.5,2,log(4)),hessian=TRUE,y=Y,K=K,X=traplocs,
11084   S=pts,D=Dtraps,inpoly=in pts)
11085 frog2<-nlm(intlik3edv2,c(-2.5,2,log(4)),hessian=TRUE,y=Y,K=K,X=traplocs,
11086   S=pts,D=Deuclid,inpoly=in pts)

```

11087 These two models fit, with the correctly specified ecological distance, constrained by  
 11088 the patch boundaries, and that with the ordinary (misspecified) Euclidean distance are

**Table 12.5.** Fitting results XXXXXXXXXXXXXXXX

Distance	neg. LL	alpha0	alpha1	log(n0)
constrained	-21.8921	-1.3380122	0.3321878	4.3530026
Euclidean	-21.1280	-1.3071132	0.3821317	4.2116319

11092 summarized in Table 12.5. We find little difference between the two models. In particular,  
 11093 150 individuals were captured and so truth is  $\log(n_0) = 3.9$ . The correct model produces  
 11094 only a slightly more accurate estimate, and it is favored by only .7 negative log-likelihood  
 11095 units. Therefore, for this single instance, the results are not too different. This is primarily  
 11096 because the distance between individuals, and traps that they are likely to be captured  
 11097 in, is well-approximated by the Euclidean distance.

## 12.8 SUMMARY AND OUTLOOK

11098 Almost all published applications of SCR models to date have been based on models for  
 11099 the encounter probability that are functions of the Euclidean distance between individual  
 11100 activity centers and traps. The obvious limitations of such models are that Euclidean distance  
 11101 is unaffected by landscape or habitat structure and implies stationary, isotropic and  
 11102 symmetrical home ranges. These are standard criticisms of the basic SCR model which we  
 11103 have seen many times in referee reports, or heard in discussions with colleagues. However,  
 11104 this should not be seen as criticism that is inherent to the basic conceptual formulation  
 11105 of SCR models, because, we have shown here that one can modify the Euclidean distance  
 11106 metric to accommodate more realistic formulations of distance that allow for inference to  
 11107 be made about landscape connectivity, and model “distance” as a function of local habitat  
 11108 characteristics. As such, effective distance between individual home range centers and traps  
 11109 varies depending on the local landscape.

11110 How animals use space and therefore how distance to a trap is perceived by individuals  
 11111 is not something that can ever be known. We can only ever conjure up models to describe  
 11112 this phenomenon and fit those models to limited data on a sample of individuals during a  
 11113 limited amount of time. Here we have shown that there is hope to estimate parameters,  
 11114 from capture-recapture data, that describe how animals use space and thereby allow for  
 11115 irregular home range geometry that is influenced by landscape structure.

11116 The simulation study of Royle et al. (2012c) demonstrated (see Table XXXX) that the  
 11117 MLE of model parameters is approximately unbiased in moderate sample sizes. Moreover,  
 11118 the effect of ignoring ecological distance and using normal Euclidean distance in the model  
 11119 for encounter probability, has the logical effect of causing negative bias in estimates of  
 11120  $N$ . This is expected because the effect is similar to failing to model heterogeneity, i.e.,  
 11121 if we mis-specify “model  $M_h$ ” (Otis et al., 1978) with “model  $M_0$ ” (Otis et al., 1978)  
 11122 then we will expect to under-estimate  $N$ . So the effect of mis-specifying the ecological  
 11123 distance metric with a standard homogeneous Euclidean distance has the same effect. As  
 11124 a practical matter, it stands to reason that many previous applications of SCR models  
 11125 based on homogeneous distance metrics have under-stated density of the focal population.

11126 In our view, this bias is not really the most important reason to consider models  
 11127 of ecological distance. Rather, inference about the structure of ecological distance is  
 11128 fundamental to many problems in applied and theoretical ecology related to modeling

11129 landscape connectivity, corridor and reserve design, population viability analysis, gene  
11130 flow, and other phenomena. Models based on least-cost path distance allow investigators  
11131 to evaluate landscape factors that influence movement of individuals over the landscape  
11132 from non-invasively collected capture-recapture data. Therefore SCR models based on  
11133 ecological distance metrics might aid in understanding aspects of space usage and move-  
11134 ment in animal populations and, ultimately, in addressing conservation-related problems  
11135 such as corridor design.

11136  
11137

---

# 13

11138  
11139

## MODELING SPATIAL VARIATION IN DENSITY

11140 Underlying every spatial capture-recapture models is a point process that describes the  
11141 number and distribution of animal activity centers within the state-space ( $\mathcal{S}$ ). A spatial  
11142 point process is characterized by an intensity parameter defined at each location in  $\mathcal{S}$ ;  
11143 and in the case of SCR models, this intensity parameter is population density. If the  
11144 intensity is constant, density is constant throughout  $\mathcal{S}$  and the point process is said to be  
11145 homogeneous. Thus far we have focused our attention on homogeneous point processes  
11146 whose realized values are the locations of the  $N$  activity centers within the state-space.  
11147 When a Poisson prior is placed on  $N$ , we have a homogeneous Poisson point process,  
11148 which is referred to as a model of “complete spatial randomness.” A similar model, that  
11149 we often use in conjunction with data augmentation and MCMC, places a binomial prior  
11150 on  $N$ . This is also a model of spatial randomness, and in this chapter we will compare  
11151 and contrast the two.

11152 The spatial randomness assumption is often viewed as restrictive because ecological  
11153 processes such as territoriality and habitat selection can result in non-uniform distributions  
11154 of organisms. We have argued, however, that this assumption is less restrictive than may  
11155 be recognized because a homogeneous point process actually allows for infinite possible  
11156 “point patterns”, or realized configurations of activity centers. Furthermore, given enough  
11157 data, the uniform prior will have very little influence on the estimated locations of activity  
11158 centers. Nonetheless, a homogeneous point process does not allow one to model population  
11159 density using covariates, which is an important objective in much ecological research. For  
11160 example, even when assuming a homogeneous point process model for the activity centers,  
11161 an estimated density surface may strongly suggest that individuals are more abundant in  
11162 one habitat than another; however, such results do not provide the basis for formally  
11163 testing hypotheses about spatial variation in density, and they could not be used to make  
11164 predictions about habitat-specific abundance in other regions. A more direct approach  
11165 is to replace the homogeneous model with an inhomogeneous model in which the point  
11166 process intensity is allowed to vary spatially.

11167 In this chapter, we cover methods for fitting inhomogeneous Poisson and binomial

11168 spatial point process models by treating the intensity parameter as a function of covariates,  
 11169 in much the same way as is done in generalized linear models. The covariates we consider  
 11170 differ from those covered in previous chapters, which were typically attributes of the  
 11171 animal (e.g. sex or age) or the trap (e.g. baited or not) and were used to model movement  
 11172 or encounter rate. In contrast, here we wish to model covariates that are defined at all  
 11173 points in  $\mathcal{S}$ , which we will refer to as state-space covariates or density covariates. These  
 11174 may include continuous covariates such as elevation, or discrete covariates such as habitat  
 11175 type. Such covariates are often formatted as raster images with a prescribed resolution  
 11176 and extent.

11177 Inhomogeneous Poisson point process models were discussed in the original formulation  
 11178 of SCR models by Efford (2004) and were described in more detail by Borchers and Efford  
 11179 (2008). We will show that an inhomogeneous binomial point process is quite similar to  
 11180 the Poisson model, but is more easily implemented in MCMC algorithms. To do so, we  
 11181 will define the data augmentation parameter  $\psi$  in terms of the point process intensity  
 11182 function, and we will replace the uniform prior on the activity centers with a prior that is  
 11183 also derived from the intensity function. Development of this prior, which does not have  
 11184 a standard form, is a central component of this chapter. First we begin with a review of  
 11185 homogeneous point process models.

### 13.1 HOMOGENEOUS POINT PROCESS REVISITED

11186 The homogeneous Poisson point process is *the* model of complete spatial randomness and  
 11187 is often used in ecology as a null model to test for departures from randomness (Cressie,  
 11188 1991; Diggle, 2003; Illian et al., 2008). The Poisson model asserts that the number of points  
 11189 in  $\mathcal{S}$  is Poisson distributed:  $N \sim \text{Poisson}(\mu|\mathcal{S}|)$  where  $\mu > 0$  is the intensity parameter  
 11190 and  $|\mathcal{S}|$  is the area of the state-space. The intensity parameter  $\mu$  is the density of points,  
 11191 and thus multiplying the intensity by the area of some region yields the expected number  
 11192 of points in that region. As with all homogeneous point process models, the  $N$  points  
 11193 are distributed uniformly, which implies that they do not interact with each other in any  
 11194 way—for example, they neither attract nor repel one another.

11195 Unlike the Poisson point process, the binomial point process assumes that  $N$  is fixed,  
 11196 not random. The distinction is illustrated by this simple R code that generates realizations  
 11197 from Poisson and binomial point processes in the unit square ( $\mathcal{S} = [0, 1] \times [0, 1]$ ):

```
11198 Area <- 1                      # Area of unit square
11199 muP <- 4                      # intensity
11200 nP <- rpois(1, muP*Area)      # number of points: random
11201 PPP <- cbind(runif(nP), runif(nP)) # Poisson point pattern
11202
11203 nB <- 4                      # number of points: fixed
11204 muB <- nB/Area                # intensity
11205 BPP <- cbind(runif(nB), runif(nB)) # binomial point pattern
```

11206 Both of these models are homogeneous because the intensity parameter is constant ( $\mu = 4$   
 11207 in both cases) and the  $N$  points do not interact with each other. This results from the  
 11208 fact that the locations of the points follow a uniform distribution on the plane. The key  
 11209 distinction is that  $N$  is random in the former and fixed in the latter.

Another difference between the Poisson and binomial models is that if the state-space is divided into  $K$  disjunct regions, the number of points in each region  $n(B_k) : k = 1, \dots, K$ ; are independent and identically distributed (i.i.d.) under the Poisson model but not under the binomial model. In the Poisson case, the counts are simply distributed as  $n(B_k) \sim \text{Poisson}(\mu|B_k|)$ , where  $|B_k|$  is the area of the region  $B_k$ . For the binomial case,  $n(B_k) \sim \text{Binomial}(N, \pi(B_k))$  where  $\pi(B_k)$  is the proportion of the state-space in  $B_k$ ; however, these counts are not i.i.d. because the number of points in one region is informative about the number of points in another region. For example, if  $N = 10$ , which would be known for a binomial point process, and if we know that there are 7 points outside the region  $B_1$ , then we can say with certainty that  $B_1 = 10 - 7 = 3$ .

Fig. 13.1 is meant to further illustrate the characteristics of the binomial model. The left panel shows a point pattern realized from a homogeneous binomial point process with  $N = 50$ . The right panel shows the same realization, except that the state-space has been discretized into 25 equally-sized disjunct regions, or pixels, and the counts in each pixel are shown. Since the pixels are the same size,  $\pi(B_k) = 1/25$ , the expected number of point in each pixel is 2:  $\mathbb{E}(n(B_k)) = N\pi(B_k) = 50/25$ , which happens to be the empirical mean in this instance. However, as previously stated, these counts are not independent realizations from a binomial distribution since  $\sum_k n(B_k) = N$ . Rather, the model for the entire vector is multinomial:  $\{n(B_1), n(B_2), \dots, n(B_K)\} \sim \text{Multinomial}(N, \{p(B_1), p(B_2), \dots, p(B_K)\})$  (Illian et al., 2008). If you need a refresher on the multinomial distribution, refer to Sec. 2.2.3, and consider the following R code, which generates counts such as those seen in Fig. 13.1:

```
11232 n.Bk <- rmultinom(1, size=50, prob=rep(1/25, 25))
11233 matrix(n.Bk, 5, 5)
```

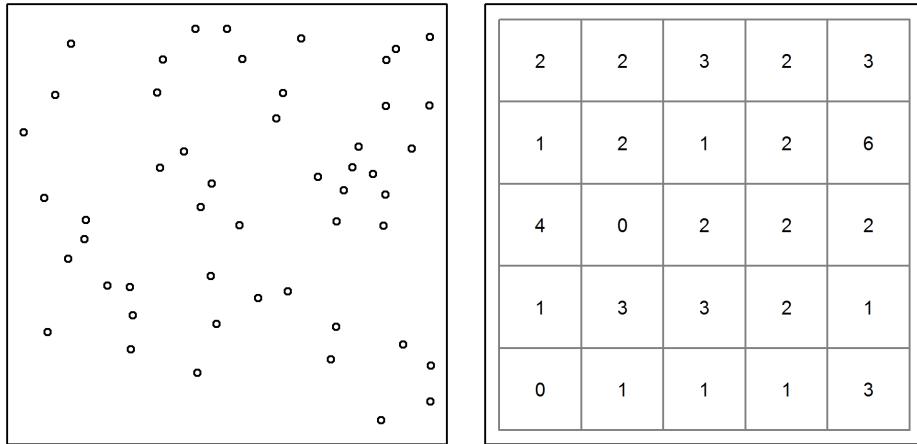
The dependence among counts has virtually no practical consequence when the number of pixels is large. For example, if there are 100 pixels, the number of points in one pixels carries very little information about the expected number of points in another pixel. However, if there are only 2 pixels, then clearly the number of points in one pixel allows one to determine how many points will occur in the remaining pixel.

The discrete representation of space shown in Fig. 13.1 is not only helpful for understanding the properties of a point process, it is also of practical importance when fitting SCR models because spatial covariates are almost always represented as rasters, i.e. grids with predetermined extent and resolution. In such cases, the definition of the prior for the point locations can be changed from the probability that a point occurs at some location in space to the probability that it occurs in some pixel of the raster. As we will explain in Sec. 13.4.2, this typically involves changing the prior from a uniform distribution to a multinomial or categorical distribution.

Up to this point in this chapter we have sketched out the basic characteristics of homogeneous Poisson and binomial point process models. Now we need to speak more specifically about their relevance to SCR models before we move on to the inhomogeneous models. In a SCR model with a homogeneous point process, the intensity parameter  $\mu$  is interpreted as population density, and  $N$  is interpreted as population size<sup>1</sup>. These interpretations are true regardless of whether we consider the Poisson model or the binomial

---

<sup>1</sup>Strictly speaking,  $N$  is the number of activity centers in  $\mathcal{S}$



**Figure 13.1.** Homogeneous binomial point process with  $N=50$  points represented in continuous and discrete space.

model, but since  $N$  is always unknown, one might wonder why we are discussing the binomial model at all.

In our work, we typically adopt the binomial model simply because it is easy to implement using MCMC and data augmentation. And while  $N$  is truly unknown, we use an upper bound  $M$  which is fixed. Thus, the standard point process we use Bayesian in analyses can be regarded in two ways. First, it is a binomial point process with  $M$  points. Second, in terms of  $N$ , it is a thinned binomial point process, where  $\psi$  is the thinning parameter. XXXX Is this thinned point process also binomial, even though  $N$  is no longer fixed? XXXX. With this in mind, the only real difference between the Poisson and binomial models, as implemented in SCR contexts, is that in the former, we have  $N \sim \text{Poisson}(\mu|\mathcal{S})$ , and in the latter we have  $N \sim \text{Binomial}(M, \psi)$ . In other words, we 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 high and  $\psi$  is very low (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 so desire—and we will so desire in the next section where the objective is to model  $\mu$  as

11277 a function of spatially-referenced covariates. First, as an exercise, execute the following  
 11278 **R** commands to familiarize yourself with some of the concepts we just covered:

```
11279 Area <- 1          # Area of state-space
11280 M <- 100           # Data augmentation size
11281 mu <- 10           # Intensity (points per area)
11282 psi <- (mu*Area)/M # Data augmentation parameter (thinning rate)
11283 N <- rbinom(M, 1, psi) # Realized value of N under binomial prior
11284 cbind(runif(N), runif(N)) # Point pattern from thinned binomial model
```

## 13.2 INHOMOGENEOUS POINT PROCESSES

11285 The principal difference between homogeneous and inhomogeneous point processes is that  
 11286 the intensity parameter  $\mu$  is allowed to vary spatially in the latter. Thus, rather than  $\mu$   
 11287 being a fixed constant, it is now a function defined at each point  $\mathbf{s} \in \mathcal{S}$ . A vast number  
 11288 of options exist for modeling spatial variation in the intensity of a point process (Cox,  
 11289 1955; Stoyan and Penttinen, 2000; Illian et al., 2008), but here we focus on modeling  $\mu$   
 11290 as a function of spatially-referenced covariates and a vector of regression coefficients  $\beta$ ; a  
 11291 function we will denote  $\mu(\mathbf{s}, \beta)$ . To be clear,  $\mu(\mathbf{s}, \beta)$ , is a function that returns the expected  
 11292 density of activity centers at location  $\mathbf{s}$ , given the covariate values at  $\mathbf{s}$ . Since the intensity  
 11293 must be positive, and because the natural logarithm is the canonical link function of the  
 11294 Poisson generalized linear model (McCullagh and Nelder, 1989), it is natural to consider  
 11295 the following model:

$$\log(\mu(\mathbf{s}, \beta)) = \beta_0 + \sum_{v=1}^V \beta_v z_v(\mathbf{s}) \quad (13.2.1)$$

11296 which says that there are  $V$  covariates and  $\beta_v$  is the regression coefficient for covariate  
 11297  $z_v(\mathbf{s})$ . This covariate,  $z_v(\mathbf{s})$ , could be any variable defined at all points in the state-  
 11298 space, such as habitat type or elevation. Eq. 13.2.1 should look familiar because it is  
 11299 the standard linear predictor used in Poisson regression. As with other GLMs, one could  
 11300 consider alternative link functions.

11301 Recall from the previous section that for a homogeneous point process, the expected  
 11302 number of points in the state-space was simply the intensity parameter multiplied by area:  
 11303  $\mathbb{E}[N] = \mu|\mathcal{S}|$ . But now that we are regarding the intensity as a function, rather than a  
 11304 scalar, this equation is not very useful. So what is  $\mathbb{E}[N]$  for an inhomogeneous point  
 11305 process? Contemplating a discrete state-space is useful for figuring this out. Imagine  
 11306 that the state-space is represented as a raster with many tiny pixels. In this case, we  
 11307 will associate  $\mathbf{s}$  with pixel ID, i.e.  $\mathbf{s}$  just references some pixel with  $V$  covariates values  
 11308 associated with it. The expected number of individuals in this pixel, say  $\mathbb{E}[n(\mathbf{s})]$ , can  
 11309 intuitively be found by evaluating the intensity function (Eq. 13.2.1) and multiplying it  
 11310 by the area of the pixel. In other words, we compute the expected number of individuals  
 11311 in a pixel by multiplying the expected value of density for that pixel by the area of the  
 11312 pixel. If we do this for each pixel in the state-space, then summing up these values gives  
 11313 us what we are after, the expected value of  $N$ . Specifically,  $\mathbb{E}[N] = \sum_{\mathbf{s} \in \mathcal{S}} \mathbb{E}[n(\mathbf{s})]$ . As  
 11314 the area of the pixels approaches zero, such that we move from discrete space back to

11315 continuous space, the summation must be replaced with an integration of the form:

$$\mathbb{E}[N] = \int_{\mathcal{S}} \mu(\mathbf{s}, \boldsymbol{\beta}) d\mathbf{s}. \quad (13.2.2)$$

11316 Together, Eqs. 13.2.1 and 13.2.2 describe a model for spatial variation in density as well  
11317 as population size. The key task in fitting such inhomogeneous point process models is to  
11318 estimate the  $\boldsymbol{\beta}$  parameters.

11319 We have now described an approach for modeling the point process intensity, yet in  
11320 order to define the likelihood or to develop an MCMC algorithm for the inhomogeneous  
11321 model, we need to specify the prior distribution for the activity centers. Recall that under  
11322 the homogeneous point process, the prior was  $\mathbf{s}_i \sim \text{Uniform}(\mathcal{S})$ , for  $i = 1, \dots, N$ , or  
11323 equivalently:

$$[\mathbf{s}_i] = 1/|\mathcal{S}| \quad (13.2.3)$$

11324 where once again  $|\mathcal{S}|$  denotes the area of the state-space. This simply indicates that an  
11325 activity center is equally likely to occur at any location in the state-space. However, if  
11326 animals exhibit habitat selection or simply occur in one region more often than another,  
11327 it would be preferable to replace this prior with one describing the spatial variation in  
11328 density. Clearly this prior should be determined in some way by the spatially-varying  
11329 intensity function  $\mu(s, \boldsymbol{\beta})$ . Since the integral of a probability density function (pdf) must  
11330 be unity, we can convert  $\mu(\mathbf{s}, \boldsymbol{\beta})$  into a pdf by dividing it by a normalizing constant. In this  
11331 case, the normalizing constant is found by integrating  $\mu(s, \boldsymbol{\beta})$  over the entire state-space.  
11332 The probability density function of the new prior is therefore:

$$[\mathbf{s}_i | \boldsymbol{\beta}] = \frac{\mu(\mathbf{s}_i, \boldsymbol{\beta})}{\int_{\mathcal{S}} \mu(\mathbf{s}, \boldsymbol{\beta}) d\mathbf{s}} \quad (13.2.4)$$

11333 Substituting the uniform prior with this new distribution allows us to fit inhomogeneous  
11334 binomial point process models to spatial capture-recapture data.

11335 As a practical matter, note that the integral in the denominator of Eq. 13.2.4 is  
11336 evaluated over space, and since we always regard space as two-dimensional (the state-  
11337 space is planar), this is a two-dimensional integral that can be approximated using the  
11338 methods discussed in Chapter 9, which include Monte Carlo integration and Gaussian  
11339 quadrature. Alternatively, if our state-space covariates are in raster format, i.e. they are  
11340 in discrete space, the integral can be replaced with a summation over all the pixels in the  
11341 raster,

$$[\mathbf{s}_i | \boldsymbol{\beta}] = \frac{\mu(\mathbf{s}_i, \boldsymbol{\beta})}{\sum_{\mathbf{s} \in \mathcal{S}} \mu(\mathbf{s}, \boldsymbol{\beta})} \quad (13.2.5)$$

11342 where  $\mathbf{s}$  is now defined as “pixel ID” rather than a point in space.

11343 Although the discrete space approach is standard practice, it is technically unjustified  
11344 because covariate values must be known for all points in space. This same problem is  
11345 present anytime that we have a sample of the spatial covariates, rather than a function  
11346 defining their value for all points in space. In such cases, it may be necessary to interpolate  
11347 the values of the covariates for points in space where they were not measured. One option  
11348 would be to use a Kriging interpolator, as demonstrated by Rathbun (1996). Another  
11349 option is to sample the spatial covariates using probabilistic sampling methods, which  
11350 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. ??.

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. 13.2.2 which results in

$$\psi = \frac{\int_S \mu(\mathbf{s}, \boldsymbol{\beta}) d\mathbf{s}}{M} \quad (13.2.6)$$

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  $M$ .

If we refer to the distribution  $[\mathbf{s}_i | \boldsymbol{\beta}]$  as “IPP”, we can write a hierarchical description of a SCR model with a Binomial encounter process and a half-normal, or Gaussian, detection function as

$$\begin{aligned} w_i &\sim \text{Bernoulli}(\psi) \\ \mathbf{s}_i &\sim \text{IPP}(\mu(\mathbf{s}, \boldsymbol{\beta})) \\ p_{ij} &= p_0 \exp(-\|\mathbf{s}_i - \mathbf{x}_j\|^2 / (2\sigma^2)) \\ y_{ij} &\sim \text{Binomial}(K, p_{ij}w_i) \end{aligned}$$

The new prior for  $\mathbf{s}_i$  and Eq. 13.2.6 are the key differences between homogeneous and inhomogeneous models.

In the next sections we walk through a few examples, building up from the simplest case where we actually observe the activity centers as though they were data. In the second example, we fit our new model to simulated data in which density is a function of a single continuous covariate. To build upon the developments in the previous chapter, we further consider the plausible case where a state-space covariate is also a covariate of ecological distance. A small simulation study indicates that both effects can be estimated. A fourth example shows an analysis in discrete space using both **secr** (Efford, 2011) and **JAGS** (Plummer, 2003). In the fifth and final example, we model the intensity of activity centers for a real dataset collected on jaguars (*Panthera onca*) in Argentina.

### 13.3 OBSERVED POINT PROCESSES

In SCR models, the points (activity centers) are not directly observed, but in other contexts they are. Examples include the locations of disease outbreaks, the locations of trees

in a forest, or the locations of radio-tracked animals. In such cases, it is straightforward to fit inhomogeneous point process models and estimate the parameters  $\beta$  from Eq. 13.2.1, 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 whose likelihood is the product of  $N$  densities given by Eq. 13.2.4 (Diggle, 2003, pg. 104). The log-likelihood is thus:

$$\ell(\beta | \{s_i\}) = \sum_{i=1}^N \log(\mu(s_i, \beta)) - \int_S \mu(s, \beta) ds.$$

Having defined the likelihood we could choose a prior distribution for  $\beta$  and obtain the posterior distribution using Bayesian methods, or we can find the maximum likelihood estimates (MLEs) using standard numerical methods as is demonstrated below.

First, we simulate some data under the model  $\mu(s, \beta) = \beta_0 + \beta_1 ELEV(s)$ , where  $ELEV(s)$  is a spatial covariate, say elevation, and  $\beta_0 = 5$  and  $\beta_1 = 2$ . It is worth emphasizing that a spatial covariate must be defined at any location  $s$ , which is demonstrated by the following R code.

```
11397 elev.fn <- function(s) {           # spatial covariate
11398   s <- matrix(s, ncol=2)          # Force s to be a matrix
11399   (s[,1] + s[,2] - 100) / 40.8 # Returns (standardized) "elevation"
11400 }
11401 # intensity function
11402 mu <- function(s, beta0, beta1) exp(beta0 + beta1*elev.fn(s=s))
11403 beta0 <- -6 # intercept of intensity function
11404 beta1 <- 1 # effect of elevation on intensity
11405 # Next line computes integral
11406 EN <- cuhre(2, 1, mu, beta0=beta0, beta1=beta1,
11407               lower=c(0,0), upper=c(100,100))$value
```

The function `elev.fn` returns the value of elevation at any location, which can be either a two-dimensional vector for the coordinates of a single point, or it can be matrix with two columns for a collection of points. The standardization bit is not necessary, but helps with the model fitting below. The next lines of the code define the intensity function  $\mu(s, \beta)$  in terms of elevation and the regression coefficients. The last line uses the `cuhre` function in the `R2Cuba` package (Hahn et al., 2010) to compute the expected value of  $N$  in a  $[0, 100] \times [0, 100]$  square state-space, which is the two-dimensional integral of Eq. 13.2.4. This integral could also be computed using a fine grid of points as we have done in previous chapters, but it is useful to gain familiarity with more efficient integration functions in R.

The R code above demonstrates how to obtain the expected value of  $N$  given a spatial covariate and the coefficients defining the intensity function. Now we need to generate a realized value of  $N$  and distribute the  $N$  points in proportion to the intensity function. This is not as simple as it was to simulate data from a homogeneous point process because the points are no longer uniformly distributed within the state-space. Instead one must resort to methods such as rejection sampling, which involves simulating data from a standard distribution and then accepting or rejecting each point using probabilities defined

11424 by the distribution of interest. For more information, readers should consult an accessible  
 11425 text such as Robert and Casella (2010). In our example, we simulate from a uniform dis-  
 11426 tribution and then accept or reject using the (scaled) probability density function  $[s_i|\beta]$   
 11427 (Eq. 13.2.4). The following **R** commands demonstrate the use of rejection sampling to  
 11428 simulate an inhomogeneous point process for the elevation covariate depicted in Fig. 13.3.

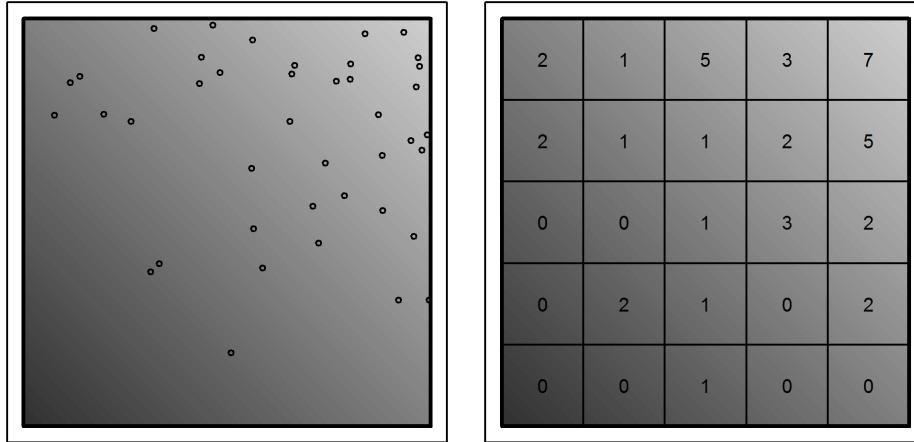
```
11429 set.seed(31025)
11430 N <- rpois(1, EN)      # Realized N
11431 s <- matrix(NA, N, 2) # This matrix will hold the coordinates
11432 elev.min <- elev.fn(c(0,0))
11433 elev.max <- elev.fn(c(100, 100))
11434 Q <- max(c(exp(beta0 + beta1*elev.min), # max of intensity function
11435             exp(beta0 + beta1*elev.max)))
11436 counter <- 1
11437 while(counter <= N) { # begin rejection sampling
11438   x.c <- runif(1, 0, 100); y.c <- runif(1, 0, 100)
11439   s.cand <- c(x.c,y.c) # proposed activity center
11440   pr <- mu(s.cand, beta0, beta1)
11441   if(runif(1) < pr/Q) { # Typically rejected if pr is low
11442     s[counter,] <- s.cand
11443     counter <- counter+1
11444   }
11445 }
```

11446 Similar methods are also implemented in the **R** package **spatstat** (Baddeley and Turner,  
 11447 2005).

11448 The 41 simulated points are shown in Fig 13.3. High elevations are represented by  
 11449 light gray and low elevations are darker. The density of points in apparently higher in  
 11450 lighter regions suggesting that these simulated animals prefer high elevations. Given these  
 11451 points, we will now estimate  $\beta_0$  and  $\beta_1$  by minimizing the negative-log-likelihood using  
 11452 **R**'s **optim** function.

```
11453 nll <- function(beta) { # negative log-likelihood
11454   beta0 <- beta[1]
11455   beta1 <- beta[2]
11456   EN <- cuhre(2, 1, mu, beta0=beta0, beta1=beta1)$value
11457   -(sum(beta0 + beta1*elev.fn(s)) - EN)
11458 }
11459 starting.values <- c(0, 0)
11460 fm <- optim(starting.values, nll, hessian=TRUE)
11461 cbind(Est=fm$par, SE=sqrt(diag(solve(fm$hessian)))) # estimates and SEs
```

11462 Maximizing the likelihood took a fraction of a second, and we obtained estimates of  
 11463  $\hat{\beta}_0 = -5.93$  and  $\hat{\beta}_1 = 0.95$ , which are very close to the data generating values. The  
 11464 95% confidence interval for  $\hat{\beta}_1$  is [0.61, 1.3] and since it does not include zero, the null  
 11465 hypothesis that  $\beta_1 = 0$ , i.e. that there is no effect of elevation on density, can be rejected.  
 11466 In addition to testing hypotheses, these results can be used to predict population size in



**Figure 13.2.** An example of a spatial covariate, say elevation, and a realization from an inhomogeneous Poisson point process with  $\mu(\mathbf{s}, \boldsymbol{\beta}) = \exp(\beta_0 + \beta_1 \text{ELEV}(\mathbf{s}))$  where  $\beta_0 = -6$  and  $\beta_1 = 1$ .

11467 new regions or create predicted density surface maps by plugging the parameter estimates  
11468 into Eqs. 13.2.1 and 13.2.2.

11469 This example demonstrates that if we had the data we wish we had, i.e. if we knew  
11470 the coordinates of the activity centers, we could easily estimate the parameters governing  
11471 the underlying point process and make inferences about spatial variation in density and  
11472 abundance. Unfortunately, in virtually all animal ecology studies, including SCR, the  
11473 locations of the  $N$  animals, or the  $N$  activity centers, cannot be directly observed. Thus  
11474 we need extra information to estimate the locations of these unobserved points. In SCR,  
11475 this information comes from the locations where each animal is captured.

## 13.4 FITTING INHOMOGENEOUS POINT PROCESS SCR MODELS

### 11476 13.4.1 Continuous space

11477 In this example, we will use the same set of points simulated in the previous section to  
11478 generate spatial capture-recapture data. Specifically, we overlay a grid of 49 traps on  
11479 the map shown in Fig. 13.3 and simulate capture histories conditional upon the activity  
11480 centers. Then, we will attempt to estimate the activity center locations as though we did  
11481 not know where they were, as is the case in real applications. We will also estimate  $\beta_0$   
11482 and  $\beta_1$  as before and see how the estimates compare when the points are not actually  
11483 observed. The following R code simulates encounter histories under a Poisson observation  
11484 model (see Chapt. 9), which could be appropriate in camera trapping studies or when  
11485 using other methods in which animals could be detected multiple times at a trap during

```

11486 a single occasion.

11487 xsp <- seq(20, 80, by=10); len <- length(xsp)
11488 X <- cbind(rep(xsp, each=len), rep(xsp, times=len)) # traps
11489 ntraps <- nrow(X); nooccasions <- 5
11490 y <- array(NA, c(N, ntraps, nooccasions)) # capture data
11491 sigma <- 5 # scale parameter
11492 lam0 <- 1 # basal encounter rate
11493 lam <- matrix(NA, N, ntraps)
11494 set.seed(5588)
11495 for(i in 1:N) {
11496   for(j in 1:ntraps) {
11497     # The object "s" was simulated in previous section
11498     distSq <- (s[i,1]-X[j,1])^2 + (s[i,2] - X[j,2])^2
11499     lam[i,j] <- exp(-distSq/(2*sigma^2)) * lam0
11500     y[i,j,] <- rpois(noccasions, lam[i,j])
11501   }
11502 }
11503 # data augmentation
11504 nz <- 80; M <- nz+nrow(y)
11505 yz <- array(0, c(M, ntraps, K))
11506 yz[1:nrow(y),,] <- y # Fill data augmentation array

```

Now that we have a simulated capture-recapture dataset  $y$  and we have augmented it to create the new data object  $yz$ , we can estimate the 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:

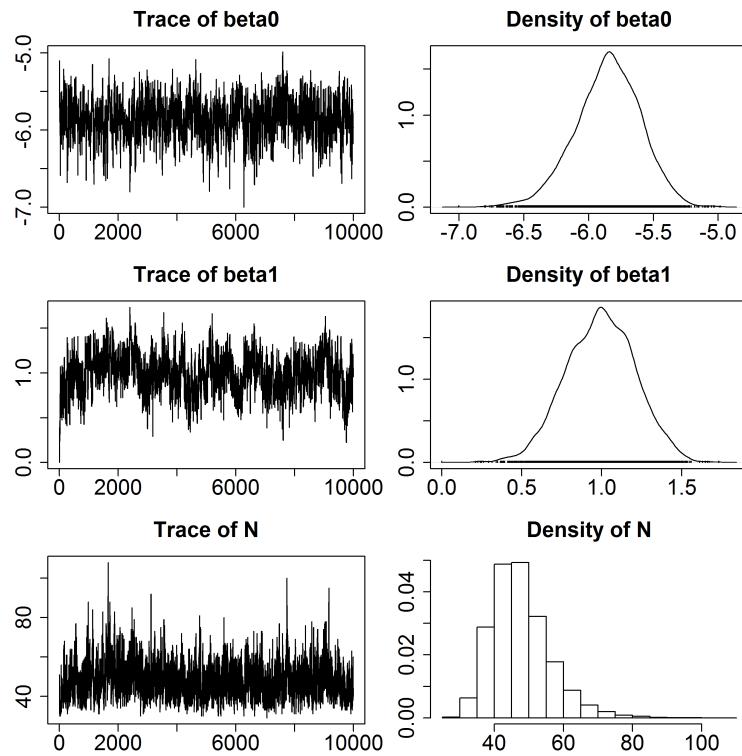
```

11514 set.seed(3434)
11515 fm1 <- scrIPP(yz, X, M, 10000, xlims=c(0,100), ylims=c(0,100),
11516           space.cov=elev.fn,
11517           tune=c(0.4, 0.2, 0.3, 0.3, 7))
11518 plot(mcmc(fm1$out))

```

which requests 10000 posterior samples and estimates the effect of the spatial covariate, elevation, on density. Currently, the function places uniform priors on the parameters  $\sigma$ ,  $\lambda_0$ ,  $\beta_0$  and  $\beta_1$ , although this could easily be modified. Note that any spatial covariate that returns a real value for any location in the state-space can be supplied using the **space.cov** argument. The resulting trace plots of the Markov chains and the posterior distributions for three parameters are shown in Fig. 13.3. The chains appear to converge rapidly but may need to be run longer to reduce Monte Carlo error.

Summaries of the posterior distributions are presented in Table 13.1. The posterior means for  $\beta_0$  and  $\beta_1$  are quite similar to MLEs from the analysis in the previous section in which we assumed no observation error. However, we see that the confidence intervals are wider. With respect to the other parameters in the model, we see that all of the



**Figure 13.3.** 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`.

**Table 13.1.** 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	0.2542	-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
$\mathbb{E}[N] = 39.9$	47.551	10.992	29.837	71.332

11530 data generating parameter fall within the 95% credible intervals. One interesting thing to  
 11531 note is that, although the point estimates for the expected and realized values of  $N$  are  
 11532 quite similar, the estimate is more precise for the realized value. This is to be expected  
 11533 because the uncertainty associated with the realized value of  $N$  is entirely determined by  
 11534 the encounter rate parameters. That is, if we could perfectly detect all of the individuals  
 11535 in  $\mathcal{S}$ , there would be no uncertainty about  $N$ . In contrast, the variance for expected value  
 11536 of  $N$  is affected by the variance of all the parameters in the model, not just the encounter  
 11537 rate parameters. See Efford and Fewster (2012) for additional discussion on the difference  
 11538 between realized and expected values of abundance.

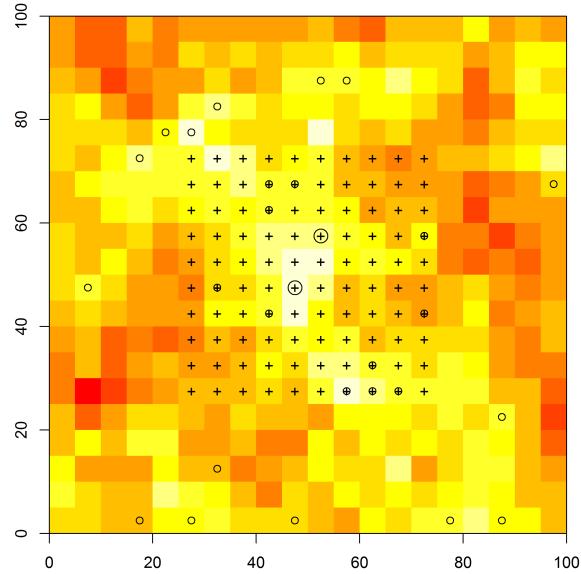
11539 Fitting continuous space inhomogenous point process models is somewhat difficult  
 11540 in **BUGS** because the “IPP” prior  $[s_i|\beta]$ —unlike the uniform prior—is not one of the  
 11541 available distributions that comes with the software. It is possible to add new distributions  
 11542 in **BUGS**, but it is somewhat cumbersome. **secr** allows users to fit continuous space  
 11543 models using linear or polynomial functions of the x- and y- coordinates, but it does not  
 11544 accept truly continuous covariates that are functions of space. However, these are not  
 11545 really important limitations because discrete space versions of the model are straight-  
 11546 forward, and virtually all spatial covariates are, or can be, defined as such.

### 11547 13.4.2 Discrete space

11548 To fit inhomogeneous point process models using covariates in discrete space, i.e. in raster  
 11549 format, we follow the same steps as outlined in Chapter 9—we define  $s_i$  as pixel ID,  
 11550 and we use the categorical distribution as a prior. This effectively changes the problem  
 11551 from estimating the coordinates of an activity center, to estimating the pixel in which  
 11552 an activity center is located. As pixel size approaches zero, these two become equivalent.  
 11553 A good example is found in (Mollet et al., 2012). Here we present an analysis of the  
 11554 simulated data shown in the Fig. 13.3. The spatial covariate, let’s call it forest canopy  
 11555 height (CANHT), was simulated using using the code shown on the help page `ch11` in  
 11556 **scrbook**. The points are the number of activity centers in each pixel, generated from  
 11557 a single realization of the inhomogeneous point process model with intensity  $\mu(s,\beta) =$   
 11558  $\exp(\beta_0 + \beta_1 \text{CANHT}(s)) \times \text{pixelArea}$ , where  $\beta_0 = -6$  and  $\beta_1 = 1$ .

11559 The **BUGS** description of the model is shown in panel 13.1. The vector `probs[]` is  
 11560 the prior probability defined by Eq. 13.2.5, which is the probability that an individual’s  
 11561 activity center is located at pixel  $s$ . `Sgrid` is the matrix of coordinates for each pixel.

11562 This model can also be fit in **secr**, which refers to the raster data as a “habitat  
 11563 mask”. **R** code to format the data and fit the models using **secr** and **JAGS** is available  
 11564 in **scrbook**—see `help(ch9secrYjags)`. Results of the comparison are shown in Table 13.2  
 11565 and are similar as expected. The differences that do exist can be explained by a variety of  
 11566 reasons. For one, there exists some Monte Carlo error in the Bayesian posterior summaries.  
 11567 There is also the fact that posterior summaries can be computed in numerous ways—  
 11568 for example, we could have presented posterior modes or medians instead of means—  
 11569 or, we could have shown highest posterior density credible intervals instead of simple  
 11570 percentiles. The posteriors would also differ if we chose more informative priors than  
 11571 the uniform distributions used here. We see no reason why these issues should be seen as  
 11572 limitations of the Bayesian analysis, rather we would argue that the posterior distribution,  
 11573 which describes the probability that the parameter equals any particular value, is a better



**Figure 13.4.** 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.

descriptor uncertainty than any particular point estimator or confidence interval.

### 13.5 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 Chapter 12.

To demonstrate, we continue with our analysis of the data shown in Fig 13.4.2. Once again, we suppose that density increases with canopy height, but this time, we also make the assumption that home range size decreases 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 Servanyt, 2005). To model this effect, we introduce the parameter  $\theta$ , which determines the "cost" of moving between pixels. If  $\theta = 0$ , then the animal perceives distance as Euclidean. If  $\theta > 0$ , then least-cost distance (LCD) is greater than Euclidean distance

---

```

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) {
    w[i] ~ dbern(psi)
    s[i] ~ dcat(probs[])
    x0g[i] <- Sgrid[s[i],1]
    y0g[i] <- Sgrid[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)) * w[i]
      y[i,j] ~ dpois(lambda[i,j])
    }
  }
  N <- sum(w[]) # Realized value of N
}

```

---

Panel 13.1: **BUGS** code for fitting inhomogeneous point process model in discrete space.

11590 (ED). In most cases, we would not expect, or should not even consider the possibility  
 11591 of  $\theta < 0$  because this implies that LCD<ED, which would mean that an animal could  
 11592 view 1000km as 1m. In addition to the fact that this is not biologically justifiable, it also  
 11593 suggests that the area of the state-space could be infinitely large. Thus, one may want to  
 11594 enforce the constraint that  $\theta$  is  $\geq 0$ . See Chapter 12 for more details.

11595 A question that arises is: Is possible to estimate  $\beta$  and  $\theta$  using standard SCR data? In  
 11596 other words, can we model spatial variation in density and connectivity at the same time,  
 11597 using standard SCR data? Currently, it is not possible to model least-cost distance using  
 11598 **JAGS** or **secr**, so we wrote our own function, **scrDED**, to fit the model using maximum  
 11599 likelihood. An example analysis is provided on the help page for the function in our  
 11600 **R** package **scrbook**. We briefly note here that the function requires the capture history  
 11601 data, the trap locations, and the raster data formatted using the **raster** package (van  
 11602 Etten, 2012). The linear model for the intensity parameter  $\mu(s, \beta)$  and the least-cost

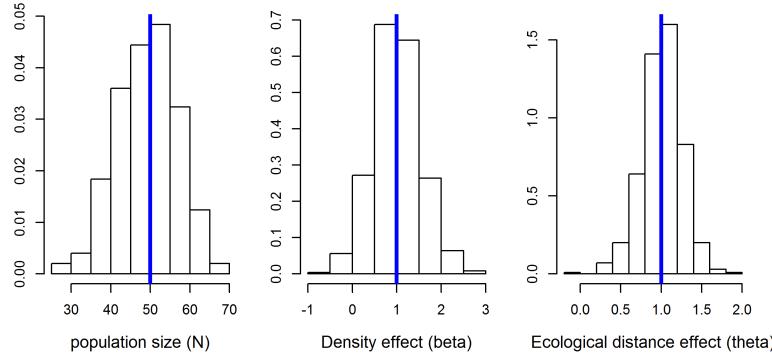
**Table 13.2.** Comparison of **secr** and **JAGS** results. Point estimates from the Bayesian analysis are posterior means. Intervals are lower and upper 95% CIs.

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

11603 distance function  $lcd(\theta)$  are specified using R's formula interface. A simple function call  
 11604 is

```
11605 fm <- scrDED(y, traplocs=X, den.formula=~elev, dist.formula=~elev,
11606 rasters=elev.raster)
```

11607 To assess the possibility of estimating both  $\beta$  and  $\theta$ , we conducted a small simulation  
 11608 study, generating 500 datasets from the model with both parameters set to 1, which  
 11609 corresponds to the conditions described above. The results indicate that it is possible to  
 11610 estimate both parameters (Fig 13.5).



**Figure 13.5.** 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.

### 13.6 THE JAGUAR DATA

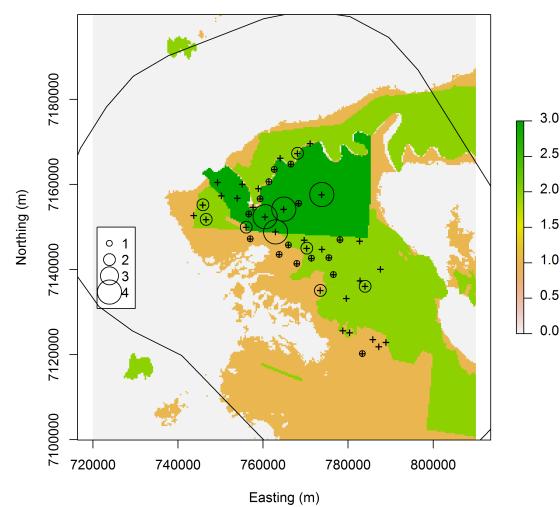
11611 Estimating density of large felines has been a priority for many conservation organizations,  
11612 but few robust methodologies existed before the advent of SCR. Distance sampling is not  
11613 feasible for such rare and cryptic species, and traditional capture-recapture methods yield  
11614 estimates that are highly sensitive to the subjective choice of the effective survey area.  
11615 SCR models provide a powerful alternative because density can be estimated directly and  
11616 data can be collected using non-invasive methods such as camera traps or hair snares.

11617 In this example, we demonstrate how readily density can be estimated for a globally  
11618 imperiled species using SCR. Furthermore, we show how inhomogeneous point process  
11619 models can be used to test important hypotheses regarding the factors affecting density.  
11620 The data come from an 8-year camera-trapping study designed to assess the impacts  
11621 of poaching on jaguar density in Argentina, near the borders of Brazil and Paraguay.  
11622 Additional information about the study is presented in Paviolo et al. (2008) and Paviolo  
11623 et al. (2009). Although jaguars themselves are occasionally killed by poachers, the larger  
11624 concern is the influence of poaching on prey species. To protect jaguars and related  
11625 species, protected areas have been established and three levels of protection are recognized  
11626 as depicted in Fig. 13.6. The dark green area is the Iguazú National Park that is patrolled  
11627 regularly by law enforcement officials. The light green areas are officially protected, but  
11628 due to resource limitations, are not patrolled as often. The beige areas are not protected  
11629 at all, and the gray areas are large soybean monocultures, which provide no habitat.

11630 To test for differences in density between the three regions, we modeled the point  
11631 process intensity parameter as a function of protection status (PROTECT), which we  
11632 treated as an ordinal variable:  $\mu(\mathbf{s}, \boldsymbol{\beta}) = \exp(\beta_0 + \beta_1 \text{PROTECT}(\mathbf{s}))$ . We hypothesized  
11633 that  $\beta_1 > 0$  indicating that jaguar density increases with protection status. In addition  
11634 to modeling spatial variation in density, we also modeled the scale parameter of the half-  
11635 normal or Gaussian encounter model as sex-specific, since male cats typically have larger  
11636 home ranges than females (Sollmann et al., 2011). Since sex is an individual-specific  
11637 covariate, and not observed for the individuals that were not captured, a prior is required  
11638 for the sex of uncaptured individuals. We used an Bernoulli prior with probability 0.5 to  
11639 describe our uncertainty about sex ratio.

11640 The geometry of the state-space differs greatly from the simple square regions that we  
11641 have considered throughout this chapter, which raises a few questions. First, how would  
11642 one integrate Eq. 13.2.4 over a complex spatial region? Earlier we used the function **cuhre**  
11643 in **R** for the two-dimensional integration, but its **lower** and **upper** arguments essentially  
11644 assume that the state-space is square. There are methods of transforming the state-space  
11645 to make this work, but once again we find that it is most convenient to work in discrete  
11646 space and sum over all the pixels defining  $\mathcal{S}$ . In this example, we restricted the state-  
11647 space to exclude the large soybean monocultures surrounding the study area, and we only  
11648 considered area south of the Iguazú River, which runs along the northern border of the  
11649 park shown in dark green in Fig. 13.6. Rather than restricting the state-space, we could  
11650 have modeled the permeability of the river using the methods described in the previous  
11651 section and in Chapter 12; however, no sampling was conducted on the northern side of  
11652 the river, and ancillary data indicates that jaguars rarely forge the waterway.

11653 We fit the model to data from a single year of data from 46 camera stations, each  
11654 consisting of a pair of cameras placed along roads or small trails. Forty-five detections of  
11655 16 jaguars (8 males and 8 females) were made over a 95-day sampling period. The mean



**Figure 13.6.** Jaguar detections at 46 camera trap stations. The three levels of protection status are no protection (beige), some protection (light green), and Iguazú National Park (dark green). Non-habitat (soybean monocultures) is shown in gray.

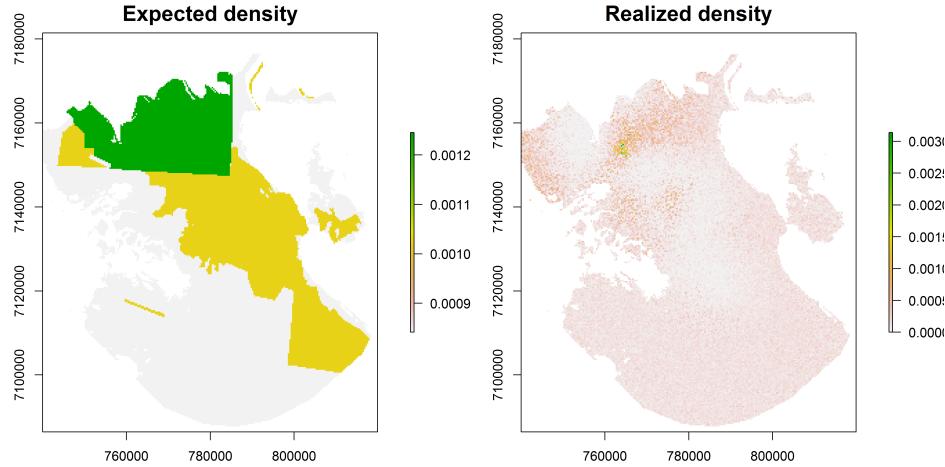
**Table 13.3.** Summaries of posterior distributions from the model of jaguar density.  $\sigma$  is the scale parameter of the half-normal detection function.  $\lambda_0$  is base-line encounter rate.  $\beta_1$  is the effect of protection status on jaguar density.  $\rho$  is the sex-ratio.  $N$  is population size. The last three parameters are the density estimates (jaguars/100km<sup>2</sup>) for the three levels of protection.

	Mean	SD	2.5%	97.5%
$\sigma_{\text{female}}$	5434.886	883.7433	4093.6069	7549.062
$\sigma_{\text{male}}$	6208.341	822.6217	4881.4060	8093.759
$\lambda_0$	0.013	0.0036	0.0068	0.021
$\beta_0$	-4.667	0.2866	-5.2527	-4.093
$\beta_1$	0.196	0.3672	-0.5179	0.961
$\rho$	0.541	0.0551	0.4286	0.644
$N$	36.428	9.6986	23.0000	61.000
$D_{\text{low}}$	0.921	0.3851	0.3789	1.894
$D_{\text{med}}$	0.775	0.3006	0.2653	1.503
$D_{\text{high}}$	1.444	0.3325	0.8791	2.110

number of sampling days at each camera station was 48.2. The raw capture data shown in Fig. 13.6 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 `jaguarDataCh9`. Parameter estimates are shown in Table13.3.

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  $\beta_0 > 0$  was 0.705, and the posterior mean of realized density was only 58% higher in the national park than in the unprotected area. Fig. 13.6 shows the estimated density surfaces. The first map is the expected density (posterior mean) 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. Conditional on the  $N$ , this is the probability distribution for the number of activity centers in each pixel of the rasterized state-space—here shown as the posterior mean. 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 town, or some other accessibility metric. It would also be interesting to model density separately for each sex. 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 worth investigation include treating PROTECT as a categorical, rather than ordinal, variable; and, it would interesting to assess the effects of roads and trails on jaguar movement using the methods described in Chapt. ???. Developing models for these extensions could be readily accomplished by modifying the fitting functions found in the



**Figure 13.7.** Estimated density surfaces from the analysis of the jaguar data.

11685 R package `scrbook`.

### 13.7 SUMMARY

11686 One of the distinguishing features of spatial capture-recapture models is that they allow  
 11687 for inference about spatial variation in density without relying on ad hoc approaches  
 11688 for determining the amount of area surveyed. The approach described in this chapter  
 11689 involves modeling the locations of activity centers as outcomes of an inhomogeneous point  
 11690 process with intensity determined by covariates defined at all locations in the state-space.  
 11691 Covariate effects can be evaluated in exactly the same way as is done in generalized linear  
 11692 models, making it easy to interpret the results.

11693 All the examples in this section included a single state-space covariate, but this was  
 11694 for simplicity only. Including multiple covariates poses no additional challenges. Similarly,  
 11695 additional model structure such sex-specific encounter rate parameters or behavioral re-  
 11696 sponses can be accommodated and fit using `secr`, **BUGS**, or by extending the functions  
 11697 in `scrbook`. It is also possible to consider covariates that affect both density and ecolog-  
 11698 ical distance. The ramifications of this are enormous for applied ecological research and  
 11699 conservation efforts because researchers can use capture-recapture data to identify areas  
 11700 where both density and landscape connectivity are high (Royle et al., 2012c). Address-  
 11701 ing such questions is simply not possible using standard, non-spatial capture-recapture  
 11702 methods.

11703 Although we focused on modeling the point process intensity as a function of covariates,  
 11704 other options for fitting inhomogeneous model exist Illian et al. (2008). Cox processes are  
 11705 models in which the intensity parameter is a function of spatial random effects. Such  
 11706 methods are useful for accommodating overdispersion, but it seems unlikely that most

11707 SCR datasets could support such complexity. Gibbs processes are another important  
11708 class of models that are distinguished by the interactions of points. Although little work  
11709 has been done on such models in the context of SCR studies (Reich et al., 2012), we  
11710 expect they will receive more attention because they can be used to model processes  
11711 such territoriality (points repel one another) or aggregation (points attract one another).  
11712 Neyman-Scott processes are another option for modeling aggregation or clustering, and  
11713 could be useful for studying gregarious species.



11714

## **Part IV**

11715

---

11716

## **Advanced SCR Models**



11717  
11718

# 14

---

11719  
11720

## DEVELOPING MARKOV CHAIN MONTE CARLO SAMPLERS

11721 In this chapter we will dive a little deeper into Markov chain Monte Carlo (MCMC)  
11722 sampling. We will construct custom MCMC samplers in **R**, starting with easy-to-code  
11723 GLMs and GLMMs and moving on to simple CR and SCR models. Finally, we will  
11724 illustrate some alternative ready-to-use software packages for MCMC sampling. We will  
11725 NOT provide exhaustive background information on the theory and justification of MCMC  
11726 sampling – there are entire books dedicated to that subject and we refer you to Robert and  
11727 Casella (2004) and Robert and Casella (2010). Rather we aim to provide you with enough  
11728 background and technical know-how to start building your own MCMC samplers for SCR  
11729 models in **R**. You will find that quite a few topics that come up in this chapter have already  
11730 been covered in previous chapters, particularly the introduction into Bayesian analysis in  
11731 Chapt. 3. To keep you from having to leaf back and forth we will in some places briefly  
11732 review aspects of Bayesian analysis, but we try to focus on the more technical issues of  
11733 building MCMC samplers relevant to SCR models.

### 11734 14.0.1 Why build your own MCMC algorithm?

11735 The standard programs we have used so far to do MCMC analyses are **WinBUGS** (Gilks  
11736 et al., 1994) and **JAGS** (Plummer, 2003). The wonderful thing about these **BUGS**  
11737 engines is that they automatically use appropriate and, most of the time, reasonably  
11738 efficient forms of MCMC sampling for the model specified by the user.

11739 The fact that we have such a Swiss Army knife type of MCMC machine begs the  
11740 question: Why would anyone want to build their own MCMC algorithm? For one, there  
11741 are a limited number of distributions and functions implemented in **BUGS**. While **Open-**  
11742 **BUGS** provides more options, some more complex models may be impossible to build  
11743 within these programs. A very simple example from spatial capture-recapture that can  
11744 give you a headache in **WinBUGS** is when your state-space is an irregular-shaped poly-  
11745 gon, rather than an ideal rectangle that can be characterized by four pairs of coordinates.

11746 It is easy to restrict activity centers to any arbitrary polygon in **R** using an ESRI shapefile  
 11747 (and we will show you an example in a little bit), but you cannot use a shape file in a  
 11748 **BUGS** model. Similarly, models of space usage that take into account ecological distance  
 11749 (Chapt. 12) cannot be implemented in the **BUGS** engines. Moreover, there are classes of  
 11750 SCR models that we have not been able to implement effectively using likelihood methods,  
 11751 and are inefficient to run in the **BUGS** engines. Examples of those models are covered in  
 11752 Chaps. 18 and 19.

11753 Sometimes implementing an MCMC algorithm in **R** may be faster than in **WinBUGS**  
 11754 - especially if you want to run simulation studies where you have hundreds or more sim-  
 11755 ulated data sets, several years' worth of data or other large models, this can be a big  
 11756 advantage.

11757 Finally, building your own MCMC algorithm is a great exercise to understand how  
 11758 MCMC sampling works. So while using the **BUGS** language requires you to understand  
 11759 the structure of your model, building an MCMC algorithm requires you to think about  
 11760 the relationship between your data, priors and posteriors, and how these can be efficiently  
 11761 analyzed and characterized. Not to mention that, if you are an **R** junkie, it can actually  
 11762 be fun. However, if you don't think you will ever sit down and write your own MCMC  
 11763 sampler, consider skipping this chapter - apart from coding it will not cover anything  
 11764 SCR-related that is not covered by other, more model-oriented chapters as well.

## 14.1 MCMC AND POSTERIOR DISTRIBUTIONS

11765 MCMC is a class of simulation methods for drawing (correlated) random numbers from  
 11766 a target distribution, which in Bayesian inference is the posterior distribution. As a re-  
 11767 minder, the posterior distribution is a probability distribution for an unknown parameter,  
 11768 say  $\theta$ , given observed data and its prior probability distribution (the probability distribu-  
 11769 tion we assign to a parameter before we observe data). The great benefit of having the  
 11770 posterior distribution of  $\theta$  is that it can be used to make probability statements about  
 11771  $\theta$ , such as the probability that  $\theta$  is equal to some value, or the probability that  $\theta$  falls  
 11772 within some range of values. The posterior distribution summarizes all we know about a  
 11773 parameter and thus, is the central object of interest in Bayesian analysis. Unfortunately,  
 11774 in many if not most practical applications, it is nearly impossible to directly compute the  
 11775 posterior. Recall Bayes' theorem:

$$[\theta|y] = \frac{[y|\theta][\theta]}{[y]}, \quad (14.1.1)$$

11776 where  $\theta$  is the parameter of interest,  $y$  is the observed data,  $[\theta|y]$  is the posterior,  $[y|\theta]$  the  
 11777 likelihood of the data conditional on  $\theta$ ,  $[\theta]$  the prior probability of  $\theta$ , and, finally,  $[y]$  is the  
 11778 marginal probability of the data, defined as

$$[y] = \int [y|\theta][\theta]d\theta$$

11779 This marginal probability is a normalizing constant that ensures that the posterior  
 11780 integrates to 1. Often, the integral is difficult or impossible to evaluate, unless you are  
 11781 dealing with a really simple model. For example, consider a Normal model, with a set of

11782  $n$  observations,  $y_i; i = 1, 2, \dots, n$ :

$$y_i \sim \text{Normal}(\mu, \sigma),$$

11783 where  $\sigma$  is known and our objective is to obtain an estimate of  $\mu$ . To fully specify the  
 11784 model in a Bayesian framework, we first have to define a prior distribution for  $\mu$ . Recall  
 11785 from Chapt. 3 that for certain data models, certain priors lead to conjugacy, i.e. if you  
 11786 choose a certain prior for your parameter, the posterior distribution will be of a known  
 11787 parametric form. The conjugate prior for the mean of a Normal model is also a Normal  
 11788 distribution:

$$\mu \sim \text{Normal}(\mu_0, \sigma_0^2)$$

11789 If  $\mu_0$  and  $\sigma_0^2$  are fixed, the posterior for  $\mu$  has the following form (for some of the algebra  
 11790 behind this, see Chapt. 2 in Gelman et al. (2004)):

$$\mu|y \sim \text{Normal}(\mu_n, \sigma_n^2) \quad (14.1.2)$$

11791 where

$$\mu_n = \left( \frac{\sigma^2}{\sigma^2 + n\sigma_0^2} \right) \times \left( \mu_0 + \frac{n\sigma_0^2}{\sigma^2 + n\sigma_0^2} \right) \times \bar{y}$$

11792 And

$$\sigma_n^2 = \frac{\sigma^2 \sigma_0^2}{\sigma^2 + n\sigma_0^2}$$

11793 We can directly obtain estimates of interest from this Normal posterior distribution, such  
 11794 as the mean  $\hat{\mu}$  and its variance; we do not need to apply MCMC, since we can recognize  
 11795 the posterior as a parametric distribution, including the normalizing constant  $[y]$ . But  
 11796 generally we will be interested in more complex models with several, say  $m$ , parameters.  
 11797 In this case, computing  $[y]$  from Eq. 14.1.1 requires  $m$ -dimensional integration, which can  
 11798 be difficult or impossible. Thus, the posterior distribution in generally only known up to  
 11799 a constant of proportionality:

$$[\theta|y] \propto [y|\theta][\theta]$$

11800 The power of MCMC is that it allows us to approximate the posterior using simulation  
 11801 without evaluating the high dimensional integrals and to directly sample from the pos-  
 11802 terior, even when the posterior distribution is unknown! The price is that MCMC is  
 11803 computationally expensive. Although MCMC first appeared in the scientific literature in  
 11804 1949 (Metropolis and Ulam, 1949), widespread use did not occur until the 1980s when  
 11805 computational power and speed increased (Gelfand and Smith, 1990). It is safe to say that  
 11806 the advent of practical MCMC methods is the primary reason why Bayesian inference has  
 11807 become so popular during the past three decades.

11808 In a nutshell, MCMC lets us generate sequential draws of  $\theta$  (the parameter(s) of in-  
 11809 terest) from distributions approximating the unknown posterior over  $T$  iterations. The  
 11810 distribution of the draw at  $t$  depends on the value drawn at  $t-1$ ; hence, the draws from a  
 11811 Markov chain<sup>1</sup>. As  $T$  goes to infinity, the Markov chain converges to the desired distri-  
 11812 bution, in our case the posterior distribution for  $\theta|y$ . Thus, once the Markov chain has  
 11813 reached its stationary distribution, the generated samples can be used to characterize the  
 11814 posterior distribution,  $[\theta|y]$ , and point estimates of  $\theta$ , its standard error and confidence  
 11815 bounds, can be obtained directly from this approximation of the posterior.

<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,  $\theta^{(t-1)}$ .

---

## 14.2 TYPES OF MCMC SAMPLING

11816 There are several general MCMC algorithms in widespread use, the most popular being  
 11817 Gibbs sampling and Metropolis-Hastings sampling, both of which were briefly introduced  
 11818 in Chapt. 3. We will be dealing with these two classes in more detail and use them to  
 11819 construct MCMC algorithms for SCR models. Also, we will briefly review alternative  
 11820 techniques that are applicable in some situations.

11821 **14.2.1 Gibbs sampling**

11822 Gibbs sampling was named after the physicist J.W. Gibbs by Geman and Geman (1984),  
 11823 who applied the algorithm to a Gibbs distribution<sup>2</sup>. The roots of Gibbs sampling can  
 11824 be traced back to work of Metropolis et al. (1953), and it is actually closely related to  
 11825 Metropolis sampling (see Chapt. 11.5 in Gelman et al. (2004), for the link between the  
 11826 two samplers). We will focus on the technical aspects of this algorithm, but if you find  
 11827 yourself hungry for more background, Casella and George (1992) provide a more in-depth  
 11828 introduction to the Gibbs sampler.

11829 Let's go back to our simple example from above to understand the motivation and  
 11830 functioning of Gibbs sampling. Recall that for a Normal model with known variance and  
 11831 a Normal prior for  $\mu$ , the posterior distribution of  $\mu|y$  is also Normal. Conversely, with a  
 11832 fixed (known)  $\mu$ , but unknown variance, the conjugate prior for  $\sigma^2$  is an Inverse-Gamma  
 11833 distribution with shape and scale parameters  $a$  and  $b$ :

$$\sigma^2 \sim \text{Inverse-Gamma}(a, b),$$

11834 With fixed  $a$  and  $b$ , the posterior  $[\sigma^2|\mu, y]$  is also an Inverse-Gamma distribution, namely:

$$\sigma^2|\mu, y \sim \text{Inverse-Gamma}(a_n, b_n), \quad (14.2.1)$$

11835 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  
 11836  $\mu$  nor  $\sigma^2$ , which is probably the more common case? The joint posterior distribution of  $\mu$   
 11837 and  $\sigma^2$  now has the general structure  
 11838

$$[\mu, \sigma^2|y] = \frac{[y|\mu, \sigma^2][\mu][\sigma^2]}{\int [y|\mu][\mu][\sigma^2]d\mu d\sigma^2}$$

11839 or

$$[\mu, \sigma^2|y] \propto [y|\mu, \sigma^2][\mu][\sigma^2]$$

11840 This cannot easily be reduced to a distribution we recognize. However, we can con-  
 11841 dition  $\mu$  on  $\sigma^2$  (i.e., we treat  $\sigma^2$  as fixed) and remove all terms from the joint posterior  
 11842 distribution that do not involve  $\mu$  to construct the full conditional distribution,

$$[\mu|\sigma^2, y] \propto [y|\mu][\mu]$$

11843 The full conditional of  $\mu$  again takes the form of the Normal distribution shown in  
 11844 Eq. 14.1.2; similarly,  $[\sigma^2|\mu, y]$  takes the form of the Inverse-Gamma distribution shown in

---

<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

Eq. 14.2.1, both distribution we can easily sample from. And this is precisely what we do when using Gibbs sampling: we break down high-dimensional problems into convenient one-dimensional problems by constructing the full conditional distributions for each model parameter separately; and we sample from these full conditionals, which, if we choose conjugate priors, are known parametric distributions. Let's put the concept of Gibbs sampling into the MCMC framework of generating successive samples, using our simple Normal model with unknown  $\mu$  and  $\sigma^2$  and conjugate priors as an example. These are the steps you need in order to build a Gibbs sampler:

**Step 0:** Begin with some initial values for  $\theta$ , say  $\theta^{(0)}$ . In our example, we have to specify initial values for  $\mu$  and  $\sigma$ , for example by drawing a random number from some Uniform distribution, or by setting them close to what we think they might be. (Note: This step is required in any MCMC sampling; chains have to start from somewhere. We will get back to these technical details a little later.)

**Step 1:** Draw  $\theta_1^{(1)}$  from the conditional distribution  $[\theta_1^{(1)} | \theta_2^{(0)}, \dots, \theta_d^{(0)}]$ . Here,  $\theta_1$  is  $\mu$ , which we draw from the Normal distribution in Eq. 14.1.2 using  $\sigma^{(0)}$  as value for  $\sigma$ .

**Step 2:** Draw  $\theta_2^{(1)}$  from the conditional distribution  $[\theta_2^{(1)} | \theta_1^{(1)}, \theta_3^{(0)}, \dots, \theta_d^{(0)}]$ . Here,  $\theta_2$  is  $\sigma$ , which we draw from the Inverse-Gamma distribution of Eq. 14.2.1, using  $\mu^{(1)}$  as value for  $\mu$ .

**Step 3, ..., d:** Draw  $\theta_3^{(1)}, \theta_4^{(1)}, \dots, \theta_d^{(1)}$  from their conditional distribution  $[\theta_3^{(1)} | \theta_1^{(1)}, \theta_2^{(1)}, \theta_4^{(0)}, \dots, \theta_d^{(0)}], \dots, [\theta_d^{(1)} | \theta_1^{(1)}, \dots, \theta_{d-1}^{(1)}]$ . In our example we have no additional parameters, so we only need step 0 through to 2.

**Repeat Steps 1 to d** for  $T =$  a large number of samples.

In terms of **R** coding, this means we have to write Gibbs updaters for  $\mu$  and  $\sigma^2$  and embed them into a loop over  $T$  iterations. The final code in the form of an **R** function is shown in Panel 14.1.

This is it! You can go ahead and simulate some data,  $y \sim \text{Normal}(5, 0.5)$  and then use the function `NormGibbs()` in the **R** package `scrbook` to run your first Gibbs sampler (note that the **R** function `rnorm` requires you to supply the standard deviation  $\sigma$  and we have written `NormGibbs` so that it returns  $\sigma$  instead of  $\sigma^2$  so you can easily compare your input value and parameter estimate).

```
11875 set.seed(13)
11876
11877 #true mean and sd are 5 and 0.5
11878 y<-rnorm(1000, 5,0.5) #data
11879
11880 mu_0<-0 #prior mean
11881 sigma2_0<-100 #prior variance
11882
11883 #Inverse-Gamma hyperparameters
11884 a<-0.1
11885 b<-0.1
11886
11887 mod=Norm.Gibbs(y, mu_0, sigma2_0, a,b,niter=10000)
```

11888 Your output, `mod`, will be a table with two columns, one per parameter, and  $T$  rows,  
 11889 one per iteration. For this 2-parameter example you can visualize the joint posterior by  
 11890 plotting samples of  $\mu$  against samples of  $\sigma$  (Fig. 14.1):

11891 `plot(out[,1], out[,2])`

11892 The marginal distribution of each parameter is approximated by examining the samples  
 11893 of this particular parameter. You can visualize it by plotting a histogram of the samples  
 11894 (Fig. 14.2 upper left and right):

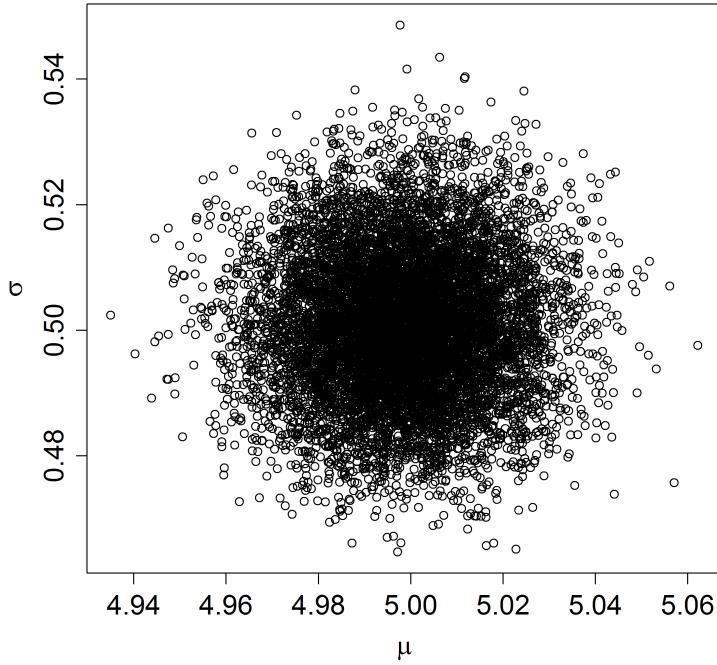
11895 `par(mfrow=c(1,2))`  
 11896 `hist(out[,1]); hist(out[,2])`

11897 Finally, recall an important characteristic of Markov chains, namely, that the chain  
 11898 has to have converged (reached its stationary distribution) in order to regard samples as  
 11899 coming from the posterior distribution. In practice, that means you have to throw out  
 11900 some of the initial samples called the burn-in. We will talk about this in more detail  
 11901 when we talk about convergence diagnostics. For now, you can use the `plot(out[,1])` or  
 11902 `plot(out[,2])` command to make a time series plot of the samples of each parameter and  
 11903 visually assess how many of the initial samples you should discard. Fig. 14.2 bottom left  
 11904 and right shows plots for the estimates of  $\mu$  and  $\sigma$  from our simulated data set; you see  
 11905 that in this simple example the Markov chain apparently reaches its stationary distribution  
 11906 very quickly – the chains look ‘grassy’ seemingly from the start. It is hard to discern a  
 11907 burn-in phase visually (but we will see examples further on where the burn-in is clearer)  
 11908 and you may just discard the first 500 draws to be sure you only use samples from the  
 11909 posterior distribution. The mean of the remaining samples are your estimates of  $\mu$  and  $\sigma$ :

11910 `summary(mod[501:10000,])`  
 11911        mu                sig  
 11912   Min.  :4.935    Min.  :0.4652  
 11913   1st Qu.:4.988   1st Qu.:0.4930  
 11914   Median :4.998   Median :0.5006  
 11915   Mean   :4.998   Mean   :0.5008  
 11916   3rd Qu.:5.009   3rd Qu.:0.5084  
 11917   Max.   :5.062   Max.   :0.5486

### 11918 14.2.2 Metropolis-Hastings sampling

11919 Although it is applicable to a wide range of problems, the limitations of Gibbs sampling are  
 11920 immediately obvious: what if we do not want to use conjugate priors or what if we cannot  
 11921 recognize the full conditional distribution as a parametric distribution, or simply do not  
 11922 want to worry about these issues? The most general solution is to use the Metropolis-  
 11923 Hastings (MH) algorithm, which also goes back to the work by Metropolis et al. (1953).  
 11924 You saw the basics of this algorithm in Chapt. 3. In a nutshell, because we do not  
 11925 recognize the posterior  $[\theta|y]$  as a parametric distribution, the MH algorithm generates  
 11926 samples from a known proposal distribution, say  $h(\theta)$ , that depends on the value of  $\theta$  at  
 11927 the previous time step,  $\theta^{t-1}$ . The candidate value  $\theta^*$  is accepted with probability.



**Figure 14.1.** Joint posterior distribution of  $\mu$  and  $\sigma$  from a Normal Model

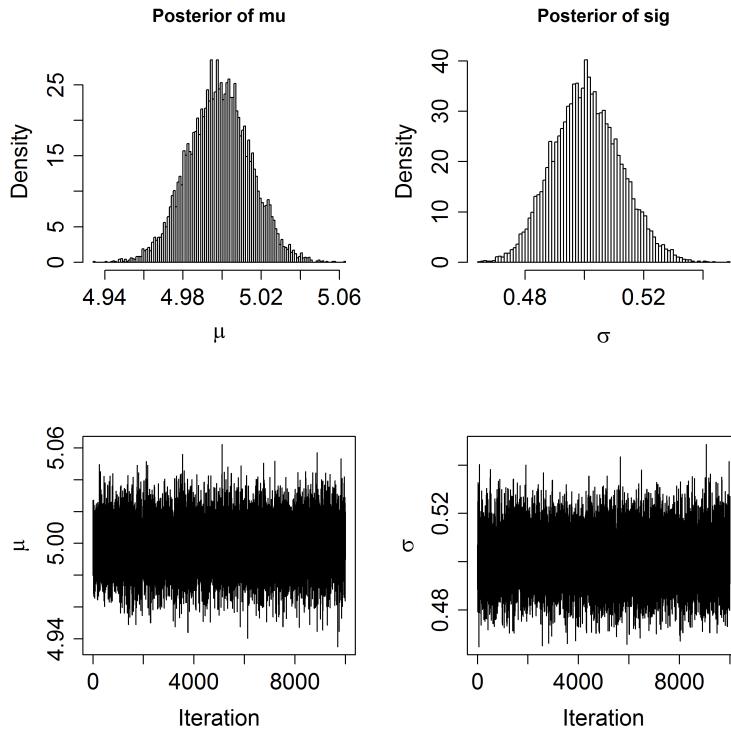
$$r = \frac{[\theta^{t-1}|y]h(\theta^*|\theta^{t-1})}{[\theta^*|y]h(\theta^t|\theta^*)}$$

Proposal distributions can be absolutely anything! You can generate candidate values from a  $\text{Normal}(0,1)$  distribution, from a  $\text{Uniform}(-3455,3455)$  distribution, or anything of proper support. Note, however, that good choices of  $h()$  are those that approximate the posterior distribution. Obviously if  $h() = [\theta|y]$  (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.

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 at the current value. A later development of the algorithm by Hastings (1970) lifted this



**Figure 14.2.** 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).

condition. Since using a symmetric proposal distribution makes life a little easier, we are going to focus on this specific case. A type of symmetric proposal useful in many situations is the so-called *random-walk* proposal distribution where candidate values are drawn from a normal distribution with mean equal to the current value and some standard deviation, say  $\delta$ , which is prescribed by the user (see below for further explanation).

**Parameters with bounded support:** Many models contain parameters that have bounded support. E.g., variance parameters live on  $[0, \infty]$ , parameters that represent probabilities live on  $[0, 1]$ , etc.. For such cases, it is sometimes convenient to use a random walk proposal distribution that can generate any real number (e.g., a normal random walk proposal). Under these circumstances you should not constrain the proposal distribution itself, but you can just reject parameters that are outside of the parameter space (sec. 6.4.1 in Robert and Casella, 2010). You will see plenty of examples of updating parameters with bounded support in this chapter.

It is worth knowing that there are alternatives to the random walk MH algorithm.

11952 For example, in the independent MH,  $\theta^*$  does not depend on  $\theta^{t-1}$ , while the Langevin  
 11953 algorithm (Roberts and Rosenthal, 1998) aims at avoiding the random walk by favoring  
 11954 moves towards regions of higher posterior probability density. The interested reader should  
 11955 look up these algorithms in Robert and Casella (2004) or Robert and Casella (2010).

11956 Building a MH sampler can be broken down into several steps. We are going to  
 11957 demonstrate these steps using a different but still simple and common model: the logit-  
 11958 normal or logistic regression model. For simplicity, assume that

$$y \sim \text{Bernoulli} \left( \frac{\exp(\theta)}{1 + \exp(\theta)} \right)$$

11959 and

$$\theta \sim \text{Normal}(\mu, \sigma)$$

11960 The following steps are required to set up a random walk MH algorithm:

11961 **Step 0:** Choose initial values,  $\theta^{(0)}$ .

11962 **Step 1:** Generate a proposed value of  $\theta$  from  $h(\theta^* | \theta^{t-1})$ . We often use a Normal proposal  
 11963 distribution, so we draw  $\theta^{(1)}$  from  $\text{Normal}(\theta^{(0)}, \delta)$ , where  $\delta$  is the variance of the Normal  
 11964 proposal distribution, the tuning parameter that we have to set.

11965 **Step 2:** Calculate the ratio of posterior densities for the proposed and the original value  
 11966 for  $\theta$ :

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

11967 In our example,

$$r = \frac{\text{Bernoulli}(y | \theta^*) \times \text{Normal}(\theta^* | \mu, \sigma)}{\text{Bernoulli}(y | \theta^{t-1}) \times \text{Normal}(\theta^{t-1} | \mu, \sigma)}$$

11968 **Step 3:** Set

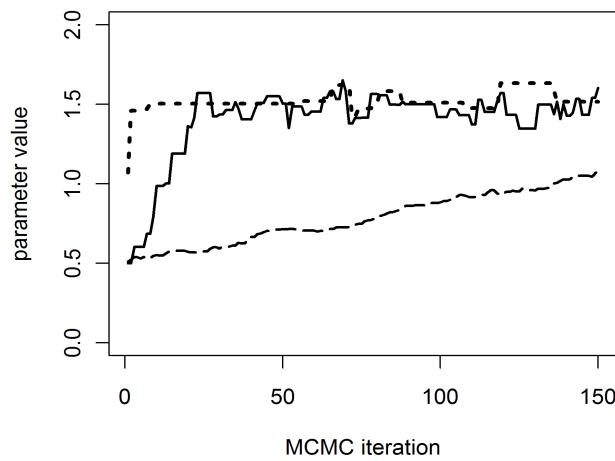
$$\begin{aligned} \theta^t &= \theta^* \text{ with probability } \min(r, 1) \\ &= \theta^{t-1} \text{ otherwise} \end{aligned}$$

11969 We can do this last step by drawing a random number  $u$  from a  $\text{Uniform}(0, 1)$  and  
 11970 accept  $\theta^*$  if  $u < r$ . Repeat for  $t = 1, 2, \dots$  a large number of samples. The **R** code for this  
 11971 MH sampler is provided in Panel 14.2.

11972 The reason we sum the logs of the likelihood and the prior, rather than multiplying  
 11973 the original values, is simply computational. The product of small probabilities can be  
 11974 numbers very close to 0, which computers do not handle well. Thus we add the logarithms,  
 11975 sum, and exponentiate to achieve the desired result. Similarly, in case you have forgotten,  
 11976  $x/y = \exp(\log(x) - \log(y))$ , with the latter being favored for computational reasons.

11977 Comparing MH sampling to Gibbs sampling, where all draws from the conditional  
 11978 distribution are used, in the MH algorithm we discard a portion of the candidate values,  
 11979 which inherently makes it less efficient than Gibbs sampling – the price you pay for its  
 11980 increased generality. In Step 1 of the MH sampler we had to choose a variance,  $\delta$ , for  
 11981 the Normal proposal distribution. Choice of the parameters that define our candidate  
 11982 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 the samples drawn during the adaptive phase are discarded. To illustrate the effects of tuning, we ran the Metropolis-within-Gibbs algorithm in Panel 14.2 with  $\delta = 0.01$ ,  $\delta = 0.2$  and  $\delta = 1$ . The first 150 iterations for  $\theta$  are shown in Fig. 14.3. 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  $\delta = 1$ .



**Figure 14.3.** 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).

12003 Other than graphically, you can easily check acceptance rates for the parameters you  
 12004 monitor (that are part of your output) using the `rejectionRate()` function of the package  
 12005 `coda` (we will talk more about this package a little later on). Do not let the term 'rejection  
 12006 rate' confuse you; it is simply  $1 - \text{acceptance rate}$ . There may be parameters – for example,  
 12007 individual values of a random effect or latent variables – that you do not want to save,  
 12008 though, and in our next example we will show you a way to monitor their acceptance rates  
 12009 with a few extra lines of code.

### 12010 14.2.3 Metropolis-within-Gibbs

12011 One weakness of the MH sampler is that formulating the joint posterior when evaluating  
 12012 whether to accept or reject the candidate values for  $\theta$  becomes increasingly complex or  
 12013 inefficient as the number of parameters in a model increases. As you already saw in  
 12014 Chapt. 3, in these cases you can simply combine MH sampling and Gibbs sampling.  
 12015 You can use Gibbs sampling to break down your high-dimensional parameter space into  
 12016 easy-to-handle one-dimensional conditional distributions and use MH sampling for these  
 12017 conditional distributions. Better yet, if you have some conjugacy in your model, you can  
 12018 use the more efficient Gibbs sampling for these parameters and one-dimensional MH for all  
 12019 the others. You have already seen the basics of how to build both types of algorithms, so  
 12020 we can jump straight into an example here and build a Metropolis-within-Gibbs algorithm.

12021 **GLMMs: Poisson regression with a random effect** Let's assume a model that gets  
 12022 us closer to the problem we ultimately want to deal with - a GLMM. Here, we assume  
 12023 we have Poisson counts,  $y_{ij}$ , from  $j = 1, 2, \dots, n$  plots in  $i$  different study sites, and we  
 12024 believe that the counts are influenced by some plot-specific covariate,  $\mathbf{x}$ , but that there is  
 12025 also a random site effect. So our model is:

$$y_{ij} \sim \text{Poisson}(\lambda_{ij})$$

$$\lambda_{ij} = \exp(\alpha_i + \beta x_{ij})$$

12027 Let's use Normal priors on  $\alpha$  and  $\beta$ ,

$$\alpha_i \sim \text{Normal}(\mu_\alpha, \sigma_\alpha)$$

12028 and

$$\beta \sim \text{Normal}(\mu_\beta, \sigma_\beta)$$

12029 In this model, we do not specify  $\mu_\alpha$  and  $\sigma_\alpha$ , but instead, estimate them as well, so we  
 12030 have to specify hyperpriors for these parameters:

$$\mu_\alpha \sim \text{Normal}(\mu_0, \sigma_0)$$

$$\sigma_\alpha^2 \sim \text{Inverse-Gamma}(a_0, b_0)$$

12031 Note that for simplicity we assume that  $\beta$  is constant across the  $i$  study sites, and for  
 12032 analysis we would set  $\mu_\beta$  and  $\sigma_\beta$ . With the model completely specified, we can compile  
 12033 the full conditionals, breaking the multi-dimensional parameter space into one-dimensional  
 12034 components:

$$\begin{aligned} [\alpha_1 | \alpha_2, \alpha_3, \dots, \alpha_i, \beta, \mathbf{y}_1] &\propto [\mathbf{y}_1 | \alpha_1, \beta][\alpha_1] \\ &\propto \text{Poisson}(\mathbf{y}_1 | \exp(\alpha_1 + \beta \mathbf{x}_1)) \times \text{Normal}(\alpha_1 | \mu_\alpha, \sigma_\alpha) \end{aligned}$$

12035 where  $\mathbf{y}_1 = (y_{11}, y_{12}, \dots, y_{1n})$  is the vector of observed counts for site  $i = 1$  and, in general,  
 12036  $\mathbf{y}_i$  is the vector of all counts for site  $i$ ; analogous,  $\mathbf{x}_i$  is the vector of all observations of  
 12037 the covariate for site  $i$ . The other full conditionals for each  $\alpha_i$  are constructed similarly:

$$\begin{aligned} [\alpha_2 | \alpha_1, \alpha_3, \dots, \alpha_i, \beta, \mathbf{y}_2] &\propto [\mathbf{y}_2 | \alpha_2, \beta][\alpha_2] \\ &\propto \text{Poisson}(\mathbf{y}_2 | \exp(\alpha_2 + \beta \mathbf{x}_2)) \times \text{Normal}(\alpha_2 | \mu_\alpha, \sigma_\alpha) \end{aligned}$$

12038 and so on for all elements of  $\alpha$ . The full-conditional for  $\beta$  is:

$$\begin{aligned} [\beta | \alpha, \mathbf{y}] &\propto [\mathbf{y} | \alpha, \beta][\beta] \\ &\propto \text{Poisson}(\mathbf{y} | \exp(\alpha + \beta \mathbf{x})) \times \text{Normal}(\beta | \mu_\beta, \sigma_\beta) \end{aligned}$$

12039 Finally, we need to update the hyperparameters for the random effects vector  $\alpha$ :

$$\begin{aligned} [\mu_\alpha | \alpha] &\propto [\alpha | \mu_\alpha, \sigma_\alpha][\mu_\alpha] \\ 12040 \quad [\sigma_\alpha | \alpha] &\propto [\alpha | \mu_\alpha, \sigma_\alpha][\sigma_\alpha] \end{aligned}$$

12041 Since we assumed  $\alpha$  to come from a Normal distribution, the choice of priors for  $\mu_\alpha$   
 12042 (Normal) and  $\sigma_\alpha^2$  (Inverse-Gamma) leads to the same conjugacy we observed in our initial  
 12043 Normal model, so that both hyperparameters can be updated using Gibbs sampling.

12044 Now let's build the updating steps for these full conditionals. Again, for the MH steps  
 12045 that update  $\alpha$  and  $\beta$  we use Normal proposal distributions with standard deviations  $\delta_\alpha$   
 12046 and  $\delta_\beta$ .

12047 First, we set the initial values  $\alpha^{(0)}$  and  $\beta^{(0)}$ . Then, starting with  $\alpha_1$ , we draw  $\alpha_1^{(1)}$  from  
 12048  $\text{Norm}(\alpha_1^{(0)}, \delta_\alpha)$ , calculate the conditional posterior density of  $\alpha_1^{(0)}$  and  $\alpha_1^{(1)}$  and compare  
 12049 their ratios,

$$r = \frac{\text{Poisson}(\mathbf{y}_1 | \exp(\alpha_1^{(1)} + \beta \mathbf{x}_1)) \times \text{Normal}(\alpha_1^{(1)} | \mu_\alpha, \sigma_\alpha)}{\text{Poisson}(\mathbf{y}_1 | \exp(\alpha_1^{(0)} + \beta \mathbf{x}_1)) \times \text{Normal}(\alpha_1^{(0)} | \mu_\alpha, \sigma_\alpha)}$$

12050 and accept  $\alpha_1^{(1)}$  with probability  $\min(r, 1)$ . We repeat this for all  $\alpha$ .

12051 For  $\beta$ , we draw  $\beta^{(1)}$  from  $\text{Norm}(\beta^{(0)}, \delta_\beta)$ , compare the posterior densities of  $\beta^{(0)}$  and  
 12052  $\beta^{(1)}$ ,

$$r = \frac{\text{Poisson}(\mathbf{y} | \exp(\alpha + \beta^{(1)} \mathbf{x})) \times \text{Normal}(\beta^{(1)} | \mu_\beta, \sigma_\beta)}{\text{Poisson}(\mathbf{y} | \exp(\alpha + \beta^{(0)} \mathbf{x})) \times \text{Normal}(\beta^{(0)} | \mu_\beta, \sigma_\beta)},$$

12053 and accept  $\beta^{(1)}$  with probability  $\min(r, 1)$ .

12054 For  $\mu_\alpha$  and  $\sigma_\alpha^2$ , we sample directly from the full conditional distributions (Eq. 14.1.2  
 12055 and Eq. 14.2.1):

$$\mu_\alpha^{(1)} \sim \text{Norm}(\mu_n, \sigma_n^2)$$

12056 where

$$\mu_n = \frac{\sigma_\alpha^{2(0)}}{\sigma_\alpha^{2(0)} + n_\alpha \sigma_0^2} \times \mu_0 + \frac{n_\alpha \sigma_0^2}{\sigma_\alpha^{2(0)} + n_\alpha \sigma_0^2} \times \bar{\alpha}^{(1)}$$

12057 and

$$\sigma_n^2 = \frac{\sigma_\alpha^{2(0)}\sigma_0}{\sigma_\alpha^{2(0)} + n\sigma_0^2}$$

12058 Here,  $\bar{\alpha}$  is the current mean of the vector  $\alpha$ , which we updated before, and  $n_\alpha$  is the  
 12059 length of  $\alpha$ . For  $\sigma_\alpha^2$  we use  $\sigma_\alpha^{2(1)} \sim \text{Inverse-Gamma}(a_n, b_n)$ , where  $a_n = n_\alpha/2 + a_0$ , and  
 12060  $b_n = 0.5 \sum_{i=1}^{n_\alpha} (\alpha_i^{(1)} - \mu_\alpha^{(1)})^2 + b_0$ .

12061 We repeat these steps over  $T$  iterations of the MCMC algorithm. Call the function  
 12062 `PoisGLMM()` in `scrbook` to check out what this algorithm looks like in **R**.

12063 In this example we may not want to save each individual  $\alpha$ , but are only interested in  
 12064 their mean and standard deviation. Since these two parameters will change as soon as the  
 12065 value for one element in  $\alpha$  changes, their acceptance rates will always be close to 1 and  
 12066 are not representative of how well your algorithm performs. To monitor the acceptance  
 12067 rates of parameters you do not want to save, you simply need to add a few lines of code  
 12068 into your updater to see how often the individual parameters are accepted. The code for  
 12069 updating  $\alpha$  from our Poisson GLMM below shows one way how to monitor acceptance of  
 12070 individual  $\alpha$ 's.

```

12071 #initiate counter for acceptance rate of alpha
12072 alphaUps<-0
12073
12074 #loop over sites, update intercepts alpha one at a time;
12075 #only data at site i contributes information
12076 #lev is the number of sites i
12077 for (i in 1:lev) {
12078   alpha.cand<-rnorm(1, alpha[i], delta_alpha)
12079   loglike<- sum(dpois (y[site==i], exp(alpha[i] + beta*x[site==i]),
12080     log=TRUE))
12081   logprior<- dnorm(alpha[i], mu_alpha,sig_alpha, log=TRUE)
12082   loglike.cand<- sum(dpois (y[site==i], exp(alpha.cand + beta *x[site==i])),
12083     log=TRUE))
12084   logprior.cand<- dnorm(alpha.cand, mu_alpha,sig_alpha, log=TRUE)
12085   if (runif(1)< exp((loglike.cand+logprior.cand) -(loglike+logprior))) {
12086     alpha[i]<-alpha.cand
12087     alphaUps<-alphaUps+1
12088   }
12089 }
12090
12091 #lets you check the acceptance rate of alpha at every 100th iteration
12092 if(iter %% 100 == 0) {
12093   cat("    Acceptance rates\n")
12094   cat("      alpha =", alphaUps/lev, "\n")
12095 }
```

---

**12096 14.2.4 Rejection sampling and slice sampling**

12097 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  
 12098 may be more efficient when applicable. **WinBUGS** applies these algorithms and we want  
 12100 you to be aware that there is more out there to approximate posterior distributions than  
 12101 Gibbs and MH. One alternative algorithm is rejection sampling. Rejection sampling is  
 12102 not an MCMC method, since each draw is independent of the others. The method can  
 12103 be used when the posterior  $[\theta|y]$  is not a known parametric distribution but can be ex-  
 12104 pressed in closed form. Then, we can use a so-called envelope function, say,  $g(\theta)$ , that  
 12105 we can easily sample from, with the restriction that  $[\theta|y] < M \times g(\theta)$ . We then sample a  
 12106 candidate value for  $\theta$  from  $g(\theta)$ , calculate  $r = [\theta|y]/M \times g(\theta)$  and keep the sample with  
 12107 the probability  $r$ .  $M$  is a constant that has to be picked so that  $r$  lies between 0 and 1, for  
 12108 example by evaluating both  $[\theta|y]$  and  $g(\theta)$  at  $n$  points and looking at their ratios. Rejec-  
 12109 tion sampling only works well if  $g(\theta)$  is similar to  $[\theta|y]$ , and packages like **WinBUGS** use  
 12110 adaptive rejection sampling (Gilks and Wild, 1992), where a complex algorithm is used to  
 12111 fit an adequate and efficient  $g(\theta)$  based on the first few draws. Though efficient in some  
 12112 situations, rejection sampling does not work well with high-dimensional problems, since  
 12113 it becomes increasingly hard to define a reasonable envelope function. For an example  
 12114 of rejection sampling in the context of SCR models, see Chapt. 13, where we use it to  
 12115 simulation non-stationary point processes.

12116 Another alternative is slice sampling (Neal, 2003). In slice sampling, we sample uni-  
 12117 formly from the area under the plot of  $[\theta|y]$ . Considering a single univariate  $\theta$ . Let's define  
 12118 an auxiliary variable,  $U \sim \text{Unif}(0, [\theta|y])$ . Then,  $\theta$  can be sampled from the vertical slice  
 12119 of  $[\theta|y]$  at  $U$  (Fig. 14.4):

$$\theta|U \sim \text{Unif}(B),$$

12120 where  $B = \{\theta : [\theta|y] \geq U\}$

12121 Slice sampling can be applied in many situations; however, implementing an efficient  
 12122 slice sampling procedure can be complicated. We refer the interested reader to Robert and  
 12123 Casella (2010, Chapt. 7) for a simple example. Both rejection sampling and slice sampling  
 12124 can be applied on one-dimensional conditional distributions within a Gibbs sampling setup.

### 14.3 MCMC FOR CLOSED CAPTURE-RECAPTURE MODEL MH

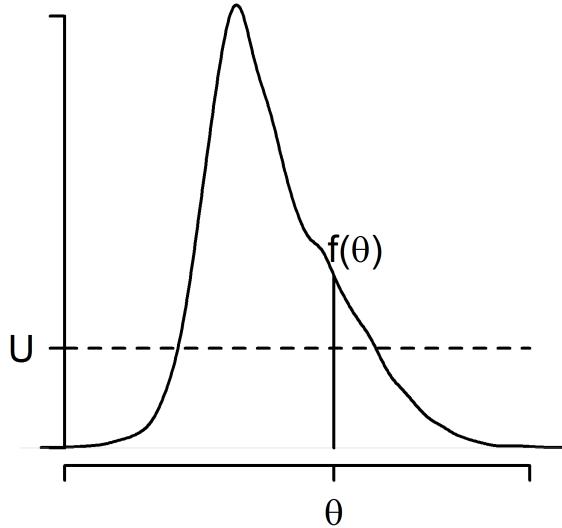
12125 By now you have seen MCMC samplers for some simple GL(M)M's. Now, to ease you  
 12126 into more complex models, we construct our own MCMC algorithm using a Metropolis-  
 12127 within-Gibbs sampler for the non-spatial model with individual heterogeneity in capture  
 12128 probability, model  $M_h$ , developed in Chapt. 4.

12129 To recapitulate: Under the non-spatial model, each of the  $n$  observed individuals is  
 12130 either detected (1) or not (0) during each of  $K$  sampling occasions. We estimate  $N$  using  
 12131 data augmentation and have a Bernoulli model for the data augmentation variables  $z_i$ .

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

12132 The binomial observation model is expressed conditional on the latent variables  $z_i$ .

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



**Figure 14.4.** 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  $B = \{\theta : [\theta|y] \geq U\}$ .

12133 Further, we prescribe a distribution for the capture probability  $p_i$ . Here we assume

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

12134 As usual, we have to go through two general steps before we write the MCMC algo-  
12135 rithm:

- 12136 (1) Identify the model with all its components (including priors)  
12137 (2) Recognize and express the full conditional distributions for all parameters

12138 Our model components are as follows:  $[y_i|p_i, z_i]$ ,  $[p_i|\mu_p, \sigma_p]$ , and  $[z_i|\psi]$  for each  $i =$   
12139  $1, 2, \dots, M$  and then prior distributions  $[\mu_p]$ ,  $[\sigma_p]$  and  $[\psi]$ . The joint posterior distri-  
12140 bution of all unknown quantities in the model is proportional to the joint distribution of  
12141 all elements  $y_i, p_i, z_i$  and also the prior distributions of the prior parameters:

$$\left\{ \prod_{i=1}^M [y_i|p_i, z_i] [p_i|\mu_p, \sigma_p] [z_i|\psi] \right\} [\mu_p, \sigma_p, \psi]$$

12142 For prior distributions, we assume that  $\mu_p, \sigma_p, \psi$  are mutually independent and for  $\mu_p$  and  
12143  $\sigma_p$  we use improper uniform priors, and  $\psi \sim \text{Unif}(0, 1)$ . This is equivalent to Beta(1, 1),  
12144 which will come in handy, as we will see in a moment. Note that the likelihood contribution  
12145 for each individual, when conditioned on  $p_i$  and  $z_i$ , does not depend on  $\psi$ ,  $\mu_p$ , or  $\sigma_p$ . As

such, the full-conditional for the structural parameter  $\psi$  only depend on the collection of data augmentation variables  $z_i$ , and that for  $\mu_p$  and  $\sigma_p$  will only depend on the collection of latent variables  $p_i; i = 1, 2, \dots, M$ . The full conditionals for all the unknowns are as follows:

(1) For  $p_i$ :

$$\begin{aligned} [p_i|y_i, \mu_p, \sigma_p, z_i] &\propto [y_i|p_i][p_i|\mu_p, \sigma_p^2] \text{ if } z_i = 1 \\ &\quad [p_i|\mu_p, \sigma_p] \text{ if } z_i = 0 \end{aligned}$$

(2) for  $z_i$ :

$$[z_i|y_i, p_i, \psi] \propto [y_i|z_i \times p_i]\text{Bernoulli}(z_i|\psi)$$

(3) For  $\mu_p$ :

$$[\mu_p|p_i, \sigma_p] \sim \left\{ \prod_i [p_i|\mu_p, \sigma_p] \right\} \times \text{const}$$

(4) For  $\sigma_p$ :

$$[\sigma_p|p_i, \mu_p] \sim \left\{ \prod_i [p_i|\mu_p, \sigma_p] \right\} \times \text{const}$$

(5) For  $\psi$ :

$$[\psi|z_i] \propto \left\{ \prod_i [z_i|\psi] \right\} \text{Beta}(1, 1)$$

Remember that Beta(1,1) is equivalent to Uniform(0,1). The beta distribution is the conjugate prior to the binomial and Bernoulli distributions and the general form of a full conditional of a beta-binomial model with  $x_i \sim \text{Bernoulli}(p)$  and  $p \sim \text{Beta}(a, b)$  is

$$[p|\mathbf{x}] \propto \text{Beta}(a + \sum_i x_i, b + n - \sum_i x_i)$$

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 term in the full conditionals for  $\mu_p$  and  $\sigma_p$  reflects the improper prior we chose for both parameters). 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.

```
12159  ### update the logit(p) parameters
12160  lp.cand<- rnorm(M,lp,1) # 1 is a tuning parameter
12161  p.cand<-plogis(lp.cand)
12162  ll<-dbinom(ytot,K,z*p, log=T)
12163  prior<-dnorm(lp,mu,sigma, log=T)
12170  llcand<-dbinom(ytot,K,z*p.cand, log=T)
12171  prior.cand<-dnorm(lp.cand,mu,sigma, log=T)
```

```

12172
12173 kp<- runif(M) < exp((llcand+prior.cand)-(ll+prior))
12174 p[kp]<-p.cand[kp]
12175 lp[kp]<-lp.cand[kp]

```

12176     The parameters  $\mu_p$  and  $\sigma_p$  are also updated using MH steps (see the code for  $\mu_p$   
12177 below). In truth, we could also sample  $\mu_p$  and  $\sigma_p^2$  directly with certain choices of prior  
12178 distributions. For example, if  $\mu_p \sim \text{Normal}(0, 1000)$  then the full conditional for  $\mu_p$  is also  
12179 Normal (see sec. 14.2.1), etc..

```

12180 p0.cand<- rnorm(1,p0,.05)
12181 if(p0.cand>0 & p0.cand<1){
12182 mu.cand<-log(p0.cand/(1-p0.cand))
12183 ll<-sum(dnorm(lp,mu,sigma,log=TRUE))
12184 llcand<-sum(dnorm(lp,mu.cand,sigma,log=TRUE))
12185 if(runif(1)<exp(llcand-ll)) {
12186 mu<-mu.cand
12187 p0<-p0.cand
12188 }
12189 }

```

12190     For  $\psi$  we can easily sample directly from the beta distribution:

```

12191 psi<-rbeta(1, sum(z) + 1, M-sum(z) + 1)

```

12192     To update the  $z_i$  we have opted for a MH updater (although they could be updated  
12193 directly from their full-conditional). Since  $z_i$  can only take the values of 0 or 1, we generate  
12194 candidate values using `z.cand<-ifelse(z==1,0,1)`. You can check out the full code by  
12195 invoking `modelMh()` from the R package `scrbook`.

#### 14.4 MCMC ALGORITHM FOR MODEL SCR0

12196 Conceptually, but also in terms of MCMC coding, it is only a small step from the non-  
12197 spatial model  $M_h$  to a fully spatial capture-recapture model. Next, we'll walk you through  
12198 the steps of building your own MCMC sampler for the basic SCR model (i.e. without any  
12199 individual, site or time specific covariates) with both a Poisson and a Binomial encounter  
12200 process. As usual, we will have to go through two general steps before we write the MCMC  
12201 algorithm:

- 12202 (1) Identify the model with all its components (including priors)
- 12203 (2) Recognize and express the full conditional distributions for all parameters

12204 It is worthwhile to go through all of step 1 for an SCR model, but you have probably  
12205 seen enough of step 2 in our previous examples to get the essence of how to express a full  
12206 conditional distribution. Therefore, we will exemplify step 2 for some parameters and tie  
12207 these examples directly to the respective R code.

12208     **Step 1 – Identify your model**

12209     Recall the components of the basic SCR model with a Poisson encounter process from  
 12210    Chapt. 9: We assume that individuals  $i$ , or rather, their activity centers  $\mathbf{s}_i$ , are uniformly  
 12211    distributed across the state space  $\mathcal{S}$ ,

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

12212    and that the number of times individual  $i$  encounters trap  $j$ ,  $y_{ij}$ , is a Poisson variable  
 12213    with mean  $\lambda_{ij}$ ,

$$y_{ij} \sim \text{Poisson}(\lambda_{ij})$$

12214    The link between individual location, movement and trap encounter rates is made by the  
 12215    assumption that  $\lambda_{ij}$ , is a decreasing function of the distance between  $\mathbf{s}_i$  and the location  
 12216    of  $j$ ,  $\mathbf{x}_j$ , say  $d_{ij} = \|\mathbf{s}_i - \mathbf{x}_j\|$ , of the half-normal form

$$\lambda_{ij} = \lambda_0 \exp(-d_{ij}^2/2\sigma^2),$$

12217    where  $\lambda_0$  is the baseline trap encounter rate at  $d_{ij} = 0$  and  $\sigma$  controls the shape of the  
 12218    half-normal function.

12219    In order to estimate the number of  $\mathbf{s}_i$  in  $\mathcal{S}$  (or any subset of  $\mathcal{S}$ ),  $N$ , we use data  
 12220    augmentation (sec. 4.2) and create  $M - n$  all-zero encounter histories, where  $n$  is the  
 12221    number of individuals we observed and  $M$  is a somewhat arbitrary number that is larger  
 12222    than  $N$ . We estimate  $N$  by summing over the auxiliary data augmentation variables,  $z_i$ ,  
 12223    which is 1 if the individual is part of the population and 0 if not, and assume that  $z_i$  is a  
 12224    Bernoulli random variable,

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

12225    To link the two model components, we modify our trap encounter model to

$$\lambda_{ij} = \lambda_0 \times \exp(-d_{ij}^2/2\sigma^2) \times z_i.$$

12226    The model has the following structural parameters, for which we need to specify priors:

12227     $\psi$ : the  $\text{Uniform}(0, 1)$  is required as part of the data augmentation procedure and in general  
 12228    is a natural choice of an uninformative prior for a probability. It will also lead to  
 12229    conjugacy as we saw in the example of model  $M_h$ , so that we can update  $\psi$  directly  
 12230    from its full conditional distribution using Gibbs sampling.

12231     $\mathbf{s}_i$ : since  $\mathbf{s}_i$  is a pair of coordinates it is two-dimensional and we use a uniform prior  
 12232    limited by the extent of our state-space over both dimensions.

12233     $\sigma$ : we can conceive several priors for  $\sigma$  but let's assume an improper prior, one that is  
 12234    Uniform over  $(-\infty, \infty)$ . We will see why this is convenient when we construct the full  
 12235    conditionals for  $\sigma$ .

12236     $\lambda_0$ : analogous, we will use a  $\text{Uniform}(-\infty, \infty)$  improper prior for  $\lambda_0$ .

12237    The parameter that is the objective of our modeling,  $N$ , is a derived parameter that we  
 12238    can obtain by summing all  $z_i$ :

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

12239    **Step 2 – Construct the full conditionals:** Having completed step 1, let's look at  
 12240    the full conditional distributions for some of these parameters. We find that with improper

12241 priors, full conditionals are proportional only to the likelihood of the observations; for  
 12242 example, consider  $\sigma$ :

$$[\sigma|\mathbf{s}, \lambda_0, \mathbf{z}, \mathbf{y}] \propto \left\{ \prod_i [y_i|\mathbf{s}_i, \lambda_0, z_i, \sigma] \right\} [\sigma]$$

12243 Since the improper prior implies that  $[\sigma] \propto 1$ , we can reduce this further to

$$[\sigma|\mathbf{s}, \lambda_0, \mathbf{z}, \mathbf{y}] \propto \left\{ \prod_i [y_i|\mathbf{s}_i, \lambda_0, z_i, \sigma] \right\}$$

12244 The R code to update  $\sigma$  is shown below. Notice that we automatically reject negative  
 12245 candidate values, since  $\sigma$  cannot be  $< 0$ .

```
12246 sig.cand <- rnorm(1, sigma, 0.1) #draw candidate value
12247 if(sig.cand>0){ #automatically reject sig.cand that are <0
12248   lam.cand <- lam0*exp(-(d*d)/(2*sig.cand*sig.cand))
12249   ll<- sum(dpois(y, lam*z, log=TRUE))
12250   llcand <- sum(dpois(y, lam.cand*z, log=TRUE))
12251   if(runif(1) < exp( llcand - ll ) ){
12252     ll<-llcand
12253     lam<-lam.cand
12254     sigma<-sig.cand
12255   }
12256 }
```

12257 These steps are analogous for  $\lambda_0$  and  $\mathbf{s}_i$  and we will use MH steps for all of these  
 12258 parameters. Similar to the random intercepts in our Poisson GLMM, we update each  
 12259  $\mathbf{s}_i$  individually. Note that to be fully correct, the full conditional for  $\mathbf{s}_i$  contains both  
 12260 the likelihood and prior component, since we did not specify an improper, but a proper  
 12261 Uniform prior on  $\mathbf{s}_i$ . However, with a Uniform distribution the probability density of  
 12262 any value is  $1/(\text{upper limit} - \text{lower limit}) = \text{constant}$ . Thus, the prior components are  
 12263 identical for both the current and the candidate value and can be ignored (formally, when  
 12264 you calculate the ratio of posterior densities,  $r$ , the identical prior component appears  
 12265 both in the numerator and denominator, so that they cancel each other out).

12266 We still have to update  $z_i$ . The full conditional for  $z_i$  is

$$[z_i|y_i, \sigma, \lambda_0, \mathbf{s}_i] \propto [y_i|z_i, \sigma, \lambda_0, \mathbf{s}_i][z_i]$$

12267 and since  $z_i \sim \text{Bern}(\psi)$ , the term has to be taken into account when updating  $z_i$ :

```
12268 zUps <- 0 #set counter to monitor acceptance rate
12269 for(i in 1:M) {
12270   #no need to update seen individuals, since their z =1
12271   if(seen[i])
12272     next
12273   zcand <- ifelse(z[i]==0, 1, 0)
12274   llz <- sum(dpois(y[i,,],lam[i,]*z[i], log=TRUE))
```

---

```

12275     llcand <- sum(dpois(y[i,], lam[i,]*zcand, log=TRUE))
12276
12277     prior <- dbinom(z[i], 1, psi, log=TRUE)
12278     prior.cand <- dbinom(zcand, 1, psi, log=TRUE)
12279     if(runif(1) < exp((llcand+prior.cand)-(llz+prior))){ 
12280         z[i] <- zcand
12281         zUps <- zUps+1
12282     }
12283 }
```

12284 The parameter  $\psi$  is a hyperparameter of the model, with an uninformative prior distribution  
 12285 of Uniform(0, 1) or Beta(1, 1), so that

$$[\psi|\mathbf{z}] \propto \text{Beta}\left(1 + \sum_i z_i, 1 + M - \sum_i z_i\right)$$

12286 These are all the building blocks you need to write the MCMC algorithm for the spatial  
 12287 null model with a Poisson encounter process. You can find the full **R** code by calling the  
 12288 function (**SCR0pois**) in the **R** package **scrbook**.

#### 12289 14.4.1 SCR model with binomial encounter process

12290 The equivalent SCR model with a binomial encounter process is very similar. Here, each  
 12291 individual  $i$  can only be detected once at any given trap  $j$  during a sampling occasion  $k$ .  
 12292 Thus

$$y_{ij} \sim \text{Binomial}(p_{ij}, K)$$

12293 Where  $p_{ij}$  is some function of distance between  $\mathbf{s}_i$  and trap location  $\mathbf{x}_j$ . Here we use:

$$p_{ij} = 1 - \exp(-\lambda_{ij})$$

12294 Recall from Chapt. 3 that this is the complementary log-log (cloglog) link function,  
 12295 which constrains  $p_{ij}$  to fall between 0 and 1. For our MCMC algorithm that means that,  
 12296 instead of using a Poisson likelihood,  $\text{Poisson}(y|\sigma, \lambda_0, \mathbf{s}, z)$ , we use a Binomial likelihood,  
 12297  $\text{Binomial}(y|\sigma, \lambda_0, \mathbf{s}, z; K)$ , in all the conditional distributions. An exemplary updating  
 12298 step for  $\lambda_0$  under a Binomial encounter model is shown below. The full MCMC code for  
 12299 the Binomial SCR with a clog-log link (**SCR0binom.cl**) can be found in the **R** package  
 12300 **scrbook**.

```

12301     lam0.cand <- rnorm(1, lam0, 0.1)
12302     #automatically reject lam0.cand that are <0
12303     if(lam0.cand >0){
12304         lam.cand <- lam0.cand*exp(-(d*d)/(2*sigma*sigma))
12305         p.cand <- 1-exp(-lam.cand)
12306         ll<- sum(dbinom(y, K, pmat *z, log=TRUE))
12307         llcand <- sum(dbinom(y, K, p.cand *z, log=TRUE))
12308         if(runif(1) < exp( llcand - ll) ){
12309             ll<-llcand
```

```

12310      pmat<-p.cand
12311      lam0<- lam0.cand
12312    }
12313  }

```

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:

```

12320  p0.cand <- rnorm(1, p0, 0.1)
12321  if(p0.cand > 0 & p0.cand < 1 ){
12322    #automatically rejects lam0.cand that are not {0,1}
12323    p.cand <- p0.cand*exp(-(d*d)/(2*sigma*sigma))
12324    ll<- sum(dbinom(y, K, pmat *z, log=TRUE))
12325    llcand <- sum(dbinom(y, K, p.cand *z, log=TRUE))
12326    if(runif(1) < exp( llcand - ll ) ){
12327      ll<-llcand
12328      pmat<-p.cand
12329      p0<- p0.cand
12330    }
12331  }

```

#### 14.4.2 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 Chapt. 4 with traditional non-spatial models (and that you will see again in Chapt. 8). 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:

```

12340 M=700
12341 trapmat<-beardata$trapmat
12342 #summarizes captures across occasions
12343 bearmat<-apply(beardata$bearArray, 1:2, sum)
12344 Xaug<-matrix(0, nrow=M, ncol=dim(trapmat)[1])
12345 Xaug[1:dim(bearmat)[1],]<-bearmat #create augmented data set

```

In addition to these data, we need to specify the outermost coordinates of the state-space. Since bears are wide ranging animals we add a 20-km buffer to the maximum and minimum coordinates of the trap array:

---

```

12349 xl<- min(trapmat[,1])- 20
12350 yl<- min(trapmat[,2])- 20
12351 xu<- max(trapmat[,1])+ 20
12352 yu<- max(trapmat[,2])+ 20

```

12353 Finally, use the MCMC code for the binomial encounter model with the clog-log link  
 12354 (`SCR0binom.cl`) and run 5000 iterations. This should take approximately 25 minutes (in  
 12355 real life we would of course run the algorithm a lot longer but for demonstration purposes  
 12356 let's stick with a number of iterations that can be run in a manageable amount of time).

```

12357 set.seed(13)
12358 mod0<-SCR0binom.cl(y=Xaug, X=trapmat, M=M, xl=xl, xu=xu, yl=yl,
12359           yu=yu, K=8, delta=c(0.1, 0.05, 2), niter=5000)

```

12360 Before, we used simple **R** commands to look at model results. However, there is a  
 12361 specific **R** package to summarize MCMC simulation output and perform some convergence  
 12362 diagnostics – package `coda` (Plummer et al., 2006). Download and install `coda`, then  
 12363 convert your model output to an `mcmc` object

```

12364 chain<-mcmc(mod0)

```

12365 which can be used by `coda` to produce MCMC specific output.

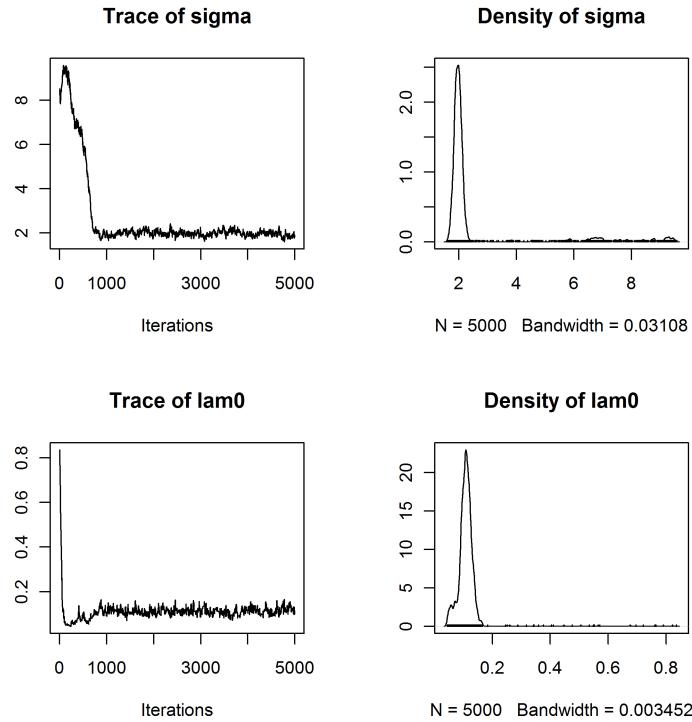
### 12366 **Markov chain time series plots**

12367 Start by looking at time series plots of your Markov chains using `plot(chain)`. This com-  
 12368 mand produces a time series plot and marginal posterior density plots for each monitored  
 12369 parameter, similar to what we did before using the `hist()` and `plot()` commands. Fig.  
 12370 14.5 shows an example of these plots for  $\sigma$  and  $\lambda_0$ . Time series plots will tell you several  
 12371 things: First, recall from sec. 14.2.2 that the way the chains move through the parameter  
 12372 space gives you an idea of whether your MH steps are well tuned. If chains were constant  
 12373 over many iterations you would need to decrease the tuning parameter of the (Normal)  
 12374 proposal distribution. If a chain moves along some gradient to a stationary state very  
 12375 slowly, you may want to increase the tuning parameter so that the parameter space is  
 12376 explored more efficiently.

12377 Second, you will be able to see if your chains converged and how many initial sim-  
 12378 ulations you have to discard as burn-in. In the case of the chains shown in Fig. 14.5,  
 12379 we would probably consider the first 750 – 1000 iterations as burn-in, as afterwards the  
 12380 chains seem to be fairly stationary.

#### 12381 **14.4.3 Posterior density plots**

12382 The `plot()` command also produces posterior density plots and it is worthwhile to look  
 12383 at those carefully. For parameters with priors that have bounds (e.g. Uniform over some  
 12384 interval), you will be able to see if your choice of the prior is truncating the posterior  
 12385 distribution. In the context of SCR models, this will mostly involve our choice of  $M$ , the  
 12386 size of the augmented data set. If the posterior of  $N$  has a lot of mass concentrated close to  
 12387  $M$  (or equivalently the posterior of  $\psi$  has a lot of mass concentrated close to 1), as in the  
 12388 example in Fig. 14.6, we have to re-run the analysis with a larger  $M$ . A diffuse posterior

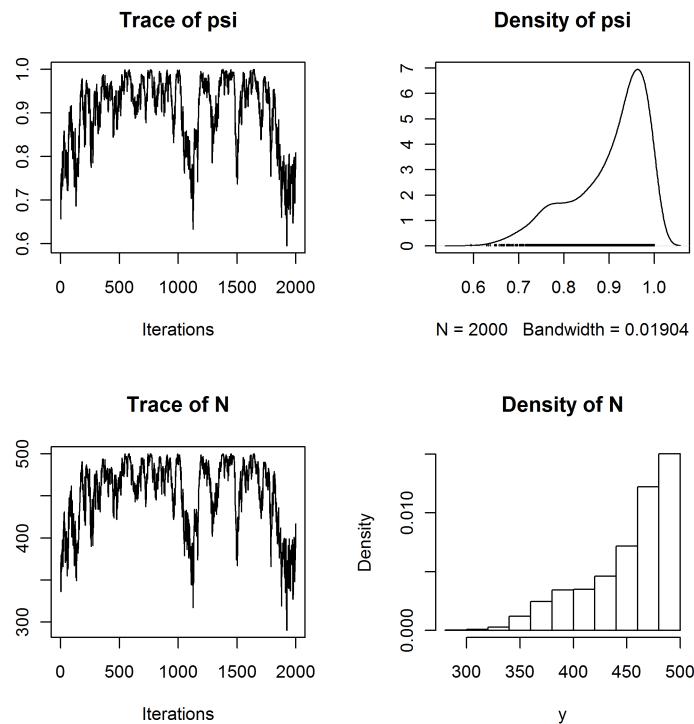


**Figure 14.5.** Time series and posterior density plots for  $\sigma$  and  $\lambda_0$  for the Fort Drum black bear data.

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.

#### 14.4.4 Serial autocorrelation and 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. If you use **WinBUGS** through the **R2WinBUGS** package, the **print()** command will automatically return the effective sample size for all monitored parameters. In the **coda** package there are several functions you can use to do so. The function **effectiveSize()**



**Figure 14.6.** 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).

12401 will directly give you an estimate of the effective sample size for the parameters:

```
12402 effectiveSize(window(chain, start=1001))
12403   sigma      lam0      psi      N
12404 93.89807 163.72311 51.96443 46.45394
```

12405 Alternatively, you can use the `autocorr.diag()` function, which will show you the  
12406 degree of autocorrelation for different lag values (which you can specify within the function  
12407 call, we use the defaults below):

```
12408 autocorr.diag(window(chain, start=1001))
12409   sigma      lam0      psi      N
12410 Lag 0  1.0000000 1.0000000 1.0000000 1.0000000
12411 Lag 1  0.9316928 0.91464875 0.9745833 0.9663320
12412 Lag 5  0.7603332 0.67445407 0.8525272 0.8500215
```

```

12413 Lag 10 0.6065374 0.48724122 0.7514657 0.7530124
12414 Lag 50 0.1122331 0.06564406 0.3811939 0.3823236

```

12415 In the present case we see that autocorrelation is especially high for the parameter  $\psi$  and  
 12416 our effective sample size for this parameter is only 52! This means we would have to run  
 12417 the model for much longer to obtain a reasonable effective sample size. Unfortunately,  
 12418 with many SCR data sets we observe high degrees of serial autocorrelation. For now, let's  
 12419 continue using this small number of samples to look at the output.

#### 12420 14.4.5 Summary results

12421 Now that we checked that our chains apparently have converged and pretending that  
 12422 we have generated enough samples from the posterior distribution, we can look at the  
 12423 actual parameter estimates. The `summary()` function will return two sets of results: the  
 12424 mean parameter estimates, with their standard deviation, the naïve standard error – i.e.  
 12425 your regular standard error calculated for  $T$  (= number of iterations) samples without  
 12426 accounting for serial autocorrelation – and the Time-series SE (in **WinBUGS** and earlier  
 12427 in this book referred to as MC error), which accounts for autocorrelation. Remember our  
 12428 rule of thumb that this error decreases with increasing chain length and should be 1% or  
 12429 less of the parameter estimate. In **WinBUGS** the MC error is only given in the log output  
 12430 within **BUGS** itself. You should adjust the `summary()` call by removing the burn-in from  
 12431 calculating parameter summary statistics. To do so, use the `window()` command, which  
 12432 lets you specify at which iteration to start 'counting'. In contrast to **WinBUGS**, which  
 12433 requires you to set the burn-in length before you run the model, this command gives us  
 12434 full flexibility to make decisions about the burn-in after we have seen the trajectories of  
 12435 our Markov chains. For our example, `summary(window(chain, start=1001))` returns the  
 12436 following output:

```

12437 Iterations = 1001:5000
12438 Thinning interval = 1
12439 Number of chains = 1
12440 Sample size per chain = 4000
12441
12442 1. Empirical mean and standard deviation for each variable,
12443 plus standard error of the mean:
12444
12445      Mean        SD  Naive SE Time-series SE
12446 sigma   1.9697  0.12534  0.0019818      0.012792
12447 lam0    0.1124  0.01521  0.0002405      0.001311
12448 psi     0.7295  0.11794  0.0018648      0.015278
12449 N       510.9190 81.99868 1.2965130     10.580567
12450
12451 2. Quantiles for each variable:
12452
12453      2.5%       25%       50%       75%     97.5%
12454 sigma   1.7288  1.8831  1.9666  2.0517  2.2240
12455 lam0    0.0863  0.1008  0.1112  0.1217  0.1449

```

---

```
12456 psi      0.5100  0.6423  0.7261  0.8170  0.9549
12457 N       359.0000 451.0000 508.0000 572.0000 668.0000
```

12458 Looking at the MC errors (column labeled **Time-series SE**), we see that in spite of the  
 12459 high autocorrelation, the MC error for  $\sigma$  is below the 1% threshold, whereas for all other  
 12460 parameters, MC errors are still above, another indication that for a thorough analysis we  
 12461 should run a longer chain.

12462 Our algorithm gives us a posterior distribution of  $N$ , but we are usually interested  
 12463 in the density,  $D$ . Density itself is not a parameter of our model, but we can derive a  
 12464 posterior distribution for  $D$  by dividing each value of  $N$  ( $N$  at each iteration) by the area  
 12465 of the state-space (here  $3032.719 \text{ km}^2$ ) and we can use summary statistics of the resulting  
 12466 distribution to characterize  $D$ :

```
12467 summary(window(chain[,4]/ 3032.719, start=1001))
12468
12469 Iterations = 1001:5000
12470 Thinning interval = 1
12471 Number of chains = 1
12472 Sample size per chain = 4000
12473
12474 1. Empirical mean and standard deviation for each variable,
12475 plus standard error of the mean:
12476
12477          Mean           SD        Naive SE Time-series SE
12478    0.1684690   0.0270380   0.0004275   0.0034888
12479
12480 2. Quantiles for each variable:
12481
12482    2.5%     25%     50%     75%   97.5%
12483 0.1184 0.1487 0.1675 0.1886 0.2203
```

12484 We see that our mean density of  $0.17/\text{km}^2$  is very similar to the estimate of  $0.18/\text{km}^2$   
 12485 obtained under the non-spatial model  $M_0$  in Chapt. 4.

#### 12486 14.4.6 Other useful commands

12487 While inspecting the time series plot gives you a first idea of how well you tuned your  
 12488 MH algorithm, use **rejectionRate()** to obtain the rejection rates (1 – acceptance rates)  
 12489 of the parameters that are written to your output:

```
12490 rejectionRate(chain)
12491      sigma      lam0      psi        N
12492 0.42988598 0.78775755 0.00000000 0.03160632
```

12493 Recall (sec. 14.2.2) that rejection rates should lie between 0.2 and 0.8, so our tuning  
 12494 seems to have been appropriate here. Draws of the parameter  $\psi$  are never rejected since  
 12495 we update it with Gibbs sampling, where all candidate values are kept. And since  $N$  is  
 12496 the sum of all  $z_i$ , all it takes for  $N$  to change from one iteration to the next are small

12497 changes in the z-vector, so the rejection rate of  $N$  is always low. If you have run several  
 12498 parallel chains, you can combine them into a single mcmc object using the `mcmc.list()`  
 12499 command on the individual chains (note that each chain has to be converted to an mcmc  
 12500 object before combining them with `mcmc.list()`). You can then easily obtain the Gelman-  
 12501 Rubin diagnostic (Gelman et al., 2004), in **WinBUGS** called Rhat, using `gelman.diag()`,  
 12502 which will indicate if all chains have converged to the same stationary distribution. For  
 12503 details on these and other functions, see the `coda` manual, which can be found (together  
 12504 with the package) on the CRAN mirror.

## 14.5 MANIPULATING THE STATE-SPACE

12505 So far, we have constrained the location of the activity centers to fall within the outermost  
 12506 coordinates of our rectangular state space by posing upper and lower bounds for  $x$  and  $y$ .  
 12507 But what if  $\mathcal{S}$  has an irregular shape – maybe there is a large water body we would like  
 12508 to remove from  $\mathcal{S}$ , because we know our terrestrial study species does not occur there. Or  
 12509 the study takes place in a clearly defined area such as an island.

12510 As mentioned before, this situation is difficult to handle in **WinBUGS**. In some  
 12511 simple cases we can adjust the state space by setting one of the coordinates of  $s_i$  to be  
 12512 some function of the other and reject candidate  $s_i$  that do not fall within this modified  
 12513 state space. In this manner, we can cut off corners of the rectangle to approximate the  
 12514 actual state space<sup>3</sup>. To visualize this approach, plot the following rectangle, representing  
 12515 your state space polygon, and line, representing, for example, the approximation of a shore  
 12516 line:

```
12517 xlim<-c(-5,5)
12518 ylim<-c(-7,7)
12519 plot(xlim, ylim, type='n')
12520 abline(a=4, b=0.4)
```

12521 The Y coordinates limiting your state space to the habitat that is suitable to the species  
 12522 you study can now be expressed as a linear function of the X coordinates, in this case,  
 12523  $Y = 4 + 0.4 \times X$ . To include this new limit in our **WinBUGS** model, we need to change  
 12524 the following:

```
12525 #draw SX and SY as before
12526 SX[i] ~ dunif(xlim[1],xlim[2])
12527 SY[i] ~ dunif(ylim[1],ylim[2])
12528 #calculate upper limit for Y given X
12529 ymax[i]<-4+0.4*SX[i]
12530 # use step function to see if location [SX, SY]
12531 # is below the Y limit (Pin = 1) or not (Pin = 0)
12532 Pin[i] <- step(ymax[i] - SY[i])
12533 In[i] ~ dbern(Pin[i])
```

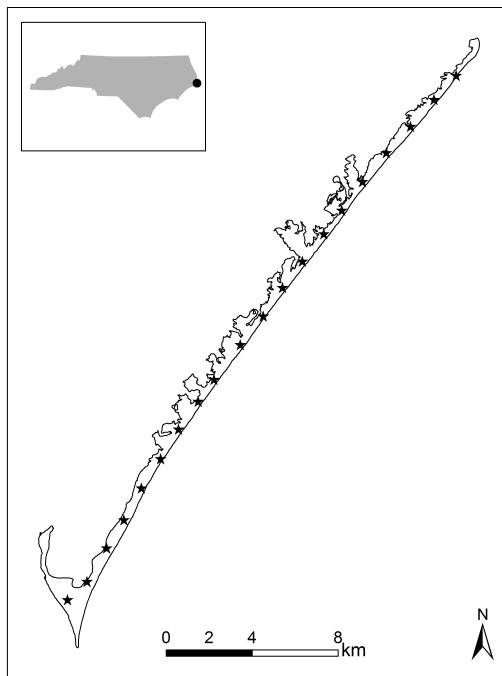
12534 In is a vector of  $M$  1's, passed as data to the model. If  $Pin = 0$ , the likelihood will be 0  
 12535 and the candidate [SX, SY] pair will be rejected. If  $Pin = 1$ , this bit of the likelihood is

---

<sup>3</sup>This idea was pitched to us by Mike Meredith, from WCS Malaysia

12536 equal to 1, and whether or not the candidate pair of coordinates is accepted depends  
 12537 only on capture history of  $i$ . This approach can be very useful in some situations but is  
 12538 clearly restricted by the functional form of the relationship between SX and SY that it  
 12539 requires.

12540 In **R**, we are much more flexible, as we can use the actual state-space polygon to  
 12541 constrain  $s_i$ . To illustrate that, let's look at a camera trapping study of raccoons (*Procyon*  
 12542 *lotor*) conducted on South Core Banks, a barrier island within Cape Lookout National  
 12543 Seashore, North Carolina (details of the study can be found in Sollmann et al. (2012b)  
 12544 and in Chapt. 19 where we present the analysis of this data set with spatial mark-resight  
 12545 models). Since camera-traps were spread across the entire length of the island, we set  
 12546 the state space to be delineated by the shore line of the island (Fig. 14.7), which clearly  
 12547 cannot easily be approximated as a rectangle. Instead, within **R** we can use an actual  
 12548 shapefile of the island.



**Figure 14.7.** 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).

12549 In other circumstances you may still want to create the state space as before, by adding  
 12550 some buffer to your trapping grid, but you may find that the resulting rectangle includes  
 12551 water bodies, paved parking lots or any other kind of habitat you know is never used by the

species you study. In order to precisely describe the state-space, these features need to be removed. You can create a precise state-space polygon in **ArcGIS** and read it into **R**, or create the polygon directly within **R**, by intersecting two shape files – one of the rectangle defining the outer limits of your state-space state and one of the landscape feature you want to remove. While you will most likely have to obtain the shapefile describing the landscape of and around your trapping grid (coastlines, water bodies etc.) from some external source, the polygon shapefile buffering your outermost trapping grid coordinates can easily be written in **R**.

If **xmin**, **xmax**, **ymin** and **ymax** mark the most extreme *x* and *y* coordinates of your trapping grid and *b* is the distance you want to buffer with, load the package **shapefiles** (Stabler, 2006) and issue the following **R** commands:

```
12563 xl= xmin-b
12564 xu= xmax+b
12565 yl= ymin-b
12566 yu= ymax+b
12567
12568         #create data frame with coordinate pairs
12569 dd <- data.frame(Id=c(1,1,1,1,1),X=c(xl,xu,xu,xl,xl),
12570 Y=c(yl,yl,yu,yu,yl))
12571 ddTable <- data.frame(Id=c(1),Name=c("Item1"))
12572         #convert to shapefile, type polygon
12573 ddShapefile <- convert.to.shapefile(dd, ddTable, "Id", 5)
12574         # name and save to location of choice
12575 write.shapefile(ddShapefile, 'c:/Test', arcgis=T)
```

You can read shapefiles into **R** loading the package **maptools** (Lewin-Koh et al., 2011) and using the function **readShapeSpatial()**. Make sure you read in shapefiles in UTM format, so that units of the trap array, the movement parameter  $\sigma$  and the state-space are all identical. Intersection of polygons can be done in **R** also, using the package **rgeos** (Bivand and Rundel, 2011) and the function **gIntersect()**. The area of your (single) polygon can be extracted directly from the state-space object **SSp**:

```
12582 area <- SSp@polygons[[1]]@Polygons[[1]]@area /1000000
```

Note that dividing by 1000000 will return the area in  $\text{km}^2$  if your coordinates describing the polygon are in UTM. If your state-space consists of several disjunct polygons, you will have to sum the areas of all polygons to obtain the size of the state-space. To include 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  $s_i$ :

---

<sup>4</sup>Remember from the previous chapter (6.4.1) 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.

```

12590     #draw candidate value
12591 Scand <- as.matrix(cbind(rnorm(M, S[,1], 2), rnorm(M, S[,2], 2)))
12592     #convert to spatial points on UTM (m) scale
12593 Scoord<-SpatialPoints(Scand*1000)
12594     # check if scand is within the polygon
12595 SinPoly<-over(Scoord,SSp)
12596
12597 for(i in 1:M) {
12598     #if scand falls within polygon, continue update
12599     if(is.na(SinPoly[i])==FALSE) {
12600 ... [rest of the updating step remains the same]

```

12601 Note that it is much more time-efficient to draw all  $M$  candidate values for  $s$  and check  
 12602 once if they fall within the state-space, rather than running the `over()` command for  
 12603 every individual pair of coordinates. To make sure that our initial values for  $s$  also fall  
 12604 within the polygon of  $\mathcal{S}$ , we use the function `runifpoint()` from the package `spatstat`  
 12605 (Baddeley and Turner, 2005), which generates random uniform points within a specified  
 12606 polygon. You'll find this modified MCMC algorithm (`SCR0poisSSp`) in the **R** package  
 12607 `scrbook`.

12608 Finally, observe that we are converting candidate coordinates of  $\mathcal{S}$  back to meters to  
 12609 match the UTM polygon. In all previous examples, for both the trap locations and the  
 12610 activity centers we have used UTM coordinates divided by 1000 to estimate  $\sigma$  on a km  
 12611 scale. This is adequate for wide ranging species like bears. In other cases you may center  
 12612 all coordinates on 0. No matter what kind of transformation you use on your coordinates,  
 12613 make sure to always convert candidate values for  $\mathcal{S}$  back to the original scale (UTM)  
 12614 before running the `over()` command.

## 14.6 INCREASING COMPUTATIONAL SPEED

12615 Using custom written MCMC algorithms in **R** is not only more flexible but can also be  
 12616 faster than using programs such as **JAGS** and especially **BUGS**. Also, **R** tends to use  
 12617 much less memory than **JAGS**, which can be crucial if you are running a large model  
 12618 but only have limited memory available. For example, you will see in Chapt. 8 that even  
 12619 with a reasonable sized data set certain parameterizations of SCR models can max out  
 12620 the memory of a 16 GB computer when using **JAGS**. These are mostly the models that  
 12621 require us to look at individual sampling occasions instead of joining observations for a  
 12622 given sampling location across the entire study, which requires us to introduce another  
 12623 for-loop into the **JAGS** model. **BUGS** is limited in the amount of memory it can access  
 12624 and thus will likely not max out your memory, but as a trade-off, it will take a long time  
 12625 to run such models. In this chapter we have provided you with the guidelines to write  
 12626 your own MCMC sampler. But beyond the material that we have covered there are a  
 12627 number of ways you can make your sampler more efficient, through parallel computing  
 12628 or by accessing an alternative computer language such as **c++**. Exploring these options  
 12629 exhaustively is beyond the scope of this book; instead, in this section we will give you  
 12630 some pointers to get started with these more advanced computational issues.

---

**12631 14.6.1 Parallel computing**

12632 If you are using a computer with several cores, you can make use of parallel computing to  
 12633 speed up overall computation. In parallel computing we execute commands simultaneously  
 12634 on different cores of the computer, instead of running them serially on one single core.  
 12635 For example, imagine you have 4 cores available and you want to implement a for-loop in  
 12636 **R**; instead of going through the loop iteration by iteration, you can prompt **R** to execute  
 12637 iterations 1 to 4 at the same time on the 4 different cores. The core that finishes first will  
 12638 then continue with iteration 5, and so on. There are several packages in **R** that allow you  
 12639 to induce parallel computing, such as **snow** (Tierney et al., 2011) and **snowfall** (Knaus,  
 12640 2010), and the more current versions of **R** (from 2.14.0 upwards) come with a pre-installed  
 12641 set of functions grouped under the name **parallel**.

12642 The MCMC algorithms developed here and in other parts of this book come with plenty  
 12643 of opportunities to parallelize computation. In various instances within the algorithm, we  
 12644 have for-loops across our augmented data set of size  $M$ , or we may have for-loops across  
 12645 sampling occasions. We also have for-loops across iterations of the algorithm, but since  
 12646 one iteration of the Markov chain depends on the preceding iteration these should always  
 12647 be run serially, not in parallel. There is another dimension we can think of, and that is  
 12648 running multiple chains of an algorithm to assess convergence. This is a comparatively  
 12649 easy implementation of parallel computing and thus provides a good starting point to  
 12650 understand how it works in **R**.

12651 Let's go back to the Ft. Drum black bear data we analyzed above with the cloglog  
 12652 version of the binomial SCR model (sec. 14.4.2) and run 3 parallel chains using **snowfall**.  
 12653 All we need to do is wrap our function **SCR0binom.cl** within another function that can  
 12654 then be executed in parallel, returning a list with one output matrix for each chain (install  
 12655 **snowfall** before executing the code below; we assume the data objects are already in your  
 12656 workspace from the previous analysis):

```
12657 library(snowfall)
12658 ## create wrapper function
12659 wrapper<-function(a){
12660   out<-SCR0binom.cl(y=Xaug, X=trapmat, M=M, xl=xl, xu=xu, yl=yl,
12661                       yu=yu, K=8, delta=c(0.1, 0.05, 2), niter=5000)
12662   return(out)
12663 }
```

12664 After creating the wrapper function we need to initialize the cluster of cores, defining  
 12665 that we want computation to be implemented in parallel and how many cores we want it  
 12666 to be run on. Here, we assume we have (at least) 3 cores, but if your computer only has 2,  
 12667 make sure to adjust the code accordingly (i.e., set **cpus=2**). In that case, 2 of the 3 chains  
 12668 will be run in parallel and whichever core finishes first will then pick up the third chain.  
 12669 Further, we have to export all **R** libraries and data to all the cores, and set up a random  
 12670 number generator, so that we do not get identical results from the different cores:

```
12671 sfInit( parallel=TRUE, cpus=3 ) #initialize cluster
12672 sfLibrary(scrbook) #export library scrbook
12673 sfExportAll() #export all data in current workspace
12674 sfClusterSetupRNG() #set up random number generator
```

---

```

12675 outL=sfLapply(1:3,wrapper) # execute 'wrapper' 3 times

12676 The object outL is a list of length 3, with one out matrix from the function SCRObinom.cl
12677 for each chain. After computation is complete, terminate the cluster using the command
12678 sfStop(). Note that the intermediate output of current values and acceptance rates in the
12679 R console is suppressed when using parallel computing. We can now look at the output
12680 as described previously using the package coda, by first defining outL to be a list of mcmc
12681 objects.

12682 library(coda)
12683 #turn output into MCMC list
12684 res<-mcmc.list(as.mcmc(outL[[1]]),as.mcmc(outL[[2]]),as.mcmc(outL[[3]]))
12685 summary(window(res, start=1001)) #remove first 1000 iterations as burn-in
12686
12687 [... some output removed ...]
12688
12689      Mean       SD  Naive SE Time-series SE
12690 sigma   1.9723  0.13093 0.0011952      0.0087055
12691 lam0    0.1115  0.01535 0.0001401      0.0009003
12692 psi     0.7130  0.10787 0.0009847      0.0077910
12693 N      499.6166 74.74934 0.6823650      5.4232653
12694
12695 2. Quantiles for each variable:
12696
12697      2.5%     25%     50%     75%   97.5%
12698 sigma  1.74339  1.8811  1.9637  2.0530  2.2618
12699 lam0   0.08443  0.1007  0.1105  0.1211  0.1438
12700 psi    0.52046  0.6350  0.7093  0.7814  0.9627
12701 N     366.00000 446.00000 497.00000 547.00000 674.0000

12702 Now that we have parallel chains we can also use the function gelman.diag to evaluate
12703 if chains have converged:
12704 gelman.diag(window(res, start=1001)) #assess chain convergence
12705
12706 Potential scale reduction factors:
12707
12708      Point est. Upper C.I.
12709 sigma      1.01      1.04
12710 lam0      1.01      1.02
12711 psi       1.07      1.21
12712 N        1.07      1.21
12713
12714 Multivariate psrf
12715
12716 1.05

12717 We can see that estimates are similar to what we observed when running a single
12718 chain (see sec. 14.4.2) 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:

```
12725 effectiveSize(window(res, start=1001))
12726
12727   sigma      lam0      psi       N
12728 272.6935 411.8384 167.4192 168.3355
```

### 14.6.2 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 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:

```

12762 gx<-seq(1,10,1)
12763 gy<-seq(1,10,1)
12764 X<-as.matrix(expand.grid(gx, gy))
12765 M<-700
12766 J<-dim(X)[1]
12767 b<-3
12768 xl<-min(gx)-b
12769 xu<-max(gx)+b
12770 yl<-min(gy)-b
12771 yu<-max(gy)+b
12772 S<-cbind(runif(M, xl, xu), runif(M, yl,yu))

```

12773 Next, we can write a “pedestrian” version of `e2dist` and check how long it takes to  
12774 calculate the squared distance matrix:

```

12775 Dfun<-function(M, J, S, X){
12776 D2<-matrix(0, nrow=M, ncol=J)
12777 for (i in 1:M){
12778 for(j in 1:J){
12779 D2[i,j]<-(S[i,1]-X[j,1])^2 + (S[i,2]-X[j,2])^2
12780 }
12781 return(D2)
12782 }
12783
12784 system.time(
12785 (D2R<-Dfun(M, J, S, X))
12786 )
12787
12788 user   system elapsed
12789 0.81    0.01   0.82

```

12790 The code to implement the same function in **C++** using the `inline` and `RcppArmadillo`  
12791 packages is shown in panel 14.3. These packages allow you to use a range of data formats  
12792 such as lists and matrices, and they take care of compiling the code in **C++** and loading  
12793 the resulting function into **R**. This is also referred to compiling **C++** code “on the fly”.  
12794 You will see that the way the code is set up is reasonably similar to **R**. One difference that  
12795 is worthy to point out is that in **C++** indexes for vectors range from 0 to  $n - 1$ , NOT  
12796 from 1 to  $n$ , as in **R**. Note that with `inline` we only need to write the core of the code and  
12797 define the type of the variables we want to pass to the function, while the `cxxfunction`  
12798 call takes care of the rest. Once your function is compiled and loaded you should check  
12799 out the full **C++** code by calling `DfunArma@code`.

12800 Executing this code shows that it is also faster than the **R** version of the distance  
12801 function or `e2dist`; in fact it is too fast for the time resolution of the `system.time()`  
12802 function to even give us a time estimate:

```

12803 system.time(
12804 (out<-DfunArma(M,J,S,X)))
12805

```

---

```
12806    user   system elapsed
12807      0       0       0
```

12808 While speed differences of less than 1 second may seem negligible, remember that  
 12809 each command has to be executed at each iteration of the Markov chain. Especially with  
 12810 time-consuming models such as those for open populations (Chapt. 15) or multi-session  
 12811 models (Chapt. 16) we believe that C++ holds large potential to make implementation  
 12812 of such models more feasible.

## 14.7 SUMMARY AND OUTLOOK

12813 In a nutshell, programs like **WinBUGS** do everything that we went through in this chapter  
 12814 (and quite a bit more). Looking through your model, they determine which parameters  
 12815 they can use standard Gibbs sampling for (i.e. for conjugate full conditional distributions).  
 12816 Then, they determine whether to use adaptive rejection sampling, slice sampling or – in  
 12817 the ‘worst’ case – Metropolis-Hastings sampling for the other full conditionals (how the  
 12818 sampler is chosen differs among softwares). For MH sampling, they will automatically  
 12819 tune the updater so that it works efficiently.

12820 Although these programs are flexible and extremely useful to perform MCMC simulations,  
 12821 it sometimes is more efficient to develop your own MCMC algorithm. Building an  
 12822 MCMC code follows three basic steps: Identify your model including priors and express  
 12823 full conditional distributions for each model parameter. If full conditionals are parametric  
 12824 distributions, use Gibbs sampling to draw candidate parameter values from those distributions;  
 12825 otherwise use Metropolis-Hastings sampling to draw candidate values from a proposal distribution and accept or reject them based on their posterior probability  
 12826 densities.

12827 These custom-made MCMC algorithms give you more modeling flexibility than existing  
 12828 software packages, especially when it comes to handling the state-space: In **BUGS**  
 12829 (and **JAGS** for that matter) we define a continuous rectangular state-space using the  
 12830 corner coordinates to constrain the Uniform priors on the activity centers **s**. But what if a  
 12831 continuous rectangle isn’t an adequate description of the state-space? In this chapter we  
 12832 saw that in **R** it only takes a few lines of code to use any arbitrary polygon shapefile as the  
 12833 state-space, which is especially useful when you are dealing with coastlines or large bodies  
 12834 of water that need removing from the state-space. Another example is the SCR **R** package  
 12835 **SPACECAP** (Gopalaswamy et al., 2012a) that was developed because implementation of an  
 12836 SCR model with a discrete state-space was inefficient in **WinBUGS**.

12837 Another situations in which using **BUGS/JAGS** becomes increasingly complicated  
 12838 or inefficient is when using point processes other than the Binomial point process (“uniformity”)  
 12839 which underlies the basic SCR model (see sec. 5.9 in Chapt. 5). In Chapt. 13 and  
 12840 XXXX BETHS PP CHAPTER XXXX you will see examples of different point processes,  
 12841 implemented using custom-made MCMC algorithms.

12842 Finally, the Chapt. 18 and 19 deal with unmarked or partially marked populations  
 12843 using hand-made MCMC algorithms to handle the (partially) latent individual encounter  
 12844 histories. While some of these models can be written in **BUGS/JAGS**, they are painstakingly  
 12845 slow; others cannot be implemented in **BUGS/JAGS** at all (e.g., the classes of  
 12846 models considered in Chaps. 12 and 13). In conclusion, while you can certainly get by

12848 using **BUGS/JAGS** for standard SCR models, knowing how to write your own MCMC  
12849 sampler allows you to tailor these models to your specific needs.

---

```
Norm.Gibbs<-function(y=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 according to Eq. 7.2
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 according to Eq. 7.3
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 14.1: 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.

---

```
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 14.2: **R** code to run a Metropolis sampler on a simple Logit-Normal model.

```
### 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 14.3: Code to compute squared distance between individual activity centers and traps in **C++** from within **R** using **inline** and **RcppArmadillo**



12850  
12851

12852

# 15

---

## OPEN POPULATION MODELS

### 15.1 INTRODUCTION

12853 All of the previous chapters focused on closed population models for estimating density  
12854 and for inference about spatial variation in density. However, a thorough understanding  
12855 of population dynamics requires information about both spatial and temporal variation  
12856 in population density and demographic parameters. In this chapter, we develop a frame-  
12857 work for inference about the processes governing spatial and temporal dynamics, namely  
12858 survival, recruitment, and movement (migration, dispersal, etc...). The ability to estimate  
12859 these parameters is critical to both basic and applied ecological research. For example,  
12860 testing hypotheses about life history trade-offs requires accurate estimates of both sur-  
12861 vival and fecundity ([citations](#)). Inference about density-dependent population regulation,  
12862 which has fascinated theoretical ecologists for well over a century, is likewise best accom-  
12863 plished by studying the factors affecting survival and fecundity, rather than the more  
12864 common approach of modeling time series data (Nichols et al., 2000). Modeling vital  
12865 rates is just as important for applied ecologists and conservation biologists, because a  
12866 mechanistic understanding of population decline requires it. Furthermore, if we know how  
12867 environmental variables affect demographic parameters, we can make predictions about  
12868 population changes under different future scenarios. We can also assess the sensitivity of  
12869 parameters such as population growth rate to variation in survival or fecundity. Although  
12870 matrix population models are often used for these purposes (Caswell, 1989; Sæther and  
12871 Bakke, 2000), the same objectives can be accomplished by computing posterior predictive  
12872 distributions as part of the MCMC algorithm.

12873 For the first time, we can fully integrate the movement of individuals onto and off of the  
12874 trap array with their encounter histories to simultaneously estimate density, survival, and  
12875 recruitment in a spatial model. For many species, such as those that are rare or not often  
12876 observed by researchers, this allows us to make inference about survival and recruitment  
12877 without having to physically capture individuals. Additionally, another reason extending  
12878 our SCR models to open populations arises purely from a sampling perspective. We often  
12879 need longer time periods to sample rare or elusive species to ensure that enough captures  
12880 and recaptures are produced. This extended time frame can quickly lead to violations

in the assumption of population closure. For example, the European wildcat study that was presented in chapter XXX insert ref XXX was conducted over a year long period. While the researchers in that study used a closed population, they did model variation in detection as a function of time. Another approach would have been to use an open population model (the spatial capture recapture open models had not been developed at the time of the wildcat study, so we'll forgive the authors for not having used this more appropriate model).

The modeling framework we will develop in this chapter is based on a formulation of 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 been many modifications and developments of the CJS and JS models including dealing with transients, multi-state, and spatially implicit models.

### 15.1.1 Overview of Population Dynamics

The most basic formulation of models for population growth stem from an idea originally used in accounting, the balance sheet. In this case, we can think of population size as a function of credits (i.e., births and immigrants) and debits (i.e., deaths and emigrants). We can then set up the population at time  $t + 1$  as a function of these four components:

$$N(t + 1) = N(t) + B(t) + I(t) - D(t) - E(t)$$

where  $N(t)$  is the population size at time  $t$ ,  $B(t)$  and  $I(t)$  are the credits (additions) from births and immigrants at time  $t$ , and  $D(t)$  and  $E(t)$  are the debits (losses) due to deaths and emigration. This balance equation model is known as the “BIDE model”. We can easily derive a simple population growth model under density independence, by assuming no immigration or emigration.

$$N(t + 1) = N(t) + N(t)r(t)$$

where  $r(t) = b(t) - d(t)$ . Here,  $b(t)$  and  $d(t)$  are the per-capita birth and death rates and thus  $r(t)$  is the per-capita growth rate. Density-dependent, age structured, stochastic effects on growth, spatially structured, and competition models (e.g., Lotka-Volterra) all are basic derivations of the BIDE model.

In closed population models, we focus on estimating  $N(t)$ , but in open population models we are interested in the dynamics that arise between years or seasons and thus we focus not only on  $N(t)$  but on these so-called “credits” and “debits” that drive the population changes. If we take the basic parameters in the BIDE model and reconceptualize them, we can relate these to the commonly used parameters in JS and CJS models, described in more detail below. For example, survival ( $\phi(t)$ ) is defined as the probability of an individual surviving from time  $t$  to  $t + 1$ , and often we call this ‘apparent’ survival because deaths and emigration cannot be separated. Mortality, the probability of dying from time  $t$  to  $t + 1$ , is  $1 - \phi(t)$ . Recruitment ( $\gamma$ ) is the probability of a new individual entering the population between  $t$  to  $t + 1$ , which includes those both those born into the population and immigrants.

**12921 15.1.2 Animal movement related to population demography**

12922 Density may influence demographic parameters such as survival rates, population growth,  
12923 etc., it is also likely that movement of individuals can influence these parameters. For  
12924 example, we know that movement of transients will affect our estimates of survival, causing  
12925 us to typically refer to estimates as “apparent survival”. This is because an animal that  
12926 appears in the population for a short period of time and then leaves is going to appear  
12927 as though it has died. Due to this problem, there has been a significant amount of  
12928 work developing models to deal with transients in both closed and open capture-recapture  
12929 models **NONE OF THESE ARE IN BIB FILE: kendalletal:1997, pradeletal:1997, hinesetal:2003, claveletal:2008**. Because we estimate movement within the SCR framework, we  
12930 can better understand the impact of animals moving onto and off of the trap array and  
12931 hence we can improve our estimates of survival by combining the traditional CJS and JS  
12932 models with the SCR model.

12933 But what if movement and space usage of individuals directly influences the survival  
12934 rates or recruitment? It is generally accepted that population structure (i.e., age, stage, or  
12935 size distribution) can affect both population size and growth over time. We also know that  
12936 how animals associate themselves in space can directly influence the age or stage structure  
12937 of a population – this can be behavioral, habitat related, or some combination of factors.  
12938 For example, if habitat is limited, some younger members of the population might have  
12939 trouble finding and/or defending a territory. Ultimately, this may lower survival for a  
12940 certain age class in the population directly impacting the population structure. Dispersal  
12941 can also affect population structure. In many animal populations, dispersal is linked with  
12942 reproduction and population regulation. Thus, movement including spatial arrangement of  
12943 activity centers and dispersal are key components to population dynamics. We start here  
12944 by showing how to extend the SCR models to open populations, but this chapter opens  
12945 the door for how we would go about incorporating space usage into models of demographic  
12946 dynamics. For example, we could incorporate space and movement into age-dependent  
12947 multistate capture-recapture models to address the impact of dispersal on recruitment or  
12948 survival.

**12950 15.1.3 Basic assumptions of JS and CJS models**

12951 Before extending the classic open models to our SCR framework, let's first look at the  
12952 basic assumptions of both models. No tag (or mark) loss is assumed in both models.  
12953 If a marked animal loses its tag or mark, then that animal cannot be recaptured and  
12954 this could appear as though the animal has died. Hence, to maintain unbiased estimates  
12955 of survival, no tag or mark loss is important. Additionally, capture and release should  
12956 be instantaneous (or as close as possible), otherwise the time interval between capture  
12957 occasions could differ for individuals and that would result in individual heterogeneity of  
12958 survival. Individuals must also be recorded accurately.

12959 In the standard CJS models, it is also assumed that all emigration from the study  
12960 area is permanent and that capture and survival probabilities are constant within each  
12961 sample occasion and group. A group can be created based on sex, age, area, etc. In  
12962 the CJS model, we condition on the captured individuals, and therefore we estimate  
12963 only the probability of recapture and the survival rates. Here, survival is considered the  
12964 “apparent” survival because emigration and mortality are confounded within the model,

thus apparent survival is always estimated lower than true survival when emigration is not zero. In the JS version of the model, we do not condition on marked individuals. Thus we can estimate survival like we do in the CJS, but now we can also model recruitment (new individuals coming into the population) and the total abundance/density of the population. Estimating more parameters does require a few more assumptions including that all individuals in the population have the same probability of capture. Under a “robust design” (Pollock, 1982), which we will demonstrate in this chapter, we can estimate heterogeneity in capture probabilities.

## 15.2 TRADITIONAL JOLLY-SEBER MODELS

There are a number of ways that researchers have formulated the JS model and while all are slightly different, the resulting estimates of abundance and the driving parameters such as survival and some form of recruitment should be the same. The most commonly used formulations are the Link-Barker, Pradel-recruitment, Burham JS, and the Pradel-l models. In all of these models, we are interested in recruitment, or how new individuals arrive into the population. Therefore one of the main differences between the various models is how new entrants into the population are parameterized.

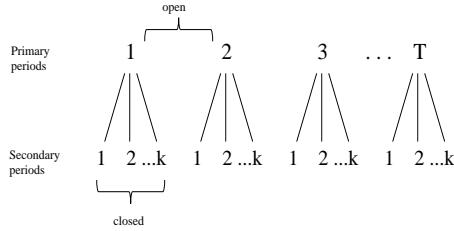
Pollock (1982) created the robust design in order to allow for heterogeneity in capture probability under the JS model. The basic idea is that there are primary occasions (e.g., years, seasons) and we allow the population to be “open” between the primary occasions. This means that individuals can enter and leave the population (i.e., births, deaths, immigration, emigration can occur) between the primary occasions. However, within a primary occasion, the population is assumed to be closed to these processes. The standard JS model does not allow for variation in detection probability between individuals or within a primary occasion because only one sample is collected per primary period. However, when multiple samples are taken within a primary occasion (we call these “secondary occasions”), then variation in detection probability can be modeled and thus our estimates of  $N$  can be improved. To that extent, we can envision the data as arising from repeated sampling over seasons or years (or *primary* periods) within which one or more samples (e.g., nights) might be taken (*secondary* periods). Fig. 15.1 demonstrates the sampling process graphically. Comparing this with all of our previous work, the sample occasions (e.g., trap nights, weeks, etc...) described in the closed population chapters are called *secondary* sampling occasions.

Based on the robust design, we can easily create a non-spatial JS model. We define  $y_{ikt}$  as the encounter history for individual  $i$  at secondary occasion  $k$  during primary occasion  $t$ . If we have a Bernoulli encounter process then we can describe the observation model, specified conditional on  $z(i, t)$ , as:

$$y_{ikt}|z(i, t) \sim \text{Bernoulli}(pz(i, t)).$$

Thus, if individual  $i$  is alive at time  $t$  ( $z(i, t) = 1$ ), then the observations are Bernoulli with detection probability  $p$  as before. Conversely, if the individual is not alive ( $z(i, t) = 0$ ), then the observations must be fixed zeros with probability 1.

Survival and recruitment in the open population are manifest in a model for the latent state variables  $z(i, t)$  describing individual mortality and recruitment events. An important aspect of the hierarchical formulation of the model that we adopt here is that the



**Figure 15.1.** 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 the secondary

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. Using this formulation simplifies the state model and also allows it to be implemented directly in the **JAGS** software Royle and Dorazio (2008). For example, considering the case  $T = 2$ , the state model is composed of the following two components: First the initial state is described by:

$$z(i, 1) \sim \text{Bern}(\psi)$$

and then a model describing the transition of individual states from  $t = 1$  to  $t = 2$ :

$$z(i, 2) \sim \text{Bern}(\phi z(i, 1) + \gamma(1 - z(i, 1))).$$

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:

$$z(i, t + 1) \sim \text{Bern}(\phi_t z(i, t) + \gamma_t(1 - z(i, t))).$$

This parameterization then 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.

---

**15.2.1 Data Augmentation for the Jolly-Seber Model**

13025 The fundamental challenge in carrying out a Bayesian analysis of this model is that the  
 13026 parameter  $N$  (the total number of individuals alive during the study) is not known. We  
 13027 have discussed and demonstrated data augmentation in many previous chapters; however,  
 13028 with the open population model, we have to take care that two issues are addressed:  
 13029 (1) the data augmentation is large enough to accommodate all potential individuals alive  
 13030 in the population during the entire study and (2) that individuals cannot die and then  
 13031 re-enter the population. To begin, let's consider the role of  $\gamma$  in the model.

13032 Data augmentation formally reparameterizes the model, replacing  $N$ , the number of  
 13033 individuals ever alive with the parameter  $\psi$  which is interpretable as the population size  
 13034 expressed as a fraction of  $M$ . That is, the expected value of  $N$  under the model is equal  
 13035 to  $\psi M$ . As a result of this reparameterization, the recruitment parameters  $\gamma_t$  are also  
 13036 relative to the number of "available recruits" on the data augmented list of size  $M$ , and  
 13037 not directly related to the population size. This is easily resolved by deriving  $N_t$ , and  
 13038  $R_t$ , the population size and number of recruits in year  $t$ , as a function of the latent state  
 13039 variables  $z(i, t)$ . In particular, the total number of individuals alive at time  $t$  is  
 13040

$$N_t = \sum_{i=1}^M z(i, t)$$

13041 and the number of recruits is

$$R_t = \sum_{i=1}^M (1 - z(i, t-1)) z(i, t)$$

13042 which is the number of individuals *not* alive at time  $t-1$  but alive at time  $t$ .

13043 In the case of just two primary periods, this process is straightforward. When the  
 13044 number of primary sample occasions is greater than 2, we must formulate the model for  
 13045 recruitment by introducing another latent variable. We do this in order to ensure that  
 13046 an individual can only be recruited once into the population. Here, this formulation of  
 13047 the model uses a set of latent indicator variables  $r(i, t)$  which describe the time interval  
 13048  $(t-1, t)$  at which individual  $i$  is recruited into the population. Let  $r(i, t) = 1$  if individual  
 13049  $i$  is recruited in time interval  $(t-1, t)$  otherwise  $r(i, t) = 0$ . To construct the recruitment  
 13050 process we make use of the standard conditional binomial construction of a removal process  
 13051 (Royle and Dorazio 2008). The initial state is given by:

$$r(i, 1) \sim \text{Bin}(1, \gamma_1)$$

13052 for  $i = 1, 2, \dots, N$ . Then, for  $t > 1$

$$r(i, t) | r(i, t-1) \dots r(i, 1) \sim \text{Bin}\left((1 - \sum_{\tau=1}^{t-1} r(i, \tau)) \times \gamma_t, 1\right)$$

13053 Each recruitment variable is conditional on whether the individual was previously  
 13054 recruited and this construction forces the recruitment variable after initial recruitment to  
 13055 be degenerate (have a sample size of 0). Then, we can describe the state variables  $z(i, t)$   
 13056 by a 1st order Markov process. For  $t = 1$ , the initial states are fixed:

$$z(i, 1) \equiv r(i, 1)$$

13057 and, for subsequent states, we have

$$z(i, t) | z(i, t - 1), r(i, t) \sim \text{Bern}(\phi_t z(i, t - 1)) + r(i, t).$$

13058 Thus, if an individual is in the population at time  $t$  (i.e.,  $z(i, t) = 1$ ), then that individual's  
 13059 status at time  $t+1$  is the outcome of a Bernoulli random variable with parameter (survival  
 13060 probability)  $\phi_t$ . If the individual, however, is not in the population at time  $t$  (i.e.,  $z(i, t) =$   
 13061 0), then the outcome is a Bernoulli random variable with probability  $\gamma_t$ , a parameter that  
 13062 is related to *per capita* recruitment. We carry out this process in **JAGS** by using the **sum()**  
 13063 and **step()** functions together to ascertain if a particular individual  $i$  was ever previously  
 13064 alive. Individuals that were ever previously alive are no longer eligible to be "recruited"  
 13065 into the population. The implementation of this model in **JAGS** is shown in panel 15.1.

---

```
model{

psi ~ dunif(0,1)
phi ~ dunif(0,1)
p.mean ~ dunif(0,1)

for(t in 1:5){
  N[t] <- sum(z[1:M,t])
  gamma[t] ~ dunif(0,1)
}

for(i in 1:M){
  z[i,1] ~ dbern(psi)
  cp[i,1] <- z[i,1]*p.mean
  Y[i,1] ~ dbinom(cp[i,1], K)
  a[i,1] <- (1-z[i,1])

  for(t in 2:5){
    a1[i,t] <- sum(z[i, 1:t])
    a[i,t] <- 1-step(a1[i,t] - 1)

    mu[i,t]<- (phi*z[i,t-1]) + (gamma[t]*a[i,t-1])
    z[i,t] ~ dbern(mu[i,t])
    cp[i,t] <- z[i,t]*p.mean
    Y[i,t] ~ dbinom(cp[i,t], K)
  }
}
}
```

---

Panel 15.1: **JAGS** model specification for the non-spatial JS model.

**13066 15.2.2 Mist-netting example**

13067 We now return to the ovenbird data collected during a mist-netting study, and initially  
 13068 presented in Chapt. 9. These data are available in the `secr` package (see, Efford et al.  
 13069 (2004); Borchers and Efford (2008)). To refresh your memory: 44 mist nets spaced 30 m  
 13070 apart on the perimeter of a 600-m x 100-m rectangle (see Fig. XXXX) were operated on  
 13071 9 or 10 non-consecutive days in late May and June for 5 years from 2005-2009.

13072 In Chapt. 9, we dealt with this dataset as a type of “multi-season” model where  
 13073 abundance in each year,  $N_t$ , was estimated separately. This is the simplest approach for  
 13074 modeling data collected over multiple years, but it does not allow for inference about  
 13075 demographic processes, as does the JS model.

13076 The first issue at hand is that each line in our 3-D encounter history array of data  
 13077 must correspond to a single individual. Previously, we were not interested in individual  
 13078 identity across years so this was not of concern; however, we need to maintain the order of  
 13079 individuals across years in order to estimate the survival and recruitment of the individual  
 13080 into the population. We organize the data set so that each row in our array represents  
 13081 just one individual across all primary periods. For the ovenbird dataset, we can organize  
 13082 the data by creating a master list of all individuals captured during the entire study.  
 13083 From this list, we can assign each individual a unique row in our dataset (in the following  
 13084 **R** commands, we do this by using the `unique()` function on the row names for each year of  
 13085 our 3-D array and use `pmatch()` to associate the data to the correct column). Additionally,  
 13086 in Chapt. 9 we carried out data augmentation for each year separately; however, we must  
 13087 consider for example that individuals captured in year  $t$  could have been alive in year  
 13088  $t - 1$ . Our data augmentation must be large enough to include individuals alive during  
 13089 any of the time periods and to account for that, we set  $M=200$ . For this example, we  
 13090 hold survival constant but allow recruitment to be time dependent (since  $\gamma$  is essentially  
 13091 a function of the data augmentation process as described above, it does not make sense  
 13092 to hold recruitment constant and we therefore make it time specific).

```
13093 library("secr")
13094 library(scrbook)
13095 data(ovenbird)
13096
13097 X<-traps<-traps(ovenCH)
13098 xlim<-c(min(X[[1]][,1])-150,max(X[[1]][,1])+150)
13099 ylim<-c(min(X[[1]][,2])-150,max(X[[1]][,2])+150)
13100 ntraps<- nrow(traps[[1]])
13101 Y<-ovenCH
13102 K<-10
13103 M<-200 # do constant data augmentation to all years
13104 Sst<-cbind(runif(M,xlim[1],xlim[2]),runif(M,ylim[1],ylim[2]))
13105 Sst<-array(Sst,dim=c(M,2,5))
13106
13107
13108 hold<- unique(c(unlist(dimnames(Y[[1]])), unlist(dimnames(Y[[2]])),
13109      unlist(dimnames(Y[[3]])), unlist(dimnames(Y[[4]])),
13110      unlist(dimnames(Y[[5]]))))
```

```

13111
13112 Yarr<-array(ntraps+1,dim=c(M,K,5))
13113 for(i in 1:5){
13114   tmp<-Y[[i]]
13115   tmp[tmp<0]<-tmp[tmp<0]*(-1) ## one guy died, we ignore that here
13116   tmp[tmp==0]<-ntraps+1
13117   nind<-nrow(tmp)
13118   nrep<-ncol(tmp)
13119   tmp2<-matrix(ntraps+1,nrow=M,ncol=10) # pad last col with NA for year 1
13120   tmp2[pmatch(unlist(dimnames(Y[[i]])) [1]), , hold], 1:nrep]<-tmp
13121   Stmp<-Sst[, , i]
13122   Stmp[pmatch(unlist(dimnames(Y[[i]])) [1]), , hold], 1:2]<-
13123     spiderplot(tmp2[pmatch(unlist(dimnames(Y[[i]])) [1]), , hold], 1:nrep),
13124     as.matrix(X[[i]]))$avg.s ##$
13125   Sst[, , i]<-Stmp
13126   Yarr[, , i]<-tmp2
13127 }

13128
13129 Yarr[Yarr < 45] <- 1
13130 Yarr[Yarr == 45] <- 0
13131 Ybin=matrix(NA, M, 5)
13132 for(t in 1:5){
13133   Ybin[, t] <- rowSums(Yarr[, , t])
13134 }
13135
13136 zst<-c(rep(1,M/2),rep(0,M/2))
13137 zst<-cbind(zst,zst,zst,zst,zst)
13138
13139 inits <- function(){list (z=zst,sigma=rnorm(1,25,100), gamma=rnorm(5,0,1)) }
13140 parameters <- c("psi","N","phi", "p.mean", "gamma")
13141 data <- list (K=10,Y=Ybin,M=M)
13142
13143 library("rjags")
13144 out1 <- jags.model("modelNSJS.txt", data, inits, n.chains=3, n.adapt=500)
13145 out2NSJS <- coda.samples(out1,parameters,n.iter=20000)

```

13146 We find in this non-spatial JS model that  $N$  is estimated to be between about 22 and  
13147 33 for each of the 5 years (see Table 15.1 for results). The posterior mean for detection  
13148 ( $p.\text{mean}$  in the model) was 0.14, it is not included in the table because the spatial models  
13149 do not have a parameter that directly corresponds to this one.

### 13150 15.2.3 Shortcomings of the traditional JS models

13151 As we have previously discussed, one of the biggest shortcomings of the non-spatial JS  
13152 model is that we estimate  $N$  but have no explicit spatial reference area for that value. As  
13153 you see in Table 15.1, the density estimate from the non-spatial JS model is listed as NA.  
13154 This is because, again, the effective sampling area is unknown leaving us to determine

that area in an ad hoc manner. Not making use of the spatial information in the data makes the estimation of density a non-formal process. As we saw in the closed models, the explicit incorporation of spatial information will allow us to provide a robust estimate of density. This improvement should also carry through in our estimation of other demographic parameters such as survival and recruitment. Also, while we can potentially model the relationship between density and the demographic parameters we are interested in by using standard JS models, we can make no inference regarding the spatial arrangement of individuals in the landscape nor the direct impact of movement.

### 15.3 SPATIAL JOLLY-SEBER MODELS

To parameterize the spatial JS models, we essentially follow all of the same steps as the non-spatial model but we also include the trap location information into our detection function. Essentially, we are using the closed population SCR model to estimate the detection parameters and initial population size, and the open component is carried out in the process of how we model the transition of  $z(i, t)$  to  $z(i, t + 1)$  which is the same as in the non-spatial JS model. To do so, we describe the Bernoulli observation model, specified conditional on  $z(i, t)$ , as we have done throughout the book:

$$y_{ijk} | z(i, t) \sim \text{Bernoulli}(p_{ijk} z(i, t)).$$

with

$$p_{ijk} = p_0 * \exp(-\alpha_1 d_{ij}^2) \quad (15.3.1)$$

where  $d_{ij} = \|s_i - x_j\|$ , the distance between  $s_i$  and  $x_j$ .

If individual  $i$  is alive at time  $t$  ( $z(i, t) = 1$ ), then the observations are Bernoulli as before. Conversely, if the individual is not alive ( $z(i, t) = 0$ ), then the observations must be fixed zeros with probability 1. We can of course consider other encounter models such as the Poisson or multinomial models described in Chapt. 9.

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, we described how this formulation of the model uses a set of latent indicator variables  $r(i, t)$  which describes if individual is recruited into the population during time  $(t-1, t)$ . Therefore,  $r(i, t) = 1$  if individual  $i$  is recruited in time interval  $(t-1, t)$  otherwise  $r(i, t) = 0$ . Determining the number of recruits into the population, can be done using two steps. 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 below.

#### 15.3.1 Mist-netting example

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

13191 during each of the sampling occasions. We already organized the data above so that  
 13192 our 3-D encounter histories are set up. The data set is then  $M = 200$  individuals by  
 13193  $K = 10$  secondary occasions by  $T = 5$  primary occasions. In the non-spatial version, we  
 13194 reduced the data to captured or not-captured; however, the encounter history array  $\text{Yarr}$ )  
 13195 contains the number of the net that each individual was captured in and contains a 45 if  
 13196 the individual was not captured. The encounter history array,  $\text{Yarr}$ ), was created above  
 13197 in the code, so we do not reproduce the code here.

```

13198 cat("
13199 model {
13200
13201 psi ~ dunif(0,1)
13202 phi ~ dunif(0,1)
13203 alpha0 ~ dnorm(0,10)
13204 sigma ~ dunif(0,200)
13205 alpha1<- 1/(2*sigma*sigma)
13206
13207 A <- ((xlim[2]-xlim[1]))*((ylim[2]-ylim[1]))
13208
13209 for(t in 1:5){
13210 N[t] <- sum(z[1:M,t])
13211 D[t] <- N[t]/A
13212 gamma[t] ~ dunif(0,1)
13213 }
13214
13215 for(i in 1:M){
13216 z[i,1] ~ dbern(psi)
13217
13218 #to estimate the number of recruits, we need a few derivations
13219 R[i,1]<- z[i,1]
13220 R[i,2]<-(1-z[i,1])*z[i,2]
13221 R[i,3]<- (1-z[i,1])*(1-z[i,2])*z[i,3]
13222 R[i,4] <-(1-z[i,1])*(1-z[i,2])*(1-z[i,3])*(1-z[i,4])*z[i,5]
13223 R[i,5] <-(1-z[i,1])*(1-z[i,2])*(1-z[i,3])*(1-z[i,4])*z[i,5]
13224
13225
13226 for(t in 1:5){
13227 S[i,1,t] ~ dunif(xlim[1],xlim[2]) # XXXX This needs to be justified XXXX
13228 S[i,2,t] ~ dunif(ylim[1],ylim[2])
13229
13230 for(j in 1:ntraps){
13231 d[i,j,t] <- pow(pow(S[i,1,t]-X[j,1],2) + pow(S[i,2,t]-X[j,2],2),1)
13232 }
13233
13234 for(k in 1:K){
13235 for(j in 1:ntraps){
13236 lp[i,k,j,t] <- exp(alpha0 - alpha1*d[i,j,t])*z[i,t]
```

```

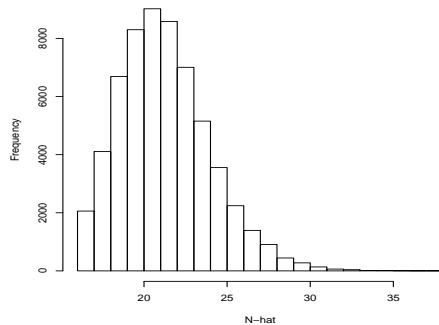
13237     cp[i,k,j,t] <- lp[i,k,j,t]/(1+sum(lp[i,,t]))
13238   }
13239   cp[i,k,ntraps+1,t] <- 1-sum(cp[i,k,1:ntraps,t]) #last cell = not captured
13240   Ycat[i,k,t] ~ dcat(cp[i,k,,t])
13241 }
13242 }
13243
13244 a[i,1]<-(1-z[i,1])
13245
13246 for(t in 2:5){
13247   a1[i,t] <- sum(z[i, 1:t])
13248   a[i,t] <- 1-step(a1[i,t] - 1)
13249
13250   mu[i,t]<- (phi*z[i,t-1]) + (gamma[t]*a[i,t-1])
13251   z[i,t]~dbern(mu[i,t])
13252   }
13253 }
13254
13255 R1<-sum(R[1:M,1])
13256 R2<-sum(R[1:M,2])
13257 R3<-sum(R[1:M,3])
13258 R4<-sum(R[1:M,4])
13259 R5<-sum(R[1:M,5])
13260 }
13261
13262 ",file="modelJS.txt")
13263 ####
13264
13265
13266 zst<-c(rep(1,M/2),rep(0,M/2))
13267 zst<-cbind(zst,zst,zst,zst,zst)
13268
13269 inits <- function(){list (z=zst,sigma=runif(1,25,100), gamma=runif(5,0,1) ,S=Sst,alpha0=runif(1,-2,-1) ) }
13270 parameters <- c("psi","alpha0","alpha1","sigma","N","D", "phi", "gamma", "R2", "R3", "R4", "R5")
13271 data <- list (X=as.matrix(X[[1]]),K=10,Ycat=Yarr,M=M,ntraps=ntraps,ylim=ylim,xlim=xlim)
13272
13273 library("rjags")
13274 out1 <- jags.model("modelJS.txt", data, inits, n.chains=3, n.adapt=500)
13275 out2JS <- coda.samples(out1,parameters,n.iter=10000)
13276

```

13277 Our results for density, alpha0, and alpha1 are rather similar to those found in the  
 13278 multi-season analysis from Chapt. 9. Since all of our parameters including alpha0 and  
 13279 alpha1 are shared between seasons, we would expect these results to be similar between the  
 13280 multi-season model and the JS model (see Table 15.1). There are some slight differences in  
 13281 the parameter estimates, for example, the density is smaller in year 4 in the multi-season  
 13282 model than in the JS model. This maybe be due to a smaller sample size in that year

**Table 15.1.** Posterior mean of model parameters for the non-spatial JS model (NS-JS), the spatial JS model (S-JS), and the spatial multi-season model (S-MS) fitted to the ovenbird data set.

	NS-JS	S-JS	S-MS
D[1]	NA	9.6e-05	9.3e-05
D[2]	NA	1.0e-04	1.0e-04
D[3]	NA	1.1e-04	1.2e-04
D[4]	NA	1.1e-04	8.9e-05
D[5]	NA	7.9e-05	7.6e-05
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	1.5e+01	NA
R3	NA	1.9e+01	NA
R4	NA	8.3e+00	NA
R5	NA	8.3e+00	NA



**Figure 15.2.** Posterior distribution of  $N_5$  from the spatial JS model for the ovenbird dataset. This graph suggests that there is no truncation of the posterior of  $N_5$ .

and the JS model is able to make use of the data a little more efficiently across the years. Because we have defined the same state space for the spatial JS model and multi-season, our estimates of  $N$  are directly comparable. However, the estimates of  $N$  under the non-spatial JS model are not directly comparable as we do not have a well-defined effective trapping area. We see from Table 15.1 that  $N$  is smallest for the non-spatial JS model across all years. This suggests that the actual effective trapping area is smaller than our state space, but we cannot know how much relative to the state-space to make useful comparisons between the  $N$ s.

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 ( $R_2$ ,  $R_3$ ,  $R_4$ , and  $R_5$ ).  $R_2$  is the number of new recruits from primary period 1 to 2;  $R_3$  is the number of new recruits from primary period 2 to 3; and so forth.  $R_2$  and  $R_3$  are almost double that of  $R_4$  and  $R_5$ , 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  $N$  to make sure they are not truncated, Fig. 15.2 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 it here, 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 literature. Some studies have found annual male ovenbird survival to be around 0.62 (Porneluzzi and Faaborg, 1999; Bayne and Hobson, 2002); however, female ovenbird survival was much lower (0.21, Bayne and Hobson (2002)). With more individuals, we

13311 could run this model with survival estimated for each sex separately. However, we should  
 13312 be careful not to over-parameterize our model based on the amount of data available.

## 15.4 TRADITIONAL CJS MODELS

13313 The Cormack-Jolly-Seber models are used extensively in the literature to estimate survival  
 13314 probabilities. There are essentially two ways to fit these models, using either a multinomial  
 13315 approach (Lebreton et al., 1992) or a state-space likelihood approach (Gimenez et al., 2007;  
 13316 Royle, 2008).

13317 We can adopt a simple hierarchical parameterization of the basic single state, non-  
 13318 spatial CJS model in which the observation model is described conditional on the latent  
 13319 state variables  $z(i, t)$  – the alive state which describe whether individual  $i$  is alive ( $z(i, t) =$   
 13320 1) or not ( $z(i, t) = 0$ ) during each of  $t = 1, 2, \dots, T$  primary periods. Let  $y_{it}$  indicate  
 13321 the observed encounter data of individual  $i$  in primary period  $t$ . The model, specified  
 13322 conditional on  $z(i, t)$ , is:

$$y_{it}|z_{it} \sim \text{Bernoulli}(p_t z_{it}).$$

13323 Analogous to the JS model, if individual  $i$  is alive at time  $t$  ( $z_{it} = 1$ ), then the observations  
 13324 are Bernoulli with probability of detection  $p_t$ . If the individual is not alive ( $z_{it} = 0$ ), then  
 13325 the observations must be fixed zeros with probability 1. In the CJS formulation, as  
 13326 opposed to the JS, we condition on first capture which means that  $z_{it}$  will be 1 when  $t$  is  
 13327 the first primary period of capture. We can denote this  $z_{if_i}$  where  $f_i$  indicates the primary  
 13328 occasion in which individual  $i$  is first captured, which can vary from  $1 \dots T$ . This ensures  
 13329 that each individual is alive upon entering the model.

13330 We can describe the "alive state" at time  $t$  for each individual as a function of the  
 13331 state at the previous time step  $t - 1$ . Because we condition on the first capture, the initial  
 13332 state is set to one:

$$z_{if_i} = 1$$

13333 and to model the transition of individual states from  $t$  to  $t + 1$  for all  $t > f_i$  we have

$$z_{it} \sim \text{Bernoulli}(\phi z_{i,t-1}).$$

13334 Because we start with  $z_{if_i} = 1$ , the individual survives with probability  $\phi$  to time  $f_i + 1$   
 13335 and so forth. Once an individual leaves the population (i.e.,  $z_{it} = 0$ ), there is no mech-  
 13336 anism for the individual to return. This means that under this specification individuals  
 13337 cannot temporarily emigrate. In the CJS model we are not estimating  $N$ , so we do not  
 13338 incorporate any data augmentation here. This version of the model is easy to construct in  
 13339 the **BUGS** (or **JAGS**) language which is shown in Panel 15.2. Variations on this basic  
 13340 model and associated code for fitting the model in **BUGS** are described in detail in Kéry  
 13341 and Schaub (2012, Chaps. 7-9).

### 15.4.1 Migratory fish example

13342 The motivation for this example stems from an interest in better understanding survival  
 13343 and movement of migratory fishes. For this example, we will use data collected on Amer-  
 13344 ican shad *Alosa sapidissima* in the New River in North Carolina, U.S.A. (see photo in  
 13345 Fig. 15.3). The data were collected and analyzed in Raabe (2012). Using a resistance

---

```
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 15.2: **JAGS** model specification for the non-spatial basic CJS model.



**Figure 15.3.** American shad caught in North Carolina, U.S.A. Credit: Joshua Raabe, North Carolina State University

**Table 15.2.** 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

13347 board weir near the river mouth, 315 fish were tagged with passive integrated transponders  
 13348 (PIT) in the spring of 2010. An array of 7 upstream PIT antennas passively recaptured  
 13349 individuals during upstream and downstream migrations. Each time a fish passed over  
 13350 the antenna, it was recorded and summarized weekly for 12 weeks. To apply the basic  
 13351 CJS model, we create the encounter history for each individual for the 12 weeks and we  
 13352 also create a vector to indicate the period of first capture.

13353 Table 15.2 shows the estimated detection probability for each of the 12 primary periods  
 13354 in the study. The posterior mean for detection probability ranges from 0.126 to 0.880,  
 13355 which could potentially be due to variation in water flow, stream depth, storms, etc...  
 13356 The weekly survival probability,  $\phi$  had a posterior mean estimate of 0.824. This estimate  
 13357 could be considered low for a weekly probability, but is likely due to the fact that the  
 13358 migration upstream can be quite energetically taxing. Additionally, the CJS model is  
 13359 only estimating apparent survival and some fish may have left the stream temporally or  
 13360 permanently. We demonstrate in panel 15.2 how to allow  $p$  to vary by time, but we could  
 13361 also allow survival,  $\phi$  to vary by time by implementing it exactly as we do  $p$ . As we move  
 13362 into the multi-state model, we can test for movement and survival by state, allow us to  
 13363 address more specific biological questions.

## 15.5 MULTI-STATE CJS MODELS

13364 The basic version of the CJS model only allows for estimation of survival and detection.  
 13365 However, researchers are often interested in addressing other ecological questions such as  
 13366 age-dependent survival rates, habitat based movements, etc. Multi-state models allow  
 13367 researchers to directly address such questions by incorporating more than one state that  
 13368 an individual may potentially be in Arnason (1972, 1973); Brownie et al. (1993). These  
 13369 possible states can be geographic location, age class, or reproductive status among many  
 13370 others. Instead of just having an encounter history for an individual, we will also have  
 13371 auxiliary information on the state of that individual at capture (e.g., breeder or non-  
 13372 breeder). Since our interest in movement of individuals, here we will consider states that  
 13373 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 our single-state model above, an individual  $i$  was either alive ( $z_{it} = 1$ ) at time  $t$  or dead ( $z_{it} = 1$ ). 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 15.3. Here,  $\phi_A$  is the probability of surviving in State A from time  $t$  to  $t + 1$  and  $\phi_B$  is analogous for State B. The movement parameters are  $\psi_{AB}$  and  $\psi_{BA}$ , where  $\psi_{AB}$  is the probability that an individual, which survived from  $t$  to  $t + 1$  in Site A, moves to State B just before  $t + 1$  and vice versa for  $\psi_{BA}$

**Table 15.3.** 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

We do not necessarily observe individuals in their given state though, so we must estimate detection 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.

To relate this back to the description of multi-state models in Chapt. 9, we can define  $s$  as the index of which state an individual is 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 such that

$$u_{it} \sim \text{dcat}(\psi)$$

where  $\psi$  is a constant vector. We observe that individual with probability  $p_0$ , that is:

$$\Pr(y_{it} = 1|u_{it}) = p_0$$

The state-transition probabilities are constant.

To extend this model, we can define  $s$  as the index of which state an individual is in and then condition the observed locations,  $u_{it}$  as a function of the state an individual is in,  $s$ . This means that whether an individual moves or not, or where it moves to, is a function of where it is located.

This commonly used model has successive movement outcomes that are *iid*

$$u_{it} \sim \text{dcat}(\psi(s_i))$$

Conditional on the state in which individual  $i$  is located, we observe that individual with probability  $p_0$ . That is:

$$\Pr(y_{it} = 1|u_{it}) = p_0$$

**Table 15.4.** 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

13405 The state-transition probabilities are still constant, conditional on  $s$ . Other models for  
 13406 these transition probabilities are possible and we will discuss those later.

13407 A slight modification of this model would define  $s$  as a “home area” for each individual.  
 13408 Then the region the animal goes to is a function, not of where he was last time, but  
 13409 which region is his home area. This model is only subtlety different from the Markovian  
 13410 model and as was shown in Chapt. 9 for closed populations models is how we make the  
 13411 technical transition from multi-state models to SCR models. Essentially increasing to a  
 13412 large number of strata, this formulation of the multi-state model becomes an SCR model  
 13413 where the “home area”  $s$  becomes the “activity center” for each individual.

13414 To program this model in **JAGS**, we use a slightly different formulation which es-  
 13415 sentially combines  $u_{it}$  and  $y_{it}$  as defined above into one observation matrix such that  
 13416  $y_{it} = 1, 2$ , or 3 where 3 indicates “not observed”. Additionally, we use  $z_{it}$  to indicate the  
 13417 true state of individual  $i$  such that  $z_{it} = 1, 2$ , or 3 where 1 indicates alive and in state 1, 2  
 13418 indicates alive and in state 2, and 3 indicates “not alive”. Using this delineation, we just  
 13419 need to set up the transition matrix based on Table 15.3 and define each item within the  
 13420 model specification, shown in Panel 15.3. Note that this can become quite cumbersome  
 13421 when dealing with models that have many states.

### 13422 15.5.1 Migratory fish example

13423 Previously, we analyzed the American shad data using a basic CJS model. However,  
 13424 the researchers were interested in movement of fish during migration and so we classified  
 13425 the stream into 2 states (regions) – “downstream” and “upstream”. Each antenna was  
 13426 assigned to a state based on the location, those below 20 river kilometers were considered  
 13427 in the downstream state. Each fish has an encounter history including whether or not the  
 13428 fish was detected during each week of the 12 week study, but also the “state” of capture  
 13429 (“downstream” or “upstream”). Again, a vector to indicate the period of first capture  
 13430 was also created. Fish captured in more than one state during the week were assigned the  
 13431 state in which they were captured most during that week.

13432 Survival between the two areas is quite different (see Table 15.4). This might suggest  
 13433 that fish moving further upstream are expending more energy and are more likely to die.  
 13434 While survival in the two states was different, it is intuitive that the average of the survival  
 13435 probabilities for A and B is essentially the same as that from the basic non-spatial CJS

---

```

model {
  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 15.3: **JAGS** model specification for a two state version of the multi-state CJS model. Code adjusted from (Kéry and Schaub, 2012, Chapt. 9).

( $\phi = 0.82$ , see Table 15.2). Also, it should be noted that  $\psi_B A$  is very high, indicating that fish in this study are returning downstream after spawning in the upstream area. These results highlight the utility in using a multi-state model to understand movement between states; here, we used spatial states, but age, class, breeding status, etc. are all possibilities. We did have to reduce the dataset however to fit this model and information on spatial location was lost in creating just two states, downstream and upstream.

## 15.6 SPATIAL CJS MODELS

In Chapt. 9, we described how SCR models are essentially a type of multi-state model with spatially structured transition probabilities. As we noted, individuals can appear in  $> 1$  states, simultaneously, which is not directly analogous to a standard multi-state model. However, building on the state-space and multi-state CJS models, we can explicitly incorporate individual movement as an individual covariate model (Royle, 2009b). To move from the basic and multi-state CJS models to the SCR version, we need only make a few changes to the model. Essentially, we will not have discrete states and thus the biggest difference is that individuals do not “transition” between a finite set of states, but instead are allowed to move in continuous space.

We may consider the same basic encounter models as described previously (i.e., Poisson, Bernoulli, or multinomial). In particular, let  $y_{ijkt}$  indicate the observed encounter data of individual  $i$  in trap  $j$ , during interval (secondary period or sub-sample)  $k = 1, 2, \dots, K$  and primary period  $t$ . We note that in some cases we may have intervals ( $K = 1$ ) which correspond to the design underlying a standard CJS or JS models whereas the case  $K > 1$  corresponds to the “robust design” (Pollock 1982). The Poisson observation model, specified conditional on  $z(i, t)$ , is:

$$y_{ijkt} | z(i, t) \sim \text{Poisson}(\lambda_0 g_{ij} z(i, t)).$$

Conversely, if the individual is not alive ( $z(i, t) = 0$ ), then the observations must be fixed zeros with probability 1. In the CJS formulation, we will condition on first capture which means that  $z(i, t)$  will be 1 when  $t$  is the first primary period of capture. We can denote this  $z(i, f_i)$  where  $f_i$  indicates the primary occasion in which individual  $i$  is first captured. This ensures that each individual is alive upon entering the model.

Modeling time-effects either within or across primary periods is straightforward. For that, we define  $\lambda_0 \equiv \lambda_0(k, t)$  and then develop models for  $\lambda_0(k, t)$  as in our closed SCR models (we note that trap-specific effects could be modeled analogously).

We follow the same model for survival as described in the non-spatial version of the CJS. The model is initialized by setting the alive state at first capture to one:

$$z(i, f_i) = 1$$

and for the transition of an individual’s alive state from  $t$  to  $t + 1$ , for all  $t > f_i$ , we have

$$z_{it} \sim \text{Bernoulli}(\phi z_{i,t-1}).$$

An individual survives with probability  $\phi$  from one time step to the next. It is easy to see that we can let survival be time specific by allowing  $\phi$  to vary with each time step:

$$z_{it} \sim \text{Bernoulli}(\phi_t z_{i,t-1}).$$

13471 In either case, once an individual leaves the population (i.e.,  $z_{it} = 0$ ), there is no  
 13472 recruitment so individuals cannot return. Again, we are not estimating  $N$  in this model,  
 13473 hence we do not need any data augmentation. This conveniently makes the model run  
 13474 faster too!

13475 **15.6.1 Migratory fish example**

13476 Going back to our American shad example, we can consider that this is exactly a spatial  
 13477 capture recapture problem. In stream networks, the placement of PIT antennas along the  
 13478 stream mimics the type of spatial data collected in terrestrial passive detector arrays such  
 13479 as camera traps, hair snares, acoustic recording devices, etc. The difference is that for  
 13480 fish and aquatic species, the stream constrains the movement of individuals to a linear  
 13481 network. Using the data from the array of 7 PIT antennas and the number of times each  
 13482 fish passed over the antenna, we can apply the SCR CJS model to evaluate movement  
 13483 up and downstream of these fishes. When we look at the individuals encountered at each  
 13484 antenna for each of the primary periods, the dimensions of the data are 315 individuals by  
 13485 7 antennas by 12 sample occasions. Individuals can encounter any antenna any number  
 13486 of times during the week, which means we just sum the encounters over the week and  
 13487 eliminate any need for explicit secondary occasions in the model. The result is a 3-D  
 13488 array instead of a 4-D array. Given the structure of the encounters, we use a Poisson  
 13489 encounter model in this example.

```
13490 library(reshape)
13491
13492 # Constants:
13493 M <- 315      # Number of individuals
13494 T <- 12       # Number of periods (weeks)
13495 nantenna <- 7 # weir, 6 antennas
13496 antenna.loc <- c(3,7,12,44,56,72,77) # antenna locations
13497
13498 # Input and format data matrix:
13499 AS10 <- read.table("AS10.txt" ,header=T)
13500 melted.rkm <- melt(AS10, id=c("TagID","RKM"))
13501 y <- cast(melted.rkm, TagID ~ RKM ~ value, fill=0, length)
13502 first=read.csv("firstcap.csv")
13503
13504 sink("ModelCJS.txt")
13505 cat("
13506
13507 model {
13508 # Priors
13509 sigma ~ dunif(0,80)
13510 sigma2 <- sigma*sigma
13511 lam0 ~ dgamma(0.1, 0.1)
13512 phi ~ dunif(0, 1) # Survival (constant across time)
13513 tauv~dunif(0, 30)
13514 tau<-1/(tauv*tauv)
```

```

13515
13516 for (i in 1:M){
13517   z[i,first[i]] <- 1
13518   S[i,first[i]] ~ dunif(0,50)
13519
13520 for(j in 1:nantenna) {
13521   D2[i,j,first[i]] <- pow(S[i,first[i]]-antenna.loc[j], 2)
13522   lam[i,j,first[i]]<- lam0*exp(- D2[i,j,first[i]]/(2*sigma2))
13523   tmp[i,j,first[i]] <- lam[i,j,first[i]]
13524   y[i,j,first[i]] ~ dpois(tmp[i,j,first[i]])
13525 }
13526
13527 for (t in first[i]+1:T) {
13528   S[i,t] ~ dunif(xl, xu) # XXXX above you have dunif(0,50)?
13529   for(j in 1:nantenna) {
13530     D2[i,j,t] <- pow(S[i,t]-antenna.loc[j], 2)
13531     lam[i,j,t] <- lam0 * exp(-D2[i,j,t]/(2*sigma2))
13532     tmp[i,j,t] <- z[i,t]*lam[i,j,t]
13533     y[i,j,t] ~ dpois(tmp[i,j,t])
13534   }
13535   phiUP[i,t] <- z[i,t-1]*phi
13536   z[i,t] ~ dbern(phiUP[i,t])
13537 }
13538 }
13539 }
13540
13541 ",fill = TRUE)
13542 sink()
13543
13544 data1<-list(y=y, first=first, M=M, T=T, xl=0, xu=80, nantenna=nantenna, antenna.loc=antenna.loc)
13545
13546 z=matrix(NA, M, T)
13547 for(i in 1:M){
13548   for(t in first[i]:12){
13549     z[i,t] <-1
13550   }
13551 }
13552
13553 inits = function() {list(z=z,phi=runif(1,0,1), lam0=runif(1,0,2),
13554                         tauv=runif(1,10, 20), sigma=runif(1,0,10)) }
13555
13556 parameters <- c("sigma", "phi", "lam0")
13557
13558 library("rjags")
13559 out1 <- jags.model("modelCJS.txt", data1, inits, n.chains=3, n.adapt=500)
13560 out2CJS <- coda.samples(out1,parameters,n.iter=20000)

```

**Table 15.5.** Results of the spatial CJS 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

13561 The baseline encounter rate,  $\lambda_0$ , was allowed to vary by week and ranged from 0.188 to  
 13562 5.555. We use the Poisson encounter model in this spatial CJS example rendering  $\lambda_0$  not  
 13563 directly comparable to  $p_0$  from the non-spatial and multi-state versions which arises as the  
 13564 detection probability based under the Binomial encounter model. The posterior mean for  
 13565  $\phi$  was 0.784 (see Table 15.5), again showing that the survival probability is generally low,  
 13566 just as we saw in the two previous example analysis of these data. Here, we are modeling  
 13567 survival probability as constant, but there is reason to believe that it might vary by time  
 13568 (similar to detection) and we might consider this additional parameterization in a more  
 13569 complete analysis of the data set. The other parameter of interest is  $\sigma$ , the movement  
 13570 parameter, which had a posterior mean of 13.954. Our system here is linear, so we do not  
 13571 think of fish as having a home range radius in this system. However,  $\sigma$  can still inform  
 13572 us about the linear distance fish are moving. One final note about this example, we have  
 13573 simplified the dataset for analysis here and some parameter estimates are different than  
 13574 found in Raabe (2012).

## 15.7 MOVING ACTIVITY CENTERS

13575 We extend the model of individual encounter histories by specifying an additional model  
 13576 component that describes the spatial distribution of individual activity centers. A plau-  
 13577 sible “null model” for the distribution of individual activity centers is to assume they  
 13578 are static over time and do not change across primary periods, i.e.,  $s_i \sim \text{Unif}(\mathcal{S})$ . It  
 13579 might seem more likely that activity centers change over time but are independent from  
 13580 year to year for a given individual such  $s_i \sim \text{Unif}(\mathcal{S})$ . This is how the spatial version  
 13581 of the JS and CJS models were formulated above. Another option would be to assume  
 13582 that  $s(i, t) \sim \text{Normal}(s(i, t - 1), \tau^2 \mathbf{I})$  for  $t > 1$  so that individual home range centers are  
 13583 perturbed randomly from their previous value.

13584 We could use this specification to model changes in home range centers with regards  
 13585 to habitat. For example, if our primary period is a season, we may expect that individuals  
 13586 move as the available food sources change. Using telemetry data and/or capture recapture

models a number of developments have been made to understand animal movement patterns relative to habitat or dynamic systems(e.g., Jonsen et al. (2005); Hooten and Wikle (2010)). Similarly, if we have an indicator of habitat that varies by season, then in SCR models we can model the location of activity centers as a function of the change in habitat. There are a number of options for modeling variation in activity centers or animal locations as a function of covariates such as habitat, season, or behavior. Other approaches to analyzing movement in a mark-recapture framework include but are not limited to diffusion and auto-regressive models(Ovaskainen (2004); Ovaskainen et al. (2008)), agent-based (Grimm et al. (2005); Hooten et al. (2010)) and dispersal kernels (Fujiwara et al. (2006)). For example, we define  $u_{ikt}$  as the individual's observed location at secondary period  $k$  in primary period  $t$ . Then  $u_{ikt} \sim \text{Normal}(\mathbf{s}(i, t), \Sigma_t)$  where  $\Sigma_t$  is the variance-covariance matrix at time  $t$ . This is the model we have assumed quite frequently throughout the book, i.e., that individual observed locations are assumed to follow a bivariate normal distribution about the activity center,  $\mathbf{s}$ . This is similar to the Guassian and Laplace dispersal kernels. We could then allow the observed locations to follow an auto-regressive model such that  $u_{ikt} \sim \text{Normal}(\rho\mathbf{s}(i, t - 1), \Sigma_t)$  XXX figure out the variance XXX. This is just one simple example, as more information becomes available and data are collected over longer time periods, the ability to use different movement models will continue to be employed in open SCR models.

Rathbun and Cressie (1994) articulate model for marked point processes where they separate out the spatial birth, growth, and survival processes for longleaf pine trees. Because of the application, these demographic parameters are slightly different than how they are often considered in wildlife, but are still analogous. Allowing birth, growth, and survival as well as density to arise from different spatially varying processes is the next stage in development of the open SCR models.

### 15.7.1 Migratory Fish Example Notes

In our American shad example above, we had reason to believe that individual movement is directly related to stream flow. When the stream flow is low, we might expect that the fish move very little, and when the stream flow is high, they might move upstream to spawn. In this case, we could model the effect of stream flow in two ways. First, we might allow  $\sigma$  to be a function of flow and to vary for each primary occasion.

$$\log(\sigma_t) = \mu_S + \alpha_2 \text{Flow}_t$$

But if we think that the change in activity centers between primary periods might be related to the overall movement of fish, then we could allow the variation in locations to be a function of flow. This means that we assume the activity centers are correlated so we have

$$\mathbf{s}(i, t) \sim \text{Normal}(\mathbf{s}(i, t - 1), \tau^2 \mathbf{I})$$

where

$$\log(\tau) = \mu_T + \alpha_2 \text{Flow}_t$$

These are just a few thoughts on simple ways to model movement as a function of habitat variables. As we discussed in the previous section, there are many other movement models that could be used.

## 15.8 SUMMARY AND OUTLOOK

13626 In this chapter we have described a framework for making inference not only about spatial  
13627 and temporal variation in population density, but also demographic parameters including  
13628 survival, recruitment, and movement. The ability to model population vital rates is es-  
13629 sential for ecology, management, and conservation; and the models described here allow  
13630 researchers to examine the spatial and temporal dynamics governing those population  
13631 parameters.

13632 As open models are further developed, mechanisms for dealing directly with dispersal  
13633 and transients will provide improved inference frameworks for understanding movement  
13634 as well as the potential to estimate *true* survival instead of only *apparent* survival. This is  
13635 a function of explicitly modeling movement, which means we can separate movement from  
13636 mortality providing a huge advantage over traditional models. Also, models of individual  
13637 dispersal can be used to examine dynamics of population dynamics relative to habitat,  
13638 density-dependence, or climatic events.

13639 Birth and death processes, as well as movement, all have the potential to be related  
13640 to the space usage of animals in the landscape. Understanding the impact of spatially  
13641 varying density on survival and recruitment will provide insights into the basic ecology  
13642 of species. With the advent of non-invasive techniques, like camera trapping and genetic  
13643 analysis of tissue, we can start to understand the population dynamics of species that are  
13644 rarely observed in the wild. As more and more data are collected, we can use the models  
13645 to explore the spatio-temporal patterns of survival, recruitment, density, and movement of  
13646 species, providing incredibly useful biological and ecological information as we face broad  
13647 changes in climate, land-use, habitat fragmentation, etc..

13648  
13649

# 16

---

13650  
13651

## STRATIFIED POPULATIONS: MULTI-SESSION AND MULTI-SITE DATA

13652 In this chapter, we describe SCR models for situations when we have multiple, distinct  
13653 groups, strata or “sessions” (multi-session models using the `secr` terminology). The mod-  
13654 els are extremely general and provide a flexible hierarchical modeling framework for mod-  
13655 eling abundance (Converse and Royle, 2012; Royle et al., 2012b). We believe that such  
13656 “stratified” populations are extremely commonplace, yet most SCR applications have been  
13657 based on models that are distinctly single-population models. This is done either by an-  
13658 alyzing separate data sets one-at-a-time or by pooling data from multiple study areas. A  
13659 standard example that arises frequently is that in which multiple distinct patches (often  
13660 refuges, parks or reserves) are sampled independently with the goal of estimating the pop-  
13661 ulation size in each reserve. It makes sense to combine the data together into a single  
13662 model that permits the sharing of information about some parameters, but provides in-  
13663 dividual estimates of abundance for each land unit. A similar situation is that in which  
13664 a number of replicate trap arrays are located within a landscape, sometimes for purposes  
13665 of evaluating management actions or landscape structure. This is extremely common in  
13666 studies of small mammals (Converse et al., 2006a,b; Converse and Royle, 2012), or in mist-  
13667 netting of birds (DeSante et al., 1995) (BETTER REF HERE WOULD BE NICE), but  
13668 there are examples of large-scale monitoring of carnivores and other species, e.g., tigers  
13669 (Jhala et al., 2011).

13670 Stratified or multi-session SCR models are also directly relevant when the grouping is  
13671 based on distinct time samples, either periods within a biological season, or even across  
13672 years. Unlike in the case of having spatial strata, with temporally defined samples, we  
13673 imagine a fully dynamic, or demographically open, model that involves survival and re-  
13674 cruitment might be suitable. We deal with those models specifically in Chapt. 15. How-  
13675 ever, the stratified (multi-session) models we deal with in this chapter can be thought of  
13676 as a primitive type of model for open systems, but in which the populations are assumed  
13677 to be *independent*. Whereas the underlying model may be one of Markovian dynamics  
13678 (survival, recruitment), we could *ignore* that dependence for convenience or perhaps the  
13679 dynamics are not distinctly estimable because individual recapture rate is low.

13680 We focus mostly on Bayesian analysis of stratified SCR models using data augmentation (Royle et al., 2012b; Royle and Converse, 2012). The technical modification of  
 13681 data augmentation to deal with such models is that it is based on a model for the joint  
 13682 distribution of the stratum-specific population sizes,  $N_g$ , *conditioned* on their total. This  
 13683 results in a multinomial distribution which we can analyze in some generality using data  
 13684 augmentation. As a practical matter, specification of this multinomial distribution for the  
 13685  $N_g$  parameters *induces* a distribution for an individual covariate, say  $g_i$ , which is “group  
 13686 membership”. This is extremely handy to analyze by MCMC in the various **BUGS**  
 13687 engines that you are familiar with by now.

13688 The **R** package **secr** fits a class of multi-session models which we have already seen  
 13689 (sec. 6.5.4) and we used this class of models to analyze the ovenbird data in **secr** (sec.  
 13690 9.3). Later in this chapter we will provide a Bayesian analysis of the ovenbird data in  
 13691 **BUGS** using an analogous class of models.

## 16.1 DATA STRUCTURE

13692 We suppose that  $g = 1, 2, \dots, G$  populations, having sizes  $N_g$ , state-spaces  $S_g$  are sampled  
 13693 using some capture-recapture method producing sample sizes of  $n_g$  unique individuals and  
 13694 encounters  $y_{ijk}$  for individual  $i = 1, 2, \dots, \sum_{g=1}^G n_g$ . Right now we won’t be concerned  
 13695 with the details of every type of capture-recapture observation model so, for context,  
 13696 we consider the Bernoulli model in which individual and trap-specific encounter frequen-  
 13697 cies are binomial counts:  $y_{ij} \sim \text{Binomial}(K, p_{ij})$ . Let  $g_i$  be a covariate (integer-valued,  
 13698 1, ...,  $G$ ) indicating the population membership of individual  $i$ . This covariate is *observed*  
 13699 for the sample of captured individuals but not for individuals that are not captured.

13700 A key idea that we develop shortly is that the assumption of certain models for the  
 13701 collection of abundance variables  $N_g$  *implies* a specific model for the population mem-  
 13702 bership variable  $g_i$ . Then, the data from all populations can essentially be pooled, and  
 13703 analyzed as data from a single population with the appropriate model on  $g_i$ , without  
 13704 having to model the  $N_g$  parameters *directly*. In this way, we can easily build hierarchical  
 13705 models for stratified populations, using an *individual* level parameterization of the model.  
 13706 Obviously this is important for SCR models as they all possess at least onee random effect  
 13707 in the form of the activity center **s**. Moreover, in the context of stratified or multi-session  
 13708 type models, the “population membership” variable  $g_i$  is a *categorical* type of individual  
 13709 covariate (Huggins 1989; Alho 1990; Royle 2009).

13710 To illustrate the prototypical data structure for stratified SCR data, we suppose that  
 13711 a population comprised of 4 sub-populations is sampled  $K = 5$  times. Then a plausible  
 13712 data set has the following structure:

```
13714     individual (i) : 1 2 3 4 5 6 7 8 9 10
13715     total encounters (y) : 1 1 3 1 1 2 2 4 1 1
13716     group (g)       : 1 1 1 2 3 3 3 3 4 4
```

13717 This data set indicates three individuals were captured in subpopulation 1 (captured 1,  
 13718 1, and 3 times), a single individual was captured in population 2, four individuals were  
 13719 captured in population 3, and two individuals were captured in subpopulation 4.

## 16.2 MULTINOMIAL ABUNDANCE MODELS

13720 The Poisson GLM is commonly used throughout ecology to model variation in abundance.  
 13721 Consider sampling  $g = 1, 2, \dots, G$  populations having unknown sizes  $N_g$ :

$$N_g \sim \text{Poisson}(\lambda_g) \quad (16.2.1)$$

13722 with

$$\log(\lambda_g) = \beta_0 + \beta_1 x_g \quad (16.2.2)$$

13723 where  $x_g$  is some measured attribute for population  $g$ . Under this Poisson model, by  
 13724 conditioning on the total population size over all  $G$  populations, the  $N_g$  variables have a  
 13725 multinomial distribution:

$$\mathbf{N} = (N_1, \dots, N_G) | \{N_T = \sum_g N_g\} \sim \text{Multinom}(\boldsymbol{\pi} | N_T). \quad (16.2.3)$$

13726 with multinomial probabilities  $\pi_g = \lambda_g / \sum_g \lambda_g$ . This relationship between Poisson and  
 13727 multinomial random variables is a standard distribution theory result.

13728 **XXXX below here needs edited and reworked XXXXX XXXX don't need  
 13729 any of this ... just the z/g model cite back to R/C papers XXXX**

13730 To devise a data augmentation scheme for this model of population size, we embed  
 13731 the multinomial for  $\{N_s\}$  into a multinomial of the same dimension but with larger, fixed  
 13732 sample size. Specifically, we introduce a latent super-population variable  $M_s$  which we  
 13733 assume has the desired Poisson distribution but with scaled mean:  $M_s \sim \text{Poisson}(A\lambda_s)$   
 13734 where  $A \gg 1$  where  $A$  exists (can be chosen) to ensure that  $M_s$  is arbitrarily larger  
 13735 than  $N_s$ . Conditional on the total super-population size  $M_T = \sum_s M_s$ , then  $\mathbf{M}$  has a  
 13736 multinomial distribution:

$$\mathbf{M} | M_T \sim \text{Multinom}(M_T; \boldsymbol{\pi}) \quad (16.2.4)$$

13737 where  $\pi_s = \lambda_s / \sum_s \lambda_s$  which are the same probabilities as for the target multinomial  
 13738 for  $\mathbf{N}$ . This multinomial model for the super-population sizes  $M_s$  is equivalent to the  
 13739 following:

$$g_i \sim \text{Categorical}(\boldsymbol{\pi})$$

13740 for  $g_i; i = 1, 2, \dots, M_T$ . Given  $\mathbf{M}$  or, equivalently,  $g_i$ , we specify a model for  $\{N_g\}$  that  
 13741 differentiates between "real" and "pseudo-" individuals by a Bernoulli sampling model:

$$N_s \sim \text{Binom}(M_s, \psi)$$

13742 where  $\psi \sim \text{Uniform}(0, 1)$ . Bernoulli sampling preserves the marginal Poisson assumption  
 13743 (Takemura 1999). That is,  $N_s$  is Poisson, unconditional on  $M_s$  and, also, conditional  
 13744 on  $N_T = \sum_s N_g$ ,  $\mathbf{N}$  has a multinomial with probabilities  $\boldsymbol{\pi}$  and index  $N_T$ . Note also  
 13745 that  $N_T \sim \text{Binom}(M_T, \phi)$  which is consistent with data augmentation applied to total  
 13746 population size  $N_T$ . This binomial sampling model can be represented, equivalently, by  
 13747 the set of Bernoulli variables:

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

13748 for  $i = 1, 2, \dots, M_T$ .

13749 The multinomial construction makes it clear that  $\psi$  is confounded with  $\exp(\beta_0)$ . By  
 13750 constructing the model conditional on the total, we lose information about the intercept

$\beta_0$ , but this is recovered in the data augmentation parameter  $\psi$ . One of these parameters has to be fixed. We can set  $\beta_0 = 0$  or else we can fix  $\psi$ . The constraint can be specified by noting that, under the binomial data augmentation model  $\mathbb{E}(N_T) = \psi M_T$  and, under the Poisson model,  $\mathbb{E}(N_T) = \sum_g \exp(\beta_0 + \beta_1 x_g)$  and so we can set

$$\psi = \frac{1}{M_T} \sum_g \exp(\beta_0 + \beta_1 x_g).$$

The equivalence of  $\psi$  and  $\beta_0$  can be thought of in terms of pooling data from the different sub-populations. In a model with *no* covariates, we could pool all of the data and estimate a single parameter  $\psi$  or  $\beta_0$  but not both. In this sense, pooling data from multiple spatial samples is justifiable (in terms of sufficiency arguments) under a Poisson assumption on local abundance (which was noted by Royle 2004b; Royle and Dorazio 2008, sec. 5.5.1).

By introducing the latent  $M_g$  structure, and the Bernoulli sampling of  $N_s$ , the model is equivalently represented by the latent variable pair  $(g_i, z_i)$  where  $g_i$  is categorical with prior probabilities  $\pi_s$  and  $z_i \sim \text{Bern}(\psi)$ . In particular, the multinomial assumption for the latent variables  $G_s$  is formulated in terms of “group membership” for each individual in the super-population of size  $M$  according to:

$$g_i \sim \text{Categorical}(\boldsymbol{\pi})$$

with  $\boldsymbol{\pi} = (\pi_1, \dots, \pi_S)$  and  $\pi_s = \lambda_s / (\sum_s \lambda_s)$ . Note that aggregating these  $M$  categorical variables yields a set of multinomial variables consistent with Eq. 16.2.4. That is, define  $G_1 = \sum_{i=1}^M I(g_i = 1)$ ,  $G_2 = \sum_{i=1}^M I(g_i = 2)$ , etc., where  $I()$  is the indicator function. The binomial sampling from the super-population,  $N_T \sim \text{Binom}(M, \psi)$  can be described at the level of the individual also, by introducing the binary variables  $z_1, \dots, z_M$  such that

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

where  $\psi$  is constrained as noted in the previous section. We implement this individual-level formulation of the model in BUGS in Panel 16.1.

A second implementation of the model is suggested by working from Eq. (16.2.3) – we can marginalize  $N_T$  over the prior  $N_T \sim \text{Binom}(M, \phi)$  to see that the  $(S+1) \times 1$  vector  $(N_1, \dots, N_S, N_{S+1})$  has, conditional on  $M$ , a multinomial distribution with cell probabilities  $\pi_s^+ = \pi_s \psi$  for  $s = 1, 2, \dots, S$  and  $\pi_{S+1}^+ = (1 - \psi)$  for the last cell which corresponds to individuals of the super-population that are not members of any of the  $S$  populations that were subject to sampling. Thus,

$$\mathbf{N}|M \sim \text{Multinom}(\boldsymbol{\pi}^+).$$

where the superscript + here indicates that  $\boldsymbol{\pi}^+$  is a larger version of  $\boldsymbol{\pi}$  from 16.2.4. In this case,

$$g_i \sim \text{Categorical}(\boldsymbol{\pi}^+) \text{ for } i = 1, \dots, M_T \quad (16.2.5)$$

The two distinct implementations are shown in Panel 16.1 for an ordinary closed population model (model  $M_0$ ).

### 16.2.1 Observation Models

Any observation model is cool here, no worries. We show a multinomial model below.

Bernoulli model.... replace binom command in WB code with a double-loop, y[i,j] etc...

---

```

model {
  # This will show that psi and b0
  #   are confounded.
  p ~ dunif(0,1)
  b0 ~ dnorm(0,.1)
  b1 ~ dnorm(0,.1)
  psi ~ dunif(0,1)
  for(s in 1:S){
    log(lam[s]) <- b0 + b1*x[s]
    gprobs[s] <- lam[s]/sum(lam[1:S])
  }
  for(i in 1:M){
    g[i] ~ dcat(gprobs[])
    z[i] ~ dbern(psi)
    y[i] ~ dbin(mu[i], J)
    mu[i] <- z[i]*p
  }
  N <- sum(z[1:M])
}

model {
  # This version constrains psi with
  #   the intercept parameter
  p ~ dunif(0,1)
  b0 ~ dnorm(0,.1)
  b1 ~ dnorm(0,.1)
  psi <- sum(lam[])/M
  for(j in 1:K){
    log(lam[j]) <- b0 + b1*x[j]
    gprobs[j] <- lam[j]/sum(lam[1:K])
  }
  for(i in 1:M){
    g[i] ~ dcat(gprobs[])
    z[i] ~ dbern(psi)
    y[i] ~ dbin(mu[i], J)
    mu[i] <- z[i]*p
  }
  N <- sum(z[1:M])
}

```

---

Panel 16.1: BUGS model specification for a capture-recapture model with constant encounter probability and Poisson subpopulation sizes,  $N_k$ , with mean depending on a single covariate  $x[j]$ . 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 on the RHS.

**13786 16.2.2 Simulating group structured capture-recapture data**

13787 It is helpful, as always, to simulate some data in order to understand the model. Suppose  
 13788 we carry-out a trapping study, say using hair-snares within baited wooden boxes for a  
 13789 species of *Mustelid*, and we establish arrays of 25 hair snares organized in an opportunistic  
 13790 along stream networks within 20 watersheds. Actually, we didn't know anything about  
 13791 SCR when we did this study but we set up hairsnares to be at least 1 stream km apart  
 13792 from each other, based on a systematic sampling of all stream and wetland boundaries.  
 13793 The main objective is to study the effect of development on mink density, measured by  
 13794 building structures per  $km^2$ , but each watershed also differs in the amount of available  
 13795 habitat which we characterize the km of stream plus km of lake, pond and major wetland  
 13796 shoreline. We imagine that

$$\log(\lambda_{ag}) = \log(area_g)\beta_0 + \beta_1 habitat + \beta_2 development$$

13797 simulate..... R script

**13798 16.2.3 Fitting in BUGS**

13799 Bernoulli observation model.....  
 13800 Fit the model in WinBUGS with about 10 lines of code.....

**13801 16.2.4 Approach B modeling  $\psi$** 

13802 Another idea here is to model the DA parameter  $\psi$  as being variable for each subject as  
 13803 a function of group-specific variables (Tenan ref....?). That is, if  $x_g$  is the value of some  
 13804 stratum covariate then we could have  $z_i \sim Bern(\psi_i)$  with

$$\text{logit}(\psi_i) = \beta_0 + \beta_1 x_{gi}$$

13805 This implies a binomial model for the stratum population sizes:

$$N(g) \sim Bin(psi(g), M)$$

13806 and also a multinomial for the vector  $N_1, \dots, N_G, M - \sum N$  with probabilities  $\psi_g$  and, for  
 13807 the last cell,  $1 - \sum_g \psi_g$ . This is almost the same multinomial as produced by the other  
 13808 approach. Also, if  $M$  is sufficiently large then the  $N(g)$  are approximately independent  
 13809 Poisson random variables with means  $\psi_g M$ .

**16.3 SPATIAL CAPTURE-RECAPTURE**

13810 We describe a model for the encounter histories conditional on knowing to which pop-  
 13811 ulation each observed individual belongs. Let  $\mathbf{y}_{ik} = (y_{i1k}, y_{i2k}, \dots, y_{ijk})$  be the spatial  
 13812 encounter history for individual  $i$ , a sequence of 0's and 1's for individual  $i$  during sample  
 13813  $k$ .

13814 A standard type of model which applies to detection devices such as hair snares  
 13815 (Borchers and Efford 2008; Gardner et al. 2010) is that in which the  $y_{ijk}$  are independent  
 13816 Bernoulli trials so that an individual can be captured in any number of the  $J$  traps during

13817 a sample occasion. In that case, the probability of encounter in trap  $j$  is modeled by some  
 13818 function of the distance between trap  $j$  (a 2-dimensional vector  $\mathbf{x}_j$ ), and the individual  
 13819 activity center  $\mathbf{s}_i$  which is regarded as a latent variable (i.e., “random effect”) in the model  
 13820 (Borchers and Efford 2008; Royle and Young 2008). For example, a common model is the  
 13821 “half-normal” model:

$$\Pr(y_{ijk} = 1) = p_{ijk} = p_0 \exp\left(-\frac{1}{2\sigma^2} \text{dist}(\mathbf{x}_j, \mathbf{s}_i)^2\right)$$

13822 Or, equivalently, we can express this as a linear function on a suitable scale:  $\log(p_{ijk}) =$   
 13823  $\alpha_0 + \alpha_1 \text{dist}(\mathbf{x}_j, \mathbf{s}_i)^2$  where  $\alpha_0 = \log(p_0)$  and  $\alpha_1 = 1/(2\sigma^2)$ .

13824 For standard live traps – also called “single catch” traps (Efford 2004), an individual  
 13825 can be captured in at most one trap. Then, the vector  $(y_{i1k}, y_{i2k}, \dots, y_{iJk}, y_{i,J+1,k})$ ,  
 13826 where the last element  $y_{i,J+1,k}$  corresponds to “not captured”, contains a single 1 and the  
 13827 remaining values are 0. This  $(J + 1) \times 1$  vector  $\mathbf{y}_{ik}$  is a multinomial trial:

$$\mathbf{y}_{ik} \sim \text{Multinomial}(n = 1; \boldsymbol{\pi}_{ik})$$

13828 where  $\boldsymbol{\pi}_{ik}$  is a  $(J + 1) \times 1$  vector where each element represents the probability of being  
 13829 encountered in a trap (for elements 1, ...,  $J$ ) or not captured at all (element  $J + 1$ ).

13830 For the multinomial case, we also model the encounter probability vector as a function  
 13831 of distance between trap locations and individual activity centers, but for this case we use  
 13832 the multinomial logit transform. The equivalent half-normal model is:

$$\text{mlogit}(\pi_{ij}) = \eta_{ij} = \alpha_0 + \alpha_1 * \text{dist}(\mathbf{x}_j, \mathbf{s}_i)^2 \quad (16.3.1)$$

13833 where  $\alpha_1 = 1/(2\sigma^2)$  and  $\sigma$  is the scale parameter of the half-normal detection model.  
 13834 Then

$$\boldsymbol{\pi}_{ij} = \exp(\eta_{ij}) / [1 + \sum_j \exp(\eta_{ij})]$$

13835 for each  $j = 1, 2, \dots, J$ , and the last cell corresponding to the event “not captured” is:

$$\pi_{i,J+1} = 1 - \sum_{j=1}^J \pi_{ij}$$

13836 It is easy to build additional covariates into this model including those that vary by  
 13837 sample occasion. For example, to model a behavioral effect (which we do in the example  
 13838 below), let  $C_{ik}$  be a covariate of previous encounter (i.e.,  $C_{ik} = 0$  before the occasion of  
 13839 first capture, and  $C_{ik} = 1$  thereafter), then

$$\text{mlogit}(\pi_{ijk}) = \eta_{ijk} = \alpha_0 + \alpha_1 * \text{dist}(\mathbf{x}_j, \mathbf{s}_i)^2 + \alpha_2 * C_{ik}$$

13840 We note, in this case, the multinomial probabilities depend not only on individual and  
 13841 trap, but also on sample occasion.

## 16.4 APPLICATION

13842 Single-catch traps  
 13843 XXX move this stuff XXXXX

13844 For the single-catch system, the independent multinomial model we employed is a  
 13845 misspecification of the true observation model. This is because competition for single-  
 13846 catch traps renders the independence assumption invalid. As Efford et al. (2009) noted,  
 13847 we expect “bias to be small when trap saturation (the proportion of traps occupied) is low.  
 13848 Trap saturation will be higher when population density is high...”, relative to trap density,  
 13849 or when net encounter probability is high. Efford et al. (2009) did a limited simulation  
 13850 study and found essentially no effective bias and concluded that estimators of density  
 13851 from the misspecified independent multinomial model are robust to the mild dependence  
 13852 induced when trap saturation is low. Conversely, properly specifying the likelihood for the  
 13853 correct single-catch system is challenging and, so far, has eluded formal characterization  
 13854 by researchers.

13855 XXXXX

13856 We applied this model to data described by Converse et al. (2006). The data were  
 13857 collected as part of an effort to understand the impacts of fuel reduction treatments on  
 13858 small mammal populations at 2 replicate study sites in northern New Mexico (Fig. 16.1;  
 13859 the Jemez Mountains Study area of the National Fire and Fire Surrogate Study; McIver  
 13860 et al. 2008), with trapping over 3 years (2001-2003) in each of 4 replicate experimental  
 13861 units per study site (i.e., number of groups  $G = 24$ , 8 units by 3 years, in this exam-  
 13862 ple). The experimental design included plans for thinning, burning, and thinning/burning  
 13863 combination treatments, as well as a control, at each study site. However, during the  
 13864 period when these data were collected, the thinning only treatment was completed on a  
 13865 single experimental unit at the JM-B study area (see Converse et al. 2006:1713), and at  
 13866 the JM-C study area, all 4 study experimental units were burned in a wildfire. Both the  
 13867 thinning treatment and the wildfire took place between the 2002 and 2003 study seasons.

13868 Trapping was conducted over 10 occasions (2 per day) at each experimental unit, with  
 13869 half the units at each site randomly selected in the first year for trapping in trapping  
 13870 session 1, which lasted 5 days, and half in session 2, an additional 5 days. The assignment  
 13871 to session then alternated over years. In 2001, the traps in each experimental unit were  
 13872 configured in a 6 by 6 grid, with 50 m between each trap. After a pilot project to assess  
 13873 the effects of trap spacing (Converse et al. 2004) the trap density was increased such that  
 13874 there was 25 m between traps, and so the grid was an 11 by 11 grid with 121 total trap  
 13875 stations. Multiple species were captured in the grids, but we base our analyses on the  
 13876 species with the largest number of captures, the deer mouse (*Peromyscus maniculatus*).

13877 The detection model is related to covariates through the multinomial logit transform in  
 13878 which the trap-specific encounter probabilities are given by Eq. 16.3.1. In the application  
 13879 we have

$$\eta_{ijk} = \alpha_{0,g_i} + \alpha_1 * C_{ik} + \alpha_{2,g_i} * d_{ij}^2$$

13880 where  $d_{ij} \equiv dist(\mathbf{s}_i, \mathbf{x}_j)$ ,  $\alpha_{0,1}, \dots, \alpha_{0,G}$  are group-specific intercepts,  $\alpha_1$  is the behavioral  
 13881 response parameter,  $C_{ik}$  is a covariate of previous encounter (i.e.,  $C_{ik} = 0$  before the  
 13882 occasion of first capture, and  $C_{ik} = 1$  thereafter), and  $\alpha_{2,g_i}$  is a group-specific coefficient on  
 13883 distance (related to  $\sigma_{g_i}$  by:  $\alpha_{2,g_i} = 1/(2\sigma_{g_i})$ ), allowing for the possibility that treatments  
 13884 influence home range size.

13885 To accommodate differences in trap array configuration (e.g.,  $6 \times 6$  vs.  $11 \times 11$  grids),  
 13886 we introduce a trap-operation matrix,  $\mathbf{A}$  where  $A_{j,k}^g = 1$  if, for group  $g$ , trap  $j$  is oper-  
 13887ational during period  $k$  and  $A_{j,k}^g = 0$  otherwise. A similar approach could be used if, in  
 13888 practice, certain traps were not operational during certain occasions. This could occur, for

example, if traps were sprung or damaged by animals. Then we include trap availability as multiplying  $\exp(\eta_{ijk})$  so that, in the multinomial logit transform, the cell probability is zeroed out for an inoperative trap.

For the abundance model, we assume that  $N_g$  is Poisson with mean

$$\lambda_g = \exp(\beta_{0,g} + \mathbf{x}'_g \boldsymbol{\beta})$$

where  $\beta_{0,g}$  is a group-specific random effect (see below),  $\mathbf{x}'_g$  is a vector of population-specific covariates, and including an intercept. In our analysis here,  $\mathbf{x}_g = (\text{year1}_g, \text{year2}_g, \text{thin}_g, \text{fire}_g)$  where **year1** and **year2** are dummy variables indicating years 2001 and 2002) i.e., **year1**<sub>*g*</sub> = 1 if group *g* occurred in 2001, **season2**<sub>*g*</sub> = 1 if group *g* occurred in 2002; **thin** and **fire** are binary treatment effects being **thin**<sub>*g*</sub> = 1 if group *g* was a thinned experimental unit, and **fire**<sub>*g*</sub> = 1 if group *g* was a burned experimental unit.

We used proper uniform prior distributions for each of the regression coefficients:  $\beta_m \sim \text{Unif}(-10, 10)$  for  $m = 1, 2, 3, 4$ ,  $\alpha_1 \sim \text{Unif}(-10, 10)$ , and  $\alpha_2 \sim \text{Unif}(-10, 10)$ . For the group-specific intercept parameters  $\beta_{0,g}$  we assumed:

$$\beta_{0,g} \sim \text{Normal}(0, \tau_\lambda)$$

with  $\sigma_\lambda = (1/\sqrt{\tau_\lambda}) \sim \text{Unif}(0, 10)$ . The mean of the normal distribution for  $\beta_{0,g}$  is 0 because the intercept of the abundance model is confounded with the data augmentation parameter  $\psi$ . That is,  $\psi$  is providing the information on the total abundance which is equivalent information to the intercept in the abundance model (Royle et al. 2012). The effect of this group-specific random effect is to induce extra-Poisson variation in the group-specific abundance parameters  $N_g$ . It is convenient to use the normal distribution on the  $\log(\lambda)$  scale here but a gamma noise term multiplying  $\lambda$  is equivalent to a negative binomial abundance model (Royle et al. 2012). For the group-specific intercept parameter  $\alpha_0$  we assumed then to be independent with normal prior

$$\alpha_{0,g} \sim \text{Normal}(\mu_p, \tau_p)$$

and flat priors on the hyperparameters  $\mu_p$  and standard-deviation:  $\mu_p \sim \text{Unif}(-10, 10)$ ,  $\sigma_p = 1/\sqrt{\tau_p} \sim \text{Unif}(0, 10)$ . We assumed a normal prior for  $\alpha_{2,g}$  also, having parameters  $\mu_{\alpha_2}$  and standard deviation  $\sigma_{\alpha_2}$ .

### 16.4.1 Results

There was a positive response of deer mouse population density to both thinning ( $\beta_2$ ) and wildfire ( $\beta_3$ ) (Table 1, Figure 1). There were also reasonably strong annual effects on density. Overall density of the species, across all groups, was estimated to be 0.00025 per  $m^2$ , or 2.5 per ha. The conclusion that both thinning and fire had a positive effect on density of deer mice was consistent with the conclusion reached by Converse et al. (2006). We also found strong trap-happy responses (i.e., animals that had been trapped previously had a higher capture probability, see  $\alpha_1$ , in Table 1).

**Table 16.1.** Point estimates (posterior mode) and 95% credible intervals for parameters in the observation process portion of the model as well as the ecological process portion of the model, for the joint estimation and modeling of density of *Peromyscus* spp. on experimental units at the Jemez Mountains Study Area, New Mexico. See text for explanation of parameters.  
**Footnotes:** (a) Only 2 fixed season effects are separately estimable. The third effect =  $-1 * (\beta_1[\text{seas}1] + \beta_1[\text{seas}2])$ . (b) Overall abundance is summed across all 24 groups, each with an implied area = 12.25 ha. (c) Overall density is reported as individuals/m<sup>2</sup>.

Parameter	Estimate	95% Lower	95% Upper
<b>Observation Process</b>			
$\mu_p$	-1.85	-2.19	-1.57
$\sigma_p$	0.55	0.37	0.88
$\alpha_1$	0.22	0.05	0.41
$\mu_{\alpha_2}$	-1.28	-1.49	-1.07
$\sigma_{\alpha_2}$	0.46	0.34	0.68
<b>Ecological Process</b>			
$\sigma_\lambda$	0.17	0.05	0.46
$\beta$ [seas 1]	-0.60	-0.80	-0.43
$\beta$ [seas 2]	-0.17	-0.33	-0.01
$\beta$ [seas 3](a)	0.79	0.59	0.97
$\beta$ [fire]	0.60	0.14	1.03
$\beta$ [thin]	0.38	0.12	0.77
N (b)	747	708	797
Density (c)	2.54x10-4	2.41x10-4	2.71x10-4

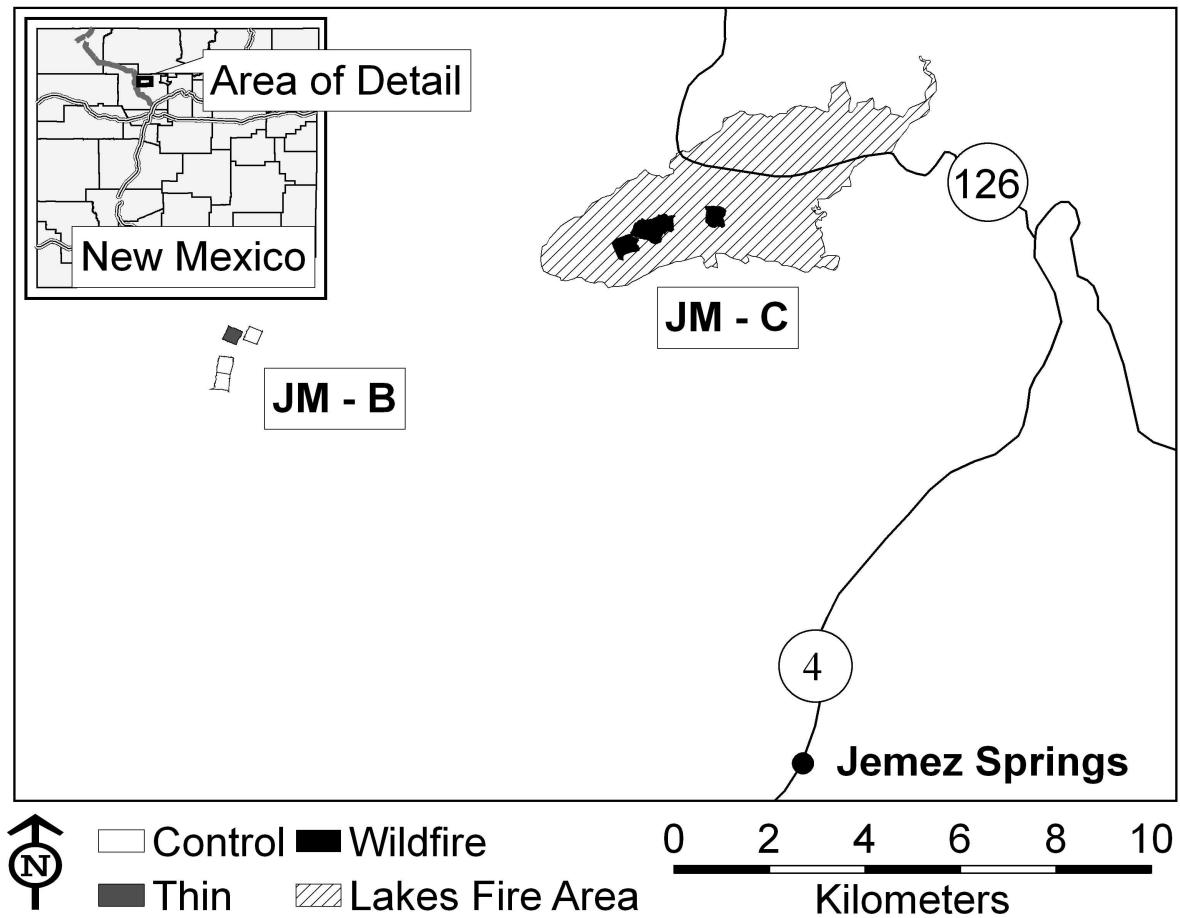


Figure 16.1. Central New Mexico Study Area from Converse et al. (2006).

**16.5 TOPICS IN MULTI-SESSION MODELS**

13922 The thing is this induces a slight bit of *dependence* among the counts but, for even a  
 13923 smallish number of populations , and of moderate size, the dependence is imperceptible  
 13924 and we think basically irrelevant from an inference standpoint.  
 13925 Over-dispersion. Everyone loves it. Could have normal random effect or Gamma  
 13926 distribution.

**16.5.1 Temporal models**

13928 The case here is we have  $g = 1, \dots, G$  samples over time but individuals are coming  
 13929 and going. We might capture some individuals over time but we ignore the individual  
 13930 recaptures across primary periods. (See chapter 15). So instead of modeling the dynamics  
 13931 at the individual level we just model net change in  $N_g$ .

**16.5.2 Dependence – is it a problem?**

13932 **XXX This should probably go in the body of the text or something XXXX**

13933 In time – ignoring the dependence of  $N_g$  probably entails a little *loss* of efficiency  
 13934 but should have no effect on anything. In space, there might be some individuals shared  
 13935 by multiple groups and we don't think that should cause any bias or anything, even in  
 13936 statement of uncertainty. So we view these models as pretty generally useful and relevant.  
 13937 A few points worth discussing: If you have grids that are in relatively close proximity you  
 13938 might want to build a model in which the total state-space is used in the model. i.e., form  
 13939 the union of the state-spaces and model that. That will be more computationally tedious  
 13940 but on the other hand it preserves the real landscape and any interactions that might be  
 13941 affecting grids simultaneously.

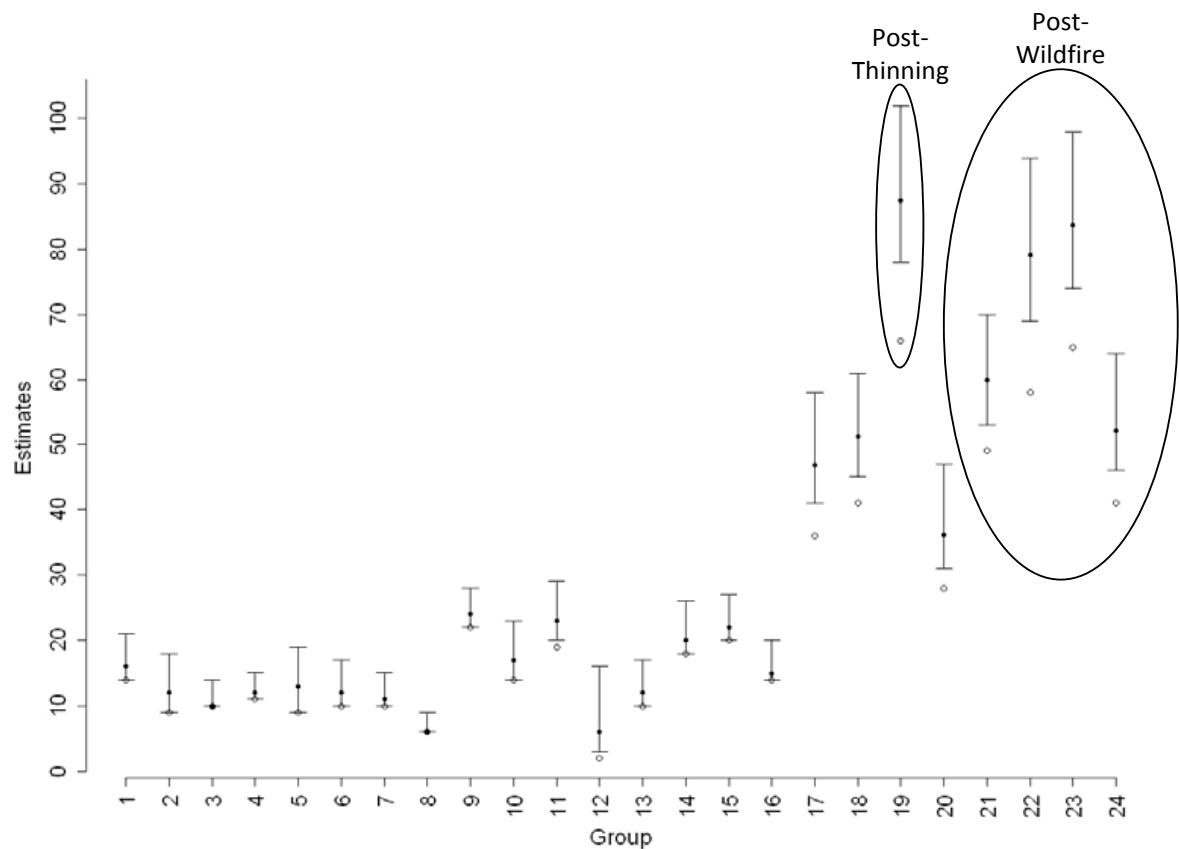
13942 Conceptually we can apply models like this which assume  $N_g$  are independent even if  
 13943 they're not... as long as we don't care about the underlying dynamics explicitly and also  
 13944 possibly with some loss of efficiency.

13945        GROUPS, STRATA, POPULATIONS, ETC...????

**16.6 MULTI-SESSION MODELS IN SECR**

13946 We talked about this back in sec. xxx and also in sec. xxxxx....

13947 The R package **secr** (Efford 2011) implements an estimator for “multiple sessions”  
 13948 that could be applied to data from multiple trap arrays or other meaningfully grouped  
 13949 data. The multi-session model in **secr** arises by an explicit Poisson assumption on  $N$ ,  
 13950 but uses a classical likelihood analysis in which the parameters  $N_g$  are removed from the  
 13951 likelihood by marginalizing the conditional-on- $N$  likelihood over a Poisson prior. One  
 13952 advantage of our Bayesian formulation based on data augmentation which enables direct  
 13953 implementation in widely available software (WinBUGS, JAGS, OpenBUGS) is that it  
 13954 is more versatile in terms of the model specification. For example, here we allowed for  
 13955 multiple group-specific random effects in the detection model, which is not accommodated  
 13956 in the **secr** package. As another (potential) example, we believe the model could be  
 13957 extended to open populations (Gardner et al. 2010) without much difficulty.



**Figure 16.2.** Abundance estimates for *Peromyscus* spp. per experimental unit (with area = 5.0625 ha) for each of 24 groups composed of 8 experimental units in year 1 (groups 1:8), and the same 8 experimental units in year 2 (groups 9:16) and year 3 (groups 17:24) at the Jemez Mountains Study Area, New Mexico. Point estimates (filled circles) are posterior modes, and error bars reflect 95% credible intervals. Also shown are the number of individuals captured per group (open circles).

**13959 16.6.1 Ovenbird data in WinBUGS?**

13960 Multi-catch observation model.... it is worth fitting this for sure.

**13961 16.6.2 Converse data in secr?**

13962 probably leave this out .... too much stuff..... but a simplified model would run a lot faster  
13963 in secr I bet.

**16.7 SUMMARY AND OUTLOOK**

13964 The other context is temporally indexed data – multi-session or group-structured models  
13965 are a simplified type of open model, one without explicit Markovian dynamics. The models  
13966 are not incorrect per se, but just simpler, reduced versions of the more general Markovian  
13967 models. We do cover the Markovian models in Chapt. 15

13968 SCR data are not always collected as single isolated studies but, rather, usually a  
13969 number of replicate trap arrays are used. Often this is motivated by specific objectives,  
13970 e.g., the trap arrays represent experimental replicates, and oftentimes just to obtain more  
13971 valid estimates of density by obtaining a representative sample of space within some region.  
13972 Thus there is a need to combine data from multiple arrays or sites in a single unified model  
13973 that accommodates explicit sources of variation in density among sites. This is naturally  
13974 accomplished by developing an explicit model for variation in  $N$ , e.g., a Poisson GLM or  
13975 similar.

13976 In this paper we extended SCR models to allow for modeling variation in  $N$  with  
13977 explicit assumptions on  $N$ . We adopt the data augmentation strategy from Converse and  
13978 Royle (2012), and Royle et al. (2012) and extended this to the spatial capture-recapture  
13979 observation model, and applied that model to data from a study of forest disturbance ef-  
13980 fects on small mammal populations. The framework for combining multiple sites is general  
13981 and will work for any kind of SCR observation model. We demonstrated a multi-catch  
13982 (ovenbird), single-catch (microtus) and a bernoulli (simulated data) model..... Imple-  
13983 mentation in a Bayesian framework allows for modeling of individual effects (and hence  
13984 makes SCR possible) but also facilitates efficient modeling of nuisance variation via hier-  
13985 archical structures (for example, on detection parameters, block effects, or time effects).  
13986 However, certain types of models can be fitted in secr easily, and .....

13987 Previously people always did ad hoc shit when confronted with multi-session types of  
13988 capture-recapture data. They would get Nhats and do regression on this. For example,  
13989 our small-mammal trapping case study comes from Converse et al (2006), who used a  
13990 3 step process to complete the analysis of these data: first a closed capture-recapture  
13991 analysis to estimate abundance, second an analysis of mean maximum distance moved  
13992 (Wilson and Anderson 1985) to allow conversion of abundance to density, and third a  
13993 weighted regression analysis of the resulting density estimates. The weighted analysis was  
13994 necessary to accomodate the non-zero sampling covariances resulting from the first 2 steps.  
13995 The analysis shown herein is both more streamlined and also integrates the improvements  
13996 that spatial capture-recapture methods bring to the estimation of capture-recapture data.  
13997 In addition, the Bayesian analysis we present makes the use of hierarchical structures  
13998 simple, such as the random effects for modeling variation in components of detection.  
13999 Converse and Royle (2012) showed that the use of random effects for modeling variation

<sup>14000</sup> in components of detection provides a good compromise between model complexity and  
<sup>14001</sup> parsimony, and can result in the lowest root mean square error in analyses of replicated  
<sup>14002</sup> capture-recapture data.



14003  
14004

14005  
14006

# 17

---

## MODELS FOR SEARCH-ENCOUNTER DATA

14007 In this chapter we discuss models for search-encounter data. These are models that arise  
14008 where you get actual location data of individuals not biased by trap locations but rather by  
14009 searching space in some fashion. In most cases both detection probability and movement  
14010 parameters are resolvable. i.e., models that preserve the these movement outcomes – the  
14011  $u_{it}$  variables. The models we differentiate here depend on a number of things related to  
14012 data structure or protocol – basically whether or not we record the exact location and how  
14013 we record it. All models have an underlying movement model which may be completely  
14014 latent or not.

14015 How exactly are these different from models for data from fixed arrays? (1) sample  
14016 units are either continuous space polygons or lines, not points; (2) we have location information  
14017 that is not biased by trap locations (but is biased by the observation device  
14018 somehow); (3) because we have direct observations of location that exist independent of  
14019 traps we can often build an explicit model of space usage or an explicit movement model.

14020 A few variations of the models exist – a long sample path through a sample region  
14021 where we note the locations of individuals seen along the way, *and their identity* (this is  
14022 different from distance sampling int hat sense). Or we could search a region systematically  
14023 and so forth. The canonical situation is Royle and Young (2008) which involved a plot  
14024 search for lizards. They assumed the plot was uniformly searched which justified the  
14025 assumption of constant  $p$  within the plot boundaries. The data set was  $\geq 1$  location  
14026 observations for each of a sample of  $n$  individuals. The recent paper by Efford XXXX  
14027 discussed likelihood analysis of similar models. In the jargon of `secr` such models are  
14028 referred to as models for *polygon detectors*. An extension of this model was described by  
14029 (Royle et al., 2011a).

14030 Search-encounter models also provide something of a bridge between the standard  
14031 models for fixed trap arrays (e.g., Chapt. 5), and the models of (Chandler and Royle,  
14032 In press) where no individual identity is present. The latter are search-encounter models  
14033 where the movement process (and outcomes) are completely latent. Another type of model  
14034 is SCR/DS – this is a SCR model with .....

## 17.1 SEARCH-ENCOUNTER SAMPLING DESIGNS

14035 For our purposes here we recognize 4 basic sampling designs, each of which might have  
14036 variations due to modification of the basic sampling protocol. In later sections of this  
14037 chapter we will do some examples but not of all of them.

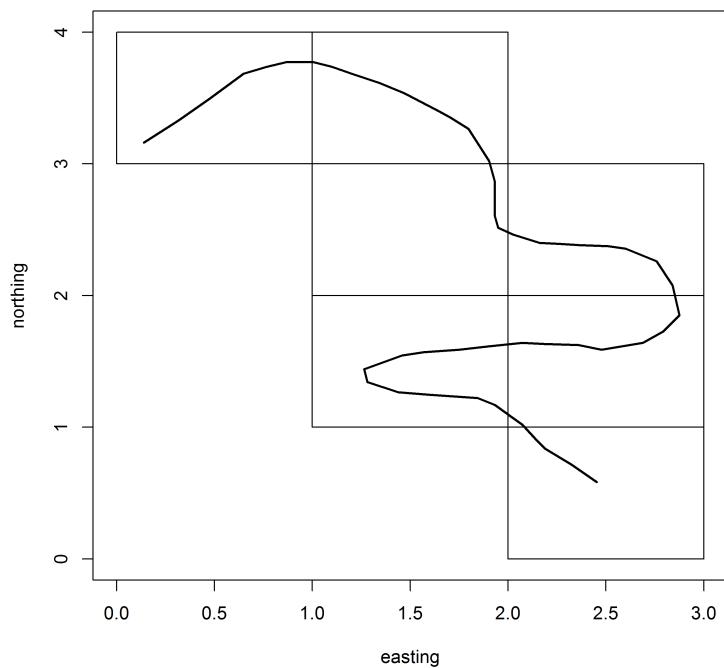
14038 **Design 1: Fixed Search Path.** The ideal situation is where we have a continuous  
14039 search-path or lines, or multiple such lines, in some region (Fig. XXXX 1 XXXXX). This  
14040 is the type of problem described by Royle et al. (2011 MEE). We assume the path or  
14041 lines are laid out a priori in some manner that is done independent of the activity centers  
14042 of individuals and the collection of data does not affect the lines. Sometimes the lines  
14043 are within well-defined polygons but the polygon boundaries are not meaningful in terms  
14044 of the observation process. A number of variations of the data collection protocol are  
14045 possible:

14046 Protocol (1a) has us just record the locations of individuals  
14047 Protocol (1b) has us record location of individuals AND location on the transect where  
14048 we observed the individual  
14049 Protocol (1c) has us record neither of those things, instead we record the closest perpen-  
14050 dicular distance. This is a typical distance sampling situation which produces exactly  
14051 a DS type of a model (or a CR-DS model). We don't recommend recording closest  
14052 perpendicular distance and we don't discuss these models too much here  
14053 Protocol (1d) . In this case, observations are restricted to the line itself. We imagine  
14054 that the line is evolving in response to search activity. It is not quite like the other  
14055 ones so let's call it "ad hoc". In this case we use small bins as traps and the length of  
14056 the line in each grid as a covariate. Unstructured survey data. Thompson et al. and  
14057 Russell et al.  
14058

14058 **Design 2: Uniform search intensity.** In this case we have one or more well-defined  
14059 sample areas (polygons), such as a quadrat or a transect, and we imagine that the area is  
14060 uniformly searched so that  $p = p_0$  is constant within the search area. Sampling produces  
14061 locations of individuals within the well-defined boundaries of the sample area. The polygon  
14062 boundaries defining the sample unit are important because it tells us that  $p = 0$  by design  
14063 outside of the boundary.

14064 Using the example from the Figure above, we could imagine that each quadrat was  
14065 uniformly searched. The individual quadrat boundaries are irrelevant and we only need  
14066 to be concerned about the "total" boundary of the composite polygon (the intersection of  
14067 all little ones). That said for analysis in BUGS it is easier to work with square polygons.  
14068 We show a simulation example here and we analyze it either using a bivariate normal  
14069 movement model or else a 2-d random walk type of model. But we don't provide a real  
14070 example as Royle and Dorazio 2008 did a reanalysis of the lizard data and see also Efford  
14071 (XXXX).

14072 **Design 3: bad implementation of 1 or 2** We set up search polygons (e.g., the grid  
14073 cells of above figure) and record locations of encountered individuals but we do not do a  
14074 uniform search of quadrats and we forgot to record the GPS path. Analysis of this design:  
14075 We imagine that we can assume a uniform search intensity here and maybe it won't be so



**Figure 17.1.** snake line.... showing design 1. more here.

14076 bad. We should do a simulation study of this somehow. I am working on methods to lay  
 14077 down some standard sets of lines for simulating data, and then ignoring the lines in doing  
 14078 an analysis. Alternative 2 for analysis: We could map each location to the CENTER of  
 14079 the grid cell and pretend this is trap array (traps at center of each grid). This was the  
 14080 idea of Kery et al. and some other papers.

14081 **Design 4: Really bad implementation of 1 or 2** In this case we screw up even  
 14082 further and forget to record the locations of individuals within a bunch of quadrats. I  
 14083 believe Richard has been thinking along these lines - using the underlying movement  
 14084 model as a latent model. There are two variations of Design 4:

14085 Protocol (4a) - We imagine that you could have counts BY individual identity within  
 14086 each quadrat. Not sure what analysis model this would be.

14087 Protocol (4b) - We don't have individual identities but just total counts. This is Chan-  
 14088 dler and Royle (2012/13).

14089 The capricailie example: search of polygons – could be search encounter with uniform  
 14090 search intensity but we ignored the polygon boundaries and just mapped each observation  
 14091 to the center point. The fisher data: we had a GPS line but it was not really fixed , it  
 14092 evolved as dogs searched around. Therefore as a practical matter the locations of samples  
 14093 were all *on* the line. We therefore mapped to a center point of a grid. We make a grid of the  
 14094 sampled area and we assume within each grid if a species is present then it is independent  
 14095 .... actually if the grid is placed INDEPENDENT of the lines then its probably safe to  
 14096 make some kind of independence assumption. Russell et al. – similar situation, they have  
 14097 a search parth but not really independent.

14098 For the rest of this chapter, we will provide some model formulations for some cases,  
 14099 provide code for simulating and analyzing the data, and some real examples but not for  
 14100 every situation. A number of published examples have been given. The Royle et al. 2011  
 14101 paper on the MHB. The Royle and Young 2008 (see also Marques et al. and Efford 2011).  
 14102 We also have the Thompson et al. XXX and Russell et al. XXXX and Capricaillie paper  
 14103 XXXXX.

14104 Possible examples to provide:

14105 Example 1: Analysis of the Swiss MHB survey using Design 1

14106 Example 1b: Lizard data. No need to analyze this as it was done in RD book. Mention  
 14107 polygon detectors in secr.

14108 Example 2: Fisher data possibly - lion data or – or Capricaillie data?

## 17.2 A MODEL FOR SEARCH-ENCOUNTER DATA

14109 We cover the basic Design 1 here which also is relevant to Design 2 as a special case....  
 14110 this comes from Royle et al. 2011.

14111 XXX t below has to be k XXXX

14112 Our approach is to parameterize a model for the encounter histories  $y_{ik}$  in terms of  
 14113  $\mathbf{u}_{ik}$ , the two-dimensional location of capture at the instant of sample,  $k$ . In contrast to  
 14114 most of the models describe in this book, we develop models for encounter probability that  
 14115 depend explicitly on the instantaneous location  $\mathbf{u}_{ik}$ , i.e.,  $p_{ik} \equiv p(\mathbf{u}_{ik}) = \Pr(y_{ik} = 1 | \mathbf{u}_{ik})$ .  
 14116 Note that  $\mathbf{u}$  is unobserved for the  $y = 0$  observations and thus we cannot analyze the

14117 conditional-on- $\mathbf{u}$  likelihood directly. Instead, we regard  $\mathbf{u}$  as random effects and assume  
 14118 a distribution for them, which allows us to handle the problem of missing  $\mathbf{u}_{ik}$  values.

14119 To develop encounter probability models for this problem we cannot just use the  
 14120 previous models because the “trap” is actually a line or collection of line segments (e.g.,  
 14121 Fig XXXX). Intuitively,  $\Pr(y_{ik} = 1 | \mathbf{u}_{ik})$  should increase as  $\mathbf{u}_{ik}$  comes “close” to the line  
 14122 segments  $\mathbf{X}$ . It seems reasonable to express closeness by some distance metric  $\|\mathbf{u}_{ik} - \mathbf{X}\| =$   
 14123  $dist(\mathbf{u}_{ik}, \mathbf{X})$  and then assume

$$\text{logit}(p_{ik}) = \alpha_0 + \alpha_1 \|\mathbf{u}_{ik} - \mathbf{X}\|.$$

14124 For the case where  $\mathbf{X}$  describes a wandering line, some kind of average distance from  $\mathbf{u}$  to  
 14125 the line might be reasonable; possible alternatives include the absolute minimum distance  
 14126 or the mean over specific segments of the line (within some distance), etc. Because the  
 14127 line  $\mathbf{X}$  is not a single point (like a camera trap) we have to somehow describe the total  
 14128 encounter probability to the line. We adopt a similar idea to the hazard modeling idea  
 14129 in survival analysis (also adopted in distance sampling by Hayes & Buckland (1983) and  
 14130 Skaug & Schweder (1999) and, in the context of arrays of fixed traps by Borchers & Efford  
 14131 (2008)). The individual is detected (analogous to mortality) if encountered at any point  
 14132 along  $\mathbf{X}$ . Naturally, covariates are modeled as affecting the hazard rate and we think of  
 14133 distance to the line as a covariate acting on the hazard. Let  $h(\mathbf{u}_{ik}, \mathbf{x})$  be the hazard of  
 14134 individual  $i$  being encountered by sampling at a point  $\mathbf{x}$  on occasion  $t$ . For example, one  
 14135 possible model assumes, for all points  $\mathbf{x} \in \mathbf{X}$ ,

$$\log(h(\mathbf{u}_{ik}, \mathbf{x})) = \alpha_0 + \alpha_1 * dist(\mathbf{u}_{ik}, \mathbf{x}). \quad (17.2.1)$$

14136 The total hazard to encounter anywhere along the survey path, for an individual located at  
 14137  $\mathbf{u}_{ik}$ , say  $H(\mathbf{u}_{ik})$ , is obtained by integrating over the surveyed line, which we will evaluate  
 14138 numerically by a discrete sum where the hazard is evaluated at the set of points  $\mathbf{x}_j$  along  
 14139 the surveyed path:

$$H(\mathbf{u}_{ik}) = \exp(\alpha_0) \left\{ \sum_{x_j} \exp(\alpha_1 * dist(\mathbf{u}_{ik}, \mathbf{x}_j)) \right\} \quad (17.2.2)$$

14140 where  $\mathbf{x}_j$  is the  $j^{th}$  row of  $\mathbf{X}$  defining the survey path as a collection of line segments  
 14141 which can be arbitrarily dense, but should be regularly spaced. Then the probability of  
 14142 encounter is

$$p_{it} \equiv p(\mathbf{u}_{it}) = 1 - \exp(-H(\mathbf{u}_{it})). \quad (17.2.3)$$

14143 This is a reasonably intuitive type of encounter probability model in that the probability  
 14144 of encounter is large when an individual’s location  $\mathbf{u}_{it}$  is close to the line in the average  
 14145 sense defined by Eq. (17.2.2), and vice versa. Note that  $p_{it}$  also depends on the sample  
 14146 path  $\mathbf{X}$ , i.e.,  $p(\mathbf{u}_{it}, \mathbf{X})$  which we suppress in our notation because  $\mathbf{X}$  is fixed for any specific  
 14147 analysis. We note that we don’t require all line segments are surveyed during each sample  
 14148 period, as this simply affects the construction of the encounter probability  $p$  for each  
 14149 sample. Thus, different line segments may be surveyed at different times, which results in  
 14150 considerable flexibility in the design of a survey. Additional covariates could be included  
 14151 in the hazard function. For example, in some situations observers might record weather  
 14152 conditions along the route, time-of-day, effort or other covariates (Kéry *et al.* 2005).

14153 This formulation of total hazard and encounter probability assumes that encounter  
 14154 at each point along the line,  $\mathbf{x}_j$ , is independent of each other point. Then, the event  
 14155 that an individual is encountered *at all* is the complement of the event that it is not  
 14156 encountered *anywhere* along the line (see also Hayes and Buckland 1983). In terms of  
 14157 the survival/hazard analogy, the survival function is  $S(\mathbf{u}_{ik}, \mathbf{x}_j) = \exp(-h(\mathbf{u}_{ik}, \mathbf{x}_j))$  and so  
 14158 the probability that an individual “survives” all  $J$  points is  $\prod_j \exp(-h(\mathbf{u}_{ik}, \mathbf{x}_j))$  and the  
 14159 encounter probability is therefore the complement of this, which is precisely the expression  
 14160 given by Eq. (17.2.3).

14161 Consider the case of a single survey point, i.e.,  $\mathbf{X} \equiv \mathbf{x}$ , which we might think of as a  
 14162 camera trap location. In this case note that Eq. (17.2.3) is equivalent to

$$\log(-\log(1 - p_{ik})) = \alpha_0 + \alpha_1 * dist(\mathbf{u}_{ik}, \mathbf{x})$$

14163 which is to say that distance is a covariate on detection that is linear on the complementary  
 14164 log-log scale, which is similar to the “trap-specific” encounter probability of our Bernoulli  
 14165 encounter probability model (see Chapt. 5). The difference is that, here, the relevant  
 14166 distance is between the “trap” (i.e. the survey lines) and the individual’s present location,  
 14167  $\mathbf{u}_{ik}$ , which is observable. On the other hand, in the context of camera traps, the distance  
 14168 is that between the trap and a latent variable,  $\mathbf{s}_i$ , representing an individual’s home range  
 14169 or activity center which is not observed.

### 14170 17.2.1 Ecological process model

14171 We have so far described the model for the encounter data in a manner that is conditional  
 14172 on the locations  $\mathbf{u}_{ik}$ , some of which are unobserved. That consideration alone justifies  
 14173 the need for a 2nd level model – a “random effects” distribution – for the  $\mathbf{u}_{ik}$  variables.  
 14174 In addition, biologically we expect that these variables should be correlated because they  
 14175 correspond to repeated measures on the same individual. To develop such a model, we  
 14176 adopt what is now customary in spatial capture-recapture problems – we assume that  
 14177 individuals are characterized by a latent variable,  $\mathbf{s}_i$ , which represents a center of activity  
 14178 or territory or simply “home range”. This leads to a natural model for the variables  $\mathbf{u}_{ik}$ .  
 14179 In particular, we can now think of  $\mathbf{u}_{ik}$  as the outcomes of a *movement process*, conditional  
 14180 on  $\mathbf{s}_i$ . Here we make use of the bivariate normal model:

$$\mathbf{u}_{ik} | \mathbf{s}_i \sim \text{Normal}(\mathbf{s}_i, \sigma^2 \mathbf{I}),$$

14181 where  $\mathbf{I}$  is the  $2 \times 2$  identity matrix. This is a primitive model of individual movements  
 14182 about their home range but, in most capture-recapture studies, we will only have one to  
 14183 several observations on each individual and thus very limited ability to estimate complex  
 14184 home range models. Therefore, we believe that the bivariate normal model will be sufficient  
 14185 for most real-life spatial capture-recapture problems.

14186 We adopt our now customary assumption for the activity centers  $\mathbf{s}$ :

$$\mathbf{s}_i \sim \text{Unif}(\mathcal{S}); \quad i = 1, 2, \dots, N.$$

14187 The usual considerations apply in specifying the state-space  $\mathcal{S}$  – either choose a large  
 14188 rectangle, or prescribe a habitat mask to restrict the potential locations of  $\mathbf{s}$ .

---

**17.2.2 Other stuff**

We have specified the model “conditional on  $N$ ”, where  $N$  is the total population of individuals residing in the state-space  $\mathcal{S}$ . We need to account for the fact that  $N$  is unknown which we do using our standard approach of data augmentation. As usual, under data augmentation, the observations  $y_{it} = 0$  correspond to an excess zero when  $z_i = 0$  and to a sampling zero when  $z_i = 1$  – in the latter case an individual is indeed a member of the population of size  $N$ . The known- $N$  observation model is modified from  $y_{it} \sim \text{Bern}(p_{it})$  (as above) to  $y_{it} \sim \text{Bern}(w_i p_{it})$  and the latent variables  $w_i$  for  $i = n + 1, \dots, M$  are updated with the remaining model parameters in the MCMC algorithm (see below).

Any model for encounter probability can be converted to a hazard model so that encounter probability based on total hazard can be derived. Royle et al. 2011 considered a bunch of other hazard models including that described previously

$$\log(h(\mathbf{u}_{it}, \mathbf{x})) = \alpha_0 + \alpha_1 * \text{dist}(\mathbf{u}_{it}, \mathbf{x}).$$

which is usually called the Gompertz hazard function in survival analysis, and it is most often written  $h(t) = a \exp(b * t)$  in which case  $\log(h(t)) = \log(a) + b * t$ . Model 2 (squared-distance) is a quadratic function of distance

$$\log(h(\mathbf{u}_{it}, \mathbf{x})) = \alpha_0 + \alpha_1 * \text{dist}(\mathbf{u}_{it}, \mathbf{x})^2.$$

We’ve used this model quite a bit in the book, and it implies a bivariate normal hazard rate model. Model 3 is from Borchers & Efford (2008):

$$h(\mathbf{u}_{it}, \mathbf{x}) = -\log(1 - \text{expit}(\alpha_0) \exp(\alpha_1 * \text{dist}(\mathbf{u}_{it}, \mathbf{x})^2))$$

which produces a normal kernel model for *probability of detection* at the point level. i.e.,  $\Pr(y = 1) = 1 - \exp(-h) = h_0 \exp(\alpha_1 * \text{dist}(\mathbf{u}_{it}, \mathbf{x})^2)$  where  $h_0 = \text{expit}(\alpha_0)$ . Model 4 is

$$\log(h(\mathbf{u}_{it}, \mathbf{x})) = \alpha_0 + \alpha_1 * \log(\text{dist}(\mathbf{u}_{it}, \mathbf{x}))$$

which is a Weibull hazard function.

### 17.3 EXAMPLES

We just simulate data and fit it in JAGS – in the repo.

Example: Simulator and WinBUGS code for this example [in repo]

```
> wbout
Inference for Bugs model at "model0.txt", fit using jags,
 3 chains, each with 5000 iterations (first 1000 discarded)
n.sims = 12000 iterations saved
      mu.vect sd.vect   2.5%    25%    50%    75%   97.5% Rhat n.eff
14216 N         94.448   5.237  81.000  92.000  96.000  98.000 100.000  1.006   520
14217 beta0     -0.539   0.743  -1.714  -1.072  -0.637  -0.102   1.258  1.042    54
14218 beta1    -11.943   2.196 -17.229 -13.236 -11.665 -10.375  -8.378  1.035    62
14219 psi        0.936   0.056   0.792   0.908   0.951   0.979   0.998  1.005   650
14220 sigma      0.340   0.043   0.268   0.309   0.337   0.366   0.436  1.004   820
```

14221 deviance 206.987 25.474 160.405 189.069 205.779 224.080 259.491 1.017 130  
 14222  
 14223 For each parameter, n.eff is a crude measure of effective sample size,  
 14224 and Rhat is the potential scale reduction factor (at convergence, Rhat=1).  
 14225  
 14226 DIC info (using the rule, pD = var(deviance)/2)  
 14227 pD = 319.4 and DIC = 526.3  
 14228 DIC is an estimate of expected predictive error (lower deviance is better).

14229 **17.3.1 Hard plot boundaries**

14230 1.3. Hard quadrat boundaries: Quadrat boundaries might be relevant or might not be.  
 14231 If they are then, for Bayesian analysis, a value of  $u$  outside the boundary has  $p$  forced to  
 14232 0, its as simple as that. In general, we define  $p[I,t] = p[I,t]*I(u \text{ in } X)$ . We see how this  
 14233 relates to the uniform search intensity model.  $P[I,t] = p_0$  then defines precisely the model  
 14234 of Royle and Young (2008).

14235 **Hard plot boundaries** – The previous development assumed that encounters can be  
 14236 made anywhere in space but that the encounter probability decreases with distance from  
 14237 the survey path. In practice, as in the MHB, we might delineate a plot which restricts  
 14238 where individuals might be observed (as in the situation considered by Royle & Young  
 14239 (2008)). For such cases we truncate the encounter probability function such as

$$p(\mathbf{u}_{it}) = (1 - \exp(-H(\mathbf{u}_{it})))I(\mathbf{u}_{it} \in \mathcal{X})$$

14240 where  $\mathcal{X}$  is the surveyed polygon and the indicator function  $I(\mathbf{u}_{it} \in \mathcal{X}) = 1$  if  $\mathbf{u}_{it} \in \mathcal{X}$  and  
 14241 0 otherwise. That is, the probability of encounter is identically 0 if an individual is located  
 14242 *outside* the plot at sample period  $t$ . Given this modified encounter probability function,  
 14243 it is clear that the model is a modified form of Royle & Young (2008) where their model  
 14244 – “uniform search intensity” – replaces the above expression with

$$p(\mathbf{u}_{it}) = p_0 I(\mathbf{u}_{it} \in \mathcal{X})$$

14245 Analysis of lizard data from Royle and Young ..... 2008

14246 **Multiple survey plots** – It is common in wildlife surveys to have multiple spatial  
 14247 sample units which need to be integrated into a single model. It is convenient if the  
 14248 population sizes for each plot are independent. In the case of the MHB data, the closest  
 14249 two plots were 10 km apart and, for this species, it is reasonable to assume independence.  
 14250 Moreover, the MHB plots represent (approximately) a random sample and thus indepen-  
 14251 dence is probably justified from a design-based perspective. With multiple plots, it is  
 14252 convenient computationally to organize the plots in some modified coordinate system that  
 14253 keeps them far enough apart so that individual movement outcomes cannot be located in  
 14254 multiple plots. This enables an implementation by data augmentation based on a single  
 14255 augmented data set. To construct the point process state-space, the 7 plots were embed-  
 14256 ded into a 30.8 km rectangular state-space having a minimum of 0.6 km buffer, which  
 14257 we judged to be sufficient given the estimate of  $\sigma$  (see below) so that individuals cannot  
 14258 appear in  $> 1$  plot during the MCMC simulation (i.e., 0.6 is large relative to the estimate  
 14259 of  $\sigma$ ).

---

**14260 17.3.2 Analysis of other protocols**

14261 Analysis of 1b is a distance-sampling like model but with an additional hierarchical struc-  
14262 ture the describes the individual location scatter about the home range center. This is  
14263 precisely a type of DS with measurement error. Analysis of 1c is a similar idea except it  
14264 represents an explicit model misspecification since one is approximating the observation  
14265 process by the nearest perpendicular to the line. Analysis of 1d is the “unstructured sur-  
14266 vey data” like from Thompson et al. or Russell et al. Note also that the capcrap paper is  
14267 a version of this - grids or polygons were sampled but no information on the search path  
14268 is available. This could be a Design 3 problem but that is excess computation I think.

14269 Protocol (1b) has us record location of individuals AND location on the transect where  
14270 we observed the individual. This is an easier problem I think, but you have to account for  
14271 “not seen” prior to  $x_0$  so maybe some kind of a cumulative hazard model or something.

14272 Protocol (1c) has us record neither of those things, instead we record the closest  
14273 perpendicular distance. This is a typical distance sampling situation which produces  
14274 exactly a DS type of a model (or a CR-DS model). We don’t recommend recording closest  
14275 perpendicular distance and we don’t discuss these models too much here

14276 Protocol (1d) . In this case, observations are restricted to the line itself. We imagine  
14277 that the line is evolving in response to search activity. It is not quite like the other ones  
14278 so let’s call it “ad hoc”. In this case we use small bins as traps and the length of the  
14279 line in each grid as a covariate. Thompson et al. and Russell et al. Simulation results

14280

---

**17.4 DESIGN 3: AD HOC IMPLEMENTATION OF DESIGN 1.**

14281 We don’t do anything new in terms of modeling here but we look at how bad do we do if  
14282 we don’t have the search path and use the USI model? We consider 4 cases: Case 1 and  
14283 2: regular searching of low and high intensity. E.g., for a 1 unit block, then we can have  
14284 a sinusoid track through each block of length 1.5 or 2 and then 4 or 5 km. For case 3 and  
14285 4 we use heterogeneity in search intensity.

**17.5 CAPRICAILLIE CRAP****17.6 DESIGN 4 – NO LOCATION INFO**

14286 We imagine a series of models for situations where we forget altogether to record location  
14287 information within the sample unit. We further assume the design was such that the  
14288 sample units represent contiguous quadrats or at least close enough together so that  
14289 individuals may be counted in multiple units. The idea here is that by being exposed to  
14290 multiple units, there is a spatial dependence induced and this spatial dependence provides  
14291 a little bit more of information about model parameters.

14292 We have two specific cases here: Imagine we have a bunch of quadrats or segments  
14293 that are contiguous and we do the surveys like above and record counts PER individual  
14294 but no other sampling information. Not sure what to do about this. The other case is  
14295 that we don’t record individual ID at all – instead we just have total count frequencies in  
14296 each plot. This model is precisely the one considered by (Chandler and Royle, In press)  
14297 and this is the focus of Chapt. 18.

14298      Comments on inference for the first situation?

**17.7 SUMMARY AND OUTLOOK**

<sup>14299</sup> Searching space for scat is , we imagine, the future of all animal sampling.  
<sup>14300</sup> SCR/DS?

14301  
14302

14303  
14304

# 18

---

## SPATIAL CAPTURE-RECAPTURE FOR UNMARKED POPULATIONS

14305  
14306  
14307  
14308  
14309  
14310  
14311  
14312

### todo: Royle-Nichols observaiton model

Traditional capture-recapture models share the fundamental assumption that each individual in a population can be uniquely identified when captured. Often, this can be accomplished by marking individuals with color bands, ear tags, or some other artifical mark that can be subsequently read in the field. For other species, such as tigers or marbled salamanders, individuals can be easily identified using only their natural markings, yet many species do not possess adequate natural markings and are difficult to capture, making it impractical to use standard capture-recapture techniques.

14313  
14314  
14315  
14316  
14317  
14318  
14319  
14320  
14321  
14322  
14323  
14324

Estimating density when individuals are unmarked can be accomplished using a variety of alternatives to capture-recapture, but many of these methods have important limitations that warrant the exploration of alternative approaches. In this chapter we highlight the work of Chandler and Royle (In press) who demonstrated that the “individual recognition” assumption of capture-recapture models is not a requirement of spatial capture-recapture models. They showed that, under certain conditions described below, spatially-correlated count data are sufficient for making inference about animal distribution and density even when no individuals are marked. The Chandler and Royle (In press) “spatial count model” (hereafter the SC model) is virtually identical to other SCR models except that the encounter histories  $\{z_{ijk}\}$  are not directly observed. Instead, the observed data are the counts realized by summing up the detections for each individual at a survey location during a sampling occasion  $n_{jk} = \sum_i z_{ijk}$ .

14325  
14326  
14327  
14328  
14329  
14330  
14331

The ability to fit SCR models to data from unmarked populations has important implications. For one, it means that SCR models can be applied to data collected using methods like points counts in which observers record simple counts of animals at an array of survey points. Camera trapping data on unmarked animals such as deer or coyotes could also also be suitable. In addition, this development has important implications for traditional SCR studies because many SCR datasets include some individuals that cannnot be identified due to poor photo quality or indistiguishable natural markings.

14332

It is also interesting to note that by disregarding individual identity, we wind up with

14333 a model that closely resembles another large class of spatial models, known as convolution  
14334 models (Wolpert and Ickstadt, 1998; Higdon, 1998). These models have been used for a  
14335 variety of purposes such as describing oceanic surface temperatures and correlation in tree  
14336 locations within managed forests. The SC model offers an improvement in some respects  
14337 over existing convolution models because it does not require arbitrary decisions about the  
14338 location and number of “support points”. We will clarify this later in the chapter, and  
14339 briefly mention how this model can be used outside of SCR contexts for general purpose  
14340 spatial modeling of correlated count data.

## 18.1 EXISTING MODELS FOR INFERENCE ABOUT DENSITY IN UNMARKED POPULATIONS

14341 When capture-recapture methods are not a viable option, researchers often collect sim-  
14342 ple count data or even detection/non-detection data to estimate population parameters.  
14343 These data are often analyzed using Poisson regression or logistic regression, perhaps with  
14344 random effects. When detection is imperfect, as it almost always is, these methods cannot  
14345 be used to obtain unbiased estimates of population size or occurrence probability. Even  
14346 when these data are used an index of abundance or occurrence, standard models may yield  
14347 unreliable results when covariates affect both the state variable and detection probability.  
14348 A classic example is the finding by Bibby and Buckland (1987) who reported that the de-  
14349 tention probability of songbirds in restocked conifer plantations was negatively associated  
14350 with vegetation height, yet population density was positively related to vegetation height.  
14351 This intuitive and common phenomenon has led to the development of a vast number  
14352 of models to estimate population size and detection probability when individuals are un-  
14353 marked. A review of these models is beyond the scope of this chapter, but we mention a  
14354 few deficiencies of existing methods that warrant the exploration of alternatives for robust  
14355 inference when standard capture-recapture methods do not apply.

14356 Distance sampling (Buckland et al., 2001), which we briefly introduced in Chapter 1,  
14357 is perhaps the most widely used method for estimating population density when individ-  
14358 uals are unmarked and detection probability is less than one. This class of methods is  
14359 known to work impeccably when estimating the number of stakes in a field or the number  
14360 of duck nests in a wetland. Distance sampling can also work very well in more interesting  
14361 situations, and is an extremely powerful method when the assumptions can be met. How-  
14362 ever, the assumptions that distance data can be recorded without error and that animals  
14363 are distributed randomly with respect to the transect can be easily violated by common  
14364 processes such as animal movement and measurement error. Although numerous methods  
14365 have been proposed to relax some of these assumptions Royle et al. (2004); Borchers et al.  
14366 (1998); Johnson (2010); Chandler et al. (2011), another issue is that distance sampling is  
14367 simply not practical in many settings. For example, many species are so rare and elusive  
14368 that they can only be reliably surveyed using methods such as camera traps.

14369 Other common sampling methods used to estimate density when individuals are un-  
14370 marked include double-observer sampling, removal sampling, and repeated counts, for  
14371 which custom models have been developed (Nichols et al., 2000; Farnsworth et al., 2002;  
14372 Royle, 2004a,b; Fiske and Chandler, 2011). To obtain reliable density estimates using  
14373 these methods, the area surveyed must be well defined and closed with respect to move-  
14374 ment and demographic processes. Given a short enough sampling interval, such as a 5-min

14375 point-count, the closure assumption may be reasonable. However, short sampling intervals  
14376 limit the number of detections, so observers generally visit each survey location multiple  
14377 times during a season. But then animal movement may invalidate the closure assumption,  
14378 and a model of temporary emigration is required (Kendall et al., 1997; Chandler et al.,  
14379 2011). Furthermore, distance-related heterogeneity in detection probability can introduce  
14380 bias in these models, although this bias is negligible when the ratio of plot size to the scale  
14381 parameter of the detection function is low (Efford and Dawson, 2009).

14382 We mention these issues not to suggest that existing models do not have value—indeed  
14383 we believe that they can be used to obtain reliable density estimates in many situations—  
14384 rather our aim is to highlight the need for alternative methods when the assumptions of  
14385 existing methods cannot be met. Additionally, the spatial count model we discuss in this  
14386 chapter serves as the foundation for a broad class of SCR models in which all or some of  
14387 the individuals cannot be uniquely identified, which is the focus of the next chapter.

## 18.2 SPATIAL CORRELATION AS INFORMATION

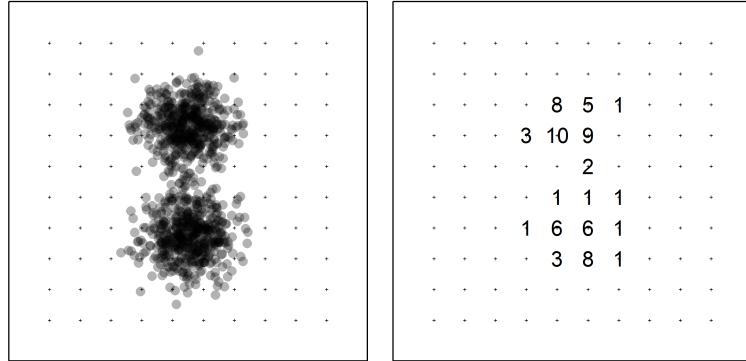
14388 All of the previous methods require some sort of auxiliary information to separately model  
14389 abundance and detection. That is, we need multiple observers or distance data or repeated  
14390 visits to ensure that model parameters are identifiable<sup>1</sup>. The same is true for SC model  
14391 (Chandler and Royle, In press), but the auxiliary information comes in the form of spatial  
14392 correlation, which requires no extra effort to collect.

14393 It is natural to be suspicious of the claim that spatial correlation is a good thing.  
14394 Indeed, elaborate methods have been devised to deal with spatial correlation as a nuisance  
14395 parameter (F Dormann et al., 2007), and ecologists have been admonished for failing to  
14396 obtain “real” replicates uncontaminated by spatial correlation (Hurlbert, 1984). The  
14397 following heuristic may be helpful.

14398 Imagine a  $10 \times 10$  grid of camera traps and a single unmarked individual exposed to  
14399 capture whose home range center lies in the center of the trapping grid. If the individual  
14400 has a small home range size relative to the extent of the trapping grid, we can imagine  
14401 what the spatial correlation structure of the encounters might look like. If the animal’s  
14402 movement is symmetric around the activity center then the number of times the individual  
14403 is detected at each trap (the trap counts) is a function of the distance between the home  
14404 range center and the trap, *i.e.* traps with the same distance from the activity center  
14405 will yield counts that are more highly correlated with one another than traps located  
14406 at different distances from the activity center. Thus, the correlation in counts tells us  
14407 something about the location of the activity center. It is relatively intuitive that spatial  
14408 correlation carries information about distribution, but what about density?

14409 Imagine now that there are two activity centers located in our trapping grid. Using  
14410 trap counts alone, can our model tell us both where the activity centers are and how  
14411 many exist in the population exposed to capture? The answer is yes, at least under  
14412 certain circumstances. Figure 18.1 illustrates the process. The map on the left shows 500  
14413 simulated movement outcomes of the two individuals. The right panel shows the total  
14414 counts made at each trap after 10 survey occasions. Assuming that animals have bivariate  
14415 normal home ranges, the fact that there are two areas in the map with high counts that

<sup>1</sup>Or we can make very strong model assumptions and get away without any auxiliary data (Lele et al., 2012; Sólymos et al., 2012)



**Figure 18.1.** Movement outcomes (left) of two individuals with slightly overlapping home ranges. Crosses represent trap (or point count) locations. The right panel shows counts at each point. It is possible to estimate density using the count data alone.

14416 dissipate in both dimensions suggests that the most likely number of individuals given  
 14417 these data is 2. Furthermore, the degree to which the counts dissipate from the two  
 14418 areas of highest intensity is information about the home range size parameter  $\sigma$ . These  
 14419 two pieces of information are enough to estimate density—again, given that a bivariate  
 14420 normal home range is a valid assumption. Departures from this assumption are discussed  
 14421 subsequently.

### 18.3 DATA

14422 One of the important benefits of the SC model is that it can be applied to data collected  
 14423 using an enormous variety of survey methods. Whereas traditional SCR models require  
 14424 spatially-referenced encounter histories, this model requires simple count data. Once  
 14425 again, suppose that we have  $J$  “traps” operated on  $K$  time periods during which no births  
 14426 or deaths occur. We use the term trap very loosely in this context. A trap is simply some  
 14427 sampling device capable of recording the number of individuals detected,  $n_{jk}$ , so traps  
 14428 could be camera traps, hair snares, or even human observers standing at some location  
 14429  $\mathbf{x}_j$ . Regardless of the sampling method, the requisite data are the counts  $n_{jk}$  and the  
 14430 coordinates of the traps  $\mathbf{x}_j$ . In some instances, we might have additional data such as  
 14431 trap-specific covariates, state-space covariates, information on the identities of a subset  
 14432 of individuals, or perhaps even distance data. Some of these extensions are covered in  
 14433 Chapters 19 and ??, but for the sake of simplicity we focus on the basic data structure in  
 14434 this chapter.

### 18.4 MODEL

14435 The state model that we consider here is the same as in the basic spatial-capture setting,  
 14436 in which we assume a homogeneous point process  $\mathbf{s}_i \sim Unif(\mathcal{S})$  where  $\mathbf{s}_i$  is the activity

center of individual  $i = 1, \dots, N$ , and  $\mathcal{S}$  is the state-space which is typically a polygon defining the region where the organism occur. This state model describes the number and locations of animals. The observation model is once again conditional on the state model and describes the encounter rate as a function of the distance between activity centers and traps.

As with all SCR models, the encounter process is specific to the sampling method, and here we consider the standard camera trapping situation in which an individual can be encountered at multiple traps during a single time period, say one night during a camera-trapping study, and it can be detected multiple times at a single trap during an occasion. This is the Poisson encounter model described in Chapt. 9. The model for the capture histories can be described by

$$z_{ijk} \sim \text{Poisson}(\lambda_{ij}). \quad (18.4.1)$$

where  $\lambda_{ij}$  is the encounter rate for individual  $i$  at trap  $j$ . A common form of this parameter is

$$\lambda_{ij} = \lambda_0 \exp(\|\mathbf{x}_j - \mathbf{s}_i\|/2\sigma^2)$$

where  $\lambda_0$  is the baseline encounter rate and  $\sigma$  is the scale parameter describing the distance-related decay in encounter rate.

When individuals cannot be uniquely identified, the  $z_{ijk}$  cannot be directly observed, which seems like a massively insurmountable problem. The solution is the same one we routinely apply when we cannot directly observe the process of interest—we regard the encounter histories as latent variables. The data are now just a reduced-information summary of the latent encounter histories. That is, they are the sample- and trap-specific totals, aggregated over all individuals:

$$n_{jk} = \sum_{i=1}^N z_{ijk}.$$

This data structure, a matrix of counts made at a collection of sampling locations on one or more occasions is extremely common in ecology. Note also that we can get by with a single occasion of data ( $J \equiv 1$ ) because under the Poisson model,

$$n_{jk} \sim \text{Poisson}(\Lambda_j) \quad (18.4.2)$$

where

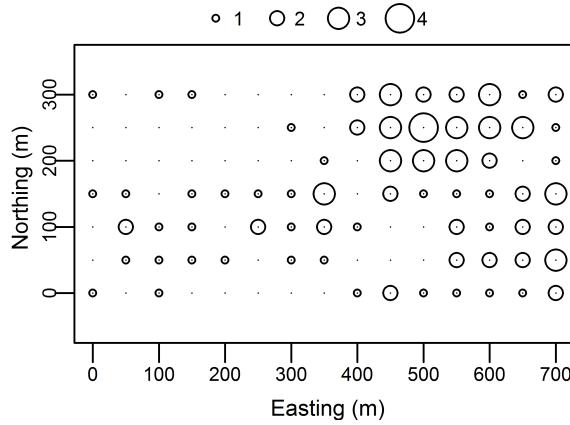
$$\Lambda_j = \lambda_0 \sum_i k_{ij},$$

and because  $\Lambda_j$  does not depend on  $t$ , we can aggregate the replicated counts, defining  $n_{j\cdot} = \sum_k n_{jk}$  and then

$$n_{j\cdot} \sim \text{Poisson}(K\Lambda_j)$$

As such,  $K$  and  $\lambda_0$  serve equivalent roles as affecting baseline encounter rate as has been noted elsewhere (Efford et al., 2009b).

This formulation of the model in terms of the aggregate count simplifies computations as the latent variables  $z_{irt}$  do not need to be updated in the MCMC estimation scheme (see below). However, retaining  $z_{irt}$  in the formulation of the model is important if some individuals are uniquely marked, in which case modifying the MCMC algorithm to include both types of data is trivial is straight-forward. This is because uniquely identifiable individuals produce observations of some of the  $z_{irt}$  variables, which we elaborate on in the subsequent chapter.



**Figure 18.2.** Spatially-correlated counts of northern parula on a 50-m grid. The size of the circle represents the total number of detections at each point.

## 18.5 NORTHERN PARULA EXAMPLE

14473 Here we re-analyze the Northern Parula (*Parula americana*) data described in Chandler  
 14474 and Royle (In press). The data were collected at 105 points located on a 50-m grid at  
 14475 the Patuxent Wildlife Research Center. Each point was surveyed 3 times during June  
 14476 2006, and Fig. 18.2 depicts the resulting spatially-correlated counts ( $n_{r,i}$ ). A total of 226  
 14477 detections were made with a maximum count of 4 during a single survey. At 38 points,  
 14478 no warblers were detected. All but one of the detections were of singing males, and this  
 14479 one observation was not included in the analysis.

14480 In our analysis of the parula data, we defined the point process state-space by buffering  
 14481 the grid of point count locations by 250 m and used  $M = 300$ .

14482 At this point in time there is no canned software to fit this model, and it is actually  
 14483 not straight-forward to use **BUGS** because of the constraints in the model<sup>2</sup>. However,  
 14484 **JAGS** has a neat distribution called the **dsum** distribution, which was designed for this  
 14485 type of situation where the observed data are a sum of random variables. Remember,  
 14486 if we have 3 detections at a point, we assume that these results as  $\sum_i z_{ijk}$ . Thus, we  
 14487 are summing up random variables. **JAGS** actually works rather well for this situation  
 14488 although it is quite slow. Another limitation of using **JAGS** is that we can't mix data from  
 14489 marked and unmarked individuals because **dsum** requires that we sum over unobserved  
 14490 quantities, not a mix of observed and unobserved nodes. Thus, we can't use **JAGS** for the  
 14491 situations considered in the next chapter, and thus we wrote our own MCMC algorithm  
 14492 which overcomes these limitations, and it is somewhat faster. Nonetheless, here is the  
 14493 **JAGS** code to analyze the NOPA data.

```
14494 model{
```

<sup>2</sup>Although it can be done using the so-called “ones-trick”

```

14495 sigma ~ dunif(0, 5)
14496 lam0 ~ dunif(0, 5)
14497 psi ~ dunif(0, 1)
14498 for(i in 1:M) {
14499     # Indicator of occurrence
14500     w[i] ~ dbern(psi)
14501     # Animal activity centers
14502     sx[i] ~ dunif(0, xSide)
14503     sy[i] ~ dunif(0, ySide)
14504     for(r in 1:nTraps) {
14505         # distance from plot center
14506         d[i, r] <- sqrt(pow(sx[i] - X[r, 1], 2) + pow(sy[i] - X[r, 2], 2))
14507         # encounter rate
14508         lam[i, r] <- lam0 * exp(-1*pow(d[i, r],2) / (2*pow(sigma,2))) * w[i]
14509         for(t in 1:nReps) {
14510             z[i, r, t] ~ dpois(lam[i, r])
14511         }
14512     }
14513 }
14514 for(r in 1:nTraps) {
14515     for(t in 1:nReps) {
14516         y[r, t] ~ dsum(z[1,r,t],z[2,r,t], ... ,z[100,r,t]) # code abbreviated
14517     }
14518 }
14519 N <- sum(w[])
14520 }
```

14521 Note that this code will not run as shown because we abbreviated the arguments to  
 14522 `dsum`. In practice, you need to provide all 100 of them, if  $M = 100$ ! This is kind of a drag,  
 14523 but you can easily create the text using `paste` in R. Maybe Martyn Plummer will throw  
 14524 us a bone and allow for a vector as an argument. Anyhow, the entire analysis is shown  
 14525 on the `???XX` help page in `scrbook`.

14526 We simulated posterior distributions using three Markov chains, each consisting of  
 14527 300000 iterations after discarding the initial 10000 draws. Convergence was satisfactory,  
 14528 as indicated by an  $\hat{R}$  statistic of  $< 1.02$  (Gelman and Rubin, 1992).

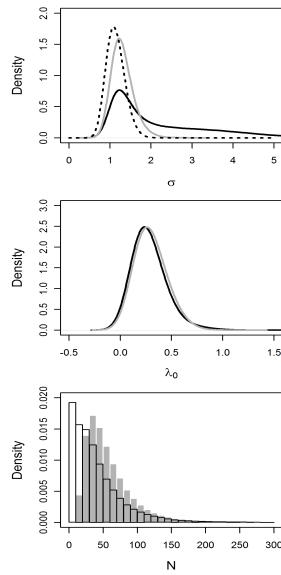
14529 The posterior distribution for  $N$  was highly skewed with a long right tail resulting in a  
 14530 wide 95% credible interval (Table 18.1). Nonetheless, the interval for density,  $D$ , includes  
 14531 estimates reported from more intensive field studies (Moldenhauer and Regelski, 1996).  
 14532 As with any SCR model, we can produce a density surface map, as shown in Fig. 18.4

## 18.6 IMPROVING PRECISION WITH PRIOR INFORMATION

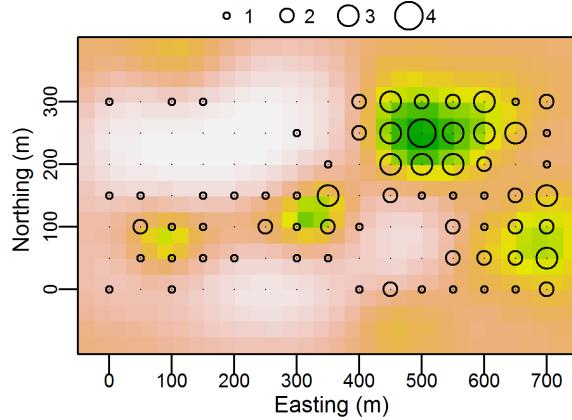
14533 We are asking a lot of a little data. Because both the activity centers and the encounter  
 14534 histories are latent variables, there is inherently high uncertainty in the data, even if it is  
 14535 “perfect” data simulated from the true model. This explains the low posterior precision  
 14536 in the parula data.

**Table 18.1.** Posterior summary statistics for spatial Poisson-count model applied to the northern parula data. Two sets of priors were considered.  $M = 300$  was used in both cases. Parulas/ha,  $D$ , is a derived parameter.

Par	Prior	Mean	SD	Mode	q0.025	q0.50	q0.975
$\sigma$	$U(0, \infty)$	2.154	1.222	1.230	0.896	1.665	5.170
$\lambda_0$	$U(0, \infty)$	0.284	0.149	0.212	0.084	0.256	0.665
$N$	$U(0, M)$	40.953	38.072	4.000	3.000	31.000	143.000
$D$	—	0.427	0.397	0.0417	0.0313	0.323	1.490
$\sigma$	$G(13, 10)$	1.301	0.258	1.230	0.889	1.266	1.908
$\lambda_0$	$U(0, \infty)$	0.298	0.132	0.240	0.098	0.279	0.603
$N$	$U(0, M)$	59.321	36.489	36.000	18.000	50.000	157.000
$D$	—	0.618	0.380	0.375	0.188	0.521	1.635



**Figure 18.3.** Effects of  $\sigma \sim \text{Gamma}(13, 10)$  prior on the posterior distributions from the northern parula model. Posteriors from model with uniform priors are shown in black, and posteriors from the informative prior model are shown in gray. The prior itself is shown as dotted line in the upper panel.



**Figure 18.4.** Estimated density surface of northern parula activity centers. The grid of point count locations with count totals is superimposed. See Fig. 1 for additional details.

14537 So why not just collect distance data or something? If you can, great—we are not  
 14538 arguing against the use of other methods. But in many cases, other models are not  
 14539 applicable. For instance, our model could be applied to camera trapping data collected  
 14540 on species without natural marks, such as pumas or coyotes. In addition, this model  
 14541 provides an important foundation for modeling data where other methods do not apply,  
 14542 and the underlying state model is so damn cool because it corresponds to what we think  
 14543 is happening in the field. Furthermore, the potential generalizations are numerous as we  
 14544 will see later in this chapter and in the next chapter. In sum, the model can be applied  
 14545 where no other models can, and it provides the foundation for important extensions, but  
 14546 how can we improve precision?

14547 Indeed, extensive information on home range size has been compiled for many species in  
 14548 diverse habitats (*e.g.*, DeGraaf and Yamasaki, 2001). It is easy to embody this information  
 14549 in a prior distribution as we demonstrated for the parula data.

14550 One benefit of a Bayesian analysis is that it can accommodate prior information on  
 14551 the home range size and encounter rate parameters, which are readily available for many  
 14552 species. To illustrate, we analyzed the parula data using a new set of priors. Whereas in the  
 14553 first analysis, all priors were improper, customary non-informative priors (see Table 18.1),  
 14554 in the second set we used an informative prior for the scale parameter  $\sigma \sim \text{Gamma}(13, 10)$ .  
 14555 We arrived at this prior using the methods described by Royle et al. (2011a) and published  
 14556 information on the warbler's home range size and detection probability (Moldenhauer and  
 14557 Regelski, 1996; Simons et al., 2009). More details on this derivation are found in ??????.  
 14558 We briefly note here that this prior includes the biologically-plausible range of values from  
 14559  $\sigma$  suggested by the published literature.

14560 This was true when considering both sets of priors, although posterior precision was  
 14561 higher under the informative set of priors. Specifically, the use of prior information reduced  
 14562 posterior density at high, biologically implausible, values of  $\sigma$ , and hence decreased the

14563 posterior mass for low values of  $N$  (Fig. 18.3).

## 18.7 DESIGN ISSUES

### 14564 18.7.1 How Much Correlation Is Enough?

14565  $\sigma$  shouldn't be too small or too large relative to trap spacing.

14566 Can we test for correlation using K-functions or something?

### 14567 18.7.2 Linear Designs

14568 Survey points are not always located on a grid with even spacing—in fact, it is rare to see  
14569 a perfect  $10 \times 10$  grid of points in any study because of habitat patchiness or rugged terrain  
14570 or what have you. Instead, points are often distributed haphazardly or using some form  
14571 of probability sampling. Such designs can still produce data amenable to the models we  
14572 consider in this chapter if individuals can be encountered at multiple points, and none of  
14573 the considerations discussed above need to be modified. But what about linear designs?

14574 In bird studies, point counts are often placed on linear transects. For example, the  
14575 Breeding Bird Survey involves surveying 50 points spaced by 0.5 miles. The mountain-top  
14576 bird survey in the White Mountain National Forest involves surveying 42 transects, each  
14577 with 20? points spaced by 250-m (King et al., 2008). For many species, the 0.5 mile spacing  
14578 of the BBS will ensure that individuals are not detected at multiple points. However, in  
14579 the moutain-top survey, it's easy to imagine that a Bicknell's Thrush (*Catharus bicknelli*)  
14580 could easily be heard from adjacent points. So can we apply our model to obtain density  
14581 estimates with such simple counts?

### 14582 18.7.3 Quadrat counts

## 18.8 ALTERNATIVE OBSERVATION MODELS

14583 Chandler and Royle (In press) focused exclusively on the Poisson observation model, but  
14584 noted that alternative models such as the Bernoulli model or the multinomial model  
14585 (Chapt. 9) should be easily accomodated. Unfortunately, our experimentation with these  
14586 models indicates that the base-line encounter probability parameter  $p_0$  is not identifiable.  
14587 At this point in time, it is not clear why this would be so. However, this situation is  
14588 similar to that of traditional mark-resight models where the unmarked individuals provide  
14589 no information about the parameters of the capture process. Under these models, capture  
14590 or re-sight probability can only be estimated by marking a subset of the population. In  
14591 the next chapter we demonstrate how data from marked and unmarked individuals can  
14592 be combined to improve precision and allow for the estimation of parameters under the  
14593 alternative observation models.

### 14594 18.8.1 Spatial point process models

14595 Our model has some direct linkages to existing point process models. We note that  
14596 the observation intensity function (i.e., corresponding to the observation locations) is a

14597 compound Gaussian kernel similar to that of the Thomas process (Thomas, 1949; Møller  
 14598 and Waagepetersen, 2004, pp. 61-62). Also, the Poisson-Gamma Convolution models  
 14599 (Wolpert and Ickstadt, 1998) are structurally similar (see also Higdon (1998) and Best  
 14600 et al. (2000)). In particular, our model is such a model but with a *constant* basal encounter  
 14601 rate  $\lambda_0$  and *unknown* number and location of “support points”, which in our case are the  
 14602 animal activity centers,  $s_i$ . We can thus regard our model as a model for *estimating*  
 14603 the location and local density of support points in such models, which we believe could  
 14604 be useful in the application of convolution models. Best et al. (2000) devise an MCMC  
 14605 algorithm for the Poisson-Gamma model based on data augmentation, which is similar  
 14606 to the component of our algorithm for updating the  $z$  variables in the conditional-on- $z$   
 14607 formulation of the model. We emphasize that our model is distinct from these Poisson-  
 14608 Gamma models in that the number *and* location of such support points are estimated.

14609 If individuals were perfectly observable then the resulting point process of locations is  
 14610 clearly a standard Poisson or Binomial (fixed  $N$ ) cluster process or Neyman-Scott process.  
 14611 If detection is uniform over space but imperfect, then the basic process is unaffected by  
 14612 this random thinning. Our model can therefore be viewed formally as a Poisson (or  
 14613 Binomial) cluster process model but one in which the thinning is non-uniform, governed  
 14614 by the encounter model which dictates that thinning rate increases with distance from  
 14615 the observation points. In addition, our inference objective is, essentially, to estimate the  
 14616 number of parents in the underlying Poisson cluster process, where the observations are  
 14617 biased by an incomplete sampling apparatus (points in space).

14618 As a model of a thinned point process, our model has much in common with classical  
 14619 distance sampling models (Buckland et al., 2001). The main distinction is that our data  
 14620 structure does *not* include observed distances, although the underlying observation model  
 14621 is fundamentally the same as in distance sampling if there is only a single replicate sample  
 14622 and  $s_i$  is defined as an individual’s location at an instant in time. For replicate samples,  
 14623 our model preserves (latent) individuality across samples and traps which is not a feature  
 14624 of distance sampling. We note that error in measurement of distance is not a relevant  
 14625 consideration in our model, and we explicitly do not require the standard distance sampling  
 14626 assumption that the probability of detection is 1 if an individual occurs at the survey  
 14627 point. More importantly, distance sampling models cannot be applied to data from many  
 14628 of the sampling designs for which our model is relevant. For example, many rare and  
 14629 endangered species can only be effectively surveyed using methods such as hair snares and  
 14630 camera traps that do not produce distance data (O’Connell et al., 2010).

## 18.9 CONCLUSION

14631 Concerns about “statistical independence” have prompted ecologists to design count-based  
 14632 studies such that observed random variables can be regarded as *i.i.d.* outcomes (Hurlbert,  
 14633 1984). Interestingly, this often proves impossible in practice, and elaborate methods have  
 14634 been devised to model spatial dependence as a nuisance parameter. Our paper presents  
 14635 a modeling framework that directly confronts this view by demonstrating that spatial  
 14636 correlation carries information about the locations of individuals, which can be used to  
 14637 estimate density even when individuals are unmarked and distance-related heterogeneity  
 14638 exists in encounter probability.

14639 In this paper, we confronted one of the most difficult challenges faced in wildlife sam-

pling — estimation of density in the absence of data to distinguish among individuals. To do so, we developed a novel class of spatially-explicit models that applies to spatially organized counts, where the count locations or devices are located sufficiently close together so that individuals are exposed to encounter at multiple devices. This design yields correlation in the observed counts, and this correlation proves to be informative about encounter probability parameters and hence density. We note that sample locations in count-based studies are typically *not* organized close together in space because conventional wisdom and standard practice dictate that independence of sample units is necessary (Hurlbert, 1984). Our model suggests that in some cases it might be advantageous to deviate from the conventional wisdom if one is interested in direct inference about density. Of course, this is also known in the application of standard spatial capture-recapture models (Borchers and Efford, 2008) where individual identity is preserved across trap encounters, but it is seldom, if ever, considered in the design of more traditional count surveys.

Our model has broad relevance to an incredible number of animal sampling problems. Our motivating problem involved bird point counts where individual identity is typically not available. The model also applies to other standard methods used to sample unmarked populations, such as camera traps or even methods that yield sign (*e.g.* scat, track) counts indexed by space. However, results of our simulation study reveal some important limitations of the basic estimator applied to situations in which none of the individuals can be uniquely identified. In particular, posterior distributions are highly skewed in typical small to moderate sample size situations and posterior precision is low.

# 19

---

14661  
14662

## SPATIAL MARK-RESIGHT MODELS FOR 14663 14664 14665 PARTIALLY IDENTIFIABLE POPULATIONS

14666 So far, we have dealt with the situation where all detected individuals are identifiable  
14667 upon encounter, and in Chapt. 18 we introduced and developed an SCR model for non-  
14668 identifiable populations, a spatial *non*-capture-recapture model, if you will. These two  
14669 extremes are common in the study of animal populations with non-invasive sampling meth-  
14670 ods. However, there is also an intermediate situation, where a part of the population is  
14671 tagged or otherwise marked and can thus be identified upon recapture, while the untagged  
14672 portion remains unidentified. In this situation so-called mark-resight models (Bartmann  
14673 et al., 1987; Arnason et al., 1991; Neal et al., 1993) can be used to estimate population  
14674 size and density combining data from both the marked and unmarked individuals.

14675 Traditionally, capture-recapture studies involve physical capture of individuals through-  
14676 out the study; new individuals are marked on every re-capture occasion. This methodology  
14677 is still widely applied in the study of species that are relatively easy to capture, such as  
14678 small mammals, but can be very costly, logically challenging and risky when dealing  
14679 with larger species. In contrast, in mark-resight studies a sample of individuals is captured  
14680 and tagged (or otherwise marked) during a single marking event. Marking is followed by  
14681 resighting surveys, upon which both the detection of marked and recognizable individuals  
14682 and unmarked animals is recorded. Resighting surveys are usually non-invasive (hence the  
14683 name resighting), so that they don't involve handling of animals. As such, mark-resight  
14684 models have a major advantage over traditional capture-recapture models in that they  
14685 only require individuals to be captured and handled once, during the initial marking.  
14686 This reduces field costs and risks for the animals (and potentially the researchers).

14687 Mark-resight models have a set of underlying assumptions, most of which are analogous  
14688 to those of capture-recapture models, e.g. demographic population closure (violation  
14689 of geographic population closure can be accommodated by some models) and no loss  
14690 or misidentification of marks (see also 5). Just like regular capture-recapture models,

14691 there are means to incorporate heterogeneity in capture probability. However, a new and  
14692 essential assumption of mark-resight models is that the tagged (or otherwise identifiable)  
14693 individuals are a representative sample of the study population, so that inference about  
14694 detection can be made for the whole population from the tagged sample. This issue is  
14695 usually addressed by using a different method for marking than for resighting, and by  
14696 marking a random sample of the population.

14697 Owing to the advantages of mark-resight over capture-recapture, especially when dealing  
14698 with hard-to-trap species, mark-resight is a popular tool in wildlife population studies.  
14699 The method has been applied for decades and to a suite of species and survey techniques,  
14700 ranging from banding and resighting Canada geese (Hestbeck and Malecki, 1989) to ear-  
14701 tagging and camera-trapping grizzly bears (Mace et al., 1994) to paintball marking and  
14702 areal resightings of large ungulates (Skalski et al., 2005).

14703 In this chapter we consider mark-resight in spatial context and develop a spatial mark-  
14704 resight (SMR) model. To motivate this model development, imagine you conduct a live-  
14705 trapping study during which you capture and mark a number of animals with individually  
14706 recognizable tags. Subsequently, you go back out to the field and conduct resighting  
14707 surveys on an array of locations, and during these resighting surveys you see some of your  
14708 tagged individuals as well as new, untagged ones. Then, for the tagged animals you obtain  
14709 the same form of spatially explicit individual encounter histories as you would in a regular  
14710 SCR study. On top of that you obtain site (and occasion) specific counts of individuals you  
14711 did not tag. Thus, spatial mark-resight is an SCR framework for populations where only  
14712 part of the individuals can be identified and the major difference between SCR and SMR  
14713 is how we include those counts of unmarked individuals in the model. In the following  
14714 sections we first provide some background information on mark-resight and the types of  
14715 data such surveys can provide. We then move on to the formal development of SMR  
14716 models, which, as we will see, are hybrids of regular SCR models and the models for data  
14717 without individual identity presented in Chapt. 18.

## 19.1 BACKGROUND

### 14718 19.1.1 Types of partial ID data

14719 Before we start exploring mark-resight approaches in more detail, we need a clear un-  
14720 derstanding of what types of mark-resight data we can have, in order to appreciate and  
14721 understand the different flavors of mark-resight models. In general, we have (at least) two  
14722 sets of data: encounter histories for identifiable individuals  $i$  at trap  $j$  and occasion  $k$ ,  
14723  $y_{ijk}$ , and counts of unidentified records for each  $j$  and  $k$ ,  $n_{jk}$ . Depending on the sampling  
14724 technique, we can conceive of three slightly different types of partial ID data.

14725 **(1) Known number of tagged individuals** If you implement your resighting survey  
14726 shortly after the marking session, you may be confident that none of the marked individuals  
14727 has died or lost its mark. Under these circumstances you know that the number of marked  
14728 individuals available for resighting,  $m$ , is equal to the number of individuals you tagged.  
14729 Alternatively, tags might be radio-transmitters, allowing you to confirm the presence or  
14730 absence of marked individuals in the resighting survey area using radio-telemetry (White  
14731 and Shenk, 2001). In both cases, you know the number of marked individuals in the  
14732 population you survey. In this situation, even though you may fail to resight some of the  
14733 tagged individuals, since you know how many there are, you can simply assign those you

14734 never resighted all-zero encounter histories - in other words, contrary to regular capture-  
14735 recapture models, in mark-resight models with a known number of tagged individuals, we  
14736 can observe all-zero encounter histories. Under these circumstances, estimating  $N$  reduces  
14737 to estimating the number of unmarked individuals,  $U$ .

14738 **(2) Unknown number of tagged individuals** If we suspect that some of the  
14739 marks may have been lost between tagging and conducting the resighting samples, we  
14740 obtain a slightly different type of mark-resight data. Here, we do not accurately know  
14741 the number of marked individuals available for resighting. As a consequence, individuals  
14742 have to be resighted at least once for us to know they are still tagged and alive and thus  
14743 available for resighting. So, contrary to the situation where we know  $m$  and analogous to  
14744 regular capture-recapture models, we cannot observe all-zero encounter histories of marked  
14745 individual. Here, estimating  $N$  involves estimating both  $m$  and  $U$ .

14746 A special case of this kind of data can arise from camera trapping. Even when dealing  
14747 with a species that has no spots or stripes, some individuals in the study population can  
14748 have natural marks that make them identifiable on pictures, such as scars or some distinct  
14749 coloration. Again, in this scenario an individual has to be photographed at least once to  
14750 be known. Here, the fact that both the “marking” method and the subsequent resighting  
14751 method are the same (although marking in this case does not involve any actual physical  
14752 marking) can be cause for concern: our sample of “marked” individuals may not be a  
14753 random sample of the population but consist of individuals that for some reason are more  
14754 likely to be photographed. In that case, a basic assumption of the mark-resight model is  
14755 violated.

14756 **(3) Unknown marked status** Finally, consider a scat or hair snare survey, where  
14757 only a part of the sample is analyzed genetically (or DNA can only be extracted from  
14758 a subset of samples due to sample quality). In this scenario, your  $n_{jk}$  can contain both  
14759 completely unknown individuals that are not represented at all in  $Y$ , but it can also contain  
14760 samples from individuals that we previously identified. The difference is that in the first  
14761 two scenarios, part of the population of individuals is identifiable, while in the second  
14762 scenario, part of the samples is identifiable. This type of data violates one of the basic  
14763 assumptions of mark-resight models, namely, that tagged individuals are always correctly  
14764 identified as such.

14765 To our knowledge there are currently no mark-resight models available that account for  
14766 possible misidentification of the marking status of individuals (although some literature is  
14767 available on misidentification of individuals in capture-recapture studies, e.g., Yoshizaki  
14768 et al., 2009; Lukacs and Burnham, 2005; Link et al., 2010). In this chapter we will ignore  
14769 this kind of data and focus instead on the two types of typical mark-resight data:

- 14770 (1) Known number of tagged individuals  
14771 (2) Unknown number of tagged individuals,

14772 For both types of data a slightly different situation arises when in some instances we  
14773 can only tell that an individual is tagged, but not who it is. You may be able to see that  
14774 an individual is tagged but the identifying feature of the tag (a number or coloration)  
14775 may have become unreadable, or may be hidden from view. In this case, in addition to  
14776 your  $y_{ijk}$  and your  $n_{jk}$  you also have a number of sightings of tagged but unidentified  
14777 individuals, say  $r_{jk}$ .

**14778 19.1.2 A short history of mark-resight models**

14779 Initially, mark-resight methods focused on radio-tagged individuals to estimate popula-  
 14780 tion size (White and Shenk, 2001). Radio-collars provide a means of determining which  
 14781 of the animals were in the study area and available for sampling, i.e. determining the  
 14782 number of marked individuals in the population. Knowing this number was a prerequisite  
 14783 for most earlier mark-resight approaches (White, 1996). The oldest mark-resight model  
 14784 is the good old Lincoln-Petersen estimator, where individuals are marked and a single  
 14785 resight/recapture occasion is carried out (Krebs, 1999). We need not identify individuals,  
 14786 but only tell apart marked from unmarked individuals. Let  $m$  be the number of marked  
 14787 individuals in the population,  $m_{(R)}$  the number of marked individuals seen on the resight-  
 14788 ing occasion, and  $n_{(R)}$  the total number of marked and unmarked individuals observed  
 14789 during resighting. Population size  $N$  is then estimated as

$$N = m \times n_{(R)} / m_{(R)}$$

14790 A suite of more elaborate models using individual capture histories over several re-  
 14791 sighting occasions were developed in the 1980s and 90s and compiled into the program  
 14792 NOREMARK (White, 1996). Apart from the basic model with known number of marked  
 14793 individuals and no individual variation in resighting probabilities (joint hypergeometric  
 14794 maximum likelihood estimator) (Bartmann et al., 1987; White and Garrot, 1990; Neal,  
 14795 1990; Neal et al., 1993), NOREMARK contains models that account for lack of geographic  
 14796 population closure (Neal et al., 1993), individual heterogeneity in resighting rates and  
 14797 sampling with replacement (i.e. individuals can be seen more than once on any occasion,  
 14798 (Minta and Mangel, 1989; Bowden, 1993)). A first mark-resight model allowing for an  
 14799 unknown number of marked individuals was developed by Arnason et al. (1991).

14800 While many of these models perform well under certain situations, they are somewhat  
 14801 limited: they do not allow for combining data across several surveys (McClintock et al.,  
 14802 2006) and not all of them are likelihood-based or allow for different parameterizations  
 14803 (e.g., including a time effect on detection), so that selection of the most appropriate  
 14804 model cannot be based on standard approaches such as AIC, but is largely left up to  
 14805 educated guesswork (McClintock et al., 2006). Recently, more flexible and generalized  
 14806 likelihood-based mark-resight models have been developed. These models can account  
 14807 for individual heterogeneity in detection, unknown number of marked individuals and  
 14808 lack of geographical closure, as well as a less than 100% individual identification rate of  
 14809 tagged individuals; they can be applied to sampling with and without replacement and  
 14810 can combine data across several primary sampling occasions in a robust design type of  
 14811 analysis (McClintock et al., 2009a,b). Since they are all likelihood-based, model selection  
 14812 among different parameterizations and model averaging based on AIC is an option. Most  
 14813 of these models have also been incorporated into the program **MARK** (McClintock and  
 14814 White, 2010).

14815 For a detailed treatment of these different non-spatial mark-resight models, we refer  
 14816 you to the original papers cited in the preceding paragraph. In short, these models are  
 14817 based on the joint likelihood of two major model components: one describing the resight-  
 14818 ing process of marked individuals (either using a Poisson or a Bernoulli observation model,  
 14819 depending on whether sampling is with or without replacement), where resighting proba-  
 14820 bilities can have both fixed effects to model individual and environmental covariates, and

14821 a random-effect component to accommodate variation in detection due to individual heterogeneity; and one describing the number of unmarked individuals observed (or, under a  
 14822 Poisson observation model, the number of times unmarked individuals are observed),  $n_t$   
 14823 ( $t$  here and in the following description denotes a primary sampling occasion, for example,  
 14824 a year or a season; for a single-season study we could easily drop this subscript) which are  
 14825 approximated as a normal distribution (McClintock et al., 2006), or a normal distribution  
 14826 left-truncated at 0 (McClintock et al., 2009a):  
 14827

$$n_t \sim \text{Normal}(E(n_t), V(n_t))$$

14828 Although this is a simplification of the actual sampling process, McClintock et al. (2006)  
 14829 found this normal distribution to be a satisfactory approximation, which allows  $N$  to enter  
 14830 the model likelihood via  $E(n_t)$  and  $V(n_t)$ .

14831 In the simplest model case without any variation in detection, the expected number  
 14832 of resightings of unmarked individuals,  $E(n_t)$ , can be written as the number of unmarked  
 14833 individuals times the expected number of detections of a single individual, which is the  
 14834 mean or expected value of the underlying observation model:

$$E(n_t) = (N - m) * \theta \quad (19.1.1)$$

14835 where  $\theta = K \times p$  for a Binomial observation model with  $K$  replicates and individual  
 14836 detection probability  $p$ , or  $\theta = \text{expected/average individual encounter rate } \lambda$  for a Poisson  
 14837 observation model. Similarly,  $V(n_t)$  depends on the underlying observation model and is  
 14838 based on the parameters that determine the individual detection probability/encounter  
 14839 rate. Combining these two components,  $N$  is directly incorporated into the joint likelihood  
 14840 of the model.

14841 While these mark-resight models are very flexible, they share the shortcomings of  
 14842 regular capture-recapture models when it comes to estimating population density (e.g.,  
 14843 Chaps. 1 and 4). As long as resightings are collected across a network or array of locations,  
 14844 however, they come with the same spatial information as recaptures in a regular  
 14845 SCR study. In the following sections we will consider mark-resight sampling in the framework  
 14846 of spatial capture-recapture. We will look at models for both known and unknown  
 14847 numbers of marked individuals, and for imperfect individual identification of marks. In the  
 14848 spatial framework, most of the information on model parameters comes from the marked  
 14849 individuals. But in sec. 19.5 we will see that, analogous to the models we developed in  
 14850 the previous Chapt. 18, the spatial correlation in counts of unmarked individuals also  
 14851 contributes information about detection and movement.

## 19.2 KNOWN NUMBER OF MARKED INDIVIDUALS

14852 We begin with the easiest situation: a known number of individuals constituting a random,  
 14853 representative sample from the population are marked and a series of resight samples are  
 14854 conducted following marking. No marks (or marked animals) are lost between marking  
 14855 and resighting, all individuals are correctly identified as marked or unmarked, and marked  
 14856 individuals are 100 % correctly identified to individual level.

14857 Recall from Chapt. 18 that without any individual identity, the observed counts at  
 14858 trap  $j$  and occasion  $k$ ,  $n_{jk}$ , represent the sum of all latent individual detections at  $j$  and

14859  $k, \sum_{i=1}^N y_{ijk}$ , where  $y_{ijk}$  are the latent individual encounter histories which we include as  
 14860 variables (or missing data) in our MCMC scheme. We can model these counts as

$$n_{jk} \sim \text{Poisson}(\Lambda_j)$$

14861 where

$$\Lambda_j = \sum_{i=1}^M (\lambda_{ij})$$

14862 Under this formulation we do not need to update the individual  $y_{ijk}$  in our model, which  
 14863 is more efficient in terms of computing. However, we can also formulate the model as  
 14864 conditional on the latent  $y_{ijk}$ . This is useful because if we have  $m$  individually known  
 14865 animals in our study population, than those  $m$   $y_{ijk}$  are no longer latent, but fully observed  
 14866 and can easily be included in the analysis to provide information on detection parameters.

14867 The formulation conditional on  $y_{ijk}$  basically brings us back to the original SCR model,  
 14868 where individual site and occasion specific counts,  $y_{ijk}$ , are modeled as

$$y_{ijk} \sim \text{Poisson}(\lambda_{ij})$$

14869 and

$$\lambda_{ij} = \lambda_0 \exp(-d_{ij}^2 / (2\sigma^2))$$

14870 Unobserved  $y_{ijk}$  are essentially missing data and have to be updated as part of the  
 14871 MCMC procedure. We can do that by using their full conditional distribution, which is  
 14872 multinomial with sample size  $n_{jk}$ :

$$y_{ujk} \sim \text{Multinomial}(n_{jk}, \lambda_{uj})$$

14873 where  $u$  is an index vector of the  $M - m$  hypothetical unmarked individuals.

14874 While in the non-spatial mark-resight analysis known individuals provide direct information  
 14875 about individual detection probability (or rate), in the spatial setting they also  
 14876 inform  $\sigma$ . Including known individuals into the analysis helps estimate model parameters  
 14877 more accurately and precisely. We will address the relationship between the number of  
 14878 marked individuals and accuracy of the estimated parameters in sec. 19.5.

### 14879 19.2.1 MCMC for a spatial mark-resight model

14880 Implementing a spatial mark-resight model in **JAGS** is not trivial, since the program  
 14881 does not accept partially observed multivariate nodes (in this case the partially observed  
 14882 individual encounter histories which we model as coming from a multinomial distribution).  
 14883 Therefore, knowing how to write your own MCMC algorithm comes in extremely handy.  
 14884 You will find that we only have to make relatively simple modifications to the MCMC  
 14885 code for the model without any individual identification presented in Chapt. 18, which,  
 14886 in turn, has much in common with the algorithms we developed for regular SCR models  
 14887 in Chapt. 14. Essentially, since we observe individual detections for the marked part  
 14888 of the population, we have to update only the unobserved part of  $\mathbf{Y}$ , and modify the  
 14889 updating steps for  $z_i$  and  $\psi$ , the parameters introduced by data augmentation, to reflect  
 14900 some contribution to our knowledge of these parameters from the  $m$  marked individuals.

14891 First, we set up an array to hold  $\mathbf{Y}$ , fill the first  $m$  rows of the array with the  $m$   
 14892 observed individual encounter histories, then update  $\mathbf{Y}$  for the unknown individuals only  
 14893 (note that the code is set up so that  $n_{jk}$  contains both pictures of marked **and** unmarked  
 14894 individuals at  $j$  and  $k$ ):

```
14895 # set up placeholders and create vectors for marked and unmarked
14896 Y <- array(NA, c(M, J, K))
14897 nMarked <- nrow(y)
14898 marked <- rep(FALSE, M)
14899   marked[1:nMarked] <- TRUE
14900   Y[1:nMarked, , ] <- y
14901 z[marked] <- 1
14902 Ydata <- !is.na(Y)
14903 for (j in 1:J) {
14904   for (k in 1:K) {
14905     if (y[j, k] == 0) {
14906       Y[, j, k] <- 0
14907       next
14908     }
14909     unmarked <- !Ydata[, j, k]
14910     nUnknown <- n[j, k] - sum(Y[!unmarked, j,k])
14911     if (nUnknown < 0)
14912       browser()
14913     probs <- lam[, j] * z
14914     probs <- probs[unmarked]
14915     probs <- probs/sum(probs)
14916     Y[unmarked, j, k] <- rmultinom(1, nUnknown, probs)
14917   }
14918 }
```

14919 When we know the number of marked individuals in the population estimating  $N$   
 14920 is reduced to estimating  $u$ . Thus, we only need to estimate the  $z_i$  for  $M - m$  unknown  
 14921 individuals and the updater for  $z_i$  becomes:

```
14922 zUps <- 0
14923 seen <- apply(Y > 0, 1, any)
14924 for (i in 1:M) {
14925   if (seen[i] | marked[i])
14926     next
14927   zcand <- ifelse(z[i] == 0, 1, 0)
14928   ll <- sum(dpois(Y[i, , ], lam[i, ] * z[i], log = TRUE))
14929   llcand <- sum(dpois(Y[i, , ], lam[i, ] * zcand,
14930     log = TRUE))
14931   prior <- dbinom(z[i], 1, psi, log = TRUE)
14932   prior.cand <- dbinom(zcand, 1, psi, log = TRUE)
14933   if (runif(1) < exp((llcand + prior.cand) - (ll +
14934     prior))) {
```

```

14935      z[i] <- zcand
14936      zUps <- zUps + 1
14937      }
14938  }

```

Observe that while we skip the update of  $z_i$  for the “seen” individuals (where `seen=TRUE` for any individual observed at least once and `seen=FALSE` otherwise), `seen` is defined based on  $\mathbf{Y}$  and  $\mathbf{Y}$  is updated at each iteration, so the  $z_i$  for the observed but unmarked individuals are still updated.

Finally, our update for  $\psi$  needs to reflect that we are effectively only estimating  $U$ . In the full conditional beta distribution we have to replace  $M$  with  $M - m$  and  $\sum z$  with  $\sum z - m$ :

```
14946  psi<-rbeta(1,1+sum(w[!marked]),1+sum(!marked)-sum(w[!marked]))
```

The remainder of the code is essentially identical to the MCMC code for regular SCR models we developed in Chapt. 14. You can find the full MCMC code (including the modeling options we’ll discuss in the following sections) in the accompanying **R** package `scrbook` by invoking `scrPID`.

### 19.2.2 Binomial encounter model

So far, we have only worked with Poisson encounter models for partially identifiable or unmarked populations. When we use a Bernoulli model instead, we have to make some changes to how we update the latent  $y_{ijk}$ , to ensure that a hypothetical individual receives at most a single observation at a given trap and occasion from the pool of  $n_{jk}$  pictures. Effectively, we move from a multinomial situation where the same individual could be drawn repeatedly, to a sampling without replacement situation (an individual drawn once at  $j$  and  $k$  cannot be drawn again); here is how we implement this in our MCMC algorithm:

```

14959  Y <- array(NA, c(M, J, K))
14960  #[...]
14961  for (j in 1:J) {
14962    for (k in 1:K) {
14963      if (y[j, k] == 0) {
14964        Y[, j, k] <- 0
14965        next
14966      }
14967      unmarked <- !Ydata[, j, k]
14968      nUnknown <- n[j, k] - sum(Y[!unmarked, j,k])
14969      if (nUnknown < 0)
14970        browser()
14971      probs <- lam[, j] * z
14972      probs <- probs[unmarked]
14973      probs <- probs/sum(probs)
14974      Y[unmarked, j, k] <- 0
14975      guys <- sample(which(unmarked), nUnknown, prob = probs)
14976      Y[guy, j, k] <- 1

```

**Table 19.1.** Posterior summaries of the spatial mark-resight model for Canada geese in North Carolina.

	Mean	SD	2.5%	50%	97.5%
$\sigma$ , females	1.06	0.02	1.02	1.06	1.10
$\sigma$ , males	1.13	0.02	1.09	1.13	1.18
$\lambda_0$	0.32	0.01	0.31	0.32	0.34
$\psi$	0.79	<0.01	0.73	0.79	0.86
$\phi$	0.43	0.02	0.40	0.43	0.47
$N$	3720.81	121	3492	3717	3961
$D$	6.68	0.22	6.27	6.68	7.11

14977                   }

14978                   }

14979 **Example: Canada geese in North Carolina**

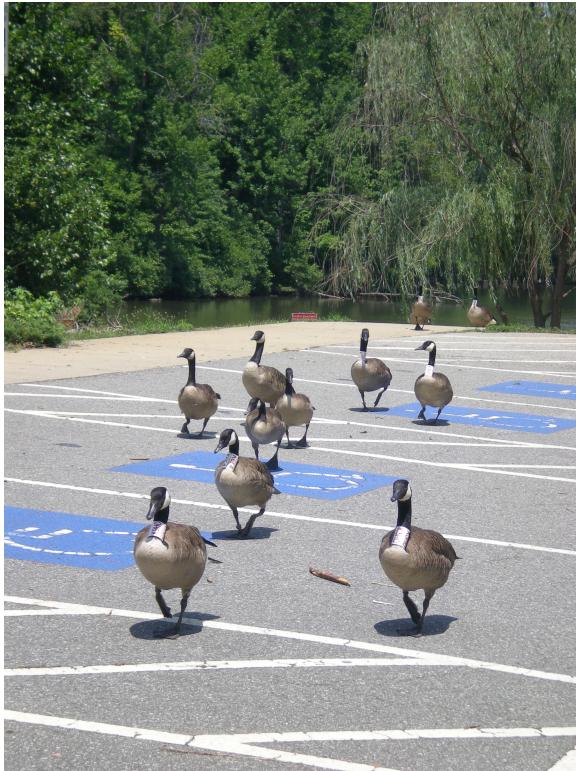
14980 We applied the spatial mark-resight model with a binomial encounter process to a  
 14981 dataset of Canada goose resightings (Rutledge, 2012) XXXget full citation with LizXXX.  
 14982 During the molt of 2008, 751 individual geese were captured and tagged with neck and  
 14983 leg bands in Greensboro, North Carolina (Fig. 19.1). Geese were resighted at 87 different  
 14984 locations on 81 resighting events over a period of 18 months. In addition to the banded  
 14985 geese, the number of unmarked geese was recorded during each resighting event. Here,  
 14986 we only looked at a subset of the data, from mid July to the end of October 2008, which  
 14987 corresponds to the first part of the post-molt season, before migratory Canada geese arrive  
 14988 in North Carolina. During this time frame, 746 of the 751 marked geese were known to  
 14989 be alive. Of those, 654 were resighted 3994 times at 40 different sites. In addition, 7944  
 14990 sightings of unmarked geese were recorded at 48 sites.

14991 In this model, we also allowed  $\sigma$  to vary between males and females. We augmented  
 14992 the data set with  $4500 - m$  all-zero encounter histories, ran 50000 MCMC iterations and  
 14993 removed a burn-in of 1000 iterations. We provide all the data (`data('canadageese')`)  
 14994 and functions (`pIDgeese`) for you to repeat this analysis but be aware that given the large  
 14995 data set it will take days to do so. The **R** code to set up the data and run 5000 iterations  
 14996 of the goose model is given as an example on the help page for `pIDgeese`. The model  
 14997 results, including the derived parameter density ( $D$ ) in individuals per  $km^2$  are shown in  
 14998 Tab. 19.2.2.

14999 We see that credible intervals of estimates are pretty narrow. Take, for example,  $\sigma$  for  
 15000 males and females: Although they differ only by 0.08, there is barely any overlap between  
 15001 the respective credible intervals, surely an effect of the large data set. The parameter  $\phi$  in  
 15002 this model is the probability of being a male, a measure of the sex ratio of the population,  
 15003 which is close to 1:1.

### 19.3 UNKNOWN NUMBER OF MARKED INDIVIDUALS

15004 Now let us consider the case where we do not know the exact number of tagged individuals  
 15005 available for resighting so that we have to capture an individual at least once to be sure  
 15006 that it is available. Unless we have a direct means of confirming the number of marked



**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)

15007 animals available for resighting, treating this number as unmarked is probably more real-  
15008 istic in most circumstances. As a consequence of not knowing the exact number of marked  
15009 individuals, we cannot observe all-zero encounter histories. When using maximum likeli-  
15010 hood inference, this situation requires a model where detection rates of known individuals  
15011 are modeled using a zero-truncated distribution (McClintock et al., 2009a). If we did not  
15012 account for the fact that zeros are unobservable, our estimates of detection rates would  
15013 be artificially inflated and estimates of population size would be negatively biased.

15014 Working with zero-truncated distributions in a spatial mark-resight setting is less  
15015 straight-forward than for non-spatial mark-resight. A marked individual only has to show  
15016 up once, anywhere on the resighting array, for us to know that it is there. When resightings  
15017 are pooled across the entire sampling grid, then the total individual counts  $\sum_j y_{ij}$  have  
15018 to be  $> 0$  for all resighted individuals and a zero-truncated distribution can be used to  
15019 model these counts. However, we are concerned with trap-specific encounters,  $y_{ij}$ , which  
15020 can easily be 0 for a resighted individual, as long as a single  $y_{ij}$  is  $> 0$ . Thus, the zero-  
15021 truncation does not apply to the individual and trap specific counts we observe, but only

15022 to the sum of these counts over all traps.

15023 As an alternative to a zero-truncated distribution, in a Bayesian framework, we can  
 15024 make use of data augmentation to estimate the number of marked individuals<sup>1</sup>. In the  
 15025 previous example, where we knew the number of marked individuals, we separate those  
 15026 individuals from the augmented population by fixing their  $z_i$  at 1 and letting  $\psi$  refer only  
 15027 to the unmarked population,  $M - m$ . All we have to do in the spatial mark-resight model  
 15028 with unknown number of marked individuals is to let our marked individuals be part of  
 15029 the augmented population again, analogous to the situation in regular SCR models:

```
15030     psi <- rbeta(1, 1 + sum(z), 1 + M - sum(z))
```

15031 Whether you have a known or an unknown number of marked individuals is included  
 15032 as an option in **scrPID**.

### 15033 A simulation example

15034 For illustration purposes we simulated a data set with  $N = 80$  individuals randomly  
 15035 distributed across a state space of 10x10 units. Of those, we randomly choose 40 to be  
 15036 marked and identifiable, and then simulate encounter data for both marked and unmarked  
 15037 individuals on an 8x8 grid with unit spacing over  $K = 5$  occasions, with  $\sigma = 0.5$  and  $\lambda_0 =$   
 15038 0.5, adopting a Poisson encounter process. To do so we use the **scrbook** function **sim.data**,  
 15039 which also allows you to create data sets from a Binomial observation process, known  
 15040 number of marked individuals, and with telemetry locations (sec. 19.6) or individual  
 15041 identification rate < 100 % (sec. 19.4). We analyzed the simulated data both assuming  
 15042 we do not know the total number of marked animals in our state space, and assuming we  
 15043 do know this number, using the **scrPID** function and running 20000 iterations. You can  
 15044 repeat the analysis by executing the R code below.

```
15045 set.seed(2501)
15046
15047 #set input values
15048 N=80
15049 lam0=0.5
15050 knownID=40
15051 rat=0.8
15052 sigma=0.5
15053 K=5
15054
15055 #create grid and state space
15056 coords<-seq(0,7, 1)
15057 grid<-expand.grid(coords, coords)
15058 trapmat<-as.matrix(grid)
15059 buff<- 3*sigma
15060 xl<-min(trapmat[,1])-buff
15061 xu<-max(trapmat[,1])+buff
15062 yl<-min(trapmat[,2])-buff
```

---

<sup>1</sup>For the interested reader, McClintock and Hoeting (2010) implement a non-spatial mark-resight model with a binomial observation model in a Bayesian framework using data augmentation

```

15063 yu<-max(trapmat[,2])+buff
15064 xlims=c(xl, xu)
15065 ylims=c(yl,yu)
15066 area<-(xu-xl)*(yu-yl)
15067
15068 #simulate data
15069 dat<-sim.pID.data(N=N, K=K, sigma=sigma, lam0=lam0, knownID=knownID,
15070 X=trapmat, xlims=xlims, ylims=ylims, obsmod= "pois",
15071 nmarked="unknown", rat=1, tel =0, nlocs=0)
15072
15073 #create initial values function for scrPID, set M and tuning parameters
15074 inits<-function(){list(S=cbind(runif(M, xlims[1], xlims[2]),
15075 runif(M, ylims[1], ylims[2])), lam0=runif(1, 0.4, 0.6),
15076 sigma=runif(1, 0.4, 0.6), psi=runif(1, 0.4, 0.6))}
15077 M<-160
15078 delta=c(0.1, 0.01, 2)
15079
15080 #run model, first m=unknown, then m=known
15081 mod<-scrPID(n=dat$n, X=trapmat, y=dat$Yobs, M=M, obsmod = "pois",
15082 nmarked="unknown", niters=20000, xlims=xlims, ylims=ylims,
15083 inits=inits(), delta=delta ) )
15084 mod2<-scrPID(n=dat$n, X=trapmat, y=dat$Yobs, M=M, obsmod = "pois",
15085 nmarked="known", niters=20000, xlims=xlims, ylims=ylims,
15086 inits=inits(), delta=delta ) )
15087

```

15088 Looking at the data, we see that of the 40 marked animals, 26 were recorded at least  
 15089 once. In terms of data that means that in the second model, where we know  $m$ , we have  
 15090 14 observed all-zero encounter histories that we cannot use in the model where we assume  
 15091  $m$  is not known. This reduction in data is reflected in the model results (Tab. 19.3). The  
 15092 estimate of  $N$  for the unknown- $m$  model shows some positive bias, although the 95 % BCI  
 15093 still includes the true value of 80. Thus, while we can formally account for the fact that we  
 15094 often do not know the number of marked individuals in the state space, we clearly loose  
 15095 quite a bit of accuracy and precision. It would be an interesting little project to quantify  
 15096 this loss in accuracy and precision in a small simulation study.

## 19.4 INDIVIDUAL IDENTIFICATION RATE < 100 %

15097 Often during resighting, it may be possible to see that an individual is tagged but impos-
 15098 sible to determine its individual identity. In such a situation in addition to the  $y_{ijk}$  and  
 15099  $n_{jk}$ , we also have site and occasion specific counts of marked but unidentified individuals,  
 15100  $r_{jk}$ . Here, the individual encounter histories of marked animals are incomplete, and if we  
 15101 used these incomplete data to inform the detection parameter of the model, we would run  
 15102 the risk of underestimating detection/trap encounter rate and overestimating abundance.  
 15103 Some non-spatial mark-resight models do not require that marked animals be identified  
 15104 individually, as long as the marking status can be observed unambiguously, but ignoring

**Table 19.2.** Posterior summaries of the spatial mark-resight model for a simulated data set analyzed with number of marked individuals  $m$  assumed to be unknown and known. First 500 iterations discarded as burn-in.

		Mean	SD	2.5%	97.5%
$m$ unknown	$\sigma$	0.521	0.029	0.470	0.583
	$\lambda_0$	0.4679	0.069	0.346	0.602
	$\psi$	0.541	0.070	0.411	0.684
	$N$	86.612	9.386	70	107
$m$ known	$\sigma$	0.514	0.0284	0.4638	0.5750
	$\lambda_0$	0.550	0.077	0.403	0.707
	$\psi$	0.332	0.066	0.212	0.468
	$N$	79.525	6.149	69	93

15105 individual level information means that we cannot accommodate heterogeneity in detection  
 15106 (McClintock and White, 2010). In a spatial framework we could ignore marked and  
 15107 unmarked status completely and apply the model by Chandler and Royle (In press) we  
 15108 discussed in Chapt. 18. But, that would mean losing important information on individual  
 15109 detection and movement. Therefore, being able to retain the individual identity of records  
 15110 that can be identified while at the same time accounting for an identification rate < 100  
 15111 % is extremely useful.

15112 McClintock et al. (2009a,b) suggest an intuitive means of correcting for this bias in a  
 15113 non-spatial model framework when dealing with a Poisson encounter model (or sampling  
 15114 with replacement). When marked but unknown resightings are part of the data, the  
 15115 expected number of records of unmarked individuals at time  $t$ ,  $n_t$ , changes from Eq.  
 15116 19.1.2 to:

$$E(n) = (N - m)\lambda + \eta/m$$

15117 Here,  $\lambda$  is the individual encounter rate estimated from the known resighted individuals  
 15118 and  $\eta$  is the number of records of marked but unidentified individuals. So, because the  
 15119 observed  $\lambda$  is known to be too low, the average number of unidentified pictures per known  
 15120 individual is added as a correction factor. This procedure assumes that the inability to  
 15121 identify a marked individual occurs at random throughout the population, which seems  
 15122 to be a reasonable assumption under most circumstances.

15123 We can relatively easily translate this concept to our spatial mark-resight models. In  
 15124 the spatial model framework we are interested in the individual and trap specific encounter  
 15125 rate,  $\lambda_{ij}$ . Further, we do not look at the sum of all records of unmarked individuals, but  
 15126 formulate the model conditional on the latent individual encounter histories. Thus, instead  
 15127 of using  $\eta/m$  as a correction factor, we need something that applies at the individual and  
 15128 trap level. If we take the sum of all correctly identified records of marked individuals,  
 15129  $\sum y_c$  and divide it by the total number of records of marked individuals,  $\sum y_m$ , we get  
 15130 the average rate of correct individual identification for marked individuals, say,  $c$ :

$$c = \sum y_c / \sum y_m$$

15131 We could then apply  $c$  as a correction factor for  $\lambda_0$  for the marked individuals.

15132 A more formal, model-based way to specify  $c$  is by assuming that

$$\sum y_c \sim \text{Binomial}(\sum y_m, c)$$

15133 and estimating  $c$  as another model parameter, so that we account for the uncertainty about  
 15134 it. If we choose an uninformative (and conjugate) beta(1, 1) prior for  $c$ , we can update it  
 15135 directly from its full conditional distribution, which is beta( $1 + \sum y_c, 1 + (\sum y_m - \sum y_c)$ ),  
 15136 within our MCMC algorithm.

15137 For the marked individuals we can then multiply  $\lambda_0$  with  $c$  to account for the fact that  
 15138 we observe incomplete individual encounter histories. Since we don't have this identifica-  
 15139 tion issue for unmarked individuals, their baseline trap encounter rate remains as before  
 15140 simply  $\lambda_0$  (or in other words, their  $c$  equals 1). Observe that now, in addition to assuming  
 15141 that failure to identify tagged individuals occurs at random throughout the population,  
 15142 we also assume that it occurs at random throughout space, i.e. our success of identifying  
 15143 a tagged individual does not depend on the trap we encounter it in. Incomplete individual  
 15144 identification of marked individuals is included as an option in the `scrPID` function and  
 15145 we show an example of using  $c$  in an analysis in sec. 19.6.

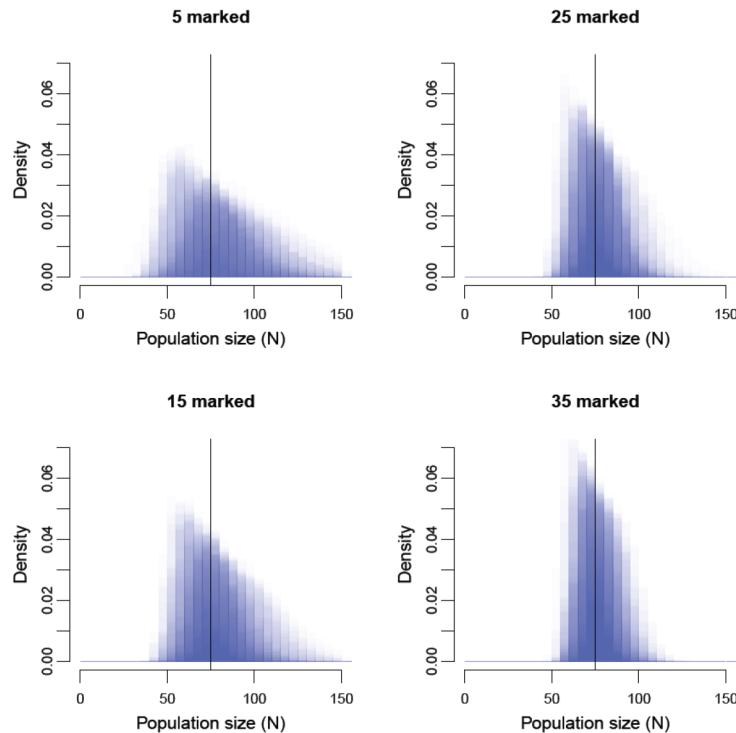
15146 **Imperfect individual identification and unknown number of marks.** The ap-  
 15147 proach described above works only if the number of marked individuals is known because,  
 15148 in that case, we can observe the all-zero encounter histories of marked individuals and  
 15149 know that all augmented individuals have to be unmarked individuals. If the number of  
 15150 marked animals is unknown, on the other hand, some of the augmented individuals may  
 15151 well be marked individuals we never observed. For those individuals we should multiply  
 15152  $\lambda_0$  with  $c$ , but we don't know who (or how many) they are. As of this moment we have  
 15153 not implemented a model with unknown number of marked individuals and imperfect  
 15154 identification of marks. It seems like one strategy to tackle that problem would be to  
 15155 estimate the number of marked and unmarked individuals separately, using two sets of  
 15156 data augmentation (i.e., estimating  $\psi_{\text{marked}}$  and  $\psi_{\text{unmarked}}$ ), but with shared detection  
 15157 parameters,  $\sigma$  and  $\lambda_0$ .

15158 As long as individuals are identified based on the same type of tags the assumption that  
 15159 failure to identify marked individuals occurs at random throughout the population should  
 15160 be valid. The assumption that failure to identify marked individuals occurs at random in  
 15161 space could be violated, for example when spatially varying habitat conditions influence  
 15162 the ability to recognize individual tags, or when an observer effect influences individual  
 15163 identification rates. While we haven't ourselves experimented with it, we believe that the  
 15164 approach described above could readily be extended to account for these differences. For  
 15165 example, identification rates could be calculated separately for different observers, or be  
 15166 modeled as functions of habitat covariates. As an alternative to the approach we present  
 15167 here, model development could explore assigning records of marked but unidentified indi-  
 15168 viduals to marked individuals in a fashion similar to how unmarked records are assigned  
 15169 to hypothetical individuals in this model, namely, based on the location of the record and  
 15170 the estimates of home range centers of marked individuals. While this is computationally  
 15171 more advanced it would make full use of the spatial information of the unmarked records.

## 19.5 HOW MUCH INFORMATION DO MARKED AND UNMARKED INDIVIDUALS CONTRIBUTE?

15172 It is intuitive that having marked individuals in the study population should lead to more  
 15173 accurate and precise parameter estimates than when no individuals are identifiable. To  
 15174 evaluate how strongly adding marked individuals to a population improves parameter

estimates, Chandler and Royle (In press) performed a simulation study. They used a  $15 \times 15$  trapping grid and simulated detection data of  $N = 75$  individuals in a  $20 \times 20$  units state-space over  $k = 5$  occasions with  $\sigma = 0.5$  and  $\lambda_0 = 0.5$ . They generated 100 datasets each for  $m = (0, 5, 15, 25, 35)$  where  $m$  is the known number of marked individuals randomly sampled from the population.



**Figure 19.2.** Overlaid posterior distributions of  $N$  from 100 simulations for four levels of marked individuals.

Without any marked individuals in the population, the posterior distribution of  $N$  turned out to be highly skewed, but its mode was still an approximately unbiased point estimator of  $N$ . As anticipated, posterior precision increased substantially with the proportion of marked individuals (Tab. 19.3 and Fig. 19.2). The posterior mode was approximately unbiased as a point estimator, and the relative root-mean squared error decreased from 0.246 when no individuals were marked to 0.085 when 35 individuals were marked (Tab. 19.3). Coverage was nominal for all values of  $m$  and posterior skew greatly diminished with increasing  $m$  (Tab. 19.3).

As we saw in the previous chapter, the spatial correlation in unmarked counts can be sufficient to obtain estimates of movement and detection parameters. However, only

**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

marked and thus identifiable individuals provide us with direct information about these parameters and may well dominate estimates. To single out the contribution of marked and unmarked individuals to parameter estimates, we re-ran the same simulations but 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 locational data 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 half-normal detection function. 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 recaptures/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  $R$  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:

$$[\sigma | \mathbf{l}, \mathbf{s}] \propto \left\{ \prod_{i=1}^m \prod_{r=1}^R \frac{1}{2\pi\sigma^2} \exp \left( -1/2 \left[ \frac{(l_{irx} - s_{ix})^2}{\sigma^2} + \frac{(l_{iry} - s_{iy})^2}{\sigma^2} \right] \right) \right\} * [\sigma]$$

and

$$[\mathbf{s}_i | \mathbf{l}, \sigma] \propto \left\{ \prod_{r=1}^R \frac{1}{2\pi\sigma^2} \exp \left( -1/2 \left[ \frac{(l_{irx} - s_{ix})^2}{\sigma^2} + \frac{(l_{iry} - s_{iy})^2}{\sigma^2} \right] \right) \right\} * [\mathbf{s}_i]$$

Under the standard mark-resight assumption that marked individuals are a representative sample of the population, the estimate of  $\sigma$  can be applied for the entire population. For the unmarked individuals  $\mathbf{s}_i$  are estimated as described before conditional on their latent encounter histories.

**R** makes it easy to implement the update of  $\sigma$  and  $\mathbf{s}_i$  based on telemetry data and the above described full conditionals within our existing MCMC algorithm. We replace the current updating step for  $\sigma$  with:

```
15233 #ntot = number of telemetry-tagged individuals
15234 #locs = list of length ntot; each element is a matrix
```

```

15235 #with telemetry locations
15236 #telID = vector with identifier for telemetry-tagged
15237 #individuals
15238
15239 sigma.cand <- rnorm(1, sigma, delta[1])
15240 if (sigma.cand > 0) {
15241
15242 llsig<-llsig.cand<-rep(NA, ntot)
15243
15244 for (x in 1:ntot) {
15245 lls[x]<-sum(dmvnorm(x=locs[[x]],mean=c(S[telID[x],1],S[telID[x],2]),
15246 sigma=cbind(c(sigma^2,0), c(0,sigma^2)), log=T))
15247 lls.cand[x]<-sum(dmvnorm(x=locs[[x]],mean=c(S[telID[x],1],S[telID[x],2]),
15248 sigma=cbind(c(sigma.cand^2,0), c(0,sigma.cand^2)), log=T))
15249 }
15250 if(runif(1) < exp( sum(lls.cand) - sum(lls) ) ){
15251   sigma<-sigma.cand
15252   lam <- lam0*exp(-(D*D)/(2*sigma.cand*sigma.cand))
15253 }
15254 }

```

15255 For the  $s_i$  we use an analogous updater for the telemetry-tagged individuals and the  
15256 regular updater for individuals without associated telemetry location information. A full  
15257 example can be found in the **R** package **scrbook**, by calling **scrPID.tel**. Note that not  
15258 all marked individuals need to be telemetry-tagged, but telemetry data used on the model  
15259 should correspond to the period over which resighting surveys were conducted (as we  
15260 discussed in Chapt. 5, both the  $s_i$  and  $\sigma$  should only be interpreted against the specific  
15261 sampling period). Further, this approach of incorporating telemetry data into a spatial  
15262 mark-resight model can easily be extended to update  $\sigma$  and  $s$  conditional on both resighting  
15263 and telemetry data and applies equally to regular SCR models where all individuals are  
15264 identifiable.

15265 **Example: Raccoons on the Outer Banks of North Carolina**

15266 Solmann et al. (2012b) applied a spatial mark-resight model with telemetry data to  
15267 a camera-trap and radio-telemetry data set from the raccoon population on South Core  
15268 Banks, a barrier island within Cape Lookout National Seashore, North Carolina. Between  
15269 May and September 2007, 131 raccoons were marked with dog collars and large indi-  
15270 vidual numbered cattle tags; 44 of these tagged individuals were equipped with radio  
15271 collars. Collared individuals were located using a VHF receiver and antenna, and their  
15272 locations were estimated approximately weekly. Twenty camera traps were set up along  
15273 the length of South Core Banks and camera trapping data collected between October 1  
15274 2007 to January 22 2008 constituted the resighting data in this analysis. During this  
15275 period 104 marked individuals, 38 radio-collared, were alive and available for resighting  
15276 with camera traps.

15277 The state-space  $\mathcal{S}$  was defined as the entire area of South Core Banks island. A  
15278 change in the number of photocaptures over the course of the study suggested a variation  
15279 of detection rate with time. Since date recording in cameras malfunctioned, photographic



**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*)

records could only be assigned to the time interval between subsequent trap checks, and these intervals between checks are referred to as sampling occasions. These occasions ranged from 2 to 43 days;  $\lambda_0$  was standardized to 7-day intervals and allowed to change with sampling occasion. Since not all pictures of marked raccoons could be identified to the individual level, the authors applied the correction factor  $c$  as described in sec. 19.4, estimated separately for each occasion.

Camera-traps recorded 117 pictures of unmarked raccoons, 33 pictures of 18 marked and identifiable raccoons, and 49 records of marked but not individually identifiable individuals (Fig. 19.3). An average of 16.32 telemetry locations (SD 4.91) were collected for each of the 38 collared individuals. Raccoon abundance on the island was estimated at 186.712 (SE 14.810) individuals, which translated to a density of 8.291 (SE 0.658) individuals per  $km^2$ . Parameter estimates are listed in Tab. 19.5.

In this study, although a large number of raccoons were tagged, photographic data of these tagged individuals were surprisingly sparse. Analysis of the photographic data set without the telemetry data did not render usable estimates as parallel Markov chains did not converge. One reason for the relatively sparse data was the camera trap study design: traps were spaced on average 1.77 km apart, which is about 3.5 times  $\sigma$ . Consequently, very few individual raccoons were photographed at more than one trap. Under these

**Table 19.5.** Summary statistics of parameter estimates 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%
$\sigma$	0.491 (0.010)	0.472	0.491	0.512
$\lambda_0$ (t=1)	0.237 (0.045)	0.158	0.234	0.335
$\lambda_0$ (t=2)	0.397 (0.081)	0.257	0.391	0.573
$\lambda_0$ (t=3)	0.108 (0.028)	0.061	0.105	0.170
$\lambda_0$ (t=4)	0.296 (0.073)	0.174	0.289	0.459
$\lambda_0$ (t=5)	0.032 (0.011)	0.015	0.030	0.056
$\lambda_0$ (t=6)	0.031 (0.009)	0.016	0.030	0.052
$c$ (t=1)	0.545 (0.085)	0.377	0.546	0.709
$c$ (t=2)	0.389 (0.112)	0.184	0.385	0.616
$c$ (t=3)	0.294 (0.107)	0.110	0.286	0.523
$c$ (t=4)	0.375 (0.162)	0.099	0.364	0.710
$c$ (t=5)	0.375 (0.161)	0.099	0.364	0.709
$c$ (t=6)	0.300 (0.138)	0.075	0.287	0.600
$N$	186.712 (14.810)	162	185	220
$D$	8.291 (0.658)	7.194	8.215	9.769

circumstances, the telemetry data provide the necessary spatial information to estimate  $\sigma$  and the activity centers of individual animals and thus make other model parameter estimable. Similarly, in a camera-trapping study on Florida panthers (*Puma concolor coryi*), Sollmann et al. (in revision), including telemetry data from the 3 individuals that were collared and known to use the study area resulted in density estimates with considerably higher precision as compared to preliminary estimates *without* telemetry location data, reducing the width of the 95 % BCI by about 60 %. Such improvements in precision of estimates is especially important when we are interested in changes in the population over time.

## 19.7 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. 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. Especially for difficult-to-study, cryptic species where often only a small sample of the population can be

15319 tagged this enables researchers to make optimal use of all existing data and obtain robust  
15320 density estimates without the need for additional invasive methods. Just as SCR, the  
15321 spatial mark-resight model framework is flexible to account for a variety of factors that  
15322 may influence individual movement and detection, as well as survey-related parameters,  
15323 and we saw one example for the Canada geese, where  $\sigma$  was sex-specific.

15324 Spatial mark-resight models are a fairly new development and much remains to be ex-  
15325 plored. We mentioned the assignment of marked but unidentified records to actual marked  
15326 individuals based on their spatial location, which provides some (though imperfect) infor-  
15327 mation of their identity (sec. 19.4). Similarly, records where the marked status cannot be  
15328 determined could potentially be included in the model as some form of overall correction  
15329 factor on detection. GPS telemetry devices and their ability to collect location data with  
15330 much higher frequency offer the opportunity to assign records of collared animals to indi-  
15331 viduals based on how close to a given camera the collared individuals were, both in space  
15332 and time. In this scenario, individual identity itself could be expressed probabilistically,  
15333 leading to an SMR model accounting for potential misidentification. All these possible  
15334 extensions can tailor SMR models to specific survey techniques. As such, the approach is  
15335 applicable to a wide range of population estimation problems when dealing with animals  
15336 that cannot be identified based on natural marks.



15337

# 20

15338

15339

15340

## 2012: A SPATIAL CAPTURE-RECAPTURE ODYSSEY

15341 Capture recapture methods have been a cornerstone of ecological modeling and analysis  
15342 for decades. Yet there are essentially no real capture-recapture data sets that come *without*  
15343 auxiliary spatial information about location of capture (but sometimes such information  
15344 is thrown into the trashcan).

15345 The big point is that we provide a framework for spatial analysis of animal populations  
15346 from individual encounter data: MOVEMENT, SPACE USAGE, SPATIAL VARIATION  
15347 IN DENSITY – much to be done: how do individuals interact? how is space usage  
15348 changing over time, etc...

15349 Topics to discuss here:

15350 (1) Strauss process model (2) Need for general purpose software.... all of the spatial  
15351 stuff + open populations in one big model. (3) Efficient computation is still an issue. (4)  
15352 Fit and model selection will continue to be important practical issues.

## 20.1 10 THESIS OR DISSERTATION TOPICS

- 15353 Future research directions:  
15354 Modeling dynamics of the point process. Transient individuals. Dispersal. Things like  
15355 that.  
15356 Calibration of GoF under meaningful alternatives  
15357 Calibration of AIC/DIC and efficacy study  
15358 Models for non-uniform point processes that exhibit clustering or repulsion  
15359 no-marking model + RSF  
15360 occupancy and counts data + SCR data (AOAS and Sollmann et al.)  
15361 Spatial genetics – can use SCR to study gene flow, related things....  
15362 SCR on dendritic networks (streams and trails).

## 20.2 THREE DIMESIONAL SPACE

- 15363 Throughout this book we have treated space as two-dimensional, meaning that activity  
15364 centers are assumed to occur on the real plane. This approximation of reality is reasonable  
15365 for many terrestrial species, but aquatic organisms, especially marine animals move about  
15366 in three-dimensional space. Treating space as three-dimensional could also conceivably  
15367 be useful in studies of flying organisms or species that use multiple strata of tall forests;  
15368 however, we suspect that two dimensional models of space should suffice in such contexts.  
15369 Regardless, a three-dimensional view of space requires that activity centers  $s_i$  are indexed  
15370 by  $x, y, z$  coordinates. In theory, this presents no problem whatsoever. In practice, estima-  
15371 tion based on integrated likelihood methods must involve a three-dimensional integration.  
15372 This will clearly be more computationally demanding, but it should be possible using  
15373 packages such as **R2Cuba**.

## 20.3 GREGARIOUS SPECIES

- 15374 Many social species move about in large groups rather than as single individuals. Even  
15375 species regarded as solitary often join family groups for some portion of their life cycle.  
15376 The consequences of gregariousness?? are x-fold....  
15377 To account for this, we change our definition of  $s_i$  from the location of an individual's  
15378 activity center, to the location of a group's activity center. We then expand our model to  
15379 include a submodel for group size, and we can estimate both the density of group activity  
15380 centers and total population size.

15381

## Part V

15382

---

15383

# Appendices



15384

## APPENDIX I - USEFUL SOFTWARE AND 15385 R PACKAGES

15386

---

15387 Throughout this book we have used a suite of software and R packages, all of which are  
15388 freely available online. To make life a little easier for you, here we provide you with a list  
15389 of all software and R packages, download links and some (hopefully) helpful tips regarding  
15390 their installation.

### 20.4 WINBUGS

15391 Although **WinBUGS** (Gilks et al., 1994) is becoming increasingly obsolete with the  
15392 faster and more flexible **OpenBUGS** and **JAGS**, there are still situations in which  
15393 the program 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  
15394 and double-click on the .exe file and follow the installation instructions on the screen. On  
15395 64 bit machines, according to the BUGS project you should download a zip file (from the  
15396 same page) and unzip it into a folder of your choice. There are a couple of additional  
15397 steps to make BUGS run. First, you need to obtain a key (which is free and valid for  
15398 life) here: '[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 key comes with instructions on how to activate it. Second, you need  
15399 to update the basic **WinBUGS** version to the most current one (which is from August  
15400 2007) following 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  
15401 ready to use after quitting and re-opening it. Remember that **WinBUGS** only runs on  
15402 Windows machines. Also, there appears to be a problem installing the program in Vista,  
15403 although we have no personal experience with this.  
15404

#### 20.4.1 WinBUGS through R

15405 While you can run **WinBUGS** as a standalone application, we recommend you access  
15406 it from within **R** using the package **R2WinBUGS** (Sturtz et al., 2005), so you can conve-  
15407 niently process your output, make graphs etc. **R2WinBUGS** also allows you to run mod-  
15408 els in **OpenBUGS** (see below). You can install the package from within **R** directly  
15409 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  
15410 manual with some examples ('[http://voterview.com/bayes\\_beach/R2WinBUGS.pdf](http://voterview.com/bayes_beach/R2WinBUGS.pdf)').  
15411

## 20.5 OPENBUGS

15415 **OpenBUGS** is the up-to-date version of **WinBUGS** and can be downloaded here:  
 15416 ''<http://www.openbugs.info/w/Downloads>'' (Windows, Mac and Linux versions are  
 15417 available). The name '**OpenBUGS**' refers to the software being open source, so users  
 15418 do not need to download a license key, like they have to for **WinBUGS** (although the  
 15419 license key for **WinBUGS** is free and valid for life). For Windows, install by double-  
 15420 clicking on the .exe file and following the instructions on the installer screen. Compared  
 15421 to **WinBUGS**, **OpenBUGS** has more built-in functions. The method of how to deter-  
 15422 mine the right updater for each model parameter has changed and the user can manually  
 15423 control the MCMC algorithm used to update model parameters. Several other changes  
 15424 have been implemented in **OpenBUGS** and a detailed list of differences between the two  
 15425 **BUGS** versions, can be found at <http://www.openbugs.info/w/OpenVsWin>. We have  
 15426 encountered convergence problems with simple scr models in this program. There is an  
 15427 extensive help archive for both **WinBUGS** and **OpenBUGS** and you can subscribe to  
 15428 a mailing list, where people pose and answer questions of how to use these programs at  
 15429 <http://www.mrc-bsu.cam.ac.uk/bugs/overview/list.shtml>

15430 **20.5.1 OpenBUGS through R**

15431 Like **WinBUGS**, **OpenBUGS** can be used as a standalone application or through **R**.  
 15432 There are several packages that allow **R** to interface with **OpenBUGS**, all of which can  
 15433 be installed directly from a cran mirror:

15434 **R2WinBUGS**: One of the options in the `bugs()` call is `program`, which lets you specify either  
 15435 **WinBUGS** or **OpenBUGS**. This is a convenient option because after having worked  
 15436 through some of this book you will likely be familiar with the format of `bugs()` output  
 15437 and other functions of the **R2WinBUGS** package.

15438 **R2OpenBUGS**: **R2OpenBUGS** (Sturtz et al., 2005) is very similar to, and actually based on,  
 15439 **R2WinBUGS** and it is unclear to us what can be gained by using the former over the latter.  
 15440 Arguments of the `bugs()` call differ slightly between the two packages and given that  
 15441 **R2WinBUGS** allows for the use of both **OpenBUGS** and **WinBUGS** it is probably easiest  
 15442 to stick with it.

15443 **BRugs**: **BRugs** (Thomas et al., 2006) can be installed from within **R** directly from a cran  
 15444 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 ac-  
 15445 cess at ''<http://www.rni.helsinki.fi/openbugs/OpenBUGS/Docu/BRugs%20Manual.html>''.  
 15446 **BRugs** has the convenient feature that all pieces of a **BUGS** analysis can be run  
 15447 from within **R**, including checking the model syntax, something that requires opening the  
 15448 **BUGS** GUI with other packages.

## 20.6 JAGS

15450 **JAGS** (Just Another Gibbs Sampler) (Plummer, 2003) runs scr models considerably faster  
 15451 than **WinBUGS**, does not have the convergence problem with simple scr models we have

15452 encountered in **OpenBUGS** but similar to the latter program, is flexible and constantly  
15453 updated. Writing a **JAGS** model is virtually identical to writing a **WinBUGS** model.  
15454 However, some functions may have slightly different names and you can look up available  
15455 functions and their use in the **JAGS** manual. One potential downside is that **JAGS** can  
15456 be very particular when it comes to initial values. These may have to be set as close to  
15457 truth as possible for the model to start. Although **JAGS** lets you run several parallel  
15458 Markov chains, this characteristic interferes with the idea of using overdispersed initial  
15459 values for the different chains. Also, we have found that when running models, sometimes  
15460 **JAGS** crashes for unclear reasons, taking **R** down with it. Oftentimes, in order to make  
15461 it run again you'll have to go through downloading and installing it again (remove the  
15462 non-functioning version first).

15463 **JAGS** has a variety of functions that are not available in **WinBUGS**. For example,  
15464 **JAGS** allows you to supply observed data for some deterministic functions of unobserved  
15465 variables. In **BUGS** we cannot supply data to logical nodes. Another useful feature is  
15466 that the adaptive phase of the model (the burn-in) is run separately from the sampling  
15467 from the stationary Markov chains. This allows you to easily add more iterations to the  
15468 adaptive phase if necessary without the need to start from 0. There are other, more  
15469 subtle differences and there is an entire manual section on differences between **JAGS** and  
15470 **OpenBUGS**.

15471 **JAGS** is available for download at '<http://sourceforge.net/projects/mcmc-jags/files/>', together with the R package **rjags** (Plummer, 2011), which allows running  
15472 **JAGS** through **R**, user and installation manuals and examples. At this site **JAGS** is  
15473 available for Windows and Mac; Linux binaries are distributed separately and you can  
15474 find links to various sources here: '<http://mcmc-jags.sourceforge.net/>'. **JAGS**  
15475 comes with a 32 bit and a 64 bit version and can be installed by double-clicking on the  
15476 .exe file and following the instructions on the installer screen. For questions and prob-  
15477 lems concerning **JAGS** there is a forum online at <http://sourceforge.net/projects/mcmc-jags/>.  
15478 forums/forum/610037.

### 15480 20.6.1 JAGS through R

15481 Unlike the two **BUGS** programs, **JAGS** does not have a GUI interface but a command  
15482 line interface that can be used to run the program as a standalone application. **JAGS**  
15483 will solely perform the MCMC simulation; analyzing and summarizing the output has to  
15484 be done outside of **JAGS**. To run **JAGS** through **R** you have two options.

15485 **rjags**: As mentioned above, **rjags** (Plummer, 2011) can be found together with **JAGS**  
15486 and was developed/is being maintained by the inventor of **JAGS**, which means it is  
15487 guaranteed to stay up to date when/as **JAGS** changes. The package can be installed from  
15488 a cran mirror and the help document can be accessed at '<http://cran.r-project.org/web/packages/rjags/rjags.pdf>'

15490 **R2jags**: Alternatively, the package **R2jags** (Su and Yajima, 2011) provides a means of  
15491 accessing **JAGS** through **R**. We prefer **rjags** for the reason named above, as well as be-  
15492 cause it stores data in a more memory-efficient way and has better **plot()** and **summary()**  
15493 methods.

## 20.7 R

15494 At the time of the preparation of this list, **R** for Windows is at version 2.15.0, which can be  
 15495 downloaded at <http://cran.r-project.org/bin/windows/base/> This site also contains help-  
 15496 ful tips on how to install **R** in Windows Vista, how to update **R** packages etc. Installation  
 15497 of **R** in Windows is straightforward: download the .exe file, double-click on it and follow  
 15498 the instructions of the Windows installer. The later versions of **R** come with versions for  
 15499 both 64 bit and 32 bit machines. The **R** site (''<http://mirrors.softliste.de/cran/>'')  
 15500 has an extensive FAQ section Hornik (2011), which includes instructions on how to install  
 15501 **R** on Unix and Mac computers.

### 15502 20.7.1 R packages

15503 This section provides an alphabetical list of useful **R** packages. There is a large number  
 15504 of **R** packages and by no means is this list intended to be complete in terms of what is  
 15505 useful. Rather, we list packages that we are familiar with and that we employ at one point  
 15506 or the other in this book. Unless explicitly stated otherwise, all packages can be installed  
 15507 directly from within **R** trough a cran mirror.

15508 **adapt**: **adapt** (Genz et al., 2007) is a package for multidimensional numerical integration.  
 15509 The package has been removed from the CRAN repository but can be obtained from  
 15510 ''<http://cran.r-project.org/src/contrib/Archive/adapt/>''.

15511 **coda**: **coda** (Plummer et al., 2006) lets you summarize and perform diagnostics on mcmc  
 15512 output. For a list and description of functions, see the manual at ''<http://cran.r-project.org/web/packages/coda/coda.pdf>''.

15514 **gdistance**: **gdistance** (van Etten, 2011) is a package for calculating distances and routes  
 15515 on geographical grids and can be used to calculate least cost path surfaces. Manual at  
 15516 ''<http://cran.r-project.org/web/packages/gdistance/gdistance.pdf>''.

15517 **igraph**: **igraph** (Csardi and Nepusz, 2006) provides routines for graphs and network anal-  
 15518 ysis. Manual at ''<http://cran.r-project.org/web/packages/igraph/igraph.pdf>''.

15519 **inline**: **inline** (Sklyar et al., 2010) allows the user to define R functions with in-lined **C**,  
 15520 **C++** or **Fortran** code. Manual at <http://cran.r-project.org/web/packages/inline/inline.pdf>.

15522 **maps**: **maptools** (?) is a library of maps. Manual at ''<http://cran.r-project.org/web/packages/maps/index.html>''.

15524 **maptools**: **maptools** (Lewin-Koh et al., 2011) provides a set of tools for reading and manip-  
 15525 ulating spatial data, especially ESRI shapefiles. Manual at ''<http://cran.r-project.org/web/packages/maptools/maptools.pdf>''.

15527 **R2cuba**: **R2cuba** (Hahn et al., 2010) is another package for multidimensional integration.  
 15528 Manual at ''<http://cran.r-project.org/web/packages/R2Cuba/R2Cuba.pdf>''.

15529 **raster**: **raster** (van Etten, 2012) provides functions for geographic analysis and modeling  
15530 with raster data. Manual at '<http://cran.r-project.org/web/packages/raster/raster.pdf>'.

15532 **Rcpp**: **Rcpp** (Eddelbuettel and François, 2011) provides R functions as well as a C++ library  
15533 which facilitate the integration of R and C++. Manual at <http://cran.r-project.org/web/packages/Rcpp/Rcpp.pdf>.

15535 **RcppArmadillo**: **RcppArmadillo** (François et al., 2011) is a templated C++ linear algebra  
15536 library, integrating the **Armadillo** library and R. Manual at <http://cran.r-project.org/web/packages/RcppArmadillo/RcppArmadillo.pdf>.

15538 **reshape**: **reshape** (Wickham and Hadley, 2007) allows you to easily manipulate, summarize  
15539 and reshape data. Manual at '<http://cran.r-project.org/web/packages/reshape/reshape.pdf>'.

15541 **rgeos**: **rgeos** (Bivand and Rundel, 2011) provides many useful functions for spatial operations  
15542 such as intersecting or buffering spatial features. Manual at '<http://cran.r-project.org/web/packages/rgeos/rgeos.pdf>'.

15544 **SCRbayes**: (Russell et al., 2012)

15545 **secr**: **secr** (Efford et al., 2009a)

15546 **shapefiles**: **shapefiles** (Stabler, 2006) allows you to read and write ESRI shapefiles  
15547 (i.e. shapefiles you would use in ArcGIS). Manual at '<http://cran.r-project.org/web/packages/shapefiles/shapefiles.pdf>'.

15549 **snow**, **snowfall**: **snow** (Tierney et al., 2011) and **snowfall** (Knaus, 2010) provide functionality  
15550 for parallel computing. The latter is a more user-friendly wrapper around the former.  
15551 Manuals at <http://cran.r-project.org/web/packages/snowfall/snowfall.pdf>  
15552 and <http://cran.r-project.org/web/packages/snow/snow.pdf>.

15553 **sp**: **sp** (Pebesma and Bivand, 2011) is a package for plotting, selecting, subsetting etc.  
15554 spatial data. **sp** and **spatstat** (see below) are complementary in many ways and data  
15555 formats can be easily converted between the two packages. Manual at '<http://cran.r-project.org/web/packages/sp/sp.pdf>'.

15557 **SPACECAP**: (Gopalaswamy et al., 2012a)

15558 **spatstat**: **spatstat** (Baddeley and Turner, 2005) is an extensive package for analyzing  
15559 spatial data. We use it, for example, to generate random points within a state space  
15560 that cannot be described as a rectangle but consists of a (or several) arbitrary polygon(s).  
15561 Manual at '<http://cran.r-project.org/web/packages/spatstat/spatstat.pdf>'.

15562 **unmarked**:

15563 #####

15564 #####

15565 **References**

15566 #####



## BIBLIOGRAPHY

15567

15568

---

- 15569 Adriaensen, F., Chardon, J. P., De Blust, G., Swinnen, E., Villalba, S., Gulinck, H.,  
15570 and Matthysen, E. (2003), "The application of 'least-cost' modelling as a functional  
15571 landscape model," *Landscape and urban planning*, 64, 233–247.
- 15572 Aitkin, M. (1991), "Posterior bayes factors," *Journal of the Royal Statistical Society.  
Series B (Methodological)*, 111–142.
- 15573 Alho, J. M. (1990), "Logistic regression in capture-recapture models," *Biometrics*, 623–  
15574 635.
- 15575 Alpízar-Jara, R. and Pollock, K. H. (1996), "A combination line transect and capture-  
15576 recapture sampling model for multiple observers in aerial surveys," *Environmental and  
15577 Ecological Statistics*, 3, 311–327.
- 15578 Amstrup, S. C., McDonald, T. L., and Manly, B. F. J. (2005), *Handbook of capture-  
15579 recapture analysis*, Princeton Univ Pr.
- 15580 Anderson, D. R., Burnham, K. P., White, G. C., and Otis, D. L. (1983), "Density Esti-  
15581 mation of Small-Mammal Populations Using a Trapping Web and Distance Sampling  
15582 Methods," *Ecology*, 64, 674–680.
- 15583 Arnason, A., Schwarz, C., and Gerrard, J. (1991), "Estimating closed population size  
15584 and number of marked animals from sighting data," *Journal of Wildlife Management*,  
15585 716–730.
- 15586 Arnason, N. A. (1972), "Parameter estimates from mark-recapture experiments on two  
15587 populations subject to migration and death," *Researches on Population Ecology*, 13,  
15588 97–113.
- 15589 — (1973), "The estimation of population size, migration rates and survival in a stratified  
15590 population," *Researches on Population Ecology*, 15, 1–8.
- 15591 Baddeley, A. and Turner, R. (2005), "Spatstat: an R package for analyzing spatial point  
15592 patterns," *Journal of Statistical Software*, 12, 1–42, ISSN 1548-7660.
- 15593 Bales, S. L., Hellgren, E. C., Leslie Jr., D. M., and Hemphill Jr., J. (2005), "Dynamics  
15594 of recolonizing populations of black bears in the Ouachita Mountains of Oklahoma,"  
15595 *Wildlife Society Bulletin*, 1342–1351.
- 15596 Balme, G. A., Slotow, R., and Hunter, L. T. B. (2010), "Edge effects and the impact of non-  
15597 protected areas in carnivore conservation: leopards in the Phinda–Mkuze Complex,  
15598 South Africa," *Animal Conservation*, 13, 315–323.
- 15599 Bartmann, R. M., White, G. C., Carpenter, L. H., and Garrott, R. A. (1987), "Aerial  
15600 Mark-Recapture Estimates of Confined Mule Deer in Pinyon-Juniper Woodland," *The  
15601 Journal of Wildlife Management*, 51, 41–46.
- 15602 Bayne, E. and Hobson, K. (2002), "Annual survival of adult American Redstarts and  
15603 Ovenbirds in the southern boreal forest," *The Wilson Bulletin*, 114, 358–367.

- 15605 Beier, P., Majka, D. R., and Spencer, W. D. (2008), "Forks in the road: choices in  
15606 procedures for designing wildland linkages," *Conservation Biology*, 22, 836–851.
- 15607 Belant, J., Van Stappen, J., and Paetkau, D. (2005), "American black bear population  
15608 size and genetic diversity at Apostle Islands National Lakeshore," *Ursus*, 16, 85–92.
- 15609 Berger, J. O., Liseo, B., and Wolpert, R. L. (1999), "Integrated likelihood methods for  
15610 eliminating nuisance parameters," *Statistical Science*, 1–22.
- 15611 Best, N. G., Ickstadt, K., and Wolpert, R. L. (2000), "Spatial Poisson Regression for  
15612 Health and Exposure Data Measured at Disparate Resolutions," *Journal of the Ameri-*  
15613 *cian Statistical Association*, 95, 1076.
- 15614 Bibby, C. J. and Buckland, S. T. (1987), "Bias of bird census results due to detectability  
15615 varying with habitat," *Acta Ecologica*, 8, 103–112.
- 15616 Bibby, C. J., Burgess, N. D., and Hill, D. A. (1992), "Bird census techniques: Academic  
15617 Press," *London, UK*, 257.
- 15618 Bivand, R. and Rundel, C. (2011), *rgeos: Interface to Geometry Engine - Open Source*  
15619 (*GEOS*), r package version 0.1-8.
- 15620 Blair, W. (1940), "A study of prairie deer-mouse populations in southern Michigan,"  
15621 *American Midland Naturalist*, 273–305.
- 15622 Bolker, B. (2008), *Ecological models and data in R*, Princeton Univ Pr.
- 15623 Bondrup-Nielsen, S. (1983), "Density estimation as a function of live-trapping grid and  
15624 home range size," *Canadian Journal of Zoology*, 61, 2361–2365.
- 15625 Borchers, D. (2012), "A non-technical overview of spatially explicit capture–recapture  
15626 models," *Journal of Ornithology*, 1–10.
- 15627 Borchers, D. L., Buckland, S. T., and Zucchini, W. (2002), *Estimating animal abundance:*  
15628 *closed populations*, vol. 13, Springer Verlag.
- 15629 Borchers, D. L. and Efford, M. G. (2008), "Spatially explicit maximum likelihood methods  
15630 for capture–recapture studies," *Biometrics*, 64, 377–385.
- 15631 Borchers, D. L., Zucchini, W., and Fewster, R. M. (1998), "Mark-recapture models for  
15632 line transect surveys," *Biometrics*, 1207–1220.
- 15633 Borchers, D. L. et al. (xxxx), "gibbons surveys stuff from ISEC," xxxx, xxxx.
- 15634 Boulanger, J. and McLellan, B. (2001), "Closure violation in DNA-based mark-recapture  
15635 estimation of grizzly bear populations," *Canadian Journal of Zoology*, 79, 642–651.
- 15636 Bowden, D. C. (1993), "A simple technique for estimating population size. Technical Re-  
15637 port 93/12." Tech. rep., Department of Statistics, Colorado State University, Fort  
15638 Collins, Colorado, USA.
- 15639 Brooks, S. P., Catchpole, E. A., and Morgan, B. J. T. (2000), "Bayesian Animal Survival  
15640 Estimation," *Statistical Science*, 15, 357–376.
- 15641 Brownie, C., Hines, J. E., Nichols, J. D., Pollock, K. H., and Hestbeck, J. B. (1993),  
15642 "Capture-recapture studies for multiple strata including non-Markovian transitions."  
15643 *Biometrics*, 49, 1173–1187.
- 15644 Buckland, S., Anderson, D., Burnham, K., Laake, J., Borcher, D., and L, T. (2001), *In-*  
15645 *roduction to distance sampling: estimating abundance of biological populations*, Oxford,  
15646 UK: Oxford University Press.
- 15647 Buckland, S. T. (2004), *Advanced distance sampling*, Oxford University Press, USA.
- 15648 Burnham, K., Anderson, D., and Laake, J. (1980), "Estimation of density from line tran-  
15649 sect sampling of biological populations," *Wildlife monographs*, 3–202.
- 15650 Burnham, K. P. and Anderson, D. R. (2002), *Model selection and multimodel inference:*

- 15651        *a practical information-theoretic approach*, Springer Verlag.
- 15652     Burnham, K. P. and Overton, W. S. (1978), "Estimation of the size of a closed population  
15653        when capture probabilities vary among animals," *Biometrika*, 65, 625.
- 15654     Burt, W. (1943), "Territoriality and home range concepts as applied to mammals," *Journal  
15655        of mammalogy*, 24, 346–352.
- 15656     Casella, G. and Berger, R. L. (2002), *Statistical inference*, Duxbury Press.
- 15657     Casella, G. and George, E. I. (1992), "Explaining the Gibbs sampler," *American Statisti-  
15658        cian*, 46, 167–174.
- 15659     Caswell, H. (1989), *Matrix population models*, Sinauer assoc. Sunderland.
- 15660     Chandler, R., Royle, J., and King, D. (2011), "Inference about density and temporary  
15661        emigration in unmarked populations," *Ecology*, 92, 1429–1435.
- 15662     Chandler, R. B. and Royle, J. A. (In press), "Spatially-explicit models for inference about  
15663        density in unmarked or partially marked populations," *Annals of Applied Statistics*.
- 15664     Clobert, J., Danchin, E., Dhondt, A., and Nichols, J. (2001), *Dispersal*, Oxford.
- 15665     Cochran, W. (2007), *Sampling techniques*, John Wiley & Sons.
- 15666     Conde, D., Colchero, F., Zarza, H., Christensen, N., Sexton, J., Manterola, C., Chávez,  
15667        C., Rivera, A., Azuara, D., and Ceballos, G. (2010), "Sex matters: Modeling male  
15668        and female habitat differences for jaguar conservation," *Biological Conservation*, 143,  
15669        1980–1988.
- 15670     Converse, S., White, G., and Block, W. (2006a), "Small mammal responses to thinning  
15671        and wildfire in ponderosa pine-dominated forests of the southwestern United States,"  
15672        *Journal of Wildlife Management*, 70, 1711–1722.
- 15673     Converse, S., White, G., Farris, K., and Zack, S. (2006b), "Small mammals and forest fuel  
15674        reduction: national-scale responses to fire and fire surrogates," *Ecological Applications*,  
15675        16, 1717–1729.
- 15676     Converse, S. J. and Royle, J. A. (2012), "Dealing with incomplete and variable detectabil-  
15677        ity in multi-year, multi-site monitoring of ecological populations," in *Design and Anal-  
15678        ysis of Long-term Ecological Monitoring Studies*, eds. Gitzen, R. R., Millspaugh, J. J.,  
15679        Cooper, A. B., and Licht, D. S., Cambridge University Press, pp. 426–442.
- 15680     Cormack, R. M. (1964), "Estimates of survival from the sighting of marked animals,"  
15681        *Biometrika*, 51, 429–438.
- 15682     Coull, B. A. and Agresti, A. (1999), "The Use of Mixed Logit Models to Reflect Hetero-  
15683        geneity in Capture-Recapture Studies," *Biometrics*, 55, 294–301.
- 15684     Cox, D. (1955), "Some statistical methods connected with series of events," *Journal of  
15685        the Royal Statistical Society. Series B (Methodological)*, 129–164.
- 15686     Cressie, N. (1991), *Statistics for spatial data*, Wiley Series in Probability and Mathematical  
15687        Statistics.
- 15688     Csardi, G. and Nepusz, T. (2006), "The igraph software package for complex network  
15689        research," *InterJournal*, Complex Systems, 1695.
- 15690     Cushman, S. A., Compton, B. W., and McGarigal, K. (2010), "Habitat fragmentation  
15691        effects depend on complex interactions between population size and dispersal ability:  
15692        Modeling influences of roads, agriculture and residential development across a range  
15693        of life-history characteristics," in *Spatial complexity, informatics, and wildlife conser-  
15694        vation*, eds. Cushman, S. A. and Huettmann, F., New York: Springer, chap. 20, pp.  
15695        369–385.
- 15696     Cushman, S. A., McKelvey, K. S., Hayden, J., and Schwartz, M. K. (2006), "Gene flow in

- 15697 complex landscapes: testing multiple hypotheses with causal modeling," *The American  
15698 Naturalist*, 168, 486–499.
- 15699 Dawson, D. K. and Efford, M. G. (2009), "Bird population density estimated from acoustic  
15700 signals," *Journal of Applied Ecology*, 46, 1201–1209.
- 15701 DeGraaf, R. M. and Yamasaki, M. (2001), *New England wildlife: habitat, natural history,  
15702 and distribution*, University Press of New England.
- 15703 DeSante, D. F., Burton, K. M., Saracco, J. F., and Walker, B. L. (1995), "Productiv-  
15704 ity indices and survival rate estimates from MAPS, a continent-wide programme of  
15705 constant-effort mist-netting in North America," *Journal of Applied Statistics*, 22, 935–  
15706 948.
- 15707 Dice, L. R. (1938), "Some census methods for mammals," *Journal of Wildlife Management*,  
15708 2, 119–130.
- 15709 — (1941), "Methods for estimating populations of mammals," *Journal of Wildlife Man-  
15710 agement*, 5, 398–407.
- 15711 Diggle, P. J. (2003), *Statistical analysis of spatial point processes*, London: Arnold, 2nd  
15712 ed.
- 15713 Dijkstra, E. W. (1959), "A note on two problems in connexion with graphs," *Numerische  
15714 mathematik*, 1, 269–271.
- 15715 Dillon, A. and Kelly, M. (2007), "Ocelot *Leopardus pardalis* in Belize: the impact of trap  
15716 spacing and distance moved on density estimates," *Oryx*, 41, 469–477.
- 15717 Dixon, P. (2002), "Bootstrap resampling," *Encyclopedia of environmetrics*.
- 15718 Dorazio, R. M. (2007), "On the choice of statistical models for estimating occurrence and  
15719 extinction from animal surveys," *Ecology*, 88, 2773–2782.
- 15720 Dorazio, R. M. and Royle, J. A. (2003), "Mixture models for estimating the size of a closed  
15721 population when capture rates vary among individuals," *Biometrics*, 351–364.
- 15722 Durban, J. and Elston, D. (2005), "Mark-recapture with occasion and individual effects:  
15723 abundance estimation through Bayesian model selection in a fixed dimensional parame-  
15724 ter space," *Journal of agricultural, biological, and environmental statistics*, 10, 291–305.
- 15725 Eddelbuettel, D. and François, R. (2011), "Rcpp: Seamless R and C++ Integration,"  
15726 *Journal of Statistical Software*, 40, 1–18.
- 15727 Efford, M. (2004), "Density estimation in live-trapping studies," *Oikos*, 106, 598–610.
- 15728 — (2011), *secr: Spatially explicit capture-recapture models*, R package version 2.3.1.
- 15729 Efford, M. and Dawson, D. (2009), "Effect of distance-related heterogeneity on population  
15730 size estimates from point counts," *The Auk*, 126, 100–111.
- 15731 — (2010), "SECR for acoustic data," .
- 15732 Efford, M. and Fewster, R. (2012), "Estimating population size by spatially explicit  
15733 capture–recapture," *Oikos*.
- 15734 Efford, M. G., Borchers, D. L., and Byrom, A. E. (2009a), "Density estimation by spatially  
15735 explicit capture–recapture: likelihood-based methods," *Modeling demographic processes  
15736 in marked populations*, 255–269.
- 15737 Efford, M. G., Dawson, D. K., and Borchers, D. L. (2009b), "Population density estimated  
15738 from locations of individuals on a passive detector array," *Ecology*, 90, 2676–2682.
- 15739 Efford, M. G., Dawson, D. K., and Robbins, C. S. (2004), "DENSITY: software for  
15740 analysing capture-recapture data from passive detector arrays," *Animal Biodiversity  
and Conservation*, 217–228.
- 15741 Efford, M. G., Warburton, B., Coleman, M. C., and Barker, R. J. (2005), "A field test of

- two methods for density estimation," *Wildlife Society Bulletin*, 33, 731–738.
- Epps, C. W., Palsbøll, P. J., Wehausen, J. D., Roderick, G. K., Ramey II, R. R., and McCullough, D. R. (2005), "Highways block gene flow and cause a rapid decline in genetic diversity of desert bighorn sheep," *Ecology Letters*, 8, 1029–1038.
- Epps, C. W., Wehausen, J. D., Bleich, V. C., Torres, S. G., and Brashares, J. S. (2007), "Optimizing dispersal and corridor models using landscape genetics," *Journal of Applied Ecology*, 44, 714–724.
- Dormann, C., M McPherson, J., B Araújo, M., Bivand, R., Bolliger, J., Carl, G., Davies, R., Hirzel, A., Jetz, W., Daniel Kissling, W., et al. (2007), "Methods to account for spatial autocorrelation in the analysis of species distributional data: a review," *Ecography*, 30, 609–628.
- Farnsworth, G. L., Pollock, K. H., Nichols, J. D., Simons, T. R., E., H. J., and R., S. J. (2002), "A removal model for estimating detection probabilities from point-count surveys," *The Auk*, 119, 414–425.
- Fecske, D., Barry, R., Precht, F., Quigley, H., Bittner, S., and Webster, T. (2002), "Habitat use by female black bears in western Maryland," *Southeastern Naturalist*, 1, 77–92.
- Fieberg, J. (2007), "Utilization distribution estimation using weighted kernel density estimators," *The Journal of wildlife management*, 71, 1669–1675.
- Fieberg, J. and Kochanny, C. (2005), "Quantifying home-range overlap: the importance of the utilization distribution," *Journal of Wildlife Management*, 69, 1346–1359.
- Fienberg, S. E., Johnson, M. S., and Junker, B. W. (1999), "Classical multilevel and Bayesian approaches to population size estimation using multiple lists," *Journal of the Royal Statistical Society of London A*, 163, 383–405.
- Fiske, I. J. and Chandler, R. B. (2011), "unmarked: An R package for fitting hierarchical models of wildlife occurrence and abundance," *Journal Of Statistical Software*, 43, 1–23.
- Forester, J., Ives, A., Turner, M., Anderson, D., Fortin, D., Beyer, H., Smith, D., and Boyce, M. (2007), "State-space models link elk movement patterns to landscape characteristics in Yellowstone National Park," *Ecological Monographs*, 77, 285–299.
- Fortin, D., Beyer, H. L., Boyce, M. S., Smith, D. W., Duchesne, T., and Mao, J. S. (2005), "Wolves influence elk movements: behavior shapes a trophic cascade in Yellowstone National Park," *Ecology*, 86, 1320–1330.
- Foster, R. J. and Harmsen, B. J. (2012), "A critique of density estimation from camera-trap data," *The Journal of Wildlife Management*, 76, 224–236.
- François, R., Eddelbuettel, D., and Bates, D. (2011), *RcppArmadillo: Rcpp integration for Armadillo templated linear algebra library*, r package version 0.2.25.
- Frary, V., Duchamp, J., Maehr, D., and Larkin, J. (2011), "Density and distribution of a colonizing front of the American black bear *Ursus americanus*," *Wildlife Biology*, 17, 404–416.
- Fujiwara, M., Anderson, K., Neubert, M., and Caswell, H. (2006), "On the estimation of dispersal kernels from individual mark-recapture data," *Environmental and Ecological Statistics*, 13, 183–197.
- García-Alaníz, N., Naranjo, E. J., and Mallory, F. F. (2010), "Hair-snare: A non-invasive method for monitoring felid populations in the Selva Lacandona, Mexico," *Tropical Conservation Science*, 3, 403–411.
- Gardner, B., Reppucci, J., Lucherini, M., and Royle, J. (2010a), "Spatially explicit inference for open populations: estimating demographic parameters from camera-trap

- 15789 studies," *Ecology*, 91, 3376–3383.
- 15790 Gardner, B., Royle, J. A., and Wegan, M. T. (2009), "Hierarchical models for estimating  
15791 density from DNA mark-recapture studies," *Ecology*, 90, 1106–1115.
- 15792 Gardner, B., Royle, J. A., Wegan, M. T., Rainbolt, R. E., and Curtis, P. D. (2010b),  
15793 "Estimating black bear density using DNA data from hair snares," *The Journal of  
15794 Wildlife Management*, 74, 318–325.
- 15795 Garshelis, D. L. and Hristienko, H. (2006), "State and provincial estimates of American  
15796 black bear numbers versus assessments of population trend," *Ursus*, 17, 1–7.
- 15797 Gelfand, A. and Smith, A. (1990), "Sampling-based approaches to calculating marginal  
15798 densities," *Journal of the American statistical association*, 85, 398–409.
- 15799 Gelman, A. (2006), "Prior distributions for variance parameters in hierarchical models,"  
15800 *Bayesian analysis*, 1, 515–533.
- 15801 Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004), *Bayesian data analysis,  
15802 second edition.*, Bocan Raton, Florida, USA: CRC/Chapman & Hall.
- 15803 Gelman, A., Meng, X. L., and Stern, H. (1996), "Posterior predictive assessment of model  
15804 fitness via realized discrepancies," *Statistica Sinica*, 6, 733–759.
- 15805 Gelman, A. and Rubin, D. B. (1992), "Inference from iterative simulation using multiple  
15806 sequences," *Statistical Science*, 7, 457–511.
- 15807 Geman, S. and Geman, D. (1984), "Stochastic relaxation, Gibbs distributions, and the  
15808 Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine  
15809 Intelligence*, PAMI-6, 721–741.
- 15810 Genz, A. S., Meyer, M. R., Lumley, T., and Maechler, M. (2007), "The adapt Package. R  
15811 package version 1.0-4," .
- 15812 Gerlach, G. and Musolf, K. (2000), "Fragmentation of landscape as a cause for genetic  
15813 subdivision in bank voles," *Conservation Biology*, 14, 1066–1074.
- 15814 Gilks, W. and Wild, P. (1992), "Adaptive rejection sampling for Gibbs sampling," *Applied  
15815 Statistics*, 41, 337–348.
- 15816 Gilks, W. R., Thomas, A., and Spiegelhalter, D. J. (1994), "A Language and Program for  
15817 Complex Bayesian Modelling," *Journal of the Royal Statistical Society. Series D (The  
15818 Statistician)*, 43, 169–177, ArticleType: primary\_article / Issue Title: Special Issue:  
15819 Conference on Practical Bayesian Statistics, 1992 (3) / Full publication date: 1994 /  
15820 Copyright 1994 Royal Statistical Society.
- 15821 Gimenez, O., Rossi, V., Choquet, R., Dehais, C., Doris, B., Varella, H., Vila, J. P., and  
15822 Pradel, R. (2007), "State-space modelling of data on marked individuals," *Ecological  
15823 Modelling*, 206, 431–438.
- 15824 Gopalaswamy, A. M. (2012), "Capture-recapture models, spatially explicit," in *Encyclopedia  
15825 of Environmentrics Second Edition*, eds. El-Shaarawi, A. H. and Piegorsch, W.,  
15826 John Wiley and Sons Ltd, Chichester, UK.
- 15827 Gopalaswamy, A. M., Royle, A. J., Hines, J., Singh, P., Jathanna, D., Kumar, N. S., and  
15828 Karanth, K. U. (2012a), "Program SPACECAP: software for estimating animal density  
15829 using spatially explicit capturerecapture models," *Methods in Ecology and Evolution*,  
15830 online early, r package version 1.0.4.
- 15831 Gopalaswamy, A. M., Royle, J. A., Delampady, M., Nichols, J. D., Karanth, K. U., and  
15832 Macdonald, D. W. (2012b), "Density estimation in tiger populations: combining infor-  
15833 mation for strong inference," *Ecology*, 93, 1741–1751.
- 15834 Gruber, D. and White, M. (1983), "Black bear food habits in Yosemite National Park,"

- 15835      *Bears: Their Biology and Management*, 1–10.
- 15836      Greig-Smith, P. (1964), *Quantitative plant ecology*, Butterworths (Washington).
- 15837      Grimm, V., Revilla, E., Berger, U., Jeltsch, F., Mooij, W., Railsback, S., Thulke, H.-H., Weiner, J., Wiegand, T., and DeAngelis, D. (2005), “Pattern-oriented modeling of agent-based complex systems: Lessons from ecology.” *Science*, 310, 987–991.
- 15840      Hahn, T., Bouvier, A., and Kiêu, K. (2010), *R2Cuba: Multidimensional Numerical Integration*, r package version 1.0-6.
- 15842      Hall, R. J., Henry, P. F. P., and Bunck, C. M. (1999), “Fifty-year trends in a box turtle population in Maryland,” *Biological Conservation*, 88, 165–172.
- 15844      Hanski, I. A. (1999), *Metapopulation Ecology*, Oxford Univiversity Press.
- 15845      Hastings, W. (1970), “Monte Carlo sampling methods using Markov chains and their applications,” *Biometrika*, 57, 97–109.
- 15847      Hawkins, C. E. and Racey, P. A. (2005), “Low population density of a tropical forest carnivore, Cryptoprocta ferox: implications for protected area management,” *Oryx*, 39, 35–43.
- 15850      Hayes, R. J. and Buckland, S. T. (1983), “Radial distance models for the line transect method.” *Biometrics*, 39, 29–42.
- 15852      Hayne, D. (1950), “Apparent home range of Microtus in relation to distance between traps,” *Journal of mammalogy*, 31, 26–39.
- 15854      Hayne, D. W. (1949), “An Examination of the Strip Census Method for Estimating Animal Populations,” *The Journal of Wildlife Management*, 13, 145–157.
- 15856      Hedley, S. L., Buckland, S. T., and Borchers, D. L. (1999), “Spatial modelling from line transect data,” *Journal of Cetacean Research and Management*, 1, 255–264.
- 15858      Hestbeck, J. B. and Malecki, R. A. (1989), “Mark-resight estimate of Canada Goose midwinter numbers,” *Journal of Wildlife Management*, 53, 749–752.
- 15860      Higdon, D. (1998), “A process-convolution approach to modelling temperatures in the North Atlantic Ocean,” *Environmental and Ecological Statistics*, 5, 173–190.
- 15862      Hobbs, N. T. (2011), “An Ecological Modeler’s Primer on JAGS,” .
- 15863      Holdenried, R. (1940), “A population study of the long-eared chipmunk (*Eutamias quadrivittatus*) in the central Sierra Nevada,” *Journal of Mammalogy*, 21, 405–411.
- 15865      Hooten, M., Johnson, D., Hanks, E., and Lowry, J. (2010), “Agent-based inference for animal movement and selection,” *Journal of agricultural, biological, and environmental statistics*, 15, 523–538.
- 15868      Hooten, M. and Wikle, C. (2010), “Statistical agent-based models for discrete spatio-temporal systems.” *Journal of the American Statistical Association*, 105, 236–248.
- 15870      Hornik, K. (2011), “The R FAQ,” ISBN 3-900051-08-9.
- 15871      Huggins, R. M. (1989), “On the statistical analysis of capture experiments,” *Biometrika*, 76, 133.
- 15873      Hurlbert, S. H. (1984), “Pseudoreplication and the design of ecological field experiments,” *Ecological monographs*, 54, 187–211.
- 15875      Illian, J., Penttinen, A., Stoyan, H., and Stoyan, D. (2008), *Statistical analysis and modelling of spatial point patterns*, Wiley.
- 15877      Ivan, J. (2012), “Density, demography, and seasonal movements of snowshoe hares in central Colorado,” Ph.D. thesis, Colorado State University.
- 15879      Jackson, R., Roe, J., Wangchuk, R., and Hunter, D. (2006), “Estimating Snow Leopard Population Abundance Using Photography and Capture-Recapture Techniques,”

- 15881        *Wildlife Society Bulletin*, 34, 772–781.
- 15882        Jett, D. A. and Nichols, J. D. (1987), “A Field Comparison of Nested Grid and Trapping  
15883        Web Density Estimators,” *Journal of Mammalogy*, 68, 888–892.
- 15884        Jhala, Y. V., Qureshi, Q., Gopal, R., and R., S. P. (2011), “Status of tigers, co-predators  
15885        and prey in India.” Tech. rep., National Tiger Conservation Authority, Govt. of India,  
15886        New Delhi and Wildlife Institute of India, Dehradun.
- 15887        Johnson (2010), “A Model-Based Approach for Making Ecological Inference from Distance  
15888        Sampling Data,” *Biometrics*, 66, 310318.
- 15889        Johnson, D. (1980), “The comparison of usage and availability measurements for evaluat-  
15890        ing resource preference,” *Ecology*, 61, 65–71.
- 15891        Johnson, D. H. (1999), “The insignificance of statistical significance testing,” *The journal  
15892        of wildlife management*, 763–772.
- 15893        Johnson, D. S., Thomas, D. L., Ver Hoef, J. M., and Christ, A. (2008), “A general  
15894        framework for the analysis of animal resource selection from telemetry data,” *Biometrics*,  
15895        64, 968–976.
- 15896        Jolly, G. M. (1965), “Explicit estimates from capture-recapture data with both death and  
15897        dilution-stochastic model,” *Biometrika*, 52, 225–247.
- 15898        Jonsen, I. D., Flemming, J. M., and Myers, R. A. (2005), “Robust state-space modeling  
15899        of animal movement data,” *Ecology*, 86, 2874–2880.
- 15900        Kadane, J. B. and Lazar, N. A. (2004), “Methods and criteria for model selection,” *Journal  
15901        of the American Statistical Association*, 99, 279–290.
- 15902        Karanth, K. U. (1995), “Estimating tiger *Panthera tigris* populations from camera-trap  
15903        data using capture–recapture models,” *Biological Conservation*, 71, 333–338.
- 15904        Karanth, K. U. and Nichols, J. D. (1998), “Estimation of Tiger Densities in India Using  
15905        Photographic Captures and Recaptures,” *Ecology*, 79, 2852–2862.
- 15906        — (2000), “Ecological status and conservation of tigers in India,” *WCS, US Fish and  
15907        Wildlife Service. Centre for Wildlife Studies. Bangalore, India*, 123.
- 15908        — (2002), *Monitoring tigers and their prey: a manual for researchers, managers and  
15909        conservationists in {T}ropical {A}sia*, Bangalore, India: Centre for Wildlife Studies.
- 15910        Kass, R. and Wasserman, L. (1996), “The selection of prior distributions by formal rules,”  
15911        *Journal of the American Statistical Association*, 1343–1370.
- 15912        Kays, R. W., Slanson, K. M., Long, R. A., MacKay, P., Zielinski, W. J., and Ray, J. C.  
15913        (2008), “Remote cameras.” *Noninvasive survey methods for carnivores*, 110–140.
- 15914        Kelly, M., Noss, A., Di Bitetti, M., Maffei, L., Arispe, R., Paviolo, A., De Angelo, C., and  
15915        Di Blanco, Y. (2008), “Estimating puma densities from camera trapping across three  
15916        study sites: Bolivia, Argentina, and Belize,” *Journal of Mammalogy*, 89, 408–418.
- 15917        Kendall, K. C., Stetz, J. B., Boulanger, J., Macleod, A. C., Paetkau, D., and White, G. C.  
15918        (2009), “Demography and genetic structure of a recovering grizzly bear population,”  
15919        *The Journal of Wildlife Management*, 73, 3–16.
- 15920        Kendall, W. L., Nichols, J. D., and Hines, J. E. (1997), “Estimating Temporary Emigration  
15921        Using Capture-Recapture Data with Pollock’s Robust Design,” *Ecology*, 78, 563–578.
- 15922        Kéry, M. (2010), *Introduction to WinBUGS for Ecologists: Bayesian Approach to Regres-  
15923        sion, ANOVA, Mixed Models and Related Analyses*, Academic Press.
- 15924        — (2011), “Towards the modeling of true species distributions,” *J. Biogeography*, 617–618.
- 15925        Kéry, M., Gardner, B., and Monnerat, C. (2010), “Predicting species distributions from  
15926        checklist data using site-occupancy models,” *Journal of Biogeography*, 37, 1851–1862.

- 15927 Kéry, M., Gardner, B., Stoeckle, T., Weber, D., and Royle, J. A. (2011), "Use of Spatial  
15928 Capture–Recapture Modeling and DNA Data to Estimate Densities of Elusive Animals,"  
15929 *Conservation Biology*, 25, 356–364.
- 15930 Kéry, M., Royle, J. A., and Schmid, H. (2005), "Modeling avian abundance from replicated  
15931 counts using binomial mixture models," *Ecological Applications*, 15, 1450–1461.
- 15932 Kéry, M. and Schaub, M. (2012), *Bayesian Population Analysis Using WinBugs*, Academic  
15933 Press.
- 15934 King, R. (2009), "Missing," *missing*, Missing.
- 15935 King, R. and Brooks, S. (2001), "On the Bayesian analysis of population size," *Biometrika*,  
15936 88, 317–336.
- 15937 King, R., Brooks, S., and Coulson, T. (2008), "Analyzing Complex Capture–Recapture  
15938 Data in the Presence of Individual and Temporal Covariates and Model Uncertainty,"  
15939 *Biometrics*, 64, 1187–1195.
- 15940 Knaus, J. (2010), *snowfall: Easier cluster computing (based on snow)*., r package version  
15941 1.84.
- 15942 Koehler, G. M. and Pierce, D. J. (2003), "Black Bear Home-Range Sizes in Washington:  
15943 Climatic, Vegetative, and Social Influences," *Journal of Mammalogy*, 84, 81–91, Article-  
15944 Type: research-article / Full publication date: Feb., 2003 / Copyright 2003 American  
15945 Society of Mammalogists.
- 15946 Kohn, M., York, E., Kamradt, D., Haught, G., Sauvajot, R., and Wayne, R. (1999),  
15947 "Estimating population size by genotyping faeces," *Proceedings of the Royal Society of  
15948 London. Series B: Biological Sciences*, 266, 657–663.
- 15949 Krebs, C. J. (1999), *Ecological methodology*, Menlo Park, CA: Benjamin/Cummings.
- 15950 Kucera and Barrett (2011), "missing," *missing*, missing.
- 15951 Kuo, L. and Mallick, B. (1998), "Variable selection for regression models," *Sankhy: The  
15952 Indian Journal of Statistics, Series B*, 65–81.
- 15953 Laird, N. M. and Ware, J. H. (1982), "Random-effects models for longitudinal data,"  
15954 *Biometrics*, 963–974.
- 15955 Langtimm, C. A., Dorazio, R. M., Stith, B. M., and Doyle, T. J. (2011), "New aerial  
15956 survey and hierarchical model to estimate manatee abundance," *The Journal of Wildlife  
15957 Management*, 75, 399–412.
- 15958 Le Cam, L. (1990), "Maximum likelihood: an introduction," *International Statistical Review/  
15959 Revue Internationale de Statistique*, 153–171.
- 15960 Lebreton, J. D., Burnham, K. P., Clobert, J., and Anderson, D. R. (1992), "Modeling  
15961 Survival and Testing Biological Hypotheses Using Marked Animals: A Unified Approach  
15962 with Case Studies," *Ecological Monographs*, 62, 67–118.
- 15963 Lele, S. R. and Keim, J. L. (2006), "Weighted distributions and estimation of resource  
15964 selection probability functions," *Ecology*, 87, 3021–3028.
- 15965 Lele, S. R., Moreno, M., and Bayne, E. (2012), "Dealing with detection error in site  
15966 occupancy surveys: What can we do with a single survey?" *Journal of Plant Ecology*.
- 15967 Lewin-Koh, N. J., Bivand, R., contributions by Edzer J. Pebesma, Archer, E., Baddeley,  
15968 A., Bibiko, H.-J., Dray, S., Forrest, D., Friendly, M., Giraudoux, P., Golicher, D., Rubio,  
15969 V. G., Hausmann, P., Hufthammer, K. O., Jagger, T., Luque, S. P., MacQueen, D.,  
15970 Niccolai, A., Short, T., Stabler, B., and Turner, R. (2011), *maptools: Tools for reading  
15971 and handling spatial objects*, r package version 0.8-10.
- 15972 Link, W. (2003), "Nonidentifiability of Population Size from Capture–Recapture Data

- 15973      with Heterogeneous Detection Probabilities," *Biometrics*, 59, 1123–1130.  
15974      — (2004), "Individual heterogeneity and identifiability in capturerecapture models." in  
15975      *Proceedings of the International Conference EURING 2003, Deutschland. Animal Bio-*  
15976      *diversity and Conservation*, eds. Senar, J. C., Dhondt, A., and Conroy, M. J., vol. 27,  
15977      pp. 87 – 91.  
15978      Link, W. A. and Barker, R. J. (1994), "Density Estimation Using the Trapping Web  
15979      Design: A Geometric Analysis," *Biometrics*, 50, 733–745.  
15980      — (2006), "Model weights and the foundations of multimodel inference," *Ecology*, 87,  
15981      2626–2635.  
15982      — (2010), *Bayesian Inference: With Ecological Applications*, London, UK: Academic  
15983      Press.  
15984      Link, W. A. and Eaton, M. J. (2011), "On thinning of chains in MCMC," *Methods in*  
15985      *Ecology and Evolution*, 3, 112–115.  
15986      Link, W. A., Yoshizaki, J., Bailey, L. L., and Pollock, K. H. (2010), "Uncovering a latent  
15987      multinomial: analysis of markrecapture data with misidentification," *Biometrics*, 66,  
15988      178–185.  
15989      Liu and Wu (1999), "Parameter expansion for data augmentation," *J Am Stat Assoc*, 94,  
15990      1264–1274.  
15991      Lukacs, P. M. and Burnham, K. P. (2005), "Estimating population size from DNA-based  
15992      closed capture-recapture data incorporating genotyping error," *Journal of Wildlife*  
15993      *Management*, 69, 396–403.  
15994      Lunn, D., Spiegelhalter, D., Thomas, A., and Best, N. (2009), "The BUGS project: Evo-  
15995      lution, critique, and future directions," *Statistics in Medicine*, 28, 3049–3067.  
15996      Lunn, D. J., Thomas, A., Best, N., and Spiegelhalter, D. (2000), "WinBUGS-a Bayesian  
15997      modelling framework: concepts, structure, and extensibility," *Statistics and Computing*,  
15998      10, 325–337.  
15999      Lyons, A., Gaines, W., and Servheen, C. (2003), "Black bear resource selection in the  
16000      northeast Cascades, Washington," *Biological Conservation*, 113, 55–62.  
16001      Mace, R., Minta, S., Manley, T., and Aune, K. (1994), "Estimating grizzly bear population  
16002      size using camera sightings," *Wildlife Society Bulletin*, 22, 74–83.  
16003      MacEachern, S. N. and Berliner, L. M. (1994), "Subsampling the Gibbs sampler," *Amer-*  
16004      *ican Statistician*, 188–190.  
16005      MacKay, P., Smith, D. A., Long, R. A., Parker, M., Long, R. A., MacKay, P., Zielinski,  
16006      W. J., and Ray, J. C. (2008), "Scat detection dogs," *Noninvasive survey methods for*  
16007      *carnivores*, 183–222.  
16008      MacKenzie, D. I., Nichols, J. D., Lachman, G. B., Droege, S., Royle, J. A., and Langtimm,  
16009      C. A. (2002), "Estimating site occupancy rates when detection probabilities are less than  
16010      one," *Ecology*, 83, 2248–2255.  
16011      MacKenzie, D. I., Nichols, J. D., Royle, J. A., Pollock, K. H., Bailey, L. L., and Hines,  
16012      J. E. (2006), *Occupancy estimation and modeling: inferring patterns and dynamics of*  
16013      *species occurrence*, Academic Press.  
16014      Maffei, L. and Noss, A. J. (2008), "How small is too small? Camera trap survey areas and  
16015      density estimates for ocelots in the Bolivian Chaco," *Biotropica*, 40, 71–75.  
16016      Magoun, A. J., Long, C. D., Schwartz, M. K., Pilgrim, K. L., Lowell, R. E., and Valken-  
16017      burg, P. (2011), "Integrating motion-detection cameras and hair snags for wolverine  
16018      identification," *The Journal of Wildlife Management*, 75, 731–739.

- 16019 Manel, S., Schwartz, M. K., Luikart, G., and Taberlet, P. (2003), "Landscape genetics:  
16020 combining landscape ecology and population genetics," *Trends in Ecology & Evolution*,  
16021 18, 189–197.
- 16022 Manly, B., McDonald, L., Thomas, D., McDonald, T., and Erickson, W. (2002), *Resource  
16023 selection by animals: statistical design and analysis for field studies*, Springer, 2nd ed.
- 16024 Marques, T., Thomas, L., Ward, J., DiMarzio, N., and Tyack, P. (2009), "Estimating  
16025 cetacean population density using fixed passive acoustic sensors: An example with  
16026 Blainvilles beaked whales," *The Journal of the Acoustical Society of America*, 125,  
16027 1982.
- 16028 Marques, T. A., Thomas, L., and Royle, J. A. (2011), "A hierarchical model for spatial  
16029 capture-recapture data: Comment," *Ecology*, 92, 526–528.
- 16030 Mayor, S., Schneider, D., Schaefer, J., and Mahoney, S. (2009), "Habitat selection at  
16031 multiple scales," *Ecoscience*, 16, 238–247.
- 16032 McCarthy, M. A. (2007), *Bayesian Methods for Ecology*, Cambridge: Cambridge Univer-  
16033 sity Press.
- 16034 McClintock, B. and Hoeting, J. (2010), "Bayesian analysis of abundance for binomial  
16035 sighting data with unknown number of marked individuals," *Environmental and Eco-  
16036 logical Statistics*, 17, 317–332.
- 16037 McClintock, B., King, Thomas, Matthiopoulos, McConnell, and Morales (2012), "A gen-  
16038 eral discrete-time modeling framework for animal movement using multi-state random  
16039 walks," *Ecological Monographs*, 82, 335–349.
- 16040 McClintock, B. and White, G. (2010), "From NOREMARK to MARK: software for esti-  
16041 mating demographic parameters using mark-resight methodology," *Journal of Ornithol-  
16042 ogy*, 1–10.
- 16043 McClintock, B., White, G., Antolin, M., and Tripp, D. (2009a), "Estimating abundance  
16044 using mark-resight when sampling is with replacement or the number of marked indi-  
16045 viduals is unknown," *Biometrics*, 65, 237–246.
- 16046 McClintock, B., White, G., and Burnham, K. (2006), "A robust design mark-resight abun-  
16047 dance estimator allowing heterogeneity in resighting probabilities," *Journal of agricul-  
16048 tural, biological, and environmental statistics*, 11, 231–248.
- 16049 McClintock, B. T., White, G. C., Burnham, K. P., and Pryde, M. A. (2009b), "A general-  
16050 ized mixed effects model of abundance for mark-resight data when sampling is without  
16051 replacement," in *Modeling demographic processes in marked populations*, ed. Thomson,  
16052 D., New York: Springer, pp. 271–289.
- 16053 McCullagh, P. and Nelder, J. A. (1989), *Generalized linear models*, Chapman & Hall/CRC.
- 16054 McRae, B. H. and Beier, P. (2007), "Circuit theory predicts gene flow in plant and animal  
16055 populations," *Proceedings of the National Academy of Sciences*, 104, 19885.
- 16056 McRae, B. H., Dickson, B. G., Keitt, T. H., and Shah, V. B. (2008), "Using circuit theory  
16057 to model connectivity in ecology, evolution, and conservation," *Ecology*, 89, 2712–2724.
- 16058 Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., Teller, E., et al. (1953), "Equa-  
16059 tion of state calculations by fast computing machines," *The journal of chemical physics*,  
16060 21, 1087–1092.
- 16061 Metropolis, N. and Ulam, S. (1949), "The Monte Carlo method," *Journal of the American  
16062 Statistical Association*, 44, 335–341.
- 16063 Millar, R. B. (2009), "Comparison of hierarchical Bayesian models for overdispersed count  
16064 data using DIC and Bayes' Factors," *Biometrics*, 65, 962–969.

- 16065 Mills, L. S., Citta, J. J., Lair, K. P., Schwartz, M. K., and Tallmon, D. A. (2000), "Estimating animal abundance using noninvasive DNA sampling: promise and pitfalls,"  
16066 *Ecological Applications*, 10, 283–294.
- 16068 Minta, S. and Mangel, M. (1989), "A Simple Population Estimate Based on Simulation  
16069 for Capture-Recapture and Capture-Resight Data," *Ecology*, 70, 1738–1751.
- 16070 Møller, J. and Waagepetersen, R. P. (2004), *Statistical inference and simulation for spatial  
16071 point processes*, Chapman & Hall/CRC.
- 16072 Mohr, C. (1947), "Table of equivalent populations of North American small mammals,"  
16073 *American midland naturalist*, 37, 223–249.
- 16074 Moldenhauer, R. R. and Regelski, D. J. (1996), "Northern Parula (*Parula americana*),"  
16075 in *The Birds of North America Online*, ed. Poole, A., Ithaca, NY: Cornell Lab of  
16076 Ornithology.
- 16077 Mollet, P., Kéry, M., Gardner, B., Pasinelli, G., and A, R. J. (2012), "Population size  
16078 estimation for capercaille (*Tetrao urogallus* L.) using DNA-based individual recognition  
16079 and spatial capture-recapture models," *missing, missing, missing*.
- 16080 Morrison, M. L., Strickland, M. D., Block, W. M., Collier, B. A., and Peterson, M. J.  
16081 (2008), *Wildlife Study Design*, Springer.
- 16082 Muller, W. G. (2007), "Collecting spatial data: Optimum design of experiments for ran-  
16083 dom fields," .
- 16084 Neal, A., White, G., Gill, R., Reed, D., and Olterman, J. (1993), "Evaluation of mark-  
16085 resight model assumptions for estimating mountain sheep numbers," *Journal of Wildlife  
16086 Management*, 436–450.
- 16087 Neal, A. K. (1990), "Evaluation of mark-resight population estimates using simulations  
16088 and field data from mountain sheep." M.S. thesis, Colorado State University, Fort  
16089 Collins, Colorado, USA.
- 16090 Neal, R. (2003), "Slice sampling," *Annals of Statistics*, 31, 705–741.
- 16091 Nelder, J. A. and Wedderburn, R. W. M. (1972), "Generalized linear models," *Journal of  
16092 the Royal Statistical Society. Series A (General)*, 370–384.
- 16093 Nichols, J. D., Hines, J. E., Sauer, J. R., Fallon, F. W., Fallon, J. E., and Heglund,  
16094 P. J. (2000), "A double-observer approach for estimating detection probability and  
16095 abundance from point counts," *The Auk*, 117, 393–408.
- 16096 Nichols, J. D. and Karanth, K. U. (2002), "Statistical concepts: assessing spatial distri-  
16097 butions," in *Monitoring tigers and their prey: a manual for researchers, managers and  
16098 conservationists in Tropical Asia*, eds. Karanth, K. U. and Nichols, J. D., Bangalore,  
16099 India: Centre for Wildlife Studies, pp. 29–38.
- 16100 Niemi, A. and Fernández, C. (2010), "Bayesian Spatial Point Process Modeling of Line  
16101 Transect Data," *Journal of Agricultural, Biological, and Environmental Statistics*, 15,  
16102 327–345.
- 16103 Norris III, J. L. and Pollock, K. H. (1996), "Nonparametric MLE under two closed capture-  
16104 recapture models with heterogeneity," *Biometrics*, 639–649.
- 16105 O'Brien, T. (2011), "Abundance, density and relative abundance: A conceptual frame-  
16106 work," in *Camera traps in animal ecology: methods and analyses*, eds. O'Connell, A.  
16107 F. J., Nichols, J. D., and Karanth, U., Tokyo, Japan: Springer Verlag, pp. 71–96.
- 16108 O'Connell, A. F., Nichols, J. D., and Karanth, U. K. (2010), *Camera traps in animal  
16109 ecology: Methods and analyses*, Springer.
- 16110 O'Hara, R. and Sillanpää, M. (2009), "A review of Bayesian variable selection methods:

- what, how and which," *Bayesian Analysis*, 4, 85–118.
- Otis, D. L., Burnham, K. P., White, G. C., and Anderson, D. R. (1978), "Statistical inference from capture data on closed animal populations," *Wildlife monographs*, 3–135.
- Ovaskainen, O. (2004), "Habitat-specific movement parameters estimated using mark-recapture data and a diffusion model," *Ecology*, 85, 242–257.
- Ovaskainen, O., Rekola, H., Meyke, E., and Arjas, E. (2008), "Bayesian methods for analyzing movements in heterogeneous landscapes from mark-recapture data," *Ecology*, 89, 542–554.
- Overton, W. S. and Stehman, S. V. (1995), "The Horvitz-Thompson theorem as a unifying perspective for probability sampling: with examples from natural resource sampling," *American Statistician*, 261–268.
- Parmenter, R. R. and MacMahon, J. A. (1989), "Animal Density Estimation Using a Trapping Web Design: Field Validation Experiments," *Ecology*, 70, 169–179.
- Parmenter, R. R., Yates, T. L., Anderson, D. R., Burnham, K. P., Dunnum, J. L., Franklin, A. B., Friggens, M. T., Lubow, B. C., Miller, M., Olson, G. S., and Others (2003), "Small-mammal density estimation: A field comparison of grid-based vs. web-based density estimators," *Ecological Monographs*, 73, 1–26.
- Patterson, T., Thomas, L., Wilcox, C., Ovaskainen, O., and Matthiopoulos, J. (2008), "State-space models of individual animal movement," *Trends in ecology & evolution*, 23, 87–94.
- Paviolo, A., De Angelo, C., Di Blanco, Y., and Di Bitetti, M. (2008), "Jaguar *Panthera onca* population decline in the Upper Paraná Atlantic Forest of Argentina and Brazil," *Oryx*, 42, 554.
- Paviolo, A., Di Blanco, Y., De Angelo, C., and Di Bitetti, M. (2009), "Protection affects the abundance and activity patterns of pumas in the Atlantic Forest," *Journal of Mammalogy*, 90, 926–934.
- Pebesma, E. and Bivand, R. (2011), *Package 'sp'*, r package version 0.9-91.
- Pledger, S. (2000), "Unified maximum likelihood estimates for closed capture-recapture models using mixtures," *Biometrics*, 434–442.
- Plummer, M. (2003), "JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling," in *Proceedings of the 3rd International Workshop on Distributed Statistical Computing (DSC 2003)*. March, pp. 20–22.
- (2011), "rjags: Bayesian graphical models using mcmc. R package version 3-5," .
- Plummer, M., Best, N., Cowles, K., and Vines, K. (2006), "CODA: Convergence Diagnosis and Output Analysis for MCMC," *R News*, 6, 7–11.
- Pollock, K., Nichols, J., Brownie, C., and Hines, J. (1990), "Statistical inference for capture-recapture experiments," *Wildlife monographs*, 3–97.
- Pollock, K. H. (1982), "A Capture-Recapture Design Robust to Unequal Probability of Capture," *Journal of Wildlife Management*, 46, 752–757.
- Porneluzi, P. A. and Faaborg, J. (1999), "Season long fecundity, survival, and viability of Ovenbirds in fragmented and unfragmented landscapes," *Conservation Biology*, 13, 1151–1161.
- Pradel, R. (1996), "Utilization of Capture-Mark-Recapture for the Study of Recruitment and Population Growth Rate," *Biometrics*, 52, 703–709.
- Raabe, J. (2012), "Factors Influencing Distribution and Survival of Migratory Fishes Fol-

- 16157 lowing Multiple Low-Head Dam Removals on a North Carolina River. PhD dissertation.” Ph.D. thesis, North Carolina State University, Raleigh, NC.
- 16158 Rathbun, S. (1996), “Estimation of Poisson intensity using partially observed concomitant variables,” *Biometrics*, 226–242.
- 16159 Rathbun, S., Shiffman, S., and Gwaltney, C. (2007), “Modelling the effects of partially observed covariates on Poisson process intensity,” *Biometrika*, 94, 153–165.
- 16160 Rathbun, S. L. and Cressie, N. A. C. (1994), “A Space-Time Survival Point Process for a Longleaf Pine Forest in Southern Georgia.” *Journal of the American Statistical Association*, 89.
- 16161 Reich, B. J., Gardner, B., and Wilting, A. (2012), “A spatial capture-recapture model for territorial species,” *Biometrics in review*, xx, xx–xxx.
- 16162 Reynolds, D. and Beecham, J. (1980), “Home range activities and reproduction of black bears in west-central Idaho,” *Bears: Their Biology and Management*, 181–190.
- 16163 Robert, C. P. and Casella, G. (2004), *Monte Carlo statistical methods*, New York, USA: Springer.
- 16164 — (2010), *Introducing Monte Carlo Methods with R*, New York, USA: Springer.
- 16165 Roberts, G. O. and Rosenthal, J. S. (1998), “Optimal scaling of discrete approximations to Langevin diffusions,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60, 255–268.
- 16166 Royle, J. (2004a), “N-Mixture Models for Estimating Population Size from Spatially Replicated Counts,” *Biometrics*, 60, 108–115.
- 16167 — (2009a), “Analysis of capture-recapture models with individual covariates using data augmentation,” *Biometrics*, 65, 267–274.
- 16168 Royle, J. and Chandler, R. (2012), “Integrating Resource Selection Information with Spatial Capture-Recapture,” *arXiv preprint arXiv:1207.3288*.
- 16169 Royle, J., Chandler, R., Sun, C., and Fuller, A. (2012a), “Integrating Resource Selection Information with Spatial Capture-Recapture,” *needs updated MEE*.
- 16170 Royle, J. and Converse, S. (2012), “Hierarchical SCR models,” .... *in review*.
- 16171 Royle, J., Converse, S., and Link, W. (2012b), “Data Augmentation for Hierarchical Capture-recapture Models,” *arXiv preprint arXiv:1211.5706*.
- 16172 Royle, J. A. (2004b), “Generalized estimators of avian abundance from count survey data,” *Animal Biodiversity and Conservation*, 27, 375–386.
- 16173 — (2006), “Site occupancy models with heterogeneous detection probabilities,” *Biometrics*, 62, 97–102.
- 16174 — (2008), “Modeling individual effects in the Cormack–Jolly–Seber model: a state–space formulation,” *Biometrics*, 64, 364–370.
- 16175 — (2009b), “Analysis of capture–recapture models with individual covariates using data augmentation,” *Biometrics*, 65, 267–274.
- 16176 — (2010), “missing,” *missing*, missing.
- 16177 Royle, J. A., Chandler, R. B., Gazenski, K. D., and Graves, T. A. (2012c), “Spatial capture-recapture for jointly estimating population density and landscape connectivity,” *Ecology*, to appear.
- 16178 Royle, J. A., Dawson, D. K., and Bates, S. (2004), “Modeling abundance effects in distance sampling,” *Ecology*, 85, 1591–1597.
- 16179 Royle, J. A. and Dorazio, R. M. (2006), “Hierarchical models of animal abundance and occurrence,” *Journal of Agricultural, Biological, and Environmental Statistics*, 11, 249–

- 16203 263.
- 16204 — (2008), *Hierarchical modeling and inference in ecology: the analysis of data from pop-*  
16205 *ulations, metapopulations and communities*, Academic Press.
- 16206 — (2012), “Parameter-expanded data augmentation for Bayesian analysis of capture-  
16207 recapture models,” *Journal of Ornithology*, 152, S521–S537.
- 16208 Royle, J. A., Dorazio, R. M., and Link, W. A. (2007), “Analysis of multinomial models with  
16209 unknown index using data augmentation,” *Journal of Computational and Graphical  
16210 Statistics*, 16, 67–85.
- 16211 Royle, J. A. and Dubovsky, J. A. (2001), “Modeling spatial variation in waterfowl band-  
16212 recovery data,” *The Journal of wildlife management*, 726–737.
- 16213 Royle, J. A. and Gardner, B. (2011), “Hierarchical models for estimating density from trap-  
16214 ping arrays,” in *Camera traps in animal ecology: methods and analyses*, eds. O’Connel,  
16215 A. J., Nichols, J., and Karanth, U., Tokyo, Japan: Springer Verlag, pp. 163–190.
- 16216 Royle, J. A., Karanth, K. U., Gopalaswamy, A. M., and Kumar, N. S. (2009a), “Bayesian  
16217 inference in camera trapping studies for a class of spatial capture-recapture models,”  
16218 *Ecology*, 90, 3233–3244.
- 16219 Royle, J. A., Kéry, M., and Guélat, J. (2011a), “Spatial capture-recapture models for  
16220 search-encounter data,” *Methods in Ecology and Evolution*, 1–10.
- 16221 Royle, J. A. and Link, W. A. (2006), “Generalized site occupancy models allowing for  
16222 false positive and false negative errors,” *Ecology*, 87, 835–841.
- 16223 Royle, J. A., Magoun, A. J., Gardner, B., Valkenburg, P., and Lowell, R. E. (2011b),  
16224 “Density estimation in a wolverine population using spatial capture–recapture models,”  
16225 *The Journal of Wildlife Management*, 75, 604–611.
- 16226 Royle, J. A. and Nichols, J. D. (2003), “Estimating abundance from repeated presence-  
16227 absence data or point counts,” *Ecology*, 84, 777–790.
- 16228 Royle, J. A., Nichols, J. D., Karanth, K. U., and Gopalaswamy, A. M. (2009b), “A  
16229 hierarchical model for estimating density in camera-trap studies,” *Journal of Applied  
16230 Ecology*, 46, 118–127.
- 16231 Royle, J. A. and Young, K. V. (2008), “A Hierarchical Model For Spatial Capture-  
16232 Recapture Data,” *Ecology*, 89, 2281–2289.
- 16233 Russell, R. E., Royle, J. A., Desimone, R., Schwartz, M. K., Edwards, V. L., Pilgrim,  
16234 K. P., and McKelvey, K. S. (2012), “Estimating abundance of mountain lions from  
16235 unstructured spatial samples,” *Journal of Wildlife Management*.
- 16236 Rutledge, M. (2012), “XXX to be determined XXX,” Ph.D. thesis, North Carolina State  
16237 University.
- 16238 Sadeghpour, M. and Ginnett, T. (2011), “Habitat selection by female American black  
16239 bears in northern Wisconsin,” *Ursus*, 22, 159–166.
- 16240 Sæther, B.-E. and Bakke, Ø. (2000), “Avian life history variation and contribution of  
16241 demographic traits to the population growth rate,” *Ecology*, 81, 642653.
- 16242 Saïd, S. and Servanty, S. (2005), “The influence of landscape structure on female roe deer  
16243 home-range size,” *Landscape ecology*, 20, 1003–1012.
- 16244 Salom-Pérez, R., Carrillo, E., Sáenz, J. C., and Mora, J. M. (2007), “Critical condition of  
16245 the jaguar *Panthera onca* population in Corcovado National Park, Costa Rica,” *Oryx*,  
16246 51–56.
- 16247 Sanathanan, L. (1972), “Estimating the size of a multinomial population,” *The Annals of  
16248 Mathematical Statistics*, 142–152.

- 16249 Schofield, M. and Barker, R. (2008), "A unified capture-recapture framework," *Journal of*  
16250 *agricultural, biological, and environmental statistics*, 13, 458–477.
- 16251 Schwartz, M. K., Copeland, J. P., Anderson, N. J., Squires, J. R., Inman, R. M., McKelvey,  
16252 K. S., Pilgrim, K. L., Waits, L. P., and Cushman, S. A. (2009), "Wolverine gene flow  
16253 across a narrow climatic niche," *Ecology*, 90, 3222–3232.
- 16254 Schwartz, M. K. and Monfort, S. L. (2008), "Genetic and endocrine tools for carnivore  
16255 surveys," *Noninvasive survey methods for carnivores*, 238–262.
- 16256 Schwarz, C. J. and Arnason, A. N. (1996), "A General Methodology for the Analysis of  
16257 Capture-Recapture Experiments in Open Populations," *Biometrics*, 52, 860–873.
- 16258 — (2005), "Jolly-Seber Models in MARK," in *Program MARK: A Gentle Introduction*,  
16259 5th ed., eds. Cooch, E. G. and White, G.
- 16260 Schwarz, C. J., Bailey, R. E., Irvine, J. R., and Dalziel, F. C. (1993), "Estimating salmon  
16261 spawning escapement using capture-recapture methods." *Canadian Journal of Fisheries  
16262 and Aquatic Sciences*, 50, 1181–1191.
- 16263 Seber, G. A. F. (1965), "A Note on the Multiple-Recapture Census," *Biometrika*, 52,  
16264 249–259.
- 16265 — (1982), *The estimation of animal abundance and related parameters*, Macmillan Pub-  
16266 lishing Co.
- 16267 Sepúlveda, M. A., Bartheld, J. L., Monsalve, R., Gómez, V., and Medina-Vogel, G. (2007),  
16268 "Habitat use and spatial behaviour of the endangered Southern river otter (*Lontra  
16269 provocax*) in riparian habitats of Chile: conservation implications," *Biological Conserva-  
16270 tion*, 140, 329–338.
- 16271 Shirk, A. J., Wallin, D. O., Cushman, S. A., Rice, C. G., and Warheit, K. I. (2010),  
16272 "Inferring landscape effects on gene flow: a new model selection framework," *Molecular  
16273 ecology*, 19, 3603–3619.
- 16274 Sillett, S., Chandler, R., Royle, J., Kéry, M., and Morrison, S. (2012), "Hierarchical  
16275 distance sampling models to estimate population size and habitat-specific abundance  
16276 of an island endemic," *Ecological Applications*.
- 16277 Sillett, T., Rodenhouse, N., and Holmes, R. (2004), "Experimentally reducing neighbor  
16278 density affects reproduction and behavior of a migratory songbird," *Ecology*, 85, 2467–  
16279 2477.
- 16280 Simons, T. R., Pollock, K. H., Wetmore, J. M., Alldredge, M. W., Pacifici, K., and  
16281 Brewster, J. (2009), "Sources of Measurement Error, Misclassification Error, and Bias  
16282 in Auditory Avian Point Count Data," in *Modeling Demographic Processes In Marked  
16283 Populations*, eds. Thomson, D. L., Cooch, E. G., and Conroy, M. J., Boston, MA:  
16284 Springer US, vol. 3, pp. 237–254.
- 16285 Skalski, J. R., Millspaugh, J. J., and Spencer, R. D. (2005), "Population estimation and  
16286 biases in paintball, mark-resight surveys of elk," *Journal of Wildlife Management*, 69,  
16287 1043–1052.
- 16288 Sklyar, O., Murdoch, D., Smith, M., Eddelbuettel, D., and François, R. (2010), *inline:*  
16289 *Inline C, C++, Fortran function calls from R*, r package version 0.3.8.
- 16290 Smith, M. H., Blessing, R., Chelton, J. G., Gentry, J. B., Golley, F. B., and McGinnis,  
16291 J. T. (1971), "Determining density for small mammal populations using a grid and  
16292 assessment lines," *Acta Theriologica*, 16, 105–125.
- 16293 Soisalo, M. K. and Cavalcanti, S. M. C. (2006), "Close-up Space in "Radio-telemetry","  
16294 *Biological Conservation*, 129, 487–496.

- 16295 Sollmann, R., Furtado, M. M., Gardner, B., Hofer, H., Jacomo, A. T. A., Trres, N. M., and  
16296 Silveira, L. (2011), "Improving density estimates for elusive carnivores: Accounting for  
16297 sex-specific detection and movements using spatial capture-recapture models for jaguars  
16298 in central Brazil," *Biological Conservation*, 144, 1017–1024.
- 16299 Sollmann, R., Gardner, B., and Belant, J. L. (2012a), "How Does Spatial Study Design  
16300 Influence Density Estimates from Spatial Capture-Recapture Models?" *PloS one*, 7,  
16301 e34575.
- 16302 Sollmann, R., Gardner, B., Chandler, R. B., Shindle, D., Onorato, D. P., Royle, J. A., and  
16303 O'Connell, A. F. (in revision), "Using multiple data sources provides density estimates  
16304 for endangered Florida panther," *Journal of Applied Ecology*.
- 16305 Sollmann, R., Gardner, B., Parsons, A., Stocking, J., McClintock, B., Simons, T., Pol-  
16306 lock, K., and O'Connell, A. (2012b), "A spatial mark-resight model augmented with  
16307 telemetry data," *Ecology*.
- 16308 Sólymos, P., Lele, S., and Bayne, E. (2012), "Conditional likelihood approach for analyzing  
16309 single visit abundance survey data in the presence of zero inflation and detection error,"  
16310 *Environmetrics*.
- 16311 Spear, S. F., Balkenhol, N., Fortin, M. J., Mcrae, B. H., and Scribner, K. (2010), "Use  
16312 of resistance surfaces for landscape genetic studies: considerations for parameterization  
16313 and analysis," *Molecular Ecology*, 19, 3576–3591.
- 16314 Spiegelhalter, D. J., Best, N. G., Carlin, B. P., and Van Der Linde, A. (2002), "Bayesian  
16315 measures of model complexity and fit," *Journal of the Royal Statistical Society. Series  
16316 B, Statistical Methodology*, 583–639.
- 16317 Stabler, B. (2006), *shapefiles: Read and Write ESRI Shapefiles*, r package version 0.6.
- 16318 Stevens Jr, D. and Olsen, A. (2004), "Spatially balanced sampling of natural resources,"  
16319 *Journal of the American Statistical Association*, 99, 262–278.
- 16320 Stoyan, D. and Penttinen, A. (2000), "Recent Applications of Point Process Methods in  
16321 Forestry Statistics," *Statistical Science*, 15, 61–78.
- 16322 Sturtz, S., Ligges, U., and Gelman, A. (2005), "R2WinBUGS: A Package for Running  
16323 WinBUGS from R," *Journal of Statistical Software*, 12, 1–16.
- 16324 Su, Y.-S. and Yajima, M. (2011), *R2jags: A Package for Running jags from R*, r package  
16325 version 0.02-17.
- 16326 Sun, C. C. (in prep), "Population estimation, genetic diversity, and structure of black  
16327 bears in southwestern New York," Master's thesis, Cornell University.
- 16328 Taberlet, P. and Bouvet, J. (1992), "Bear conservation genetics," *Nature*, 358, 197–197.
- 16329 Tanner, M. A. and Wong, W. H. (1987), "The calculation of posterior distributions by  
16330 data augmentation," *J Am Stat Assoc*, 82, 528–540.
- 16331 Thomas, A., O'Hara, B., Ligges, U., and Sturtz, S. (2006), "Making BUGS Open," *R  
16332 News*, 6, 12–17.
- 16333 Thomas, M. (1949), "A generalization of Poisson's binomial limit for use in ecology,"  
16334 *Biometrika*, 36, 18–25.
- 16335 Thompson, C., Royle, J., and Garner, J. (2012), "A framework for inference about car-  
16336 nivore density from unstructured spatial sampling of scat using detector dogs," *The  
16337 Journal of Wildlife Management*, 76, 863–871.
- 16338 Thompson, S. K. (2002), *Sampling*, New York: Wiley.
- 16339 Tierney, L., Rossini, A. J., Li, N., and Sevcikova, H. (2011), *snow: Simple Network of  
16340 Workstations*, r package version 0.3-7.

- 16341 Tilman, D. and Kareiva, P. (1997), *Spatial ecology: the role of space in population dynamics*  
16342 and interspecific interactions, vol. 30, Princeton University Press.
- 16343 Tischendorf, L. and Fahrig, L. (2000), "On the usage and measurement of landscape  
16344 connectivity," *Oikos*, 90, 7–19.
- 16345 Tischendorf, L., Grez, A., Zaviezo, T., Fahrig, L., and ALLIANCE, R. (2005), "Mechanisms  
16346 affecting population density in fragmented habitat," *Ecology and Society*.
- 16347 Tobler, M., Carrillo-Percastegui, S., Leite Pitman, R., Mares, R., and Powell, G. (2008),  
16348 "An evaluation of camera traps for inventorying large-and medium-sized terrestrial  
16349 rainforest mammals," *Animal Conservation*, 11, 169–178.
- 16350 Tobler, M. W., Hibert, F., Debeir, L., and Hansen, C. (2012), "Density and sustainable  
16351 harvest estimates for the lowland tapir in the Amazon of French Guiana using a spatial  
16352 capture-recapture model," *none yet*, none, none.
- 16353 Tracy, J. A. (2006), "Individual-based movement modeling as a tool for conserving con-  
16354 nectivity," *Connectivity conservation*.
- 16355 Trolle, M. and Kéry, M. (2003), "Estimation of ocelot density in the Pantanal using  
16356 capture-recapture analysis of camera-trapping data," *Journal of Mammalogy*, 84, 607–  
16357 614.
- 16358 — (2005), "Camera-trap study of ocelot and other secretive mammals in the northern  
16359 Pantanal," *Mammalia*, 69, 409–416.
- 16360 Tufto, J., Andersen, R., and Linnell, J. (1996), "Habitat use and ecological correlates of  
16361 home range size in a small cervid: the roe deer," *Journal of Animal Ecology*, 715–724.
- 16362 Tyre, A. J., Tenhumberg, B., Field, S. A., Niejalke, D., Parris, K., and Possingham,  
16363 H. P. (2003), "Improving precision and reducing bias in biological surveys: estimating  
16364 false-negative error rates," *Ecological Applications*, 13, 1790–1801.
- 16365 Valiere, N. and Taberlet, P. (2000), "Urine collected in the field as a source of DNA for  
16366 species and individual identification," *Molecular Ecology*, 9, 2150–2152.
- 16367 van Etten, J. (2011), *Package gdistance*, r package version 1.1-2.
- 16368 van Etten, R. J. H. . J. (2012), *raster: Geographic analysis and modeling with raster data*,  
16369 r package version 1.9-67.
- 16370 Venables, W. and Ripley, B. (2002), *Modern applied statistics with S*, Springer verlag.
- 16371 Venables, W., Smith, D., and Team, R. D. C. (2012), "An introduction to R," .
- 16372 Ver Hoef, J. and Boveng, P. (2007), "Quasi-poisson vs. negative binomial regression: How  
16373 should we model overdispersed count data?" *Ecology*, 88, 2766–2772.
- 16374 Ver Hoef, J. M. (2012), "Who Invented the Delta Method?" *The American Statistician*,  
16375 66, 124–127.
- 16376 Wallace, R. B., Gomez, H., Ayala, G., and Espinoza, F. (2003), "Camera trapping for  
16377 jaguar (*Panthera onca*) in the Tuichi Valley, Bolivia," *Journal of Neotropical Mammal-*  
16378 *ogy*, 10, 133–139.
- 16379 Warton, D. and Shepherd, L. (2010), "Poisson point process models solve the pseudo-  
16380 absence problem for presence-only data in ecology," *The Annals of Applied Statistics*,  
16381 4, 1383–1402.
- 16382 Wegan, M., Curtis, P., Rainbolt, R., and Gardner, B. (2012), "Temporal sampling frame  
16383 selection in DNA-based capture mark-recapture investigations." *Ursus*, 23, 42 – 51.
- 16384 Wegan, M. T. (2008), "Aversive conditioning, population estimation, and habitat prefer-  
16385 ence of black bears (*Ursus americanus*) on Fort Drum Military Installation in northern  
16386 New York," Ph.D. thesis, Cornell University, Jan.

- 16387 Wegge, P., Pokheral, C. P., and Jnawali, S. R. (2004), "Effects of trapping effort and trap  
16388 shyness on estimates of tiger abundance from camera trap studies," *Animal Conserva-*  
16389 *tion*, 7, 251–256.
- 16390 White, G. (1996), "NOREMARK: population estimation from mark-resighting surveys,"  
16391 *Wildlife Society Bulletin*, 24, 50–52.
- 16392 White, G. and Bennetts, R. (1996), "Analysis of frequency count data using the negative  
16393 binomial distribution," *Ecology*, 2549–2557.
- 16394 White, G. and Shenk, M. (2001), "Population estimation with radio-marked individuals." in  
16395 *Radio tracking adn animal populations*, eds. Millspaugh, J. and Marzluff, J., San Diego,  
16396 USA: Academic Press, pp. 329–350.
- 16397 White, G. C. and Garrot, R. (1990), *Analysis of wildlife radiolocation data*, New York,  
16398 USA: Academic Press.
- 16399 White, G. C. and Shenk, T. M. (2000), *Population estimation with radio-marked animals*,  
16400 San Diego, California: Academic Press.
- 16401 Whitman, J., Ballard, W., and Gardner, C. (1986), "Home range and habitat use by  
16402 wolverines in southcentral Alaska," *The Journal of wildlife management*, 460–463.
- 16403 Wickham and Hadley (2007), "Reshaping data with the reshape package," *Journal of*  
16404 *Statistical Software*, 21.
- 16405 Williams, B. K., Nichols, J. D., and Conroy, M. J. (2002), *Analysis and management of*  
16406 *animal populations: modeling, estimation, and decision making*, Academic Pr.
- 16407 Wilson, K. R. and Anderson, D. R. (1985a), "Evaluation of a Density Estimator Based  
16408 on a Trapping Web and Distance Sampling Theory," *Ecology*, 66, 1185–1194.
- 16409 — (1985b), "Evaluation of Two Density Estimators of Small Mammal Population Size,"  
16410 *Journal of Mammalogy*, 66, 13–21.
- 16411 With, K. and Crist, T. (1995), "Critical thresholds in species' responses to landscape  
16412 structure," *Ecology*, 2446–2459.
- 16413 Wolpert, R. L. and Ickstadt, K. (1998), "Poisson/gamma random field models for spatial  
16414 statistics," *Biometrika*, 85, 251 –267.
- 16415 Woods, J. G., Paetkau, D., Lewis, D., McLellan, B. N., Proctor, M., and Strobeck, C.  
16416 (1999), "Genetic tagging of free-ranging black and brown bears," *Wildlife Society Bul-*  
16417 *letin*, 27, 616–627.
- 16418 Worton, B. (1989), "Kernel methods for estimating the utilization distribution in home-  
16419 range studies," *Ecology*, 70, 164–168.
- 16420 Wright, J., Barker, R., Schofield, M., Frantz, A., Byrom, A., and Gleeson, D. (2009), "In-  
16421 corporating Genotype Uncertainty into Mark–Recapture-Type Models For Estimating  
16422 Abundance Using DNA Samples," *Biometrics*, 65, 833–840.
- 16423 Yang, H. C. and Chao, A. (2005), "Modeling Animals' Behavioral Response by Markov  
16424 Chain Models for Capture–Recapture Experiments," *Biometrics*, 61, 1010–1017.
- 16425 Yoshizaki, J., Pollock, K. H., Brownie, C., and Webster, R. A. (2009), "Modeling misiden-  
16426 tification errors in capture-recapture studies using photographic identification of evolv-  
16427 ing marks," *Ecology*, 90, 3–9.
- 16428 Zeller, K., McGarigal, K., and Whiteley, A. (2012), "Estimating landscape resistance to  
16429 movement: a review," *Landscape Ecology*, 1–21.
- 16430 Zuur, A. F., Ieno, E. N., Walker, N. J., Saveliev, A. A., and Smith, G. M. (2009), *Mixed*  
16431 *effects models and extensions in ecology with R*, Springer Verlag.
- 16432 Zylstra, E., Steidl, R., and Swann, D. (2010), "Evaluating survey methods for monitoring

16433      a rare vertebrate, the Sonoran desert tortoise," *The Journal of Wildlife Management*,  
16434      74, 1311–1318.

# INDEX

---

- 16435 full conditional distribution, 57
- 16436 Gibbs sampling, 57
- 16437 maptools, 145
- 16438 Monte Carlo error, 64
- 16439 R package
  - 16440 sp, 145
- 16441 R package
  - 16442 maptools, 145
- 16443 shapefile, 145