

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	Preface	13
13	Acknowledgments	14
14	I Background and Concepts	1
15	1 Introduction	3
16	1.1 The Study of Populations by Capture-Recapture . . . . .	4
17	1.2 Lions and Tigers and Bears, Oh My: Genesis of Spatial Capture-Recapture Data . . . . .	5
19	1.2.1 Camera trapping . . . . .	5
20	1.2.2 DNA sampling . . . . .	6
21	1.2.3 Acoustic sampling . . . . .	7
22	1.2.4 Search-encounter methods . . . . .	7
23	1.3 Capture-Recapture for Modeling Encounter Probability . . . . .	8
24	1.3.1 Example: Fort Drum bear study . . . . .	9
25	1.3.2 Inadequacy of non-spatial capture-recapture . . . . .	12
26	1.4 Historical Context: A Brief Synopsis . . . . .	13
27	1.4.1 Buffering . . . . .	13
28	1.4.2 Temporary emigration . . . . .	14
29	1.5 Extension of Closed Population Models . . . . .	15
30	1.5.1 Towards spatial explicitness: Efford's formulation . . . . .	15
31	1.5.2 Abundance as the aggregation of a point process . . . . .	16
32	1.5.3 The activity center concept . . . . .	17
33	1.5.4 The state-space . . . . .	17
34	1.5.5 Abundance and density . . . . .	18
35	1.6 Characterization of SCR models . . . . .	18
36	1.7 Summary and Outlook . . . . .	19
37	2 Statistical Models and SCR	21
38	2.1 Random Variables and Probability Distributions . . . . .	22
39	2.1.1 Stochasticity in ecology . . . . .	22
40	2.1.2 Properties of probability distributions . . . . .	26

41	2.2	Common Probability Distributions . . . . .	28
42	2.2.1	The binomial distribution . . . . .	28
43	2.2.2	The Bernoulli distribution . . . . .	29
44	2.2.3	The multinomial and categorical distributions . . . . .	30
45	2.2.4	The Poisson distribution . . . . .	32
46	2.2.5	The uniform distribution . . . . .	32
47	2.2.6	Other distributions . . . . .	33
48	2.3	Statistical Inference and Parameter Estimation . . . . .	35
49	2.4	Joint, Marginal, and Conditional Distributions . . . . .	38
50	2.5	Hierarchical Models and Inference . . . . .	41
51	2.6	Characterization of SCR Models . . . . .	43
52	2.7	Summary and Outlook . . . . .	47
53	<b>3</b>	<b>GLMs and Bayesian Analysis</b>	<b>49</b>
54	3.1	GLMs and GLMMs . . . . .	50
55	3.2	Bayesian Analysis . . . . .	52
56	3.2.1	Bayes' rule . . . . .	52
57	3.2.2	Principles of Bayesian inference . . . . .	53
58	3.2.3	Prior distributions . . . . .	55
59	3.2.4	Posterior inference . . . . .	56
60	3.2.5	Small sample inference . . . . .	56
61	3.3	Characterizing Posterior Distributions by MCMC Simulation . . . . .	58
62	3.3.1	What goes on under the MCMC hood . . . . .	58
63	3.3.2	Rules for constructing full conditional distributions . . . . .	60
64	3.3.3	Metropolis-Hastings algorithm . . . . .	60
65	3.4	Bayesian Analysis Using the BUGS Language . . . . .	61
66	3.4.1	Linear regression in WinBUGS . . . . .	62
67	3.5	Practical Bayesian Analysis and MCMC . . . . .	64
68	3.5.1	Choice of prior distributions . . . . .	64
69	3.5.2	Convergence and so-forth . . . . .	66
70	3.5.3	Bayesian confidence intervals . . . . .	69
71	3.5.4	Estimating functions of parameters . . . . .	69
72	3.6	Poisson GLMs . . . . .	69
73	3.6.1	Example: Breeding Bird Survey data . . . . .	70
74	3.6.2	Doing it in WinBUGS . . . . .	72
75	3.6.3	Constructing your own MCMC algorithm . . . . .	73
76	3.7	Poisson GLM with Random Effects . . . . .	76
77	3.8	Binomial GLMs . . . . .	78
78	3.8.1	Binomial regression . . . . .	79
79	3.8.2	Example: waterfowl banding data . . . . .	80
80	3.9	Bayesian Model Checking and Selection . . . . .	81
81	3.9.1	Goodness-of-fit . . . . .	82
82	3.9.2	Model selection . . . . .	84

---

83	3.10 Summary and Outlook . . . . .	85
84	<b>4 Closed Population Models</b>	<b>87</b>
85	4.1 The Simplest Closed Population Model: Model $M_0$ . . . . .	88
86	4.1.1 The core capture-recapture assumptions . . . . .	90
87	4.1.2 Conditional likelihood . . . . .	91
88	4.2 Data Augmentation . . . . .	92
89	4.2.1 DA links occupancy models and closed population models . .	92
90	4.2.2 Model $M_0$ in BUGS . . . . .	94
91	4.2.3 Formal development of data augmentation (DA) . . . . .	95
92	4.2.4 Remarks on data augmentation . . . . .	96
93	4.2.5 Example: Black bear study on Fort Drum . . . . .	98
94	4.3 Temporally Varying and Behavioral Effects . . . . .	101
95	4.4 Models with Individual Heterogeneity . . . . .	102
96	4.4.1 Analysis of model $M_h$ . . . . .	104
97	4.4.2 Analysis of the Fort Drum data with model $M_h$ . . . . .	105
98	4.4.3 Comparison with MLE . . . . .	106
99	4.5 Individual Covariate Models: Toward Spatial Capture-Recapture .	108
100	4.5.1 Example: Location of capture as a covariate . . . . .	110
101	4.5.2 Fort Drum bear study . . . . .	111
102	4.5.3 Extension of the model . . . . .	113
103	4.5.4 Invariance of density to $B$ . . . . .	116
104	4.5.5 Toward fully spatial capture-recapture models . . . . .	116
105	4.6 Distance Sampling: A Primitive SCR Model . . . . .	116
106	4.6.1 Example: Sonoran desert tortoise study . . . . .	119
107	4.7 Summary and Outlook . . . . .	121
108	<b>II Basic SCR Models</b>	<b>123</b>
109	<b>5 Fully Spatial Capture-Recapture Models</b>	<b>125</b>
110	5.1 Sampling Design and Data Structure . . . . .	126
111	5.2 The binomial observation model . . . . .	127
112	5.2.1 Definition of home range center . . . . .	129
113	5.2.2 Distance as a latent variable . . . . .	129
114	5.3 The Binomial Point Process Model . . . . .	129
115	5.3.1 The state-space of the point process . . . . .	131
116	5.3.2 Connection to model $M_h$ and distance sampling . . . . .	133
117	5.4 The Implied Model of Space Usage . . . . .	134
118	5.4.1 Bivariate normal case . . . . .	136
119	5.4.2 Empirical analysis . . . . .	136
120	5.4.3 Relevance of understanding space usage . . . . .	138
121	5.4.4 Contamination due to behavioral response . . . . .	138

122	5.5	Simulating SCR Data . . . . .	139
123	5.5.1	Formatting and manipulating real data sets . . . . .	140
124	5.6	Fitting Model SCR0 in BUGS . . . . .	141
125	5.7	Unknown N . . . . .	144
126	5.7.1	Analysis using data augmentation in WinBUGS . . . . .	145
127	5.7.2	Implied home range area . . . . .	148
128	5.7.3	Realized and expected density . . . . .	148
129	5.8	The Core SCR Assumptions . . . . .	150
130	5.9	Wolverine Camera Trapping Study . . . . .	151
131	5.9.1	Practical data organization . . . . .	151
132	5.9.2	Fitting the model in WinBUGS . . . . .	155
133	5.9.3	Summary of the wolverine analysis . . . . .	156
134	5.9.4	Wolverine space usage . . . . .	157
135	5.10	Using a Discrete Habitat Mask . . . . .	158
136	5.10.1	Evaluation of coarseness of habitat mask . . . . .	159
137	5.10.2	Analysis of the wolverine camera trapping data . . . . .	160
138	5.11	Summarizing Density and Activity Center Locations . . . . .	160
139	5.11.1	Constructing density maps . . . . .	161
140	5.11.2	Example: Wolverine density map . . . . .	164
141	5.11.3	Predicting where an individual lives . . . . .	165
142	5.12	Effective Sample Area . . . . .	165
143	5.13	Summary and Outlook . . . . .	168
144	<b>6</b>	<b>Likelihood Analysis of Spatial Capture-Recapture Models</b>	<b>173</b>
145	6.1	MLE with Known N . . . . .	174
146	6.1.1	Implementation (simulated data) . . . . .	175
147	6.2	MLE when N is Unknown . . . . .	179
148	6.2.1	Integrated likelihood under data augmentation . . . . .	182
149	6.2.2	Extensions . . . . .	183
150	6.3	Classical Model Selection and Assessment . . . . .	183
151	6.4	Likelihood Analysis of the Wolverine Camera Trapping Data . . . . .	184
152	6.4.1	Sensitivity to integration grid and state-space buffer . . . . .	185
153	6.4.2	Using a habitat mask (Restricted state-space) . . . . .	186
154	6.5	DENSITY and the R Package <b>secr</b> . . . . .	188
155	6.5.1	Encounter device types and detection models . . . . .	190
156	6.5.2	Analysis using the <b>secr</b> package . . . . .	191
157	6.5.3	Likelihood analysis in the <b>secr</b> package . . . . .	194
158	6.5.4	Multi-session models in <b>secr</b> . . . . .	195
159	6.5.5	Some additional capabilities of <b>secr</b> . . . . .	196
160	6.6	Summary and Outlook . . . . .	198

---

161	<b>7 Modeling Encounter Probability</b>	<b>201</b>
162	7.1 Encounter Probability Models . . . . .	202
163	7.1.1 Bayesian analysis with <code>bear.JAGS</code> . . . . .	204
164	7.1.2 Bayesian analysis of encounter probability models . . . . .	204
165	7.2 Modeling Covariate Effects . . . . .	206
166	7.2.1 Date and time . . . . .	208
167	7.2.2 Trap-specific covariates . . . . .	210
168	7.2.3 Behavior or trap response by individual . . . . .	210
169	7.2.4 Individual covariates . . . . .	212
170	7.3 Individual Heterogeneity . . . . .	215
171	7.3.1 Models of heterogeneity . . . . .	215
172	7.3.2 Heterogeneity induced by variation in home range size . . . . .	217
173	7.4 Likelihood Analysis in <code>secr</code> . . . . .	217
174	7.4.1 Notes for fitting standard models . . . . .	218
175	7.4.2 Sex effects . . . . .	219
176	7.4.3 Individual heterogeneity . . . . .	220
177	7.4.4 Model selection in <code>secr</code> using AIC . . . . .	220
178	7.5 Summary and Outlook . . . . .	221
179	<b>8 Model Selection and Assessment</b>	<b>223</b>
180	8.1 Model Selection by AIC . . . . .	224
181	8.1.1 AIC analysis of the wolverine data . . . . .	224
182	8.2 Bayesian Model Selection . . . . .	228
183	8.2.1 Model selection by DIC . . . . .	228
184	8.2.2 DIC analysis of the wolverine data . . . . .	229
185	8.2.3 Bayesian model averaging with indicator variables . . . . .	230
186	8.2.4 Choosing among detection functions . . . . .	234
187	8.3 Evaluating Goodness-of-Fit . . . . .	235
188	8.4 The Two Components of Model Fit . . . . .	237
189	8.4.1 Testing uniformity or spatial randomness . . . . .	237
190	8.4.2 Assessing fit of the observation model . . . . .	240
191	8.4.3 Does the SCR model fit the wolverine data? . . . . .	241
192	8.5 Quantifying Lack-of-fit and Remediation . . . . .	244
193	8.6 Summary and Outlook . . . . .	245
194	<b>9 Alternative Observation Models</b>	<b>247</b>
195	9.1 Poisson Observation Model . . . . .	248
196	9.1.1 Poisson model of space usage . . . . .	249
197	9.1.2 Poisson relationship to the Bernoulli model . . . . .	250
198	9.1.3 A cautionary note on modeling encounter frequencies . . . . .	251
199	9.1.4 Analysis of the Poisson SCR model in BUGS . . . . .	252
200	9.1.5 Simulating data and fitting the model . . . . .	253
201	9.1.6 Analysis of the wolverine study data . . . . .	255

202	9.1.7	Count detector models in the <b>secr</b> package . . . . .	256
203	9.2	Independent Multinomial Observations . . . . .	256
204	9.2.1	Multinomial resource selection models . . . . .	258
205	9.2.2	Simulating data and analysis using JAGS . . . . .	258
206	9.2.3	Multinomial relationship to the Poisson . . . . .	262
207	9.2.4	Avian mist-netting example . . . . .	262
208	9.3	Single-catch traps . . . . .	269
209	9.3.1	Inference for single-catch systems . . . . .	270
210	9.3.2	Analysis of Efford's possum trapping data . . . . .	270
211	9.4	Acoustic sampling . . . . .	273
212	9.4.1	The signal strength model . . . . .	274
213	9.4.2	Implementation in <b>secr</b> . . . . .	276
214	9.4.3	Implementation in <b>BUGS</b> . . . . .	276
215	9.4.4	Other types of acoustic data . . . . .	277
216	9.5	Summary and Outlook . . . . .	277
217	<b>10</b>	<b>Sampling Design</b>	<b>279</b>
218	10.1	General Considerations . . . . .	279
219	10.1.1	Model-based not design-based . . . . .	279
220	10.1.2	Sampling space or sampling individuals? . . . . .	279
221	10.1.3	Scope of inference vs. state-space . . . . .	279
222	10.2	Study design for (spatial) capture-recapture . . . . .	279
223	10.3	Trap spacing and array size relative to animal movement . . . . .	279
224	10.3.1	Example: Black bears from Pictured Rocks National Lakeshore: .	279
225	10.3.2	Final musings: SCR models, trap spacing and array size . . .	280
226	10.4	Spacing of traps with telemetered individuals . . . . .	280
227	10.5	Sampling over large scales . . . . .	280
228	10.6	Model-based Spatial Design . . . . .	280
229	10.6.1	Formalization of the Design Problem for SCR Studies . . .	280
230	10.6.2	An Optimal Design Criterion for SCR . . . . .	280
231	10.6.3	Optimization of the criterion . . . . .	280
232	10.6.4	Illustration . . . . .	280
233	10.7	Covariate models . . . . .	280
234	10.8	Summary and Outlook . . . . .	280
235	<b>III</b>	<b>Advanced SCR Models</b>	<b>281</b>
236	<b>11</b>	<b>Modeling Spatial Variation in Density</b>	<b>283</b>
237	11.1	Homogeneous point process revisited . . . . .	283
238	11.2	Inhomogeneous point processes . . . . .	283
239	11.3	Observed Point Processes . . . . .	284

---

241	11.4 Fitting inhomogeneous point process SCR models . . . . .	284
242	11.4.1 Continuous space . . . . .	284
243	11.4.2 Discrete space . . . . .	284
244	11.5 Ecological Distance and Density Covariates . . . . .	284
245	11.6 The Jaguar Data . . . . .	284
246	11.7 Summary and Outlook . . . . .	284
247	<b>12 Modeling Landscape Connectivity</b>	<b>287</b>
248	12.1 Shortcomings of Euclidean Distance Models . . . . .	287
249	12.2 Least-Cost Path Distance . . . . .	287
250	12.2.1 Example of Computing Cost-weighted distance . . . . .	287
251	12.3 Simulating SCR Data using Ecological Distance . . . . .	287
252	12.4 Likelihood Analysis of Ecological Distance Models . . . . .	287
253	12.4.1 Example of SCR with Least-Cost Path . . . . .	288
254	12.5 Bayesian Analysis . . . . .	288
255	12.6 Simulation Evaluation of the MLE . . . . .	288
256	12.6.1 Simulation Results . . . . .	288
257	12.7 Distance In an Irregular Patch . . . . .	288
258	12.7.1 Basic Geographic Analysis in R . . . . .	288
259	12.8 Summary and Outlook . . . . .	288
260	<b>13 Integrating Resource Selection with Spatial Capture-Recapture Models</b>	<b>291</b>
261	13.1 A Simple Model of Space Usage . . . . .	291
262	13.1.1 Poisson use model . . . . .	291
263	13.1.2 Thinning . . . . .	291
264	13.1.3 Capture-recapture Data . . . . .	291
265	13.2 The Joint RSF/SCR Likelihood . . . . .	291
266	13.3 Application: New York Black Bear Study . . . . .	291
267	13.4 Simulation Study . . . . .	291
268	13.5 Summary and Outlook . . . . .	291
270	<b>14 Stratified Populations: Multi-session and Multi-site Data</b>	<b>293</b>
271	14.1 Data Structure . . . . .	293
272	14.2 Multinomial Abundance Models . . . . .	293
273	14.2.1 Observation Models . . . . .	293
274	14.2.2 Simulating group structured capture-recapture data . . . . .	293
275	14.2.3 Fitting in BUGS . . . . .	293
276	14.2.4 Approach B modeling $\psi$ . . . . .	293
277	14.3 Spatial Capture-Recapture . . . . .	293
278	14.4 Application . . . . .	294
279	14.4.1 Results . . . . .	294
280	14.5 Topics in Multi-Session models . . . . .	296

281	14.5.1 Temporal models . . . . .	296
282	14.5.2 Dependence – is it a problem? . . . . .	296
283	14.6 Multi-session models in <b>secr</b> . . . . .	296
284	14.6.1 Ovenbird data in WinBUGS? . . . . .	296
285	14.6.2 Converse data in <b>secr</b> ? . . . . .	296
286	14.7 Summary and Outlook . . . . .	296
287	<b>15 Models for Search-Encounter Data</b>	<b>297</b>
288	15.1 Search-Encounter sampling designs . . . . .	297
289	15.2 A Model for Search-Encounter Data . . . . .	297
290	15.2.1 Ecological process model . . . . .	297
291	15.2.2 Other stuff . . . . .	297
292	15.3 Examples . . . . .	297
293	15.3.1 Hard plot boundaries . . . . .	297
294	15.3.2 Analysis of other protocols . . . . .	297
295	15.4 Design 3: Ad hoc implementation of Design 1. . . . .	297
296	15.5 Capricaillie crap . . . . .	297
297	15.5.1 model . . . . .	297
298	15.6 Design 4 – no location info . . . . .	297
299	15.7 Summary and Outlook . . . . .	297
300	<b>16 Open Population Models</b>	<b>299</b>
301	16.1 Introduction . . . . .	299
302	16.1.1 Overview of Population Dynamics . . . . .	299
303	16.1.2 Animal movement related to population demography . . . . .	299
304	16.1.3 Basic assumptions of JS and CJS models . . . . .	299
305	16.2 Traditional Jolly-Seber Models . . . . .	299
306	16.2.1 Data Augmentation for the Jolly-Seber Model . . . . .	299
307	16.2.2 Mist-netting example . . . . .	299
308	16.2.3 Shortcomings of the traditional JS models . . . . .	299
309	16.3 Spatial Jolly-Seber Models . . . . .	299
310	16.3.1 Mist-netting example . . . . .	300
311	16.4 Traditional CJS models . . . . .	300
312	16.4.1 Migratory fish example . . . . .	300
313	16.5 Multi-state CJS models . . . . .	300
314	16.5.1 Migratory fish example . . . . .	300
315	16.6 Spatial CJS models . . . . .	300
316	16.6.1 Migratory fish example . . . . .	300
317	16.7 Moving Activity Centers . . . . .	300
318	16.7.1 Migratory Fish Example Notes . . . . .	300
319	16.8 Summary and Outlook . . . . .	300

---

320	<b>IV Super-Advanced SCR Models</b>	<b>303</b>
321	<b>17 Developing Markov Chain Monte Carlo Samplers</b>	<b>305</b>
322	17.0.1 Why build your own MCMC algorithm? . . . . .	305
323	17.1 MCMC and posterior distributions . . . . .	305
324	17.2 Types of MCMC sampling . . . . .	305
325	17.2.1 Gibbs sampling . . . . .	305
326	17.2.2 Metropolis-Hastings sampling . . . . .	306
327	17.2.3 Metropolis-within-Gibbs . . . . .	306
328	17.2.4 Rejection sampling and slice sampling . . . . .	306
329	17.3 MCMC for closed capture-recapture Model Mh . . . . .	306
330	17.4 MCMC algorithm for model SCR0 . . . . .	306
331	17.4.1 SCR model with binomial encounter process . . . . .	306
332	17.4.2 Looking at model output . . . . .	306
333	17.4.3 Posterior density plots . . . . .	306
334	17.4.4 Serial autocorrelation and effective sample size . . . . .	306
335	17.4.5 Summary results . . . . .	306
336	17.4.6 Other useful commands . . . . .	306
337	17.5 Manipulating the state-space . . . . .	306
338	17.6 Increasing computational speed . . . . .	306
339	17.6.1 Parallel computing . . . . .	306
340	17.6.2 Using C++ . . . . .	306
341	17.7 Summary and Outlook . . . . .	306
342	<b>18 Spatial Capture-Recapture for Unmarked Populations</b>	<b>309</b>
343	18.1 Existing Models for Inference About Density in Unmarked Populations	309
344	18.2 Spatial Correlation as Information . . . . .	309
345	18.3 Data . . . . .	309
346	18.4 Model . . . . .	309
347	18.5 Northern Parula Example . . . . .	309
348	18.6 Improving Precision with Prior Information . . . . .	309
349	18.7 Design issues . . . . .	309
350	18.7.1 How Much Correlation Is Enough? . . . . .	309
351	18.7.2 Linear Designs . . . . .	309
352	18.7.3 Quadrat counts . . . . .	309
353	18.8 Alternative Observation Models . . . . .	309
354	18.8.1 Spatial point process models . . . . .	309
355	18.9 Conclusion . . . . .	309
356	<b>19 Spatial Mark-Resight Models for partially identifiable populations</b>	<b>311</b>
357	19.1 Background . . . . .	312
358	19.1.1 Types of partial ID data . . . . .	312

360	19.1.2 A short history of mark-resight models . . . . .	314
361	19.2 Known number of marked individuals . . . . .	315
362	19.2.1 MCMC for a spatial mark-resight model . . . . .	316
363	19.2.2 Binomial encounter model . . . . .	318
364	19.3 Unknown number of marked individuals . . . . .	319
365	19.4 Imperfect identification of marked individuals . . . . .	322
366	19.5 How much information do marked and unmarked individuals con-	
367	tribute? . . . . .	324
368	19.6 Incorporating telemetry data . . . . .	327
369	19.7 Summary and Outlook . . . . .	330
370	<b>20 2012: A Spatial Capture-Recapture Odyssey</b>	<b>333</b>
371	20.1 10 thesis or dissertation topics . . . . .	334
372	20.2 Three dimesional space . . . . .	334
373	20.3 Gregarious species . . . . .	334
374	<b>V Appendices</b>	<b>335</b>
375	20.4 WinBUGS . . . . .	337
376	20.4.1 WinBUGS through R . . . . .	337
377	20.5 OpenBUGS . . . . .	338
378	20.5.1 OpenBUGS through R . . . . .	338
379	20.6 JAGS . . . . .	338
380	20.6.1 JAGS through R . . . . .	339
381	20.7 R . . . . .	340
382	20.7.1 R packages . . . . .	340
383	<b>Bibliography</b>	<b>343</b>

## Preface

384

385



388

## Part I

389

---

390

# Background and Concepts



391  
392

# 1

---

393

## INTRODUCTION

394 Space plays a vital role in virtually all ecological processes (Tilman and Kareiva,  
395 1997; Hanski, 1999; Clobert et al., 2001). The spatial arrangement of habitat can  
396 influence movement patterns during dispersal, habitat selection, and survival. The  
397 distance between an organism and its competitors and prey can influence activity  
398 patterns and foraging behavior. Further, understanding distribution and spatial  
399 variation in abundance is necessary in the conservation and management of popu-  
400 lations. The inherent spatial aspect of *sampling* populations also plays an important  
401 role in ecology as it strongly affects, and biases, how we observe population struc-  
402 ture (Seber, 1982; Buckland et al., 2001; Borchers et al., 2002; Williams et al.,  
403 2002). However, despite the central role of space and spatial processes to both  
404 understanding population dynamics and how we observe or sample populations, a  
405 coherent framework that integrates these two aspects of ecological systems has not  
406 been fully realized either conceptually or methodologically.

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

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

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

## 1.1 THE STUDY OF POPULATIONS BY CAPTURE-RECAPTURE

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

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

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

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

---

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

461 As the name implies, the primary feature of SCR models that distinguishes  
462 them from traditional CR methods is that they make use of the spatial information  
463 inherent to capture-recapture studies. Encounter histories that are associated with  
464 auxiliary information on the location of capture, are *spatial encounter histories*.  
465 This auxiliary information is informative about spatial processes including the spa-  
466 tial organization of individuals, variation in density, resource selection and space  
467 usage, and movement. As we will see, SCR models allow us to overcome critical  
468 deficiencies of non-spatial methods, and integrate ecological theory with encounter  
469 history data. As a result, this greatly expands the practical utility and scientific  
470 relevance of capture-recapture methods, and studies that produce encounter history  
471 data.

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

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

#### **486 1.2.1 Camera trapping**

487 Considerable recent work has gone into the development of camera-trapping method-  
488 ologies. For a historical overview of this method see Kays et al. (2008) and Kucera  
489 and Barrett (2011). Several recent synthetic works have been published includ-  
490 ing Nichols and Karanth (2002), and an edited volume by O’Connell et al. (2010)  
491 devoted solely to camera trapping concepts and methods. As a method for estimat-  
492 ing abundance, some of the earliest work that relates to the use of camera trapping  
493 data in capture-recapture models originates from Karanth and colleagues (Karanth,  
494 1995; Karanth and Nichols, 1998, 2000).

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

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

508 **1.2.2 DNA sampling**

509 DNA obtained from hair, blood or scat is now routinely used to obtain individual  
510 identity and encounter history information about individuals (Taberlet and Bouvet,  
511 1992; Kohn et al., 1999; Woods et al., 1999; Mills et al., 2000; Schwartz and Monfort,  
512 2008). A common method is based on the use of “hair snares” (Fig. 1.2) which are  
513 widely used to study bear populations (Woods et al., 1999; Garshelis and Hristienko,  
514 2006; Kendall et al., 2009; Gardner et al., 2010b). A sample of hair is obtained as  
515 individuals pass under or around barbed-wire (or other physical mechanism) to take  
516 bait. Hair snares and scent sticks have also been used to sample felid populations  
517 (García-Alaníz et al., 2010; Kéry et al., 2010) and other species. Research has  
518 even shown that DNA information can be extracted from urine deposited in the  
519 wild (e.g., in snow; see Valiere and Taberlet (2000)) and as a result this may prove  
520 another future data collection technique where SCR models are useful.

521 **1.2.3 Acoustic sampling**

522 Many studies of birds (Dawson and Efford, 2009), bats, and whales (Marques et al.,  
523 2009) now collect data using devices that record vocalizations. When vocalizations  
524 can be identified by individual from multiple recording devices, spatial encounter  
525 histories are produced that are amenable to the application of SCR models (Dawson  
526 and Efford, 2009; Efford et al., 2009b). Recently, these ideas have been applied to  
527 data on direction or distance to vocalizations by multiple simultaneous observers  
528 and related problems (D. Borchers, ISEC 2012 presentation).

529 **1.2.4 Search-encounter methods**

530 There are other methods which don’t fall into a nice clean taxonomy of “devices”.  
531 Spatial encounter histories are commonly obtained by conducting manual searches

of geographic sample units such as quadrats, transects or road or trail networks. For example, DNA-based encounter histories can be obtained from scat samples located along roads or trails or by specially trained dogs (MacKay et al., 2008) searching space (Fig. 1.3). This method has been used in studies of martens, fishers (Thompson et al., 2012), lynx, coyotes, birds (Kéry et al., 2010), and many other species. A similar data structure arises from the use of standard territory or spot mapping of birds Bibby et al. (1992) or area sampling in which space is searched by observers to physically capture individuals. This is common in surveys that involve reptiles and amphibians, e.g., we might walk transects picking up box turtles (Hall et al., 1999), or desert tortoises (Zylstra et al., 2010), or search space for lizards (Royle and Young, 2008).

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

### 1.3 CAPTURE-RECAPTURE FOR MODELING ENCOUNTER PROBABILITY

We briefly introduced techniques used for the study of animal populations. These methods produce individual encounter history data, a record of where and when each individual was captured. We refer to this as a *spatial encounter history*. Historically, auxiliary spatial information has been ignored, and encounter history data have been *summarized* to simple “encounter or not” for the purpose of applying ordinary CR models. The basic problem with these ordinary (or “non-spatial”) capture-recapture models is they don’t have any sense of space in them, the spatial information is summarized out of the data set, so we aren’t able to use such models for studying things such as movement, or resource selection, etcdots. Instead, ordinary capture-recapture models usually resort to models of “encounter probability,” which is a nuisance parameter, seldom of any ecological relevance. We show an example here that is in keeping with the classical application of ordinary capture-recapture models.

#### 1.3.1 Example: Fort Drum bear study

Here we confront the simplest possible capture-recapture problem – but one of great applied interest – estimating density from a standard capture-recapture study. We use this as a way to introduce some concepts and motivate the need for spatial capture-recapture models by confronting technical and conceptual problems that we encounter. The data come from a study to estimate black bear abundance on the Fort Drum Military Installation in upstate New York ( Wegan (2008), see also Chapt. 4 for more details). The specific data used here are encounter histories on 47 individuals obtained from an array of 38 baited “hair snares” during June and July 2006. The study area and locations of the 38 hair snares are shown in Fig.

569 1.4. Barbed wire traps (see Fig. 1.2) were baited and checked for hair samples  
570 each week for eight weeks. Analysis of these data appears in Gardner et al. (2009)  
571 and Gardner et al. (2010b), and we use the data in a number of analyses in later  
572 chapters.

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

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

595 The mean female home range radius was estimated (Wegan, 2008) for this study  
596 region to be 2.19 km, and the area of the convex hull buffered by 2.19 km is  
597  $277.01 \text{ km}^2$ . (**R** commands to compute the convex hull, buffer it, and compute the  
598 area are given in the **R** package **scrbook** which accompanies the book). Hence,  
599 the estimated density here is approximately  $0.178 \text{ bears/km}^2$  using the estimated  
600 population size obtained by model  $M_0$ . We could assert that the problem has been  
601 solved, go home, and have a beer. But then, on the other hand, maybe we should  
602 question the use of the estimated home range radius – after all, this is only the  
603 female home range radius and the home ranges change for many reasons. Instead,  
604 we may decide to rely on a buffer width based on one-half mean maximum distance  
605 moved (MMDM) estimated from the actual hair snare data as is more customary  
606 (Dice, 1938). In that case the buffer width is 1.19 km, and the resulting estimated  
607 density is increased to  $0.225 \text{ bears/km}^2$  about 27 % larger. But wait – some studies  
608 actually found the full MMDM (Parmenter et al., 2003) to be a more appropriate  
609 measure of movement (e.g. Soisalo and Cavalcanti (2006)). So maybe we should use  
610 the full MMDM which is 2.37 km, pretty close to the telemetry-based estimate and  
611 therefore providing a similar estimate of density ( $0.171 \text{ bears/km}^2$ ). So in trying to

612 decide how to buffer our trap array we have already generated 3 density estimates.  
 613 The crux of the matter is obvious: Although it is intuitive that  $N$  should scale with  
 614 area – the number of bears should go up as area increases and go down as area  
 615 decreases – in this ad hoc approach of accounting for animal movement  $N$  remains  
 616 the same, no matter what area we assert was sampled. The number of bears and the  
 617 area they live in are not formally tied together within the model, because estimating  
 618  $N$  and estimating the area  $N$  relates to are two completely independent analytical  
 619 steps which are unrelated to one another by a formal model.

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

```
628 # traps: 1 2 3 4 5 6 9
629 # bears: 23 13 6 2 1 1 1
```

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

650 So what are we left with? Our density estimates span a range from 0.17 to  
 651 0.43 bears/km<sup>2</sup> depending on which estimator of  $N$  we use and what buffer strip  
 652 we apply. Should we feel strongly about one or the other? Which buffer should

we prefer? AIC favors model  $M_h$ , but did it adequately account for the differences in exposure of individuals to the trap array? Are we happy with a purely phenomenological model for heterogeneity? It assumes that all individuals are independent and identically distributed (*iid*) draws from some distribution, but does not account for the explicit mechanism of induced heterogeneity. And, further, we have information about that (trap of capture) which model  $M_h$  ignores. And if we choose one type of buffer, how do we compare our density estimates to those from other studies that may opt for a different kind of buffer? The fact that  $N$  does not scale with  $A$ , as part of the model, renders this choice arbitrary.

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

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

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

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

The capture-recapture models that we used apply to truly closed populations – a population of goldfish in a fish bowl. Yet here we are applying them to a population of bears that inhabit a rich two-dimensional landscape of varied habitats, exposed to trapping by an irregular and sparse array of traps. It seems questionable that the same model that is completely sensible for a population of goldfish in a bowl, should also be the right model for this population of bears distributed over a broad landscape. Ordinary capture-recapture methods are distinctly non-spatial. They don’t admit spatial indexing of either sampling (the observation process) or of individuals (the ecological process). This leads immediately to a number of practical deficiencies: (1) Ordinary CR models do not provide a coherent basis

for estimating density, a problem we struggled with in the black bear study. (2) Ordinary CR model and sampling methods *induce* a form of heterogeneity that can only at best be approximated by classical models of latent heterogeneity. SCR models formally accommodate heterogeneity due to the juxtaposition of individuals with the encounter devices. (3) Ordinary CR models do not accommodate trap-level covariates which exist in a large proportion of real studies; (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.

## 1.4 HISTORICAL CONTEXT: A BRIEF SYNOPSIS

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

### 1.4.1 Buffering

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

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

722 (Holdenried, 1940; Hayne, 1949), nested grids (Otis et al., 1978), and assessment lines  
723 (Smith et al., 1971)."

724 The idea of using 1/2 mean maximum distance moved ("MMDM" Wilson and  
725 Anderson, 1985b) to create a buffer strip seems to be the standard approach even  
726 today, presumably justified by Dice's suggestion to use 1/2 the home range diam-  
727 eter, with the mean over individuals of the maximum distance moved being an  
728 estimator of home range diameter. Alternatively, some studies have used the full  
729 MMDM (e.g. Parmenter et al. (2003)), because the trap array might not provide  
730 a full coverage of the home range (home ranges near the edge should be trun-  
731 cated) and so 1/2 MMDM should be biased smaller than the home range radius.  
732 And, sometimes home range size is estimated by telemetry (Karanth, 1995; Bales  
733 et al., 2005). Use of MMDM summaries to estimate home range radius is usually  
734 combined with an AIC-based selection from among the closed-population models in  
735 Otis et al. (1978) which most often suggests heterogeneity in detection (model  $M_h$ ).  
736 Almost all of these early methods were motivated by studies of small mammals us-  
737 ing classical "trapping grids" but, more recently, their popularity in the study of  
738 wildlife populations has increased with the advent of new technologies, especially  
739 related to non-invasive sampling methods such as camera trapping. In particular,  
740 the series of papers by Karanth and Nichols (Karanth, 1995; Karanth and Nichols,  
741 1998, 2002) has led to fairly widespread adoption of these ideas.

#### 742 **1.4.2 Temporary emigration**

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

761 Another interesting idea is that of using some summary of "average location"  
762 as an individual covariate in standard capture-recapture models. Boulanger and

763 McLellan (2001) use distance-to-edge (DTE) as a covariate in the Huggins-Alho  
764 type of model. Ivan (2012) uses this approach in conjunction with an adjustment  
765 to the estimated  $N$  obtained by estimating the proportion of time individuals are  
766 “on the area formally covered by the grid” using radio telemetry. We do not dwell  
767 too much on these different variations but we do note that the use of DTE as an  
768 individual covariate amounts to some kind of intermediate model between simple  
769 closed population models and fully spatial capture-recapture models, which we  
770 address directly in Chapt. 4.

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

## 1.5 EXTENSION OF CLOSED POPULATION MODELS

780 The deficiency with classical closed population models is that they have no spatial  
781 context.  $N$  is just an integer parameter that applies equally well to estimating the  
782 number of unique words in a book, the size of some population that exists in a  
783 computer, or a bucket full of goldfish. The question of *where* the  $N$  items belong  
784 is central both to interpretation of data and estimates from all capture-recapture  
785 studies and, in fact, to the construction of spatial capture-recapture models con-  
786 sidered in this book. Surely it must matter whether the  $N$  items exist as words in  
787 a book, or goldfish in a bowl, or tigers in a patch of forest! That classical closed  
788 population models have no spatial context leads to a number of conceptual and  
789 methodological problems or limitations as we have encountered previously. More  
790 important, ecologists seldom care only about  $N$  – space is often central to objec-  
791 tives of many population studies – movement, space usage, resource selection, how  
792 individuals are distributed in space and in response to explicit factors related to  
793 landuse or habitat. Because space is central to so many real problems, this is proba-  
794 bly the number 1 reason that many ecologists don’t bother with capture-recapture.  
795 They haven’t seen capture-recapture methods as being able to solve their problems.  
796 Thus, the essential problem is that classical closed population models are too sim-  
797 ple – they ignore the spatial attribution of traps and encounter events, movement  
798 and variability in exposure of individuals to trap proximity. These problems can be  
799 addressed formally by the development of more general capture-recapture models.

**800 1.5.1 Towards spatial explicitness: Efford's formulation**

801 The solution to the various issues that arise in the application of ordinary capture-  
802 recapture models is to extend the closed population model so that  $N$  becomes  
803 spatially explicit. Efford (2004) was the first to formalize an explicit model for  
804 spatial capture-recapture problems in the context of trapping arrays. He adopted  
805 a Poisson point process model to describe the distribution of individuals and essen-  
806 tially a distance sampling formulation of the observation model which describes the  
807 probability of detection as a function of individual location, regarded as a latent  
808 variable governed by the point process model. While earlier (and contemporary)  
809 methods of estimating density from trap arrays have been ad hoc in the sense of  
810 lacking a formal description of the spatial model, Efford achieved a formalization  
811 of the model, describing explicit mechanisms governing the spatial distribution of  
812 individuals and how they are encountered by traps, but adopted a more or less  
813 ad hoc framework for inference under that spatial model using a simulation based  
814 method known as inverse prediction (Gopalaswamy, 2012).

815 Recently, there has been a flurry of effort devoted to formalizing inference un-  
816 der this model-based framework for the analysis of spatial capture-recapture data  
817 (Borchers and Efford, 2008; Royle and Gardner, 2011; Borchers, 2012; Gopalaswamy,  
818 2012). There are two distinct lines of work which adopt the model-based formula-  
819 tion in terms of the underlying point process but differ primarily by the manner in  
820 which inference is achieved. One approach (Borchers and Efford, 2008) uses classi-  
821 cal inference based on likelihood (see Chapt. 6), and the other (Royle and Young,  
822 2008) adopts a Bayesian framework for inference (Chaps. 5 and 17).

**823 1.5.2 Abundance as the aggregation of a point process**

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

832 Specifically, let  $\mathbf{s}_i; i = 1, 2, \dots, N$  be the locations of all individuals in the popu-  
833 lation. One of the key features of SCR models is that the point locations are latent,  
834 or unobserved, and we only obtain imperfect information about the point locations  
835 by observing individuals at trap or observation locations. Thus, the realized loca-  
836 tions of individuals represent a type of “thinned” point process, where the thinning  
837 mechanism is not random but, rather, biased by the observation mechanism. It is  
838 also natural to think about the observed point process as some kind of a compound  
839 or aggregate point process with a set of “parent” nodes being the locations of in-

840 individual home ranges or their centroids, and the observed locations as “offspring”  
841 - i.e., a Poisson cluster process (PCP). In that context, density estimation in SCR  
842 models is analogous to estimating the number of parents of a Poisson cluster process  
843 (Chandler and Royle, In press).

844 Most of the recent developments in modeling and inference from spatial en-  
845 counter history data, including most methods discussed in this book, are predicated  
846 on the view that individuals are organized in space according to a relatively simple  
847 point process model. More specifically, we assume that the collection of individ-  
848 ual activity centers are independent and identically distributed random variables  
849 distributed uniformly over some region. This is consistent with the assumption  
850 that the activity centers represent the realization of a Poisson point process or, if  
851 the total number of activity centers is fixed, then this is usually referred to as a  
852 binomial point process.

853 **1.5.3 The activity center concept**

854 In the context of SCR models, and because most animals we study by capture-  
855 recapture are not sessile, there is not a unique and precise mathematical definition  
856 of the point locations  $s$ . Rather, we imagine these to be the centroid of individ-  
857 uals home ranges, or the centroid of an individual’s activities during the time of  
858 sampling, or even its average location measured with error (e.g., from a long series  
859 of telemetry measurements). In general, this point is unknown for any individual  
860 but if we could track an individual over time and take many observations then we  
861 could perhaps get a good idea of where that point is. We’ll think of the collection  
862 of these points as defining the spatial distribution of individuals in the population.

863 We use the terms home range or activity center interchangeably. The term  
864 “home range center” suggests that models are only relevant to animals that exhibit  
865 behavior of establishing home ranges or territories, or central place foragers, and  
866 since not all species do that, perhaps the construction of SCR models based on this  
867 idea is flawed. However, the notion of a home range center is just a conceptual  
868 device and we don’t view this concept as being strictly consistent with classical  
869 notions of animal territories. Rather our view is that a home range or territory  
870 is inherently dynamic, temporally, and thus it is a transient quantity - where the  
871 animal lived during the period of study, a concept that is completely analogous to  
872 the more conventional notion of utilization distributions. Therefore, whether or not  
873 individuals of a species establish home ranges is irrelevant because, once a precise  
874 time period is defined, this defines a distinct region of space that an individual must  
875 have occupied.

876 **1.5.4 The state-space**

877 Once we introduce the collection of activity centers,  $s_i; i = 1, 2, \dots, N$ , then the  
878 question “what are the possible values of  $s$ ?” needs to be addressed because the

879 individual  $\mathbf{s}_i$  are *unknown*. As a technical matter, we will regard them as random  
 880 effects and in order to apply standard methods of statistical inference we need to  
 881 provide a distribution for these random effects. In the context of the point process  
 882 model, the possible values of the point locations referred to as the “state-space” of  
 883 the point process and this is some region or set of points which we will denote by  
 884  $\mathcal{S}$ . This is analogous to what is sometimes called the *observation window* for  $\mathbf{s}$  in  
 885 the point process literature. The region  $\mathcal{S}$  serves as a prior distribution for  $\mathbf{s}_i$  (or,  
 886 equivalently, the random effects distribution). In animal studies, as a description  
 887 of where individuals that could be captured are located, it includes our study area,  
 888 and should accommodate all individuals that could have been captured in the study  
 889 area. In the practical application of SCR models, in most cases estimates of density  
 890 will be relatively insensitive to choice of state-space which we discuss further in  
 891 Chapt. 5 and elsewhere.

892 **1.5.5 Abundance and density**

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

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

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

## 1.6 CHARACTERIZATION OF SCR MODELS

908 Formulation of capture-recapture models conditional on the latent point process is  
 909 the critical and unifying element of *all* SCR models. However, SCR models differ  
 910 in how the underlying process model is formulated, and its complexity. Most of the  
 911 development and application of SCR models has focused on their use to estimate  
 912 density and touting the fact that they resolve certain specific technical problems  
 913 related to the use of ordinary capture-recapture models. This is achieved with a sim-  
 914 ple process model being a basic point process of independently distributed points.

At the same time, there are models of CR data that focus exclusively on *movement* modeling, or models with explicit dynamics (Ovaskainen, 2004; Ovaskainen et al., 2008). Conceptually, these are akin to spatial versions of so-called Cormack-Jolly-Seber (CJS) models in the traditional capture-recapture literature, except they involve explicit mathematical models of movement based on diffusion or Brownian motion. Finally, there are now a very small number of papers that focus on *both* movement and density simultaneously (Royle and Young, 2008; Royle et al., 2011a; Royle and Chandler, 2012) or population dynamics and density (Gardner et al., 2010b).

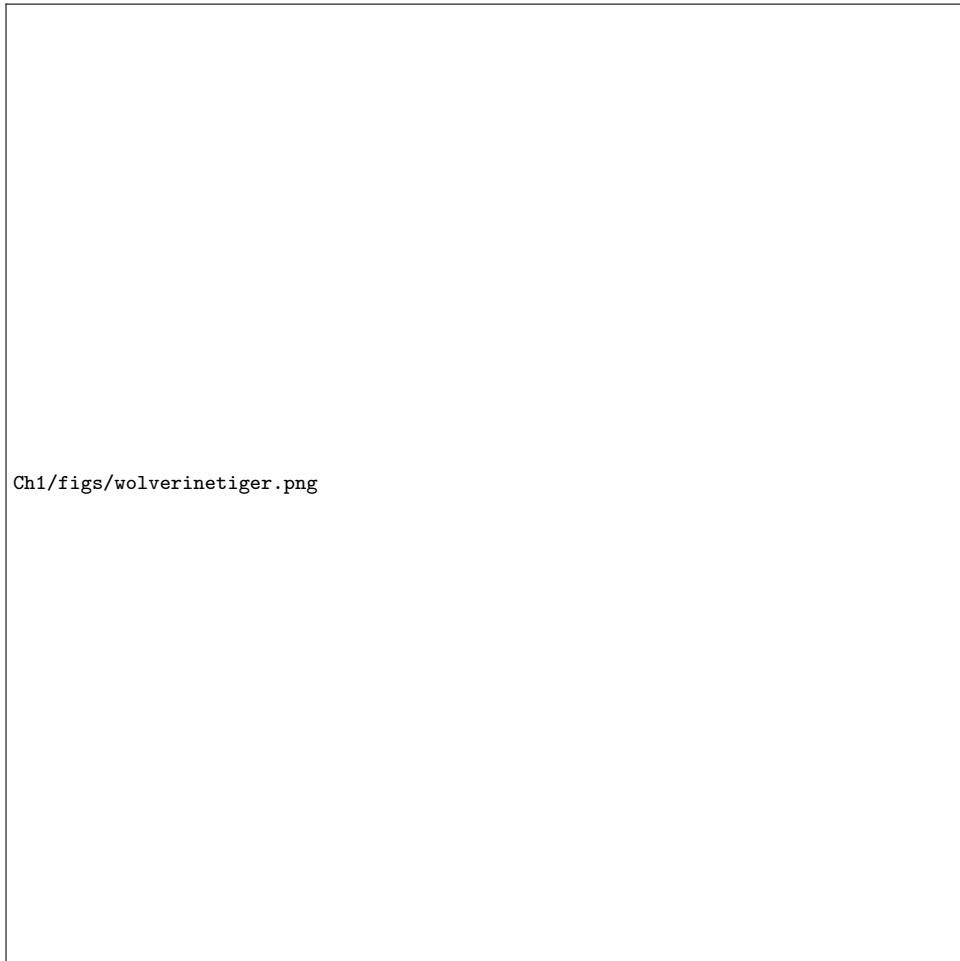
A key thing is that these models, whether focused just on density, or just on movement, or both, are similar models in terms of the underlying concepts, the latent structure, and the observation model. They differ primarily in terms of the ecological focus. Understanding movement is an important topic in ecology, but models that strictly focus on movement will be limited by two practical considerations: (1) most capture-recapture data e.g., by camera trapping or whatever, produces only a few observations of each individual (between 1-5 would be typical). So there is not too much information about complex movement models. (2) Typically people have an interest in density of individuals and therefore we need models that can be extrapolated from the sample to the unobserved part of 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 information on fine-scale movements that should be useful.

## 1.7 SUMMARY AND OUTLOOK

Spatial capture-recapture models are an extension of traditional 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.

As we have shown already, SCR models are not simply an extension of a technique to resolve certain technical problems. Rather, they provide a coherent, flexible framework for making ecological processes explicit in models of individual encounter history data, and for studying animal populations processes such as individual movement, resource selection, space usage, population dynamics, and density. Historically, researchers studied these questions independently, using ostensibly unrelated study designs and statistical procedures. For example, resource selection function (RSF) models for resource selection, state-space models for movement,

density using closed capture-recapture methods, and population dynamics with various “open” capture-recapture models. SCR can bring all of these problems together into a single unified framework for modeling and inference. Most importantly, spatial capture-recapture models promise the ability to integrate explicit ecological theories directly into the models so that we can directly test hypotheses about either space usage (e.g., Chapt. 13), landscape connectivity (Chapt. 12), movement, or spatial distribution (Chapt. 11). 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). In the following chapters we develop a comprehensive synthesis and extension of spatial capture-recapture models as they presently exist, and we suggest areas of future development and needed research.



Ch1/figs/wolverinetiger.png

**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*).

**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* )

**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.*

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

967  
968

## 2

969

# STATISTICAL MODELS AND SCR

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

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

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

996 simple **R** code throughout this chapter so that those who struggle with equations  
997 and mathematical notation can learn by doing. As mentioned in the Preface, we  
998 rely on **R** because it provides tremendous flexibility for analyzing data and because  
999 it is free. We do not, however, try to explain how to use **R** because there are so  
1000 many good references already, including Venables and Ripley (2002); Bolker (2008);  
1001 Venables et al. (2012).

1002 After covering some basic concepts of hierarchical modeling, we end the chapter  
1003 by describing spatial capture-recapture models using hierarchical modeling nota-  
1004 tion. This makes the concepts outlined in the previous chapter more precise, and  
1005 it highlights the fact that SCR models include explicit models for the ecological  
1006 processes of interest (e.g. spatial variation in density) and the observation process,  
1007 which describes how individuals are encountered.

## 2.1 RANDOM VARIABLES AND PROBABILITY DISTRIBUTIONS

### 1008 2.1.1 Stochasticity in ecology

1009 Few ecological processes can be described using purely deterministic models, and  
1010 thus we need a formal method for drawing conclusions from data while acknowl-  
1011 edging the stochastic nature of ecological systems. This is the role of statistical  
1012 inference, which is founded on the laws of probability. For our purposes, it suffices  
1013 to be familiar with a small number of concepts from probability theory—the most  
1014 important of which is the concept of a random variable, say  $X$ . A random variable  
1015 is a variable whose realized value is the outcome of some stochastic process. To  
1016 be more precise, a random variable is characterized by a function that describes  
1017 the probability of observing the value  $x$ . This probability function can be written  
1018  $\Pr(X = x|\theta)$  where  $\theta$  is a parameter, or set of parameters of the function. If  $x$  is  
1019 discrete, e.g. binary or integer, then we call the probability function a probability  
1020 mass function (pmf). If  $x$  is continuous, the function is called a probability density  
1021 function (pdf).

1022 To clarify the concept of a random variable, let  $X$  be the number of American  
1023 shad (*Alosa sapidissima*) caught after  $K = 20$  casts at the shad hole on Deerfield  
1024 River in Massachusetts. Suppose that we had a good day and caught  $x = 7$  fish.  
1025 If there were no random variation at play, we would say that the probability of  
1026 catching a fish, which we will call  $p$ , is  $p = 7/20 = 0.35$ , and we would always  
1027 expect to catch 7 shad after 20 casts. In other words, our deterministic model is  
1028  $x = 0.35 \times K$ . In reality, however, we can be pretty sure that this deterministic  
1029 model would not be very good. Even if we knew for certain that  $p \equiv 0.35$ , we would  
1030 expect some variation in the number of fish caught on repeated fishing outings.  
1031 To describe this variation, we need a model that acknowledges uncertainty (i.e.,  
1032 stochasticity), and specifically we need a model that describes the probability of  
1033 catching  $x$  fish given  $K$  and  $p$ ,  $\Pr(X = x|K, p)$ . Since  $x$  is discrete, not continuous,  
1034 we need a pmf. Before contemplating which pmf is most appropriate in this case,

1035 we need to first mention a few issues related to notation.

1036 Statisticians make things easier for themselves, and more complicated for ev-  
 1037 eryone else, by using different notation for probability distributions. Sometimes  
 1038 you will see  $\Pr(X = x|K, p)$  expressed as  $f(X|K, p)$  or  $f(X; K, p)$  or  $p(X|K, p)$  or  
 1039  $\pi(X|K, p)$  or  $\mathbb{P}(X|K, p)$  or  $[X|K, p]$  or even just  $[X]!$  Just remember that these  
 1040 expressions all have the same meaning—they are all probability distributions that  
 1041 tell us the probability of observing any possible realization of the random variable  
 1042  $X$ . In this book, we will almost always use bracket notation (the last two examples  
 1043 above) to represent arbitrary probability distributions. Hence, from here on out,  
 1044 when you see  $[X|K, p]$ , just remember that this is equivalent to the more traditional  
 1045 expression  $\Pr(X = x|K, p)$ . In addition, from here on, to achieve a more concise  
 1046 presentation, we will no longer use uppercase letters to denote random variables  
 1047 and lowercase letters for realized values. Rather, we will define a random vari-  
 1048 able by some symbol ( $x, N$ , etc...) and let the context determine whether we are  
 1049 talking about the random variable itself, or realized values of it. In some limited  
 1050 cases, we will want upper- and lower-case letters to represent different variables.  
 1051 For example, we will often let  $N$  denote population size and  $n$  denote the number  
 1052 of individuals actually detected.

1053 When we wish to be specific about a probability distribution, we will do so in  
 1054 one of two ways, one mathematically precise and one symbolic. Before explaining  
 1055 these two options, let's choose a specific distribution as a model for the data in our  
 1056 example. In this case, the natural choice for  $[x|K, p]$  is the binomial distribution,  
 1057 the mathematically precise representation of which is

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

1058 The right-hand side of this equation is the binomial pmf (described in more detail  
 1059 in Sec. 2.2), and plugging in values for the parameters  $K$ , and  $p$  will return the  
 1060 probability of observing any realized value of the random variable  $x$ . This is precise,  
 1061 but it is also cumbersome to write repetitively, and it may make the eyes glaze over  
 1062 when seen too often. Thus, we will often simplify Eq. 2.1.1 using the symbolic  
 1063 notation:

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

1064 The “ $\sim$ ” symbol is meant to represent a stochastic relationship, and can be read  
 1065 “is distributed as.” Another reason for using this notation is that it resembles the  
 1066 syntax of the **BUGS** language, which we will frequently use to conduct Bayesian  
 1067 inference.

1068 Note that once we choose a probability distribution, we have chosen a model. In  
 1069 our example, we have specified our model as  $x \sim \text{Binomial}(K, p)$ , and because we  
 1070 are assuming that the parameters are known, we can make probability statements  
 1071 about future outcomes. Continuing with our fish example, we might want to know  
 1072 the probability of catching  $x = 7$  again after  $K = 20$  casts on a future fishing

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

1073 outing, assuming that we know  $p = 0.35$ . Evaluating the binomial pmf returns a  
 1074 probability of approximately 0.18, as show using this bit of **R** code:

```
1075 > dbinom(7, 20, 0.35)
1076 [1] 0.1844012
```

1077 By definition, the pmf allows us to evaluate the probability of observing any  $x$  given  
 1078  $K = 20$  and  $p = 0.35$ , thus the distribution of the random variable can be visualized  
 1079 by evaluating it for all values of  $x$  that have non-negligible probabilities, as can be  
 1080 easily done in **R**:

```
1081 plot(0:20, dbinom(0:20, 20, 0.35), type="h", ylab="Probability",
1082       xlab="Number of shad caught (X)")
```

1083 the result of which is shown in Fig. 2.1 with some extra details.

1084 The purpose of this little example is to show that once we specify a model for the  
 1085 random variable(s) being studied, we can begin drawing conclusions, i.e. making  
 1086 inferences, about the processes of interest, even in the face of uncertainty. Prob-  
 1087 ability distributions are essential to this process, and thus we need to understand  
 1088 them in more depth.

### 1089 **2.1.2 Properties of probability distributions**

1090 A pdf or a pmf is a function like any other function in the sense that it has one  
 1091 or more arguments whose values determine the result of the function. However,  
 1092 probability functions have a few properties that distinguish them from other func-  
 1093 tions. The first is that the function must be non-negative for all possible values of  
 1094 the random variable, i.e.  $[x] \geq 0$ . The second requirement is that the integral of  
 1095 a pdf must be unity,  $\int_{-\infty}^{\infty} [x] dx = 1$ , and similarly for a pmf, the summation over  
 1096 all possible values is unity,  $\sum_x [x] = 1$ . The following **R** code demonstrates this for  
 1097 the normal and binomial distributions:

```
1098 > integrate(dnorm, -Inf, Inf, mean=0, sd=1)$value
1099 [1] 1
1100 > sum(dbinom(0:5, size=5, p=0.1))
1101 [1] 1
```

1102 This requirement is important to remember when one develops a non-standard  
 1103 probability distribution. For example, in Chapt. 11 and 13, we work with resource  
 1104 selection functions whose probability density function is not one that is pre-defined  
 1105 in software packages such as **R** or **BUGS**.

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

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

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, or mean, 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:

```
1113 > mean(rbinom(10000, 20, 0.3))
1114 [1] 6.9865
```

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

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

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

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

```
1125 > integrate(function(x) x*dnorm(x, 3, 1), -Inf, Inf)
1126 3 with absolute error < 0.00033
```

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

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

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

```
1135 > sum(dbinom(0:100, 20, 0.35)*0:100)
1136 [1] 7
```

This is great. But of what use is it? One very important concept to understand is that when we fit models, we are often modeling changes in the expected value of

1139 some random variable. For example, in Poisson regression, we model the expected  
 1140 value of the random variable, which may be a function of environmental variables.

1141 The ability to model the expected value of a random variable gets us very far,  
 1142 but we also need a model for the variance of the random variable. The variance  
 1143 describes the amount of variation around the expected value. Specifically,  $\text{Var}(x) =$   
 1144  $\mathbb{E}((x - \mathbb{E}(x))^2)$ . Clearly, if the variance is zero, the variable is not random as  
 1145 there is no uncertainty in its outcome. For some distributions, notably the normal  
 1146 distribution, the variance is a parameter to be estimated. Thus, in ordinary linear  
 1147 regression, we estimate both the expected value  $\mu = \mathbb{E}(x)$ , which may be a function  
 1148 of covariates, and the variance  $\sigma^2$ , or similarly the residual standard error  $\sigma$ . For  
 1149 other distributions, the variance is not an explicit parameter to be estimated, and  
 1150 instead, the mean to variance ratio is fixed. In the case of the Poisson distribution,  
 1151 the mean is equal to the variance,  $\mathbb{E}(x) = \text{Var}(x) = \lambda$ . A similar situation is true  
 1152 for the binomial distribution—the variance is determined by the two parameters  $K$   
 1153 and  $p$ ,  $\text{Var}(x) = Kp(1 - p)$ . In our earlier example with  $K = 20$  and  $p = 0.35$ , the  
 1154 variance is 4.55. Toying around with these ideas using random number generators  
 1155 may be helpful. Here is some code to illustrate some of these basic concepts:

```
1156 > 20*0.35*(1-0.35)           # Exact variance, Var(x)
1157 [1] 4.55
1158 > x <- rbinom(100000, 20, 0.35)
1159 > mean((x-mean(x))^2)        # Monte Carlo approximation
1160 [1] 4.545525
```

## 2.2 COMMON PROBABILITY DISTRIBUTIONS

1161 We got a little ahead of ourselves in the previous sections by using the binomial  
 1162 and Poisson distributions without describing them in detail. A solid understanding  
 1163 of the binomial, Poisson, multinomial, uniform, and normal (or Gaussian) distri-  
 1164 butions is absolutely essential throughout the remainder of the book. We will  
 1165 occasionally make use of other distributions such as the beta, log-normal, gamma,  
 1166 Dirichlet, etc... that can be helpful when modeling capture-recapture data, but  
 1167 these distributions can be readily understood once you are comfortable with the  
 1168 more commonly used distributions described in this section.

### 2.2.1 The binomial distribution

1169 The binomial distribution plays a critical role in ecology. It is used for purposes  
 1170 as diverse as modeling count data, survival probability, occurrence probability, and  
 1171 capture probability, just to name a few. To describe the properties of the binomial  
 1172 distribution, and related distributions, we will introduce a new example. Suppose  
 1173 we are conducting a bird survey at a site in which  $N = 10$  chestnut-sided warblers  
 1174 (*Setophaga pensylvanica*) occur, and each of these individuals has a detection prob-  
 1175 ability of  $p = 0.5$ . The binomial distribution is the natural choice for describing

1177 the number of individuals that we would expect to detect ( $n$ ) in this situation, and  
 1178 using our notation, we can write the model as:  $n \sim \text{Binomial}(10, 0.5)$ . When  $p < 1$ ,  
 1179 we can expect that we will observe a different number of warblers on each of  $K$   
 1180 replicate survey occasions. To see this, we simulate data under this simple model  
 1181 with  $K = 3$ .

```
1182 > n <- rbinom(3, size=10, prob=0.5) # Generate 3 binomial outcomes
1183 > n                                     # Display the 3 values
1184 [1] 6 4 8
```

1185 The vector of counts will typically differ each time you issue this command; however,  
 1186 we know the probability of observing any value of  $n_k$  because it is defined by the  
 1187 binomial pmf. As we demonstrated earlier, in **R** this probability can be found using  
 1188 the `dbinom` function. For example, the probability of observing  $n_k = 5$  is given by:

```
1189 > dbinom(5, 10, 0.5)
```

1190 This simply evaluates the function shown in Table 2.1. We could do the same more  
 1191 transparently, but less efficiently, using any of the following:

```
1192 > n <- 5; N <- 10; p <- 0.5
1193 > factorial(N)/(factorial(n)*factorial(N-n))*p^n*(1-p)^(N-n)
1194 > exp(lgamma(N+1) - (lgamma(n+1) + lgamma(N-n+1)))*p^n*(1-p)^(N-n)
1195 > choose(N, n)*p^n*(1-p)^(N-n)
```

1196 Note that the last three lines of code differ only in how they compute the binomial  
 1197 coefficient  $\binom{N}{n}$ , which is the number of different ways we could observe  $n = 5$  of  
 1198 the  $N = 10$  chestnut-sided warblers at the site. The binomial coefficient, which is  
 1199 read “ $N$  choose  $n$ ” is defined as

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

1200 Now that we know how to simulate binomial data and compute the probabilities  
 1201 of observing any particular outcome  $n$ , conditional on the parameters  $N$  and  
 1202  $p$ , we can contemplate the relevance of the binomial distribution in spatial capture-  
 1203 recapture models. One important application of the binomial distribution is as a  
 1204 model encounter frequencies. Indeed, one of the most important encounter models  
 1205 in SCR will be referred to as the “binomial encounter model”, in which the number  
 1206 of times individual  $i$  is captured at “trap”  $j$  after  $K$  survey occasions is modeled as  
 1207  $y_{ij} \sim \text{Binomial}(K, p_{ij})$ . Here,  $p_{ij}$  is the encounter probability determined, in part,  
 1208 by the distance between an animal’s activity center and the trap location. This  
 1209 binomial encounter model is described in detail in Sec. 7.1. Another important ap-  
 1210 plication of the binomial distribution is as a prior for the population size parameter  
 1211 in Bayesian analyses, as is discussed in Chapt. 4.

---

**2.2.2 The Bernoulli distribution**

Above, we showed 3 alternatives to `dbinom` for evaluating the binomial pmf. 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  $\mathbf{y} = (0, 0, 1, 1, 1, 1, 1, 0, 0, 0)$ , indicating 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 possible vectors  $\mathbf{y}$  with 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 abbreviation “*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. 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 that 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

1247 use `dbinom` with the `size` argument set to 1. For example, `dbinom(1, 1, 0.3)`  
 1248 returns the Bernoulli probability of observing  $n = 1$  given  $p = 0.3$ .

1249 **2.2.3 The multinomial and categorical distributions**

1250 The binomial distribution is used when we are accumulating a binary response—  
 1251 that is, one in which there are two possible categories such as success/failure or  
 1252 captured/not-captured. The multinomial distribution is a multivariate extension  
 1253 of the binomial used when there are  $G > 2$  categories. The multinomial distribution  
 1254 can be thought of as a model for placing  $N$  items in the  $G$  categories, which are  
 1255 also called bins or cells. Each bin has its own probability  $\pi_g$  and these probabilities  
 1256 must sum to one. In ecology,  $N$  is often population size or the number of individuals  
 1257 detected, but the definition of the  $G$  bins varies among applications. For example,  
 1258 in distance sampling, when the distance data are aggregated into intervals, the  
 1259 bins are the distance intervals, and the cell probabilities are functions of detection  
 1260 probability in each interval (Royle et al., 2004).

1261 The multinomial distribution is widely used to model data from traditional,  
 1262 non-spatial capture-recapture studies. Earlier we let  $y_{ik}$  denote a binary random  
 1263 variable indicating if warbler  $i$  was detected on survey  $k$ . The vector of observations  
 1264 for an individual,  $\mathbf{y}_i$ , is often referred to as the individual’s “encounter history”.  
 1265 The number of possible encounter histories depends on  $K$ , the number of survey  
 1266 occasions. Specifically, there are  $2^K$  possible encounter histories<sup>1</sup>. If we tabulate the  
 1267 number of individuals with each encounter history, the frequencies can be modeled  
 1268 using the multinomial distribution.

1269 Going back to our chestnut-sided warbler example, suppose the 10 individuals  
 1270 are marked and we make  $K = 2$  visits to the site such that there are  $2^K = 4$  pos-  
 1271 sible encounter histories: (11, 10, 01, 00), where, for example, “10” is the encounter  
 1272 history for an individual detected on the first visit but not the second. If  $p = 1$ ,  
 1273 then the encounter history for each of the 10 individuals must be “11”. That is, we  
 1274 would detect each individual on both occasions. In this case, we the data would be:  
 1275  $\mathbf{h} = (10, 0, 0, 0)$ , which indicates that all 10 warblers had the first encounter history.  
 1276 The corresponding cell probabilities would be  $\boldsymbol{\pi} = (1, 0, 0, 0)$ . What about the sit-  
 1277 uation where  $p < 1$ , e.g.  $p = 0.3$ ? In this case, the probability of observing the  
 1278 capture history “11” (detected on both occasions) is  $p \times p = 0.3 \times 0.3 = 0.09$ . The  
 1279 probability of observing “10” is  $p \times (1 - p) = 0.21$ . Following this logic, the vector  
 1280 of cell probabilities is  $\boldsymbol{\pi} = (0.09, 0.21, 0.21, 0.49)$ . We can simulate data under this  
 1281 model as follows:

```
1282 > caphist.probs <- c("11"=0.09, "10"=0.21, "01"=0.21, "00"=0.49)
1283 > drop(rmultinom(1, 10, caphist.probs))
1284 11 10 01 00
```

<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  $2^K - 1$

---

```
1285 0 3 2 5
```

1286 The result of our simulation is that zero individuals were observed with the capture  
 1287 history “11” and 5 individuals were observed with the capture history “00”. The  
 1288 other 5 individuals were observed one out of the two occasions. This is not such a  
 1289 surprising outcome given  $p = 0.3$ .

1290 As in non-spatial capture-recapture studies, the multinomial distribution turns  
 1291 out to be very important in spatial capture-recapture studies. However,  $N$  is not  
 1292 defined as population size. Rather, we use the multinomial distribution when an  
 1293 individual can only be captured in a single trap during an occasion. Thus  $N = 1$   
 1294 and the cell probabilities are the probabilities of being captured in each trap. A  
 1295 thorough discussion of this point can be found in Chapt. 9. Another application  
 1296 of the multinomial distribution in SCR models is discussed in Chapt. 11 where we  
 1297 discuss how to model the probability that an individual’s activity center is located  
 1298 in one of the cells of a raster defining the spatial region of interest.

1299 Just as the Bernoulli distribution is the elemental form of the binomial distri-  
 1300 bution (being the case  $N = 1$ ), the categorical distribution is essentially equivalent  
 1301 to the multinomial distribution with size parameter  $N \equiv 1$ . The only difference is  
 1302 that, rather than returning a vector with a single element equal to 1, it returns the  
 1303 element *location* where the 1 occurs. For example, if  $\mathbf{y} = (0, 0, 1, 0)$  is an outcome  
 1304 of a multinomial distribution with  $N = 1$ , then the categorical outcome would be  
 1305 3 because the 1 is located in third position in the vector. Thus, in spatial capture-  
 1306 recapture models, we might use either the multinomial distribution with  $N = 1$   
 1307 or the categorical distribution. The various **BUGS** engines describe the categori-  
 1308 cal distribution by the declaration `dcat` and, in **R**, we can simulate categorical  
 1309 outcomes using the function `sample` or as so:

```
1310 > which(rmultinom(1, 1, c(0.1, 0.7, 0.2)) == 1)
1311 [1] 2
```

#### 1312 2.2.4 The Poisson distribution

1313 The Poisson distribution is the canonical model for count data in ecology. More  
 1314 generally, the Poisson distribution is a model for random variables taking on non-  
 1315 negative, integer values. Although it is a simple model having just one parameter,  
 1316  $\lambda = \mathbb{E}(x) = \text{Var}(x)$ , its applications are highly diverse, including as a model of  
 1317 spatial variation in abundance or as a model for the frequency of behaviors over  
 1318 time. Just as logistic regression is the standard generalized linear model (GLM)  
 1319 used to model binary data, Poisson regression is the default GLM for modeling  
 1320 count data and variation in  $\lambda$ .

1321 The Poisson distribution is related to both the binomial and multinomial distri-  
 1322 butions, and the following three bits of trivia are occasionally worth knowing. First,  
 1323 it is the limit of the binomial distribution as  $N \rightarrow \infty$  and  $p \rightarrow 0$ , which means that  
 1324 for high values of  $N$  and low values of  $p$ ,  $\text{Poisson}(N \times p)$  is approximately equal

1325 to  $\text{Binomial}(N, p)$ . Second, if  $\{n_1 \sim \text{Poisson}(\lambda_1), \dots, n_K \sim \text{Poisson}(\lambda_K)\}$  then the  
 1326 vector of counts is multinomial,  $\{n_1, \dots, n_K\} \sim \text{Multinomial}(\sum_k n_k, \{\frac{\lambda_1}{\sum_k \lambda_k}, \dots, \frac{\lambda_K}{\sum_k \lambda_k}\})$ .  
 1327 Third, the sum of two Poisson random variables  $x_1 \sim \text{Poisson}(\lambda_1)$  and  $x_2 \sim$   
 1328  $\text{Poisson}(\lambda_2)$  is also Poisson:  $x_1 + x_2 \sim \text{Poisson}(\lambda_1 + \lambda_2)$ .

1329 The Poisson distribution has two important uses in spatial capture-recapture  
 1330 models: (1) as a prior distribution for the population size parameter  $N$ , and (2) as a  
 1331 model for the frequency of captures in a trap. In the first context, the Poisson prior  
 1332 for  $N$  results in a Poisson point process for the location of the  $N$  activity centers  
 1333 in the region of interest. This topic is discussed in Chapt. 5 and Chapt 11. The  
 1334 second use of the Poisson distribution in spatial capture-recapture is to describe  
 1335 data from sampling methods in which an individual can be detected multiple times  
 1336 at a trap during a single occasion. For example, in camera trapping studies we  
 1337 might obtain multiple pictures of the same individual at a trap during a single  
 1338 sampling occasion. Thus,  $\lambda$  in this case would be defined as the expected number  
 1339 of detections or captures per occasion.

1340 **2.2.5 The uniform distribution**

1341 The lowly uniform distribution is a continuous distribution whose only two pa-  
 1342 rameters are the lower and upper bounds that restrict the possible values of the  
 1343 random variable  $x$ . These bounds are almost always known, so there is typically  
 1344 nothing to estimate. Nonetheless, the uniform distribution is one of the most widely  
 1345 used distributions, especially among Bayesians who frequently use it to as a “non-  
 1346 informative” prior distribution for a parameter. For example, if we have a capture  
 1347 probability parameter  $p$  that we wish to estimate, but we have no prior knowl-  
 1348 edge of what value it may take in the range  $[0,1]$ , we will often use the prior  
 1349  $p \sim \text{Uniform}(0,1)$ . This states that  $p$  is equally likely to take on any value between  
 1350 zero and one. Prior distributions are described in more detail in the next chapter.

1351 Another common usage of the uniform distribution is as a prior for the coor-  
 1352 dinates of points in the real plane, i.e. in two-dimensional space. Such a use of  
 1353 the uniform distribution implies that a point process is “homogeneous”, meaning  
 1354 that the location of one point does not affect the location of another point and  
 1355 that the expected density of points is constant throughout the region. Thus, to  
 1356 simulate a realization from a homogeneous Poisson point process in the unit square  
 1357  $[0, 1] \times [0, 1]$ , we could use the following R code:

```
1358 D <- 100      # points per unit area
1359 A <- 1        # Area of unit square
1360 N <- rpois(1, D*A)
1361 plot(s <- cbind(runif(N), runif(N)))
```

1362 where  $s$  is a matrix of coordinates with  $N$  rows and 2 columns. We will often  
 1363 represent the uniform point process using the following notation:

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

1364 where  $\mathcal{S}$  is some specific unit of space called the state-space of the random variable  
 1365  $\mathbf{s}$ . It would be more correct to somehow distinguish this two-dimensional uniform  
 1366 distribution for the univariate one. That is, it might be more clear to use notation  
 1367 such as  $\mathbf{s} \sim \text{Uniform}_2(\mathcal{S})$  instead, but this is somewhat cumbersome, so we will opt  
 1368 for the former expression.

### 1369 2.2.6 Other distributions

1370 The other continuous distributions that are regularly encountered in SCR models  
 1371 are primarily used as priors in Bayesian analyses, and thus we will avoid a lengthy  
 1372 discussion of their properties. The normal distribution, also called the Gaussian  
 1373 distribution, is perhaps the most widely recognized and applied probability model  
 1374 in statistics, but it plays only a minor role in SCR models other than as a model for  
 1375 signal strength in acoustic SCR models (Efford et al., 2009b; Dawson and Efford,  
 1376 2009), and see Sec. 9.4. Nonetheless, it is the canonical prior for any continuous  
 1377 random variable with infinite support, and thus it is often used as a prior when  
 1378 applying Bayesian methods. One common usage is as a prior for the  $\beta$  coefficients  
 1379 of a linear model defining some parameter as a function of covariates (usually on  
 1380 a transformed scale). An example, including a cautionary note, is provided in  
 1381 Sec. 3.5.1. Be aware that although the normal distribution is typically parameterized  
 1382 in terms of the variance parameter  $\sigma^2$ , in the **BUGS** language, the inverse of  
 1383 the variance, or precision, is used instead,  $\tau = 1/\sigma^2$ . In **R**, the **dnorm** function  
 1384 requires the standard deviation  $\sigma$ , rather than the variance  $\sigma^2$ .

1385 The bivariate normal distribution is a generalization of the normal distribution  
 1386 and a special case of the multivariate normal distribution whose pdf is shown in  
 1387 Table 2.1. The bivariate normal distribution is used to model two (possibly) dependent  
 1388 continuous variables whose symmetric variance-covariance matrix is denoted  
 1389  $\Sigma$ . In SCR models, we most often use this model as a rudimentary description of  
 1390 movement outcomes about a home range center. If there is no correlation, then the  
 1391 model reduces to two independent normal draws along the coordinate axes. The  
 1392 following code generates bivariate normal outcomes with no correlation ( $\rho = 0$ ), as  
 1393 well as outcomes in which the correlation is  $\rho = 0.9$ .

```
1394 library(mvtnorm)
1395 set.seed(3)
1396 mu <- c(0,0)
1397 Sigma <- matrix(c(1, .9, .9, 1), 2, 2)
1398 X1 <- cbind(rnorm(50, mu[1], Sigma[1,1]), # No correlation (rho=0)
1399           rnorm(50, mu[2], Sigma[2,2]))
1400 X2 <- rmvnorm(50, mu, Sigma) # rho=0.9
```

1401 Fig. 2.2 shows the simulated points.

1402 Several of the parameters in capture-recapture models do not have infinite support,  
 1403 but instead are probabilities restricted to the range [0, 1], or are positive

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

1404 valued living between zero and  $\infty$ . The beta distribution is the standard prior  
 1405 used for probabilities because it can be used to express either a lack of knowledge  
 1406 or very precise knowledge about a parameter. For example, a Beta(1,1) distribu-  
 1407 tion is equivalent to a Uniform(0,1) distribution. However, unlike the uniform  
 1408 distribution, the beta distribution can be used as an informative prior; for exam-  
 1409 ple if published estimates of detection probability exist we can choose parameters  
 1410 of the beta distribution to reflect that. To gain some familiarity with the beta  
 1411 distribution, execute the following R commands:

```
1412 curve(dbeta(x, 1, 1), col="black", ylim=c(0,5))
1413 curve(dbeta(x, 10, 10), col="blue", add=TRUE)
1414 curve(dbeta(x, 10, 20), col="darkgreen", add=TRUE)
```

1415 Other parameters in SCR models are continuous but positive-valued and can be  
 1416 modeled using the gamma distribution. As with the beta distribution, the gamma  
 1417 distribution is typically favored over the uniform distribution when one is interested  
 1418 in using an informative prior. It is also frequently used as a vague prior for the  
 1419 inverse of variance parameters, but it is wise to compare this prior to a uniform to  
 1420 assess its influence on the posterior.

### 2.3 STATISTICAL INFERENCE AND PARAMETER ESTIMATION

1421 If the parameters of a statistical model were known with absolute certainty, then it  
 1422 would be possible to use pdfs and pmfs to make direct probability statements about  
 1423 unknowns such as future outcomes. However, we almost never know the actual  
 1424 values of parameters, and instead we have to estimate them from observations  
 1425 (i.e., data). Our inferences must then acknowledge the uncertainty associated with  
 1426 our imperfect knowledge of the parameters. Doing so is most often accomplished  
 1427 using one of two approaches: classical (frequentist) inference or Bayesian inference.  
 1428 These two modes of inference regard the uncertainty about parameters in entirely  
 1429 different ways. In the next chapter, we will review some of the important concepts  
 1430 in Bayesian inference, so here, we will focus on the frequentist perspective.

1431 Suppose we count oak trees at  $J$  sites, and the resulting data  $\{y_1, \dots, y_J\}$  can  
 1432 be assumed to be *iid* outcomes from some distribution, such as the Poisson with  
 1433 unknown parameter  $\lambda$ . We want to estimate this parameter. In classical inference,  
 1434 the only uncertainty about  $\lambda$  is that attributable to sampling. For instance, we can  
 1435 imagine repeatedly sampling the population (sites in this example) and obtaining  
 1436 sample-specific estimates of  $\lambda$ . Typically, we entertain the idea that there are an  
 1437 infinite number of possible samples and so we could obtain an infinite number of  
 1438 estimates:  $\{\hat{\lambda}_1, \hat{\lambda}_2, \dots, \hat{\lambda}_\infty\}$ . If these estimates are produced using the method

of maximum likelihood, and as  $n$  tends to infinity, the distribution of estimates, called the sampling distribution, will be normally distributed with  $\mathbb{E}(\hat{\lambda}) = \lambda$ . The standard deviation of the sampling distribution is called the standard error, which can also be estimated as part of the maximum likelihood procedure. Of course, we almost always have just a single sample of data, and hence a single  $\hat{\lambda}$  and a single estimate of the standard error. However, under the assumption of a normally distributed sampling distribution, we can construct a confidence interval that will include the true value of  $\lambda$  with coverage probability  $1 - \alpha$ , where  $\alpha$  is a prescribed value like 0.05. An important point is that there is no uncertainty associated with the actual parameter—it is regarded as a fixed value, and hence probability is only used to characterize the estimator via its sampling distribution.

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

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

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

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

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

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

```
1466 > lambda <- 3 # Actual parameter value
1467 > y1 <- rpois(100, lambda) # Realized values (data)
1468 > negLogLike1 <- function(par) -sum(dpois(y1, par, log=TRUE))
```

```

1473 > starting.value <- c('lambda'=1)
1474 > optim(starting.value, negLogLike1)$par # MLE
1475   lambda
1476 3.039844

```

1477 Explicitly maximizing the likelihood, numerically, isn't actually necessary here because the MLE of  $\lambda$  is given by the mean of the observations. A more interesting example is when there are covariates of  $\lambda$ . For example, suppose  $\lambda$  is a function of elevation and vegetation height according to:  $\log(\lambda_i) = \beta_0 + \beta_1 \text{ELEV}_i + \beta_2 \text{VEGHT}_i$ . This is a standard Poisson regression problem, with likelihood:

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

1482 This likelihood is almost identical to the previous one except that  $\lambda$  is now a function, and so we need to estimate the parameters of the function, i.e. the  $\beta$ 's.  
 1483 Some code to fit this model to simulated data is shown here:

```

1485 > nsites <- 100
1486 > elevation <- rnorm(100)
1487 > veght <- rnorm(100)
1488 > beta0 <- 1
1489 > beta1 <- -1
1490 > beta2 <- 0
1491 > lambda <- exp(beta0 + beta1*elevation + beta2*vegght)
1492 > y2 <- rpois(nsites, lambda)
1493 > negLogLike2 <- function(pars) {
1494   +   beta0 <- pars[1]
1495   +   beta1 <- pars[2]
1496   +   beta2 <- pars[3]
1497   +   lambda <- exp(beta0 + beta1*elevation + beta2*vegght)
1498   +   -sum(dpois(y2, lambda, log=TRUE))
1499 }
1500 > starting.values <- c('beta0'=0, 'beta1'=0, 'beta2'=0)
1501 > optim(starting.values, negLogLike2)$par
1502   beta0      beta1      beta2
1503 0.98457756 -1.03025173 -0.01218292

```

1504 We see that the maximum likelihood estimates (MLEs) are very close to the true  
 1505 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:

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

1506 This is an example of a mixed effects model or a hierarchical model. How do we  
 1507 estimate the parameters of a model that includes random effects? Earlier the like-  
 1508 lihood function was written as the product of probabilities determined by a single  
 1509 pmf or pdf,  $[y|\lambda]$ , but now we have an additional random variable, and we are forced  
 1510 to think about conditional relationships, because  $y$  depends upon  $\beta_i$  and  $\beta_i$  depends  
 1511 upon other parameters, specifically  $\mu$  and  $\sigma^2$ . This type of conditional dependence  
 1512 among parameters is the essence of hierarchical models, and statistical analysis  
 1513 of hierarchical models requires that we discuss joint distributions, marginal distri-  
 1514 butions and conditional distributions. These concepts will be used extensively in  
 1515 Chapt. 6 where we demonstrate how to estimate parameters of hierarchical models  
 1516 using maximum likelihood.

## 2.4 JOINT, MARGINAL, AND CONDITIONAL DISTRIBUTIONS

1517 So far we have restricted our attention to situations in which we wish to make  
 1518 inference about a single random variable. However, in ecology, we often are inter-  
 1519 ested in multiple random variables and how they are related. Let  $Y$  be a random  
 1520 variable that may or may not be independent of  $X$  (here again we will distinguish  
 1521 between random variables and realized values for conceptual clarity). Inference  
 1522 about these two random variables can be made using the joint, marginal, or condi-  
 1523 tional distributions—or, we may make use of all of them depending on the question  
 1524 being asked. In the case of discrete random variables, the joint distribution is the  
 1525 probability that  $X$  takes on the value  $x$  and that  $Y$  takes on the value  $y$ , which  
 1526 is written  $[X = x, Y = y]$ . To clarify this concept, let's go back to our original  
 1527 example where  $X$  was the number of fish caught after 20 casts, which we said  
 1528 was an *iid* binomial random variable. Now, let's suppose that  $X$  depends on the  
 1529 random variable  $Y$ , which is the number of other fisherman at the hole. Specifi-  
 1530 cally, let's say that the probability of catching a fish  $p$  is related to  $Y$  according  
 1531 to  $\text{logit}(p) = -0.6 + -2y$ . Furthermore, let's make the intuitive assumption that  
 1532 the number of fishermen at the hole is a Poisson random variable with mean 0.6,  
 1533 i.e.  $Y \sim \text{Poisson}(0.6)$ . Our model is now fully specified, and so we can answer the  
 1534 question: “what is the probability of catching  $x$  fish and of there being  $y$  fishermen  
 1535 at the hole”. This joint distribution is given by the product of the binomial pmf  
 1536 (with  $p$  determined by  $y$ ) and the Poisson pmf with  $\lambda = 0.6$ . The following R code  
 1537 creates the joint distribution.

```
1538 > X <- 0:20 # All possible values of X
1539 > Y <- 0:10 # All possible values of Y
```

```

1540 > lambda <- 0.6
1541 > p <- plogis(-0.62 + -2*Y) # p as function of Y
1542 > round(p,2)
1543 [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
1544 > joint <- matrix(NA, length(X), length(Y))
1545 > rownames(joint) <- paste("X=", X, sep="")
1546 > colnames(joint) <- paste("Y=", Y, sep="")
1547 >
1548 > # Joint distribution [X,Y]
1549 > for(i in 1:length(Y)) {
1550 +   joint[,i] <- dbinom(X, 20, p[i]) * dpois(Y[i], lambda)
1551 +
1552 > round(joint,2)
1553   Y=0 Y=1 Y=2 Y=3 Y=4 Y=5 Y=6 Y=7 Y=8 Y=9 Y=10
1554 X=0  0.00 0.08 0.08 0.02  0  0  0  0  0  0  0
1555 X=1  0.00 0.12 0.02 0.00  0  0  0  0  0  0  0
1556 X=2  0.01 0.08 0.00 0.00  0  0  0  0  0  0  0
1557 X=3  0.02 0.04 0.00 0.00  0  0  0  0  0  0  0
1558 X=4  0.04 0.01 0.00 0.00  0  0  0  0  0  0  0
1559 X=5  0.07 0.00 0.00 0.00  0  0  0  0  0  0  0
1560 X=6  0.09 0.00 0.00 0.00  0  0  0  0  0  0  0
1561 X=7  0.10 0.00 0.00 0.00  0  0  0  0  0  0  0
1562 X=8  0.09 0.00 0.00 0.00  0  0  0  0  0  0  0
1563 X=9  0.06 0.00 0.00 0.00  0  0  0  0  0  0  0
1564 X=10 0.04 0.00 0.00 0.00  0  0  0  0  0  0  0
1565 X=11 0.02 0.00 0.00 0.00  0  0  0  0  0  0  0
1566 X=12 0.01 0.00 0.00 0.00  0  0  0  0  0  0  0
1567 X=13 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
1568 X=14 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
1569 X=15 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
1570 X=16 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
1571 X=17 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
1572 X=18 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
1573 X=19 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
1574 X=20 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0

```

1575 This matrix tells us the probability of all possible combinations of  $x$  and  $y$ , and  
 1576 we see that the most likely value is  $(X = 1, Y = 1)$ , i.e. we will catch 1 fish and  
 1577 there will be 1 other fisherman. This matrix also demonstrates the law of total  
 1578 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:

```
1584 > margX <- rowSums(joint)
1585 > round(margX, 2)
1586   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
1587 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
1588 X=15 X=16 X=17 X=18 X=19 X=20
1589 0.00 0.00 0.00 0.00 0.00
```

1590 Bad news—the most likely value is  $X = 0$ . However, the chances of catching 1 fish  
1591 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:

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

1592 That is, the conditional distribution of  $X$  given  $Y$  is the joint distribution divided  
1593 by the marginal distribution of  $Y$ .

```
1594 > XgivenY <- joint/matrix(margY, nrow(joint), ncol(joint), byrow=TRUE)
1595 > round(XgivenY, 2)
1596   Y=0  Y=1  Y=2  Y=3  Y=4  Y=5  Y=6  Y=7  Y=8  Y=9  Y=10
1597   X=0  0.00 0.25 0.82 0.97  1  1  1  1  1  1  1
1598   X=1  0.00 0.36 0.16 0.03  0  0  0  0  0  0  0
1599   X=2  0.01 0.25 0.02 0.00  0  0  0  0  0  0  0
1600   X=3  0.03 0.11 0.00 0.00  0  0  0  0  0  0  0
```

---

1601	X=4	0.07	0.03	0.00	0.00	0	0	0	0	0	0	0
1602	X=5	0.13	0.01	0.00	0.00	0	0	0	0	0	0	0
1603	X=6	0.17	0.00	0.00	0.00	0	0	0	0	0	0	0
1604	X=7	0.18	0.00	0.00	0.00	0	0	0	0	0	0	0
1605	X=8	0.16	0.00	0.00	0.00	0	0	0	0	0	0	0
1606	X=9	0.12	0.00	0.00	0.00	0	0	0	0	0	0	0
1607	X=10	0.07	0.00	0.00	0.00	0	0	0	0	0	0	0
1608	X=11	0.03	0.00	0.00	0.00	0	0	0	0	0	0	0
1609	X=12	0.01	0.00	0.00	0.00	0	0	0	0	0	0	0
1610	X=13	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
1611	X=14	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
1612	X=15	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
1613	X=16	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
1614	X=17	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
1615	X=18	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
1616	X=19	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
1617	X=20	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0

1618 Note that we have 11 probability distributions for  $X$ , one for each possible value of  
 1619  $Y$ , and each pmf sums to unity as it should. Note also that if you show up at the  
 1620 hole and there are  $> 2$  fisherman, your chance of catching a fish is very low. Go  
 1621 home. These concepts are explained in more detail in other texts such as Casella  
 1622 and Berger (2002), Royle and Dorazio (2008), and Link and Barker (2010), but  
 1623 hopefully, the code shown here complements the equations and makes it easier for  
 1624 non-statisticians to understand these concepts.

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

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

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

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

1625 From here on out, when you see such notation, you should immediately grasp  
 1626 the fact that  $Y$  is a random variable independent of  $X$ , but  $X$  depends upon  
 1627  $Y$  through  $p$ . Now you have the tools to make probability statements about the  
 1628 random variables in this system. The one caveat faced in reality is that we typically  
 1629 do not know the values of the parameters, and instead we have to estimate them.  
 1630 Maximum likelihood methods for hierarchical models are covered in Chapt. 6.

## 2.5 HIERARCHICAL MODELS AND INFERENCE

1631 The term hierarchical modeling (or hierarchical model) has become something of  
 1632 a buzzword over the last decade with hundreds of papers published in ecological

1633 journals using that term. So then, what exactly is a hierarchical model, anyhow?  
1634 Obviously, this term stems from the root “hierarchy” which means:

1635 **Definition:** *hierarchy* (noun) – a series of ordered groupings of people or things  
1636 within a system;

1637 In the case of a hierarchical model (hierarchical being the adjective form of hi-  
1638 erarchy), the “things” are probability distributions, and they are ordered according  
1639 to their conditional probability structure. Thus, a hierarchical model is *an ordered*  
1640 *series of models, ordered by their conditional probability structure.*

1641 A canonical hierarchical model in ecology is this elemental model of species  
1642 occurrence or distribution (MacKenzie et al., 2002; Tyre et al., 2003; Kéry, 2011):

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

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

1643 where  $y_i$  = observation of presence/absence at a site  $i$  and  $z_i$  = occurrence status  
1644 ( $z_i = 1$  if a species occurs at site  $i$  and  $z_i = 0$  if not). Note that if  $p = 1$ , then we  
1645 would perfectly observe  $z$  and the model would no longer be hierarchical—it would  
1646 be a simple logistic regression model. Note also that this hierarchical model has an  
1647 important conceptual distinction between other types of classical multi-level models  
1648 such as repeated measures on subjects, in that  $z_i$  is an actual state of nature. In  
1649 that sense,  $z$  is a random variable that is the outcome of a “real” process. Royle  
1650 and Dorazio (2008) used the term *explicit* hierarchical model to describe this type of  
1651 model to distinguish from hierarchical models (*implicit* hierarchical models) where  
1652 the latent variables don’t correspond to an actual state of nature—but rather just  
1653 soak up variation that is unmodeled by explicit elements of the model. At best,  
1654 latent variables in such models are surrogates for something of ecological relevance  
1655 (“time effects”, “space effects” etc.).

1656 With these examples, we expand on our definition of a hierarchical model as we  
1657 will use it in this book:

1658 **Definition:** *Hierarchical Model:* A model with explicit component models that de-  
1659 scribe variation in the data due to (spatial/temporal) variation in *ecological process*,  
1660 and due to *imperfect observation* of the process.

1661 Most models considered in this book describe the encounter of individuals con-  
1662 ditional on the “activity center” of the individual, which is a latent variable (i.e.,  
1663 unobserved random effect). The definition of an activity center will be context-  
1664 dependent as discussed in Chapt. 5, but often it can be thought of as an individual’s  
1665 home range center. The collection of these latent variables represents the outcome  
1666 of an ecological process describing how individuals distribute themselves over the  
1667 landscape. Moreover, how individuals are encountered in traps is, in some cases,

1669 the result of a model governing movement. As such, these models are examples of  
 1670 hierarchical models that contain formal model components representing both eco-  
 1671 logical process and also the observation of that process. That is, they are explicit  
 1672 hierarchical models (Royle and Dorazio, 2008) as opposed to implicit hierarchical  
 1673 models.

## 2.6 CHARACTERIZATION OF SCR MODELS

1674 For the purposes of this book, an SCR model is any “individual encounter model”  
 1675 (not just “capture-recapture”!) where auxiliary spatial information is also obtained.  
 1676 To be more precise we could as well use the term “spatial capture and/or recap-  
 1677 ture” but that is slightly unwieldy and, besides, it also abbreviates to SCR. The  
 1678 class of SCR models includes traditional capture-recapture models with auxiliary  
 1679 spatial information and even some models that do not even require “recapture”  
 1680 (e.g., distance sampling). There is even a class of models (Chapt. 18) which don’t  
 1681 require capture or unique identification of individuals.

1682 Conceptually, SCR models involve a collection of random variables,  $\mathbf{s}$ ,  $\mathbf{u}$  and  
 1683  $y$  where  $\mathbf{s}$  is the activity center, or home range center,  $\mathbf{u}$  is the location of the  
 1684 individual at the time of sampling, which we may think of as a realization from some  
 1685 movement model, and  $y$  is the “response variable”—what the observer records. For  
 1686 example,  $y = 1$  means “detected” and  $y = 0$  means “not detected”, but many other  
 1687 types of responses are possible (Chapt 9). A broad class of models for estimating  
 1688 density are unified by a hierarchical model involving explicit models for animal  
 1689 activity centers  $\mathbf{s}$ , movement outcomes  $\mathbf{u}$ , and encounter data  $y$ . In some cases, we  
 1690 don’t observe  $y$  but rather summaries of  $y$ , say  $n(y)$ , yet it might be convenient  
 1691 in such cases to retain an explicit focus on  $y$  in terms of model construction. We  
 1692 thus introduce a sequence of models—a hierarchical model—to relate these random  
 1693 variables, which can be written as

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

1694 Every model we talk about in this book has a subset of these components although  
 1695 we never fit the full model because we have not encountered a situation requiring  
 1696 that we do so. However, a detailed description of this model and its various com-  
 1697 ponents is the subject of this book, and we will not pretend to condense hundreds  
 1698 of pages of material into the next few paragraphs. However, we give a cursory  
 1699 overview here to whet the appetite and provide some indication of where we are  
 1700 going. Don’t worry if some of this material doesn’t sink in just yet—we will walk  
 1701 through it slowly in the subsequent chapters.

1702 Let’s begin with the model  $[\mathbf{s}]$  that describes the distribution of the activity  
 1703 centers of each animal in the spatial region  $\mathcal{S}$  (the state-space as we called it previ-  
 1704 ously). As will be explained in Chapt. 5 and Chapt. 11,  $[\mathbf{s}]$  defines a spatial point  
 1705 process, which may be inhomogeneous if there exists spatial variation in density, or

1706 it may be homogeneous if density is constant throughout  $\mathcal{S}$ . In the later case, we can  
 1707 write  $[\mathbf{s}] = \text{Uniform}(\mathcal{S})$ , which is to say that the  $N$  activity centers are uniformly  
 1708 distributed in the polygon  $\mathcal{S}$ . A point process is also a model for the number of indi-  
 1709 viduals in the population  $N$ . So we could write  $[\mathbf{s}|\mu]$  where  $\mu$  is an intensity param-  
 1710 eter defined as the number of points per unit area. In other words,  $\mu$  is population  
 1711 density, and we often model population size as either  $N \sim \text{Poisson}(\mu A(\mathcal{S}))$ , where  
 1712  $A(\mathcal{S})$  is the area of the state-space; or,  $N \sim \text{Binomial}(M, \psi)$  where  $\psi = \mu A(\mathcal{S})/M$   
 1713 and  $M$  is some large integer used simply as a convenience measure when conducting  
 1714 Bayesian analysis. As it turns out, there is very little practical difference in the  
 1715 Poisson prior versus a binomial models for  $N$  (Chapt. 11).

1716 The model  $[\mathbf{u}|\mathbf{s}]$  describes the locations of animals conditional on their activity  
 1717 center. In the original formulation of SCR models (Efford, 2004), this model com-  
 1718 ponent was intentionally ignored. Indeed when movement is not of direct interest,  
 1719 or when  $\mathbf{s}$  is defined in a way not related to a home range center, it may be prefer-  
 1720 able to ignore this model component (Borchers, 2012). In other cases, we might use  
 1721 an explicit model, such as the bivariate normal model (Royle and Young, 2008).

1722 The third component of the model,  $[y|\mathbf{u}]$ , describes how the observed data—the  
 1723 so-called capture-histories—arise conditional on the locations of animals. However,  
 1724 as mentioned previously, most SCR models do not contain a movement model, and  
 1725 thus, we typically entertain the model  $[y|\mathbf{s}]$  instead of  $[y|\mathbf{u}]$ . This encounter model  
 1726 generally has at least two parameters, say  $p_0$  and  $\sigma$ , describing the probability of  
 1727 capturing or detecting an individual given the distance between  $\mathbf{s}$  and the trap.  
 1728 The most basic model is often called the half-normal model, although we typically  
 1729 refer to it as the Gaussian model since, in two-dimensional space, it is the kernel  
 1730 of a bivariate normal distribution. The model is  $p_{ij} = p_0 \exp(-\|\mathbf{x}_j - \mathbf{s}_i\|/(2\sigma^2))$   
 1731 where  $p_0$  is the capture probability when the activity center occurs at the trap  
 1732 location  $\mathbf{x}_j$ , and  $\sigma$  is a spatial scale parameter determining how rapidly capture  
 1733 probability declines with distance. One common design leads to the model  $[y_{ij}|\mathbf{s}_i] =$   
 1734 Bernoulli( $p_{ij}$ ). Chapt. 5 and Chapt. 9 describe many other possible encounter  
 1735 models.

1736 When individuals are marked by biologists or have natural markings permit-  
 1737 ting individual recognition,  $y_{ij}$  is the observed data. However, some or all of the  
 1738 individuals cannot be uniquely identified, then we cannot record this individual-  
 1739 specific encounter history data. Instead, the data might be simply the number of  
 1740 detections at a trap or perhaps binary detection/non-detection data at each trap on  
 1741 each survey occasion. We call this reduced information data  $n(y)$ , and Chapt. 18  
 1742 and Chapt. 19 describe models for  $[n(y)|y]$  that still allow for density estimation.  
 1743 The basic strategy is to view  $y$  as “missing data” and to use the spatial correlation  
 1744 in the counts, or other sources of information, to provide information about these  
 1745 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:

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

1746 These “assumptions” are statistical statements of three basic hypotheses that (1)  
 1747 population size  $N$  is Poisson distributed (2) activity centers are uniformly dis-  
 1748 tributed in two-dimensional space, and (3) capture probability is a function of the  
 1749 distance between the activity and the trap. Each of these model components can  
 1750 be modified as needed to match specific hypotheses, study designs, and data struc-  
 1751 tures. For example, spatial variation in abundance or density can be easily modeled  
 1752 as a function of habitat covariates (Chapt. 11).

1753 We realize that many the model description in Eq. 2.6.2 may not be self-evident  
 1754 to some ecologists. However, it is absolutely essential that one can understand  
 1755 such a model description—not just for being able to read this book, but also for  
 1756 understanding any statistical model in ecology. One of the best ways of familiarizing  
 1757 oneself with this notation is to translate it into **R** code that simulates outcomes  
 1758 from the model. The following code is an example.

```
1759 set.seed(36372)
1760 Area <- 1 # area of state-space (unit square)
1761 x <- cbind(rep(seq(.1,.9,.2), each=5), # trap locations
1762             rep(seq(.1,.9,.2), times=5))
1763 p0 <- 0.3 # baseline capture probability
1764 sigma <- 0.05 # Gaussian scale parameter
1765 mu <- 50 # population density
1766 N <- rpois(1, mu*Area) # population size
1767 s <- cbind(runif(N, 0, 1), # activity centers in unit square
1768             runif(N, 0, 1))
1769 K <- 5
1770 y <- matrix(NA, N, nrow(x)) # capture data
1771 for(i in 1:N) {
1772   d.ij <- sqrt((x[,1] - s[i,1])^2 + # distance between x and s[i]
1773                 (x[,2] - s[i,2])^2)
1774   p.ij <- p0*exp(-d.ij^2 / (2*sigma^2)) # capture probability
1775   y[i,] <- rbinom(nrow(x), K, p.ij) # capture history for animal i
1776 }
```

1777 Fig. 2.3 shows the results of this simulation from a basic, yet very useful, SCR  
 1778 model.

1779 Having briefly explained each of the model components in Eq. 2.6.1, and having  
 1780 shown how a subset of these components results in a basic SCR model, we can  
 1781 now discuss other relevant arrangements. Examples include: (1) Classical distance  
 1782 sampling (Buckland et al., 2001; Borchers et al., 2002), (2) Spatial capture-recapture  
 1783 models with fixed arrays of traps (Efford, 2004; Borchers and Efford, 2008; Royle

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

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. 13 and 12).

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. These models are described in Chapt. chapt.search-encounter.

## 2.7 SUMMARY AND OUTLOOK

1813 Spatial capture-recapture models are hierarchical models, and hierarchical models  
1814 are models of multiple random variables that are conditionally related. It is there-  
1815 fore important that the basic rules of modeling random variables are understood,  
1816 and we hope that this chapter has made some of the basic concepts accessible to  
1817 ecologists with rudimentary background in statistics. If some of this material still  
1818 seems difficult to grasp, we recommend working with the provided **R** code, which  
1819 is perhaps the best way of making the equations more tangible.

1820 In some respects, it is possible to understand the jist of SCR without knowing  
1821 anything about marginal and conditional relationships. One can always fit models  
1822 using canned software and interpret the output without understanding the guts of  
1823 the model or the details of the estimation process. For some applied ecologists,  
1824 this may be perfectly fine, and this book is meant to be useful for both statistical  
1825 novices and ecologists with more advanced quantitative skills. In most chapters, we  
1826 begin with a basic conceptual discussion, then we explain the technical details that  
1827 require an understanding of the concepts in this chapter, and finally we end with  
1828 one or more worked examples. For those not interested in the technical details,  
1829 we recommend focusing on the chapter introductions and the examples. However,  
1830 taking the time to understand the concepts presented in this chapter can only  
1831 increase one's ability to tackle the unique and complex problems that often present  
1832 themselves when modeling spatial and temporal aspects of population dynamics.

1833  
1834

---

1835

# 3

## CLOSED POPULATION MODELS

1836 In this chapter we introduce ordinary *non-spatial* capture-recapture (CR) models  
1837 for estimating population size in closed populations. A closed population is one  
1838 whose size,  $N$ , does not change during the study. Two forms of closure are often  
1839 discussed: demographic closure, meaning that no births or deaths occur, and geo-  
1840 graphic closure, which states that no individuals move onto or off of the sampled  
1841 area during the study. Although few populations are actually closed except during  
1842 very short time intervals, closed population CR models serve as the basis for the  
1843 development of the rest of the models presented in this book, including the models  
1844 for open populations discussed in Chapt. 16.

1845 We begin with the most basic capture-recapture model, colloquially referred  
1846 to as “model  $M_0$ ” (Otis et al., 1978), in which encounter probability is strictly  
1847 constant in all respects (across individuals, and replicates). This allows us to high-  
1848 light the basic structure of closed population models as binomial GLMs. We then  
1849 consider some important extensions of ordinary closed population models that ac-  
1850 commodate various types of “individual effects” — either in the form of explicit,  
1851 observed covariates (sex, age, body mass) or unstructured “heterogeneity” in the  
1852 form of an individual random effect, which represent unobserved or unmeasured co-  
1853 variates. A special type of individual covariate models is distance sampling, which  
1854 could be thought of as the most primitive spatial capture-recapture model. All of  
1855 these different types of closed population models are closely related to binomial  
1856 (or logistic) regression-type models. In fact, when  $N$  is known, they are precisely  
1857 logistic regression models.

1858 We emphasize Bayesian analysis of capture-recapture models and we accomplish  
1859 this using a method related to classical “data augmentation” from the statistics  
1860 literature (e.g., Tanner and Wong, 1987). This is a general concept in statistics  
1861 but, in the context of capture-recapture models where  $N$  is unknown, it has a

1862 consistent implementation across classes of capture-recapture models and one that  
 1863 is really convenient from the standpoint of doing MCMC (Royle et al., 2007; Royle  
 1864 and Dorazio, 2012). We use data augmentation throughout this book and thus  
 1865 emphasize its conceptual and technical origins and demonstrate applications to  
 1866 closed population models. We refer the reader to Kéry and Schaub (2012, ch. 6)  
 1867 for an accessible and complementary development of Bayesian analysis of ordinary,  
 1868 i.e., nonspatial closed population models.

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

1869 To start looking at the simplest capture-recapture model, let's suppose there exists  
 1870 a population of  $N$  individuals which we subject to repeated sampling, say over  $K$   
 1871 "occasions", such as trap nights, where individuals are captured, marked, released,  
 1872 and subsequently recaptured. We suppose that individual encounter histories are  
 1873 obtained, and these are of the form of a sequence of 0's and 1's indicating capture  
 1874 ( $y = 1$ ) or not ( $y = 0$ ) during any sampling occasion. As an example, suppose  $K = 5$   
 1875 sampling occasions, then an individual captured during occasion 2 and 3 but not  
 1876 otherwise would have an encounter history of the form  $\mathbf{y} = (0, 1, 1, 0, 0)$ . Thus,  
 1877 the observation  $\mathbf{y}_i$  for each individual ( $i = 1, 2, \dots, N$ ) is a vector having elements  
 1878 denoted by  $y_{ik}$  for  $k = 1, 2, \dots, K$ . Usually this is organized as a row of a matrix  
 1879 with elements  $y_{ik}$ , see Table 4.1. Except where noted explicitly, we suppose that  
 1880 observations are independent within individuals and among individuals. Formally,  
 1881 this allows us to say that  $y_{ik}$  are independent and identically distributed ("iid")  
 1882 Bernoulli random variables and we may write  $y_{ik} \sim \text{Bernoulli}(p)$ . Consequently,  
 1883 for this very simple model in which  $p$  is constant (i.e., there are no individual or  
 1884 temporal covariates that affect  $p$ ) the original binary detection variables can be  
 1885 aggregated into the total number of encounters for each individual<sup>1</sup>,  $y_i = \sum_k y_{ik}$ ,  
 1886 and the observation model changes from a Bernoulli distribution to a binomial  
 1887 distribution based on a sample of size  $K$ . That is

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

1888 for every individual in the population  $i = 1, 2, \dots, N$ , where  $N$  is the number of  
 1889 individuals in the population (i.e., population size).

1890 We emphasize the central importance of the basic Bernoulli encounter model –  
 1891 an individual is either encountered in a sample, or not – which forms the cornerstone  
 1892 of almost all of classical capture-recapture models, including many spatial capture-  
 1893 recapture models discussed in this book.

1894 Evidently, the basic capture-recapture model is a simplistic version of a logistic-  
 1895 regression model with only an intercept term ( $\text{logit}(p) = \text{constant}$ ). To say that all

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

1896 capture-recapture models are just logistic regressions is a slight over-simplification.  
1897 In fact, we are proceeding here as if we knew  $N$ . In practice we don't, of course,  
1898 and estimating  $N$  is actually the central objective. But, by proceeding as if  $N$   
1899 were known, we can specify a simple model and then deal with the fact that  $N$  is  
1900 unknown using standard methods that you are already familiar with (i.e., GLMs -  
 see Chapt. 3).

**Table 3.1.** 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

1901  
1902 Assuming individuals in the population are encountered independently, the joint  
1903 probability distribution of the observations is the product of  $N$  binomials

$$\Pr(y_1, \dots, y_N | p) = \prod_{i=1}^N \text{Binomial}(y_i | K, p). \quad (3.1.1)$$

1904 We emphasize that this expression is conditional on  $N$ , in which case we get to  
1905 observe the  $y_i = 0$  observations and the resulting data are just *iid* binomial counts.  
1906 Because this is a binomial regression model of the variety described in Chapt. 3,  
1907 fitting this model using a **BUGS** engine poses no difficulty.

1908 Equation 4.1.1 can be simplified even further if we reformat the observations  
1909 as encounter frequencies. Specifically, let  $n_k$  denote the number of individuals  
1910 captured exactly  $k$  times after  $K$  survey occasions,  $n_k = \sum_{i=1}^N I(y_i = k)$  where  $I()$   
1911 is the indicator function evaluating to 1 if its argument is true and 0 otherwise. For  
1912 sake of illustration, we converted the data from Table 4.1 to this format (Table 4.2).  
1913 What is important to note is that if we know  $N$ , then we know  $n_0$ , i.e. the number  
1914 of individuals not captured. In this case, an alternative and equivalent expression  
1915 to Eq. 4.1.1 is

$$\Pr(y_1, \dots, y_N | p) = \prod_{k=0}^K \pi_k^{n_k} \quad (3.1.2)$$

1916 where  $\pi_k = \Pr(y = k)$  under the binomial model with parameter  $p$  and sample size  
1917  $K$ . The essential problem in capture-recapture, however, is that  $N$  is *not* known

**Table 3.2.** Data from Table 4.1 formatted 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	2	3	1	0

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

$$n_0, n_1, \dots, n_K \sim \text{Multinomial}(N, \pi_0, \pi_1, \dots, \pi_K) \quad (3.1.3)$$

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

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

### 3.1.1 The core capture-recapture assumptions

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

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

The remainder of their classic work is dedicated to relaxing assumption 3. While assumptions 1 and 2 are undoubtedly necessary for inference from basic CR methods

1949 to be valid, and while they are also assumed by most of the models we present in  
 1950 the following chapters, we refrain from repeatedly making such statements. Our  
 1951 opinion is that all model assumptions are apparent when a model is clearly specified,  
 1952 and it is both redundant and impossible to list all the things not allowed by the  
 1953 model. For example, closed population models also assume that other sources of  
 1954 error do not occur, but it is not necessary to enumerate each possibility. Rather, it  
 1955 is necessary to make clear statements such as

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

1956 This simple model description carries a tremendous amount of information, and  
 1957 it leaves very little left to say with respect to assumptions. Although we will not  
 1958 always show the *iid* symbol, it will be assumed unless otherwise noted, and this  
 1959 assumption is critical for valid inference. It implies that the encounter of one indi-  
 1960 vidual does not affect the encounter of another individual, and encounter does not  
 1961 affect future encounter. Under this assumption, it is easy to write down the likeli-  
 1962 hood of the parameters and obtain parameter estimates; however, whether or not it  
 1963 is true depends upon biological and sampling issues. If this assumption is deemed  
 1964 false, the model can be discarded in favor of a more realistic alternative. However,  
 1965 once we have settled on our model, statistical inference proceeds by assuming the  
 1966 model is truth—not an approximation to truth—but actual truth.

1967 In spite of the fact that we assume that all models are truth, but we acknowledge  
 1968 that all models are wrong due to their assumptions, assumptions should not be  
 1969 viewed as a necessary evil. In fact, one way to view assumptions is as embodiments  
 1970 of our ecological hypotheses. If we make these assumptions too complex or too  
 1971 specific, then we will never be able to study general phenomena that hold true  
 1972 across space and time. Furthermore, in practice, we will rarely have enough data  
 1973 to estimate the parameters of highly complex models.

### 1974 3.1.2 Conditional likelihood

1975 We saw that the closed population model is a simple logistic regression model if  $N$   
 1976 is known and, when  $N$  is unknown, the model is multinomial with index or sample  
 1977 size parameter  $N$ . This multinomial model, being conditional on  $N$ , is sometimes  
 1978 referred to as the “joint likelihood” the “full likelihood” or the “unconditional like-  
 1979 lihood” (sometimes “model” in place of “likelihood”) (Sanathanan, 1972; Borchers  
 1980 et al., 2002). This formulation differs from the so-called “conditional likelihood”  
 1981 approach in which the likelihood of the observed encounter histories is devised con-  
 1982 ditional on the event that an individual is captured at least once. To construct this  
 1983 likelihood, we have to recognize that individuals appear or not in the sample based  
 1984 on the value of the random variable  $y_i$ , that is, if and only if  $y_i > 0$ . The obser-  
 1985 vation model is therefore based on  $\Pr(y|y > 0)$ . For the simple case of model  $M_0$ ,  
 1986 the resulting conditional distribution is a “zero truncated” binomial distribution

which accounts for the fact that we cannot observe the value  $y = 0$  in the data set. Both the conditional and unconditional models are legitimate modes of analysis in all capture-recapture types of studies. They provide equally valid descriptions of the data and, for many practical purposes provide equivalent inferences, at least in large sample sizes (Sanathanan, 1972).

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

**Table 3.3.** 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

## 3.2 DATA AUGMENTATION

We consider a method of analyzing closed population models using parameter-expanded data augmentation (PX-DA), which we abbreviate to “data augmentation” or DA, which 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, 2009), open population models (Royle and Dorazio, 2008, 2012; Gardner et al., 2010a), spatial capture-recapture models (Royle and Young, 2008; Royle et al., 2009a; Gardner et al., 2009), and many others. Kéry and Schaub (2012, Chaps. 6 and 10) provide a good introduction to data augmentation in the context of closed and open 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  $M^2$ . 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.”

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

There is a close correspondence between so-called “occupancy” models and closed population models (see Royle and Dorazio, 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 for the site, which in the case of occupancy studies, are *observed*. However, some of the zero vectors will typically correspond to sites where the species in fact *does* occur. Thus, while the zeros are observed, there are too many of them and, in a sense, the inference problem is to partition the zeros into “structural” (fixed) and “sampling” (or stochastic) zeros, where the former are associated with unoccupied sites and the latter with occupied sites where the species went undetected. More formally, inference is focused on the parameter  $\psi$ , the probability that a site is occupied.

In contrast to occupancy studies, in classical closed population studies, we observe a data set as in the middle panel of Table 4.4 where *no* zeros are observed. The inference problem is, essentially, to estimate how many sampling zeros there are – or should be – in a “complete” data set. This objective (how many sampling zeros?) is precisely the same for both types of problems if an upper limit  $M$  is specified for the closed population model. The only distinction being that, in occupancy models,  $M$  is set by design (i.e., the number of sites in the sample), whereas a natural choice of  $M$  for capture-recapture models may not be obvious. However, the choice of  $M$  induces a uniform prior for  $N$  on the integers  $[0, M]$  (Royle et al.,

---

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

2055 2007). Then, one can analyze capture-recapture models by adding  $M - n$  all-zero  
 2056 encounter histories to the data set and regarding the augmented data set, essen-  
 2057 tially, as a site-occupancy data set, where the occupancy or data augmentation  
 2058 parameter ( $\psi$ ) takes the place of the abundance parameter ( $N$ ).

2059 Thus, the heuristic motivation of data augmentation is to fix the size of the  
 2060 data set by adding *too many* all-zero encounter histories to create the data set  
 2061 shown in the right panel of Table 4.4, and then analyze the augmented data set  
 2062 using an occupancy type model which includes both “unoccupied sites” (in capture-  
 2063 recapture, augmented individuals that are not members of the real population that  
 2064 was sampled) as well as “occupied sites” (in capture-recapture, individuals that are  
 2065 members of the population but that were undetected by sampling) at which detec-  
 2066 tions did not occur. We call these  $M - n$  all-zero histories “potential individuals”  
 2067 because they exist to be recruited (in a non-biological sense) into the population,  
 2068 for example during an analysis by MCMC.

2069 To analyze the augmented data set, we recognize that it is a zero-inflated ver-  
 2070 sion of the known- $N$  data set. That is, some of the augmented all-zero rows are  
 2071 sampling zeros (corresponding to actual individuals that were missed) and some are  
 2072 “structural” zeros, which do not correspond to individuals in the population. For  
 2073 a basic closed-population model, the resulting likelihood under data augmentation  
 2074 - that is, for the data set of size  $M$  - is a simple zero-inflated binomial likelihood.  
 2075 The zero-inflated binomial model can be described “hierarchically”, by introducing  
 2076 a set of binary latent variables,  $z_1, z_2, \dots, z_M$ , to indicate whether each individual  
 2077  $i$  is ( $z_i = 1$ ) or is not ( $z_i = 0$ ) a member of the population of  $N$  individuals exposed  
 2078 to sampling. We assume that  $z_i \sim \text{Bernoulli}(\psi)$  where  $\psi$  is the probability that  
 2079 an individual in the data set of size  $M$  is a member of the sampled population –  
 2080 in the sense that  $1 - \psi$  is the probability of a “structural zero” in the augmented  
 2081 data set. The zero-inflated binomial model which arises under data augmentation  
 2082 can be formally expressed by the following set of assumptions (we include typical  
 2083 priors for a Bayesian analysis):

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

2084 for  $i = 1, \dots, M$ , where  $I(y = 0)$  is a point mass at  $y = 0$ . It is sometimes  
 2085 convenient to express the conditional-on- $z$  observation model concisely in just one  
 2086 step:

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

2087 and we understand this to mean, if  $z_i = 0$ , then  $y_i$  is necessarily 0 because its  
 2088 success probability is  $z_i p = 0$ .

2089 Note that, under data augmentation,  $N$  is no longer an explicit parameter of  
 2090 this model. In its place, we estimate  $\psi$  and functions of the latent variables  $z$ . In  
 2091 particular, under the assumptions of the zero-inflated model,  $z_i \stackrel{iid}{\sim} \text{Bernoulli}(\psi)$ ;  
 2092 therefore,  $N$  is a function of these latent variables:

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

2093 Further, we note that the latent  $z_i$  parameters *can be* removed from the model by  
 2094 integration, in which case the joint probability of the data is

$$\Pr(y_1, \dots, y_M | p, \psi) = \prod_{i=1}^M (\psi * \text{Binomial}(y_i | K, p) + I(y_i = 0)(1 - \psi)) \quad (3.2.1)$$

2095 Interpreted as a likelihood, we can directly maximize this expression to obtain the  
 2096 MLEs of the structural parameters  $\psi$  and  $p$  or those of other more complex models  
 2097 (e.g., see Royle, 2006). We could estimate these parameters and then use them  
 2098 to obtain an estimator of  $N$  using the so-called “Best unbiased predictor” (see  
 2099 Royle and Dorazio, 2012). Normally, however, we will analyze the model in its  
 2100 “conditional-on- $z$ ” form using methods of MCMC either in the **BUGS** engines or  
 2101 using our own MCMC algorithms (see Chapt. 17).

### 2102 3.2.2 Model $M_0$ in **BUGS**

2103 It is helpful to understand data augmentation by seeing what its effect is on imple-  
 2104 menting model  $M_0$ . For this model, in which we can aggregate the encounter data  
 2105 to individual-specific encounter frequencies, the augmented data are given by the  
 2106 vector of frequencies  $(y_1, \dots, y_n, 0, 0, \dots, 0)$  where the augmented values of  $y = 0$   
 2107 represent the encounter frequency for potential individuals  $y_{n+1}, \dots, y_M$ . The zero-  
 2108 inflated model of the augmented data combines the model of the latent variables,  
 2109  $z_i \sim \text{Bernoulli}(\psi)$ . The **BUGS** model description of the closed population model  
 2110  $M_0$  is shown in Panel 4.1. The last line of the model specification provides the  
 2111 expression for computing  $N$  from the data augmentation variables  $z_i$ . Note that,  
 2112 to improve readability of code snippets (especially of large ones), we will sometimes  
 2113 deviate from our standard notation a bit. In this case we use `nind` for  $n$  (the number  
 2114 of encountered individuals), and  $M = nind + nz$  is the total size of the augmented  
 2115 data set. In other cases we might also use `nocc` in place of  $K$  and `ntraps` in place  
 2116 of  $J$ . We find that word definitions make code easier to understand, especially  
 2117 without having to read surrounding text.

2118 Specification of a more general model in terms of the individual encounter obser-  
 2119 vations  $y_{ik}$  is not much more difficult than for the individual encounter frequencies.  
 2120 We define the observation model by a double loop and change the indexing of  
 2121 quantities accordingly, i.e.,

**Table 3.4.** Hypothetical occupancy data set (left), capture-recapture data in standard form (center), and capture-recapture data augmented with all-zero capture histories (right).

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

```

2122 for(i in 1:(nind+nz)){
2123   z[i] ~ dbern(psi)
2124   for(k in 1:K){
2125     mu[i,k] <- z[i]*p
2126     y[i,k] ~ dbin(mu[i,k],1)
2127   }
2128 }
```

2129 In this manner, it is straightforward to incorporate covariates on  $p$  for both  
2130 individuals and sampling occasions (see discussion of this below and also Chapt.  
2131 7) as well as to devise other extensions of the model, including models for open  
2132 populations (see Chapt. 16).

### 2133 3.2.3 Formal development of data augmentation (DA)

2134 Use of parameter-expanded data augmentation (PX-DA), or DA for short, for solv-  
2135 ing inference problems with unknown  $N$  can be justified as originating from the  
2136 choice of a uniform prior on  $N$ . The Uniform(0,  $M$ ) prior for  $N$  is innocuous in  
2137 the sense that the posterior associated with this prior is equal to the likelihood

---

```

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

2138 for sufficiently large  $M$ . One way of inducing the  $\text{Uniform}(0, M)$  prior on  $N$  is by  
 2139 assuming the following hierarchical prior:

$$\begin{aligned} N &\sim \text{Binomial}(M, \psi) \\ \psi &\sim \text{Uniform}(0, 1). \end{aligned} \tag{3.2.2}$$

2140 The model assumptions, specifically the multinomial model (Eq. 4.1.3) and Eq.  
 2141 4.2.2, may be combined to yield a reparameterization of the conventional model  
 2142 that is appropriate for the augmented data set of known size  $M$ :

$$(n_1, n_2, \dots, n_K) \sim \text{Multinomial}(M, \psi\pi_1, \psi\pi_2, \dots, \psi\pi_K) \tag{3.2.3}$$

2143 This expression arises by removing  $N$  from Eq. 4.1.3 by integrating over the bi-  
 2144 nomial prior distribution for  $N$ . Thus, the models we analyze under data aug-  
 2145mentation arise formally by removing the parameter  $N$  from the ordinary closed-  
 2146 population model, which is conditional on  $N$ , by integrating over a binomial prior  
 2147 distribution for  $N$ .

2148 Note that the  $M - n$  unobserved individuals in the augmented data set have  
 2149 probability  $\psi\pi(0) + (1 - \psi)$ , indicating that these unobserved individuals are a  
 2150 mixture of individuals that are sampling zeros ( $\psi\pi_0$ ), and belong to the population  
 2151 of size  $N$ , and others that are “structural zeros” (occurring in the augmented data  
 2152 set with probability  $1 - \psi$ ). In Eq. 4.2.3,  $N$  has been eliminated as a formal

2153 parameter of the model by marginalization (integration) and replaced with the  
2154 new parameter  $\psi$ , the data augmentation parameter. However, the full likelihood  
2155 containing both  $N$  and  $\psi$  can also be analyzed (see Royle et al., 2007).

2156 **3.2.4 Remarks on data augmentation**

2157 Data augmentation may seem like a strange and mysterious black-box, and likely  
2158 it is unfamiliar to most people, even to many of those with substantial experience  
2159 with capture-recapture models. However, it really is just a formal reparameteriza-  
2160 tion of capture-recapture models in which  $N$  is marginalized out of the ordinary  
2161 (conditional-on- $N$ ) model (by summation over a binomial prior). As a result, we  
2162 could refer to the resulting model as the “binomial-integrated likelihood” to reflect  
2163 that an estimator could be obtained from the ordinary likelihood, integrated over  
2164 a binomial prior. Other such “integrated likelihood” models are sensible. For ex-  
2165 ample, we could place a Poisson prior on  $N$  with mean  $\Lambda$  and marginalize  $N$  over  
2166 the Poisson prior. This produces a likelihood in which  $\Lambda$  replaces  $N$ , instead of  $\psi$   
2167 replacing  $N$ . We note that this type of marginalization (over a Poisson prior) is  
2168 done by the **R** package **secr** for analysis of spatial capture-recapture models (see  
2169 Sec. 6.5.3).

2170 We emphasize the motivation for data augmentation being that it produces a  
2171 data set of fixed size, so that the parameter dimension in any capture-recapture  
2172 model is also fixed. As a result, MCMC is a relatively simple proposition using  
2173 standard Gibbs Sampling. And, in particular, capture-recapture models become  
2174 trivial to implement in **BUGS**. Consider the simplest context—analyzing model  
2175  $M_0$  using the occupancy-type model. In this case, DA converts model  $M_0$  to a  
2176 basic occupancy model, and the parameters  $p$  and  $\psi$  have known full-conditional  
2177 distributions (in fact, beta distributions) that can be sampled from directly. Fur-  
2178 thermore, the data augmentation variables, i.e., the collection of  $z$ ’s, can be sampled  
2179 from Bernoulli full conditionals. MCMC is not much more difficult for complicated  
2180 models—sometimes the hyperparameters need to be sampled using a Metropolis-  
2181 Hastings step (e.g., Chapt. 17), but nothing more sophisticated than that is re-  
2182 quired.

2183 Potential sensitivity of parameter estimates to  $M$  (especially of  $N$ ) might be  
2184 cause for some concern. The guiding principle is that it should be chosen large  
2185 enough so that the posterior for  $N$  is not truncated, but it should not be too large  
2186 due to the increased computational burden. It seems likely that the properties of  
2187 the Markov chains should be affected by  $M$  and so some optimal choice of  $M$  might  
2188 exist (Gopalaswamy, 2012). Formal analysis of this is needed.

2189 There are other approaches to analyzing models with unknown  $N$ , using re-  
2190 versible jump MCMC (RJMCMC) or other so-called “trans-dimensional” (TD)  
2191 algorithms (King and Brooks, 2001; Durban and Elston, 2005; King et al., 2008;  
2192 Schofield and Barker, 2008; Wright et al., 2009). What distinguishes DA from RJM-  
2193 CMC and related TD methods is that DA is used to create a distinctly new model

that is unconditional on  $N$  and we (usually) analyze the unconditional model. The various TD/RJMCMC approaches seek to analyze the conditional-on- $N$  model in which the dimension of the parameter space is a function of  $N$ , and will therefore typically vary at each iteration of the MCMC algorithm. TD/RJMCMC approaches might appear to have the advantage that one can model  $N$  explicitly or consider alternative priors for  $N$ . However, despite that  $N$  is removed as an explicit parameter in DA, it is possible to develop hierarchical models that involve structure on  $N$  (Converse and Royle, 2012; Royle et al., 2012c; Royle and Converse, in review) which we consider in Chapt. 14. Furthermore, data augmentation is often easier to implement than RJMCMC, and the details of the DA implementation are the same for all capture-recapture problems.

### 3.2.5 Example: Black bear study on Fort Drum

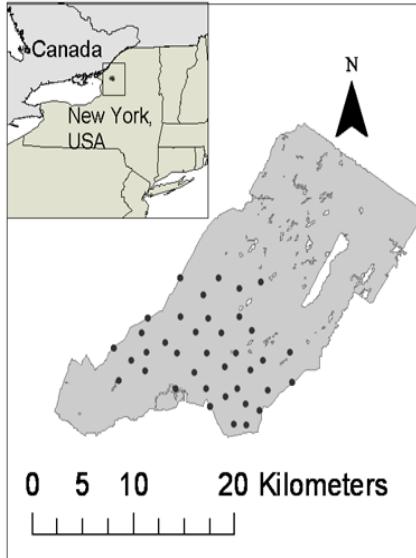
To illustrate the analysis of model  $M_0$  using data augmentation, we use a data set collected at Fort Drum Military Installation in upstate New York by P.D. Curtis and M.T Wegan of Cornell University and their colleagues at the Fort Drum Military Installation. These data have been analyzed in various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

```

2219 > library(scrbook)
2220 > data(beardata)          # load the bear data and extract components
2221 > trapmat <- beardata$trapmat
2222 > nind <- dim(beardata$bearArray)[1]
2223 > K <- dim(beardata$bearArray)[3]
2224 > ntraps <- dim(beardata$bearArray)[2]
2225
2226 > M <- 175
2227 > nz <- M-nind
2228 > Yaug <- array(0, dim=c(M,ntraps,K))
2229
2230 > Yaug[1:nind,,] <- Beardata$bearArray
2231 > y <- apply(Yaug,c(1,3),sum) # summarize by ind x rep
2232 > y[y>1] <- 1               # toss out multiple encounters per occasion
2233                               # b/c traditional CR models ignore space

```

2234 The raw data object, `beardata$bearArray` is a 3-dimensional array `nind ×`



**Figure 3.1.** Fort Drum Black bear study area and the 38 baited hair snare locations operated for 8 weeks during June and July, 2006.

2235  $n_{traps} \times K$  of individual encounter events (i.e.,  $y_{ijk} = 1$  if individual  $i$  was encountered in trap  $j$  during occasion  $k$ , and 0 otherwise). For fitting model  $M_0$  (or  $M_h$ ,  
 2236 see below), it is sufficient to reduce the data to individual encounter frequencies  
 2237 which we have re-labeled “y” above. The **BUGS** model file along with commands  
 2238 to fit the model are as follows:  
 2239

```

2240 > set.seed(2013)                      # to obtain the same results each time
2241 > library(R2WinBUGS)                   # load R2WinBUGS, set-up:
2242 > data0 <- list(y=y, M=M, K=K)         # data ....
2243 > params0 <- c('psi','p','N')          # parameters ....
2244 > zst <- c(rep(1,nind),rbinom(M-nind, 1, .5)) # inits ....
2245 > inits <- function(){ list(z=zst, psi=runif(1), p=runif(1)) }
2246
2247 > cat("                                "
2248 model{
2249   psi ~ dunif(0, 1)
2250   p ~ dunif(0,1)
2251
2252   for (i in 1:M){
2253     z[i] ~ dbern(psi)
```

```

2255   for(k in 1:K){
2256     tmp[i,k] <- p*z[i]
2257     y[i,k] ~ dbin(tmp[i,k],1)
2258   }
2259 }
2260 N<-sum(z[1:M])
2261 }
2262 ",file="modelM0.txt")
2263
2264 ## Run the model:
2265 > fit0 <- bugs(data0, inits, params0, model.file="modelM0.txt", n.chains=3,
2266   n.iter=2000, n.burnin=1000, n.thin=1, debug=TRUE, working.directory=getwd())

```

2267 This produces the following posterior summary statistics:

```

2268 > print(fit0,digits=2)
2269 Inference for Bugs model at "modelM0.txt", fit using WinBUGS,
2270 3 chains, each with 2000 iterations (first 1000 discarded)
2271 n.sims = 3000 iterations saved
2272      mean    sd  2.5%   25%   50%   75% 97.5% Rhat n.eff
2273 psi      0.29  0.04  0.22  0.26  0.29  0.31  0.36    1  3000
2274 p       0.30  0.03  0.25  0.28  0.30  0.32  0.35    1  3000
2275 N       49.94 1.99 47.00 48.00 50.00 51.00 54.00    1  3000
2276 deviance 489.05 11.28 471.00 480.45 488.80 495.40 513.70    1  3000
2277
2278 [... some output deleted ...]

```

2279 **WinBUGS** did well in choosing an MCMC algorithm for this model – we have  
2280  $\hat{R} = 1$  for each parameter, and an effective sample size of 3000, equal to the total  
2281 number of posterior samples<sup>3</sup>. We see that the posterior mean of  $N$  under this  
2282 model is 49.94 and a 95% posterior interval is (48, 54). We revisit these data later  
2283 in the context of more complex models.

2284 In order to obtain an estimate of density,  $D$ , we need an area to associate  
2285 with the estimate of  $N$ , and in Chapt. 1 we already went through a number of  
2286 commonly used procedures to conjure up such an area, including buffering the trap  
2287 array by the home range radius, often estimated by the mean maximum distance  
2288 moved (MMDM) (Parmenter et al., 2003), 1/2 MMDM (Dice, 1938) or directly from  
2289 telemetry data (Wallace et al., 2003). Typically, the trap array is defined by the  
2290 convex hull around the trap locations, and this is what we applied a buffer to. We  
2291 computed the buffer by using a telemetry-based estimate of the mean female home  
2292 range radius (2.19 km) (Bales et al., 2005) instead of using an estimate based on our  
2293 relatively more sparse recapture data. For the Fort Drum study, the convex hull has  
2294 an area of 157.135 km<sup>2</sup>, and the buffered convex hull has an area of 277.011 km<sup>2</sup>.  
2295 To create this we used functions contained in the **R** package **rgeos** and created a

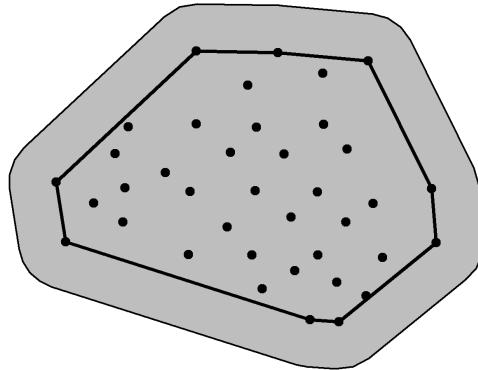
---

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

2296 utility function `bcharea` which is in our **R** package `scrbook`. The commands are  
 2297 as follows:

```
2298 > library(rgeos)
2299
2300 > bcharea <- function(buff,traplocs){
2301   p1 <- Polygon(rbind(traplocs,traplocs[1,]))
2302   p2 <- Polygons(list(p1=p1),ID=1)
2303   p3 <- SpatialPolygons(list(p2=p2))
2304   p1ch <- gConvexHull(p3)
2305   bp1 <- (gBuffer(p1ch, width=buf))
2306   plot(bp1, col='gray')
2307   plot(p1ch, border='black', lwd=2, add=TRUE)
2308   gArea(bp1)
2309 }
2310
2311 > bcharea(2.19,traplocs=trapmat)
```

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



**Figure 3.2.** Convex hull of the bear hair snare array at Fort Drum, NY, buffered by mean female home range radius (2.19 km).

2313 To conjure up a density estimate under model  $M_0$ , we compute the appropriate  
 2314 posterior summary of the ratio of  $N$  and the prescribed area ( $277.011 \text{ km}^2$ ):

```
2315 > summary(fit0$sims.list$N/277.011)
 2316   Min. 1st Qu. Median Mean 3rd Qu. Max.
 2317 0.1697 0.1733 0.1805 0.1803 0.1841 0.2130
 2318
 2319 > quantile(fit0$sims.list$N/277.011,c(0.025,0.975))
 2320   2.5% 97.5%
 2321 0.1696684 0.1949381
```

2322 which yields a density estimate of about  $0.18 \text{ ind/km}^2$ , and a 95% Bayesian con-  
 2323 fidence interval of  $(0.170, 0.195)$ . Our estimate of density should be reliable if we  
 2324 have faith in our stated value of the “sampled area”. Clearly though this is largely  
 2325 subjective, and not something we can formally evaluate (or estimate) from the data  
 2326 based on model  $M_0$ .

### 3.3 TEMPORALLY VARYING AND BEHAVIORAL EFFECTS

2327 The purpose of this chapter is mainly to emphasize the central importance of the  
 2328 binomial model in capture-recapture and so we have considered models for individ-  
 2329 ual encounter frequencies—the number of times individuals are captured out of  $K$   
 2330 occasions. Sometimes we can’t aggregate the encounter data for each individual,  
 2331 such as when encounter probability varies over time among samples. Time-varying  
 2332 responses that are relevant in many capture-recapture studies are “effort” such as  
 2333 amount of search time, number of observers, or trap nights, or encounter probability  
 2334 varying over time, as a function of date or season (Kéry et al., 2010) due to species  
 2335 behavior. A common situation in many animal studies is that in which there exists  
 2336 a “behavioral response” to trapping (even if the animal is not physically trapped).

2337 Behavioral response is an important concept in animal studies because individ-  
 2338 uals might learn to come to baited traps or avoid traps due to trauma related to  
 2339 being encountered. There are a number of ways to parameterize a behavioral re-  
 2340 sponse to encounter. The distinction between persistent and ephemeral was made  
 2341 by Yang and Chao (2005) who considered a general behavioral response model of  
 2342 the form:

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

2343 where  $x_{ik}$  is a covariate indicator variable of previous capture (i.e.,  $x_{ik} = 1$  if cap-  
 2344 tured in any previous period). Therefore, encounter probability changes depending  
 2345 on whether an individual was captured in the immediate previous period (a Marko-  
 2346 vian or ephemeral behavioral response; (Yang and Chao, 2005)), described by the  
 2347 term  $\alpha_1 y_{i,k-1}$  or in *any* previous period (persistent behavioral response), described  
 2348 by the term  $\alpha_2 x_{ik}$ . Because spatial capture-recapture models allow us to include  
 2349 trap-specific covariates, we can describe a 3rd type of behavioral response—a lo-  
 2350 cal behavioral response that is trap-specific (Royle et al., 2011b). In this local

2351 behavioral response, the encounter probability is modified for an individual trap  
 2352 depending on previous capture in that trap. Models with temporal effects are  
 2353 easy to describe and analyze in the **BUGS** language and we provide a number of  
 2354 examples in Chapt. 7 and elsewhere.

### 3.4 MODELS WITH INDIVIDUAL HETEROGENEITY

2355 Models in which encounter probability varies by individual have a long history in  
 2356 capture-recapture and, indeed, this so-called “model  $M_h$ ” is one of the elemental  
 2357 capture-recapture models in (Otis et al., 1978). Conceptually, we imagine that  
 2358 the individual-specific encounter probability parameters,  $p_i$ , are random variables  
 2359 distributed according to some probability distribution,  $[\theta]$ . We denote this basic  
 2360 model assumption as  $p_i \sim [\theta]$ . This type of model is similar in concept to extending  
 2361 a GLM to a GLMM but in the capture-recapture context  $N$  is unknown. The basic  
 2362 class of models is often referred to as “model  $M_h$ ” (“h” for heterogeneity), but really  
 2363 this is a broad class of models, each being distinguished by the specific distribution  
 2364 assumed for  $p_i$ . There are many different varieties of model  $M_h$  including parametric  
 2365 and various non-parametric approaches (Burnham and Overton, 1978; Norris and  
 2366 Pollock, 1996; Pledger, 2004). One important practical matter is that estimates of  
 2367  $N$  can be extremely sensitive to the choice of heterogeneity model (Fienberg et al.,  
 2368 1999; Dorazio and Royle, 2003; Link, 2003). Indeed, Link (2003) showed that in  
 2369 some cases it’s possible to find models that yield precisely the same expected data,  
 2370 yet produce wildly different estimates of  $N$ . In that sense,  $N$  for most practical  
 2371 purposes is not identifiable across classes of different heterogeneity models, and  
 2372 this should be understood before fitting any such model. One solution to this  
 2373 problem is to seek to model explicit factors that contribute to heterogeneity, e.g.,  
 2374 using individual covariate models (See 4.5 below). Indeed, spatial capture-recapture  
 2375 models do just that, by modeling heterogeneity due to the spatial organization  
 2376 of individuals in relation to traps or other encounter mechanism. For additional  
 2377 background and applications of model  $M_h$  see Royle and Dorazio (2008, Chapt. 6)  
 2378 and Kéry and Schaub (2012, Chapt. 6).

2379 We will work with a specific type of model  $M_h$  here which is a natural extension  
 2380 of the basic binomial observation model of model  $M_0$  so that

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

2381 where  $\mu$  is a fixed parameter (the mean) to be estimated, and  $\eta_i$  is an individual  
 2382 random effect assumed to be normally distributed:

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

2383 We could as well combine these two steps and write  $\text{logit}(p_i) \sim \text{Normal}(\mu, \sigma_p^2)$ . This  
 2384 “logit-normal mixture” was analyzed by Coull and Agresti (1999) and elsewhere.  
 2385 It is a natural extension of the basic model with constant  $p$ , as a mixed GLMM,

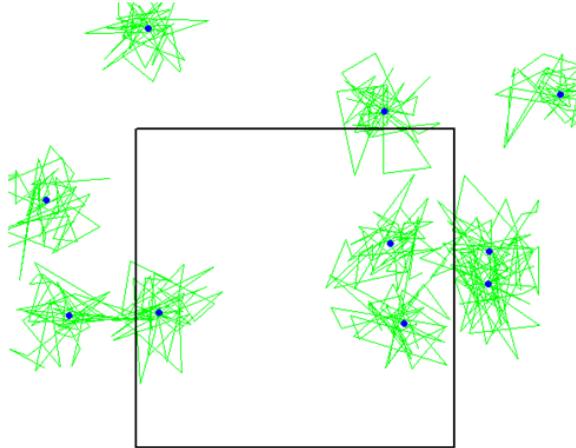
and similar models occur throughout statistics. It is also natural to consider a beta prior distribution for  $p_i$  (Dorazio and Royle, 2003) and so-called “finite-mixture” models are also popular (Norris and Pollock, 1996; Pledger, 2004). In the latter, individuals are assumed to belong to a finite number of latent classes, each of which has its own capture probability.

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

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

$$p_i = p_0\phi_i$$

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



**Figure 3.3.** 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.

#### 2424    3.4.1 Analysis of model $M_h$

2425    If  $N$  is known, it is worth taking note of the essential simplicity of model  $M_h$  as a bi-  
 2426    nomial GLMM. This is a type of model that is widely applied throughout statistics  
 2427    using standard methods of inference based either on integrated likelihood (Laird  
 2428    and Ware, 1982; Berger et al., 1999), which we discuss in Chapt. 6, or standard  
 2429    Bayesian methods. However, because  $N$  is not known, inference is somewhat more  
 2430    challenging. We address that here using Bayesian analysis based on data augmentation.  
 2431    Although we use data augmentation in the context of Bayesian methods here,  
 2432    we note that heterogeneity models formulated under DA are easily analyzed by  
 2433    conventional likelihood methods as zero-inflated binomial mixtures (Royle, 2006)  
 2434    and more traditional analysis of model  $M_h$  based on integrated likelihood, without  
 2435    using data augmentation, has been considered by Coull and Agresti (1999), Dorazio  
 2436    and Royle (2003), and others.

2437    As with model  $M_0$ , we have the Bernoulli model for the zero-inflation variables:  
 2438     $z_i \sim \text{Bernoulli}(\psi)$  and the model of the observations expressed conditional on these  
 2439    latent variables  $z_i$ . For  $z_i = 1$ , we have a binomial model with individual-specific

2440  $p_i$ :

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

2441 and otherwise  $y_i|z_i=0 \sim I(y=0)$ , i.e., a point mass at  $y=0$ . Further, we  
2442 prescribe a distribution for  $p_i$ . Here we assume

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

2443 For prior distributions we assume  $p_0 = \text{logit}^{-1}(\mu) \sim \text{Uniform}(0, 1)$  and, for the  
2444 standard deviation  $\sigma \sim \text{Uniform}(0, B)$  for some large  $B$ . Another common default  
2445 prior is to assume  $\tau = 1/\sigma^2 \sim \text{Gamma}(1, 1)$ , although we usually choose  $\sigma \sim$   
2446  $\text{Uniform}(0, B)$ .

### 2447 3.4.2 Analysis of the Fort Drum data with model $M_h$

2448 Here we provide an analysis of the Fort Drum bear survey data using the logit-  
2449 normal heterogeneity model, and we used data augmentation to produce a data  
2450 set of  $M = 700$  individuals. We have so far mostly used **WinBUGS** but we are  
2451 now transitioning to the use of **JAGS** run from within **R** using the useful packages  
2452 **R2jags** or **rjags**. The function **jags** from the **R2jags** package runs essentially like  
2453 the **bugs** function which we demonstrate here for setting up and running model  $M_h$   
2454 for the Fort Drum bear data:

```
2455 [... get data as before ....]
2456
2457 > set.seed(2013)
2458
2459 > cat("
2460 model{
2461   p0 ~ dunif(0,1)           # prior distributions
2462   mup <- log(p0/(1-p0))
2463   sigmap ~ dunif(0,10)
2464   taup <- 1/(sigmap*sigmap)
2465   psi ~ dunif(0,1)
2466
2467   for(i in 1:(nind+nz)){
2468     z[i] ~ dbern(psi)       # zero inflation variables
2469     lp[i] ~ dnorm(mup,taup) # individual effect
2470     logit(p[i]) <- lp[i]
2471     mu[i] <- z[i]*p[i]
2472     y[i] ~ dbin(mu[i],K)    # observation model
2473   }
2474
2475   N<-sum(z[1:(nind+nz)])
2476 }
2477 ",file="modelMh.txt")
```

---

```

2478 > data1 <- list(y=y, nz=nz, nind=nind, K=K)
2479 > params1 <- c('p0','sigmap','psi','N')
2480 > inits <- function(){ list(z=as.numeric(y>=1), psi=.6, p0=runif(1),
2481           sigmap=runif(1,.7,1.2),lp=rnorm(M,-2)) }
2482 > library(R2jags)
2483 > wabout <- jags(data1, inits, params1, model.file = "modelMh.txt", n.chains = 3,
2484           n.iter = 1010000, n.burnin = 10000, working.directory = getwd())

```

2485 We provide an **R** function `modelMhBUGS` in the package `scrbook` which will fit  
 2486 the model using either **JAGS** or **WinBUGS** as specified by the user. In addition,  
 2487 for fun, we construct our own MCMC algorithm using a Metropolis-within-Gibbs  
 2488 algorithm for model  $M_h$  in Chapt. 17, where we also develop MCMC algorithms for  
 2489 spatial capture-recapture models. Using `modelMhBUGS`, we ran 3 chains of 1 *million*  
 2490 iterations (mixing is poor for this model and this data set), which produced the  
 2491 posterior distribution for  $N$  shown in Fig. 4.4. Posterior summaries of parameters  
 2492 are given in Table 4.5.

**Table 3.5.** Posterior summaries from model  $M_h$  fitted to the Fort Drum black bear data. Results were obtained using **WinBUGS** running 3 chains, each with 1010000 iterations, discarding the first 10000 for a total of three *million* posterior samples.

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

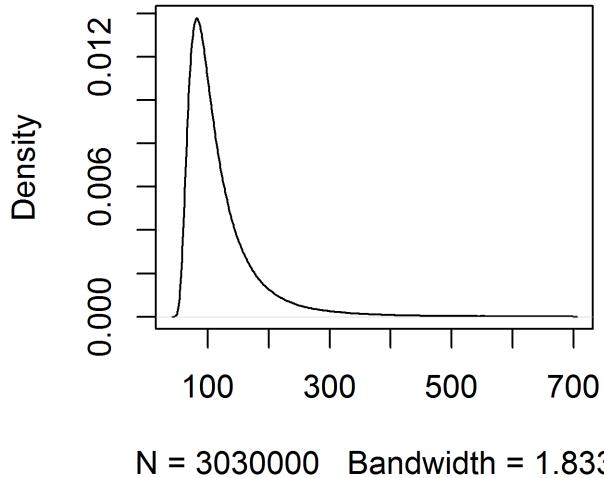
2493 We used  $M = 700$  for this analysis and we note that while the posterior mass  
 2494 of  $N$  is concentrated away from this upper bound (Fig. 4.4), the posterior has an  
 2495 extremely long right tail, with some MCMC draws at the upper boundary  $N = 700$ ,  
 2496 suggesting that an even higher value of  $M$  may be called for. To characterize the  
 2497 posterior distribution of density we produce the relevant summaries of the posterior  
 2498 distribution of  $D = N/277.11$  (recall the buffered area of the convex hull is 277.11  
 2499  $km^2$ ):

```

2500 > summary(wabout$sims.list$N/277.11)
2501   Min. 1st Qu. Median Mean 3rd Qu. Max.
2502 0.1696 0.2959 0.3681 0.4428 0.4944 2.5260
2503
2504 > quantile(wabout$sims.list$N/277.11,c(0.025,0.50,0.975))
2505   2.5%      50%     97.5%
2506 0.2237379 0.3680849 1.1511674

```

2507 Therefore, the point estimate, characterized by the posterior median, is around 0.37  
 2508 bears per square km and a 95% Bayesian credible interval is (0.224, 1.151).



**Figure 3.4.** Posterior of  $N$  for Fort Drum bear study data under the logit-normal version of model  $M_h$ .

### 2509 3.4.3 Comparison with MLE

2510 The posterior of  $N$  is highly skewed; therefore, we see that the posterior mean  
 2511 ( $N = 122.7$ ) is considerably higher than the posterior median ( $N = 102$ ). Further,  
 2512 it may be surprising that these posterior summaries do not compare well with the  
 2513 MLE. We used the **R** code contained in Panel 6.1 from Royle and Dorazio (2008) to  
 2514 obtain the MLE of  $\log(n_0)$ , the logarithm of the number of uncaptured individuals,  
 2515 is  $\widehat{\log(n_0)} = 3.86$  and therefore  $\hat{N} = \exp(3.86) + 47 = 94.47$ , which is larger than the  
 2516 mode shown in Fig. 4.4. To see this, we compute the posterior mode, by finding  
 2517 the posterior value of  $N$  with the highest mass. Because  $N$  is discrete, we can  
 2518 use the `table()` function in **R** and find the most frequent value<sup>4</sup>. If we want to  
 2519 smooth out some of the Monte Carlo error a bit, we can use a smoother of some sort  
 2520 applied to the tabled posterior frequencies of  $N$ . Here we use a smoothing spline (**R**  
 2521 function `smooth.spline`) with the degree of smoothing chosen by cross-validation  
 2522 (the `cv=TRUE` argument):

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

```

2523 > N <- table(jout$BUGSoutput$sims.list$N)
2524 > xg <- as.numeric(names(N))
2525
2526 > sp <- smooth.spline(xg,N,cv=TRUE)
2527
2528 > sp
2529
2530 Call:
2531 smooth.spline(x = xg, y = N, cv = TRUE)
2532
2533 Smoothing Parameter spar= 0.09339815 lambda= 8.201724e-09 (17 iterations)
2534 Equivalent Degrees of Freedom (Df): 121.1825
2535 Penalized Criterion: 2544481
2536 PRESS: 5903.4

```

2537 We obtain the mode of the smoothed frequencies as follows:

```

2538 sp$x[sp$y==max(sp$y)]
2539 [1] 82

```

2540 We don't dwell too much on the difference between the MLE and features of the  
 2541 posterior, but we do note here that the posterior distribution for the parameters of  
 2542 this model, for the Fort Drum data set, are very sensitive to the prior distributions.  
 2543 In the present case, the use of a Uniform(0, 1) prior for  $p_0 = \text{logit}^{-1}(\mu)$  is somewhat  
 2544 informative—in particular, it is not at all “flat” on the scale of  $\mu$ , and this affects  
 2545 the posterior. We generally always recommend use of a Uniform(0, 1) prior for  
 2546  $\text{logit}^{-1}(\mu)$  in such models. That said, we were surprised at this result, and we  
 2547 experimented with other prior configurations including putting a flat prior on  $\mu$   
 2548 directly. This kind of small sample instability has been widely noted in model  
 2549  $M_h$  (Fienberg et al., 1999; Dorazio and Royle, 2003), as has extreme sensitivity to  
 2550 the specific form of model  $M_h$  (Link, 2003). In summary, while the mode is well-  
 2551 defined, the data set is relatively sparse and hence inferences are poor and sensitive  
 2552 to model choice.

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

2553 A standard situation in capture-recapture models is when a covariate which is  
 2554 thought to influence encounter probability is measured for each individual. These  
 2555 are often called “individual covariate models” but, in keeping with the classical  
 2556 nomenclature on closed population models, Kéry and Schaub (2012) referred to  
 2557 this class of models as “model  $M_x$ ” (the  $x$  here being an explicit covariate). As  
 2558 with other closed population models, we begin with the basic binomial observation  
 2559 model:

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

2560 To model the covariate, we use a logit model for encounter probability of the form:

2561

$$\text{logit}(p_i) = \alpha_0 + \alpha_1 x_i \quad (3.5.1)$$

2562 where  $x_i$  is the covariate value for individual  $i$  and the parameters  $\boldsymbol{\alpha} = (\alpha_0, \alpha_1)$   
 2563 are the regression coefficients. Classical examples of covariates influencing detec-  
 2564 tion probability are type of animal (juvenile/adult or male/female), a continuous  
 2565 covariate such as body mass, or a discrete covariate such as group or cluster size.  
 2566 For example, in models of aerial survey data, it is natural to model the detection  
 2567 probability of a group as a function of the observation-level individual covariate,  
 2568 “group size” (Royle, 2008; Langtimm et al., 2011).

2569 Model  $M_x$  is similar in structure to model  $M_h$ , except that the individual effects  
 2570 are *observed* for the  $n$  individuals that appear in the sample. These models are  
 2571 important here because spatial capture-recapture models can be described precisely  
 2572 as a form of model  $M_x$ , where the covariate describes *where* the individual is located  
 2573 in relation to the trapping array. Specifically, SCR models *are* individual covariate  
 2574 models, but where the individual covariate is only observed imperfectly (or partially  
 2575 observed) for each captured individual. Unlike model  $M_h$ , in SCR models (and  
 2576 model  $M_x$ ) we do have some direct information about the latent variable, which  
 2577 comes from the spatial locations/distribution of individual recaptures.

2578 Traditionally, estimation of  $N$  in model  $M_x$  is achieved using methods based on  
 2579 ideas of unequal probability sampling (i.e., Horvitz-Thompson estimation<sup>5</sup>; Huggins  
 2580 (1989), Alho (1990) and Borchers et al. (2002)). An estimator of  $N$  is

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

2581 where  $\tilde{p}_i$  is the probability that individual  $i$  appeared in the sample. This quantity  
 2582 is  $\tilde{p}_i = \Pr(y_i > 0)$  and, in closed population capture-recapture models, it can be  
 2583 computed as:

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

2584 where  $p_i$  is a function of parameters  $\alpha_0$  and  $\alpha_1$  according to Eq. 4.5.1. In practice,  
 2585 parameters are estimated from the conditional-likelihood of the observed encounter  
 2586 histories which is, for observation  $y_i$ ,

$$\mathcal{L}_c(\boldsymbol{\alpha}|y_i) = \frac{\text{Binomial}(y_i|\boldsymbol{\alpha})}{\tilde{p}_i}. \quad (3.5.2)$$

2587 This derives from a straightforward application of the law of total probability.  
 2588 Conceptually, we partition  $\Pr(y)$  according to  $\Pr(y) = \Pr(y|y > 0)\Pr(y > 0) +$   
 2589  $\Pr(y|y = 0)\Pr(y = 0)$ . For any positive value of  $y$  the 2nd term is necessarily

<sup>5</sup>For a quick summary of the idea see:

[http://en.wikipedia.org/wiki/Horvitz-Thompson\\_estimator](http://en.wikipedia.org/wiki/Horvitz-Thompson_estimator)

2590 0, and so we rearrange to obtain  $\Pr(y|y > 0) = \Pr(y)/\Pr(y > 0)$  which, in the  
 2591 specific case where  $\Pr(y)$  is the binomial probability mass function (pmf) produces  
 2592 Eq. 4.5.2.

2593 Here we take a formal model-based approach to Bayesian analysis of such models  
 2594 based on the joint likelihood using data augmentation (Royle, 2009). Classical  
 2595 likelihood analysis of the so-called “full likelihood” is covered by Borchers et al.  
 2596 (2002). For Bayesian analysis of model  $M_x$ , because the individual covariate is  
 2597 unobserved for the  $n_0 = N - n$  uncaptured individuals, we require a model to  
 2598 describe variation in  $x$  among individuals, essentially allowing the sample to be  
 2599 extrapolated to the population. For example, if we have a continuous trait measured  
 2600 on each individual, then we might assume that  $x$  has a normal distribution:

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

2601 Data augmentation can be applied directly to this class of models. In particular,  
 2602 reformulation of the model under DA yields a basic zero-inflated binomial model of  
 2603 the following form, for each  $i = 1, 2, \dots, M$ :

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

2604 Fully spatial capture-recapture models use this formulation with a latent covariate  
 2605 that is directly related to the individual detection probability (see next section).  
 2606 As with the previous models, implementation is trivial in the **BUGS** language.  
 2607 The **BUGS** specification is very similar to that for model  $M_h$ , but we require the  
 2608 distribution of the covariate to be specified, along with priors for the parameters of  
 2609 that distribution.

### 2610 3.5.1 Example: Location of capture as a covariate

2611 Here we consider a special type of model  $M_x$  that is especially relevant to spatial  
 2612 capture-recapture. Intuitively, some measure of distance from home range center  
 2613 to traps for an individual should be a reasonable covariate to explain heterogeneity  
 2614 in encounter probability, i.e., individuals with more exposure to traps should have  
 2615 higher encounter probabilities and vice versa. So we can imagine *estimating* such a  
 2616 quantity, say average distance from home range center to “the trap array”, and then  
 2617 using it as an individual covariate in capture-recapture models. A version of this  
 2618 idea was put forth by Boulanger and McLellan (2001) (see also Ivan (2012)), but  
 2619 using the Huggins-Alho estimator and with covariate “distance from home range  
 2620 center to edge” of the trapping array, where the home range center is estimated by  
 2621 the average capture location. This is intuitively appealing because we can imagine,

in some kind of an ideal situation where we have a dense grid of traps over some geographic region, that the average location of capture would be a decent estimate (heuristically) of an individual's home range center. We provide an example of this type of approach using a fully model-based analysis of the version of model  $M_x$  described above, analyzed by data augmentation. We take a slightly different approach than that adopted by Boulanger and McLellan (2001). By analyzing the full likelihood and placing a prior distribution on the individual covariate, we will resolve the problem of having an ill-defined sample area. After you read later chapters of this book, it will be apparent that SCR models represent a formalization of this heuristic procedure.

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

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

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

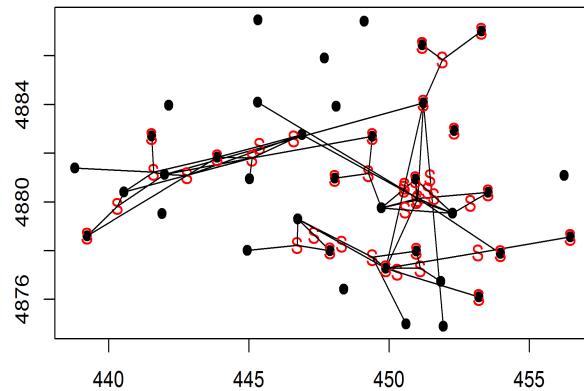
where  $B$  is a specified constant, which we may choose to be arbitrarily large. For example,  $B$  should be at least a home range diameter past the furthest trap from the centroid of the array.

### 3.5.2 Fort Drum bear study

We have to do a little bit of data processing to fit this individual covariate model to the Fort Drum data. We need to compute the individual covariate  $\mathbf{x}_i$  (distance from the centroid of the trapping array) using the R function `spiderplot` provided in `scrbook`. This function also produces the keen plot shown in Fig. 4.5 which we call

2660 a “spider plot”. The **R** commands for obtaining the individual covariate “distance  
 2661 from trap centroid” (the variable `xcent` returned by `spiderplot`) and making the  
 2662 spider plot are as follows:

```
2663 > library(scrbook)
2664 > data(beardata)
2665 > toad <- spiderplot(beardata$bearArray,beardata$trapmat)
2666 > xcent <- toad$xcent
```



**Figure 3.5.** Spider plot of the Fort Drum study data. The black dots represent the 47 trap locations with the “S” symbols being the average capture location of each bear. i.e., its estimated home range center. All traps in which a bear was captured are connected to its estimated home range center with a line.

2667 For the analysis of these data using the individual covariate “distance from  
 2668 centroid” we used  $x_i \sim \text{Uniform}(0, B)$  with  $B = 11.5 \text{ km}^2$ , which is about the  
 2669 distance from the array center to the furthest trap. Once we choose a value for  $B$ ,  
 2670 the direct implication is that the population size parameter,  $N$ , applies to the area  
 2671 within 11.5 units of the trap centroid. Therefore, the model associates a precise  
 2672 area within which the population of  $N$  individuals resides. We will see shortly  
 2673 that  $N$  does, in fact, scale with our choice of  $B$  to reflect the changing area over  
 2674 which the  $N$  individuals of the model reside. The **BUGS** model specification and  
 2675 **R** commands to package the data and fit the model are as follows:

```
2676 cat("
```

```

2677 model{
2678   p0 ~ dunif(0,1)                      # prior distributions
2679   alpha0 <- log(p0/(1-p0))
2680   psi ~ dunif(0,1)
2681   beta ~ dnorm(0,.01)
2682
2683 for(i in 1:(nind+nz)){
2684   xcent[i] ~ dunif(0,B)
2685   z[i] ~ dbern(psi)                     # DA variables
2686   lp[i] <- alpha0 + beta*xcent[i] # individual effect
2687   logit(p[i]) <- lp[i]
2688   mu[i] <- z[i]*p[i]
2689   y[i] ~ dbin(mu[i],K)                 # observation model
2690 }
2691
2692 N <- sum(z[1:(nind+nz)])
2693 }
2694 ",file="modelMcov.txt")
2695
2696 data2 <- list(y=y,nz=nz, nind=nind, K=K, xcent=xcent,B=11.5)
2697 params2 <- c('p0','psi','N','beta')
2698 inits <- function() {list(z=zst, psi=psi, p0=runif(1), beta=rnorm(1) ) }
2699 fit2 <- bugs(data2, inits, params2, model.file="modelMcov.txt",
2700                 n.chains=3, n.iter=11000, n.burnin=1000, n.thin=1)

```

2700 This produces the posterior summary statistics in Table 4.6.

**Table 3.6.** Posterior summaries from the individual covariate model (model  $M_x$ ) with covariate “distance from the centroid of the trap array”, fitted to the Fort Drum black bear data. Results were obtained using WinBUGS running 3 chains, each with 11000 iterations, discarding the first 1000 for a total of 30000 posterior samples.

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

2701 We note that the estimated  $N$  is much lower than obtained by model  $M_h$  but  
2702 there is a good explanation for this which we discuss in the next section. That issue  
2703 notwithstanding, it is worth pondering how this model could be an improvement  
2704 (conceptually or technically) over some other model/estimator including  $M_0$  and  
2705  $M_h$  considered previously. Well, for one, we have accounted formally for hetero-  
2706 geneity due to spatial location of individuals relative to exposure to the trap array,  
2707 characterized by the centroid of the array. Moreover, we have done so using a model  
2708 that is based on an explicit mechanism, as opposed to a phenomenological one such

2709 as model  $M_h$ . In addition, and importantly, using our new model, *the estimated*  
 2710  *$N$  applies to an explicit area which is defined by our prescribed value of  $B$ .* That  
 2711 is, this area is a fixed component of the model and the parameter  $N$  therefore has  
 2712 explicit spatial context, as the number of individuals with home range centers less  
 2713 than  $B$  from the centroid of the trap array. As such, the implied “effective area”  
 2714 of the trap array for a given  $B$  is a precisely defined quantity—it is that of a circle  
 2715 with radius  $B$ .

2716 **3.5.3 Extension of the model**

2717 The model developed in the previous section is not a very good model for one  
 2718 important reason: Imposing a uniform prior distribution on  $x$  implies that density  
 2719 is *not constant* over space. In particular, this model implies that density *decreases*  
 2720 as we move away from the centroid of the trap array. That is,  $x_i \sim \text{Uniform}(0, B)$   
 2721 implies constant  $N$  in each distance band from the centroid but obviously the *area*  
 2722 of each distance band is increasing. This is one reason we have a lower estimate of  
 2723 density than that obtained previously from model  $M_h$  (Sec. 4.4.2) and also why, if  
 2724 we were to increase  $B$ , we would see density continue to decrease.

2725 Fortunately, we are not restricted to use of this specific distribution for the  
 2726 individual covariate. Clearly, it is a bad choice and, therefore, we should think  
 2727 about whether we can choose a better distribution for  $B$ —one that doesn’t imply a  
 2728 decreasing density as distance from the centroid increases. Conceptually, what we  
 2729 want to do is impose a prior on distance from the centroid,  $x$ , such that abundance  
 2730 should be proportional to the amount of area in each successive distance band as  
 2731 you move farther away from the centroid, so that density is *constant*. In fact, theory  
 2732 exists which tells us we should choose  $[x] = 2x/B^2$ . This can be derived by noting  
 2733 that  $F(x) = \Pr(X < x) = (\pi x^2)/(\pi * B^2)$ . Then,  $f(x) = dF/dx = 2 * x/(B^2)$ .  
 2734 This is a sort of triangular distribution in density induced because the incremental  
 2735 area in each additional distance band increases linearly with radius (i.e., distance  
 2736 from centroid). This can be verified empirically as follows:

```
2737 > u <- runif(10000,-1,1)
2738 > v <- runif(10000,-1,1)
2739 > d <- sqrt(u*u+v*v)
2740 > hist(d[d<1])
2741 > hist(d[d<1],100)
2742 > hist(d[d<1],100,probability=TRUE)
2743 > abline(0,2)
```

2744 It would be useful if we could describe this distribution directly in **BUGS** but  
 2745 there is not a built-in way to do so. However, we can implement a discrete version  
 2746 of the pdf<sup>6</sup>. To do this, we break  $B$  into  $L$  distance classes of width  $\delta$ , with  
 2747 probabilities proportional to  $2 * x$ . In particular, if we denote the cut-points by

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

2748  $g_1 = 0, g_2, \dots, g_{L+1} = B$  and the interval midpoints are  $m_i = g_{i+1} - \delta$ . Then the  
 2749 interval probabilities are, approximately<sup>7</sup>,  $p_i = \delta(2m_i/B^2)$ , which we can compute  
 2750 once and then pass them to **BUGS** as data. The **R** commands for doing all of this  
 2751 (noting that we have already loaded and processed the Fort Drum bear data) are  
 2752 given in the following **R/BUGS** script:

```

2753 > delta <- .2
2754 > xbin <- xcent%/%delta + 1                      # Put x in bins
2755 > midpts <- seq(delta,Dmax,delta)
2756 > xprobs <- delta*(2*midpts/(B*B))
2757 > xprobs <- xprobs/sum(xprobs)

2758 > cat("
2759 model{
2760   p0 ~ dunif(0,1)                                # Prior distributions
2761   alpha0 <- log(p0/(1-p0))
2762   psi ~ dunif(0,1)
2763   beta ~ dnorm(0,.01)

2765   for(i in 1:(nind+nz)){
2766     xbin[i] ~ dcat(xprobs[])
2767     z[i] ~ dbern(psi)                            # DA variables
2768     lp[i] <- alpha0 + beta*xbin[i]*delta        # Individual covariate model
2769     logit(p[i]) <- lp[i]
2770     mu[i] <- z[i]*p[i]
2771     y[i] ~ dbin(mu[i],K)                         # Observation model
2772   }
2773
2774 N <- sum(z[1:(nind+nz)])                      # N is derived
2775 }
2776 ",file="modelMcov.txt")

```

2778 In the model description, the variable  $x$  (observed distance from centroid of  
 2779 the trap array) has been rounded or binned (placed into a distance bin) so that  
 2780 the discrete version of the pdf of  $x$  can be used, as described previously. The new  
 2781 variable labeled **xbin** is then the *integer category* in units of  $\delta$  from 0. Thus, to  
 2782 convert back to distance in the expression for  $lp[i]$ , **xbin[i]** has to be multiplied  
 2783 by  $\delta$ . To fit the model, keeping in mind that the data objects required below have  
 2784 been defined in previous analyses of this chapter, we do this:

```

2785 > data2 <- list(y=y, nz=nz, nind=nind, K=K, xbin=xbin, xprobs=xprobs,
2786   delta=delta)
2787 > params2 <- c('p0','psi','N','beta')
2788 > inits <- function() {list(z=z, psi=psi, p0=rnorm(1),beta=rnorm(1) ) }
2789 > fit <- bugs(data2, inits, params2, model.file="modelMcov.txt",
2790   working.directory=getwd(), debug=FALSE, n.chains=3,

```

---

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

2791           n.iter=11000, n.burnin=1000, n.thin=2)

2792       By specification of  $B$ , this model induces a clear definition of area in which the  
 2793       population of  $N$  individuals reside. The parameter  $N$  of the model is the population  
 2794       size that applies to the particular value of  $B$  and, as such, we will see that  $N$  scales  
 2795       with our choice of  $B$ . This might be disconcerting to some—we can get whatever  
 2796       value of  $N$  we want by changing  $B$ ! However, it is intuitively reasonable that, as  
 2797       we increase the area under consideration, there should be more individuals in it.  
 2798       Fortunately, we find empirically, that while  $N$  is highly sensitive to the prescribed  
 2799       value of  $B$ , density appears invariant to  $B$  as long as  $B$  is sufficiently large. We  
 2800       fit the model for a set of values of  $B$  from  $B = 12$  (restricting values of  $x$  to be in  
 2801       close proximity to the trap array) on up to 20. The results are given in Table 4.7.

**Table 3.7.** Analysis of Fort Drum bear hair snare data using the individual covariate model, for different values of  $B$ , the upper limit of the uniform distribution of ‘distance from centroid of the trap array’. “Density” is the posterior mean of density.

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

2802       We see that the posterior mean and SD of density (individuals per square km)  
 2803       appear insensitive to choice of  $B$  once we reach about  $B = 17$  or so. The estimated  
 2804       density of 0.25 per  $\text{km}^2$  is actually quite a bit lower than we reported using model  
 2805        $M_h$  for which no relevant “area” quantity is explicit in the model (and so we had  
 2806       to make it up). Using MLEs of  $N$  in conjunction with buffer strips (see Tab. 1.1)  
 2807       our estimates were in the range of  $0.32 - 0.43$  and see Sec. 4.4 above. On the  
 2808       other hand our estimate of  $\hat{D} = 0.25$  here (based on the posterior mean) is higher  
 2809       than that reported from model  $M_0$  using the buffered area ( $\hat{D} = 0.18$ ). There is  
 2810       no basis really for comparing or contrasting these various estimates. In particular,  
 2811       application of models  $M_0$  and  $M_h$  are distinctly *not* spatially explicit models—the  
 2812       area within which the population resides is not defined under either model. There is  
 2813       therefore no reason at all to think that the estimates produced under either closed  
 2814       population model, based on a buffered “trap area”, are justifiable by any theory.  
 2815       In fact, we would get exactly the same estimate of  $N$  no matter what we declare  
 2816       the area to be. On the other hand, the individual covariate model uses an explicit  
 2817       model for “distance from centroid” that is a reasonable and standard null model—  
 2818       it posits, in the absence of direct information, that individual home range centers  
 2819       are randomly distributed in space and that probability of detection depends on the  
 2820       distance between home range center and the centroid of the trap array. Under this

2821 definition of the system, we see that density is invariant to the choice of area, which  
 2822 seems like a desirable feature.

2823 **3.5.4 Invariance of density to  $B$**

2824 Under model  $M_x$ , and also under models that we consider in later chapters, a  
 2825 general property of the estimators is that while  $N$  increases with the prescribed  
 2826 area of the model (defined by  $B$  in this model), we expect that density estimators  
 2827 should be invariant to this area. In the model used above, we note that  $\text{Area}(B) =$   
 2828  $\pi B^2$  and  $\mathbb{E}(N(B)) = \lambda \text{Area}(B)$  and thus  $\mathbb{E}(\text{Density}(B)) = \lambda$ , i.e., constant. This  
 2829 should be interpreted as the *prior* density. Absent data, then realizations under the  
 2830 model will have density  $\lambda$  regardless of what  $B$  is prescribed to be. As we verified  
 2831 empirically above, posterior summaries of density are also invariant to  $B$  as long  
 2832 as the prescribed area is sufficiently large.

2833 **3.5.5 Toward fully spatial capture-recapture models**

2834 While the use of an individual covariate model resolves two important problems  
 2835 inherent in almost all capture-recapture studies (induced heterogeneity and absence  
 2836 of a precise relationship between  $N$  and area), is not ideal for all purposes because  
 2837 it does not make full use of the spatial information in the data set, i.e., the trap  
 2838 locations and the locations of each individual encounter, so that we cannot use  
 2839 this model to model trap-specific effects (e.g., trap effort or type). Moreover, we  
 2840 applied this model for “data” being the average observed encounter location, and  
 2841 equated that summary to the home range center  $s_i$ . Intuitively, taking the average  
 2842 encounter location as an estimate of home range center makes sense but more so  
 2843 when the trapping grid is dense and expansive relative to typical home range sizes  
 2844 which might not be reasonable in practice. Moreover, this approach also ignored  
 2845 the variable precision with which each  $s_i$  is estimated. Finally, it ignores that  
 2846 estimates of  $s_i$  around the “edge” (however we define that) are biased because the  
 2847 observations are truncated—we can only observe locations interior to the array.

2848 However, there is hope to extend this model in order to resolve these remaining  
 2849 deficiencies. In the next chapter we provide a further extension of this individual  
 2850 covariate model that definitively resolves the *ad hoc* nature of the approach we took  
 2851 here. In that chapter we build a model in which  $s_i$  are regarded as latent variables  
 2852 and the observation locations (i.e., trap specific encounters) are linked to those  
 2853 latent variables with an explicit model. We note that the model fitted previously  
 2854 could be adapted easily to deal with  $s_i$  as a latent variable, simply by adding a  
 2855 prior distribution for  $s_i$ . This is actually easier, and less ad hoc in a number of  
 2856 respects, and you should try it out.

**3.6 DISTANCE SAMPLING: A PRIMITIVE SCR MODEL**

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

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

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

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

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

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

wherein  $B > 0$  is a known constant, being the upper limit of data recording by the observer (i.e., the point count radius, or transect half-width). Specification of this

2894 distance sampling model in the **BUGS** language is shown in Panel 4.2, taken from  
 2895 Royle and Dorazio (2008).

---

```

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

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

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

```

---

Panel 3.2: Distance sampling model in **BUGS** for a line transect situation, using a half-normal detection function.

2896 As with the individual covariate model in the previous section, the distance  
 2897 sampling model can be equivalently specified by putting a prior distribution on  
 2898 individual *location* instead of distance between individual and observation point  
 2899 (or transect). Thus we can write the general distance sampling model as

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

2900 along with

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

2901 where  $\mathbf{x}_0$  is a fixed point (or line) and  $\mathbf{u}_i$  is the individual's location, which is  
 2902 observed for the sample of  $n$  individuals. In practice it is easier to record distance  
 2903 instead of location. Basic math can be used to argue that if individuals have a  
 2904 uniform distribution in space, then the distribution of Euclidean distance is also  
 2905 uniform. In particular, if a transect of length  $L$  is used and  $x$  is distance to the  
 2906 transect then  $F(x) = \Pr(X \leq x) = L * x / L * B = x / B$  and  $f(x) = dF/dx = (1/B)$ .  
 2907 For measurements of radial distance, we provided the analogous argument in the  
 2908 previous section.

2909 The preceding paragraph makes it clear that distance sampling is a special case  
 2910 of spatial capture-recapture models, such as those derived from model  $M_x$  of the  
 2911 previous section, where the encounter probability is related directly to *distance*,  
 2912 which is a reduced information summary of *location*,  $\mathbf{u}$ . Some intermediate forms  
 2913 of SCR/DS models can be described (Royle et al., 2011a). In the context of our  
 2914 general characterization of SCR models (Chapt. 2.6), we suggested that every SCR

2915 model can be described, conceptually, by a hierarchical model of the form:

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

2916 Distance sampling ignores the part of the model pertaining to  $\mathbf{s}$ , and deals only  
 2917 with the model components for the observed data  $\mathbf{u}$ <sup>8</sup>. Thus, we are left with a  
 2918 hierarchical model of the form

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

2919 In contrast, as we will see in the next chapters, many SCR models (Chapt. 5)  
 2920 ignore  $\mathbf{u}$  and condition on  $\mathbf{s}$ , which is not observed:

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

2921 Since  $[\mathbf{u}]$  and  $[\mathbf{s}]$  are both assumed to be uniformly distributed, these are equivalent  
 2922 models! The main differences have to do with interpretation of model components  
 2923 and whether or not the latent variables are observable (in distance sampling they  
 2924 are).

2925 So why bother with SCR models when distance sampling yields density esti-  
 2926 mates and accounts for spatial heterogeneity in detection? For one, imagine trying  
 2927 to collect distance sampling data on species such as jaguars or tigers! Clearly,  
 2928 distance sampling requires that one can collect large quantities of distance data,  
 2929 which is not always possible. For tigers, it is much easier, efficient, and safer to  
 2930 employ camera traps or track plates and then apply SCR models. Furthermore, as  
 2931 we will see in Chapt. 15, SCR models can make use of distance data, allowing us  
 2932 to study distribution, movement, and density. Thus, SCR models are more general  
 2933 and versatile than distance sampling models (which clearly are a special case), and  
 2934 can accommodate data from virtually all animal survey designs.

### 2935 3.6.1 Example: Sonoran desert tortoise study

2936 We illustrate the application of distance sampling models using data on the Sonoran  
 2937 desert tortoise (*Gopherus agassizii*), shown in Fig. 4.6, collected along transects  
 2938 in southern Arizona (see Zylstra et al. (2010) for details). The data are from 120  
 2939 square transects having four 250-m sides, although we ignore this detail in our  
 2940 analysis here and regard them as 1 km transects, and we pooled the detection  
 2941 data from all 120 transects. The histogram of encounter distances from the 65  
 2942 encountered individuals is shown in Fig. 4.7

2943 Commands for reading in and organizing the data for analysis using **Win-**  
**2944 BUGS** are given in the help file `?tortoise` provided with the `scrbook` package.  
 2945 To compute density, the total sampled area of the transects `striparea` is input  
 2946 as data, and computed as: 120 (transects) multiplied by the length (1000 m) and  
 2947 half-width ( $B = 40$  m), then multiplied by 2, and divided by 10000 to convert to

---

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



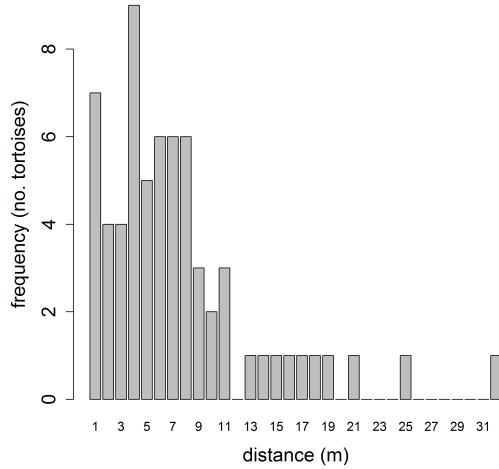
**Figure 3.6.** Desert tortoise in its native habitat (*Photo credit: Erin Zylstra, Univ. of Arizona*).

units of individuals per ha. We also provide commands for analyzing the data with **unmarked** (Fiske and Chandler, 2011) using hierarchical distance sampling models (Royle et al., 2004).

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

**Table 3.8.** Posterior summaries from the tortoise distance sampling data. Results were obtained using **WinBUGS** running 3 chains, each with 3000 iterations and the first 1000 discarded, thinning by 2.

parameter	Mean	SD	2.5%	50 %	97.5%	Rhat	n.eff
$\alpha_1$	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



**Figure 3.7.** Distance histogram of  $n = 65$  Sonoran desert tortoise detections from a total of 120 km of survey transect.

### 3.7 SUMMARY AND OUTLOOK

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

Spatial capture-recapture models that we will consider in the rest of the chapters of this book are closely related to individual covariate models (model  $M_x$ ). Naturally, spatial capture-recapture models arise by defining individual covariates based on observed locations of individuals—we can think of using some function of mean encounter location as an individual covariate. We did this in a novel way, by

2976 using distance to the centroid of the trapping array as a covariate. We analyzed  
2977 the *full likelihood* using data augmentation, and placed a prior distribution on the  
2978 individual covariate which was derived from an assumption that individual loca-  
2979 tions are, a priori, uniformly distributed in space. This assumption provides for  
2980 invariance of the density estimator to the choice of population size area (induced  
2981 by maximum distance from the centroid of the trap array). The model addressed  
2982 some important problems in the use of closed population models: it allows for het-  
2983 erogeneity in encounter probability due to the spatial juxtaposition of individuals  
2984 with the array of traps, and it also provides a direct estimate of density because  
2985 area is a feature of the model (via the prior on the individual covariate). The model  
2986 is still not completely general, however, because it does not make full use of the  
2987 spatial encounter histories, which provide direct information about the locations  
2988 and density of individuals.

2989 A specific individual covariate model that is in widespread use is classical dis-  
2990 tance sampling. The model underlying distance sampling is precisely a special  
2991 kind of SCR model—but one without replicate samples. Understanding distance  
2992 sampling and individual covariate models more broadly provides a solid basis for  
2993 understanding and analyzing spatial capture-recapture models. In fact if, instead  
2994 of placing an explicit model on *distance* in the classical distance sampling model,  
2995 we were to place the prior distribution on *location*,  $\mathbf{s}$ , of each individual, then the  
2996 form of the distance sampling model more closely resembles the SCR model we  
2997 introduce in the next chapter.



2998

## Part II

2999

---

3000

# Basic SCR Models



3001  
3002

---

3003

# 4

3004

## FULLY SPATIAL CAPTURE-RECAPTURE MODELS

3005 In the previous chapter, we discussed models that could be viewed as primitive  
3006 spatial capture-recapture models. We looked at a basic distance sampling model,  
3007 and we also considered a classical individual covariate modeling approach in which  
3008 we defined a covariate to be the distance from the (estimated) home range center to  
3009 the center of the trap array. The individual covariate model that we conjured up was  
3010 “spatial” in the sense that it included some characterization of where individuals  
3011 live but, on the other hand, only a primitive or no characterization of trap location.  
3012 That said, there is only a small step from this model to spatial capture-recapture  
3013 models that we consider in this chapter, which fully recognize the spatial attribution  
3014 of both individual animals *and* the locations of encounter devices.

3015 Capture-recapture models must accommodate the spatial organization of indi-  
3016 viduals and the encounter devices because the encounter process occurs at the level  
3017 of individual traps. Failure to consider the trap-specific data is one of the key defi-  
3018 ciencies with classical ad-hoc approaches which aggregate encounter information to  
3019 the resolution of the entire trap array. We have previously addressed some problems  
3020 that this causes including induced heterogeneity in encounter probability, imprecise  
3021 notation of “sample area” and not being able to accommodate trap-specific effects  
3022 or trap-specific missing values. In this chapter we resolve these issues by developing  
3023 our first fully spatial capture-recapture model. This model is not too different from  
3024 that considered in Sec. 4.5 but, instead of defining the individual covariate to be  
3025 distance to the centroid of the array we define  $J$  individual covariates - the distance  
3026 to *each* trap. And, instead of using estimates of individual locations  $\mathbf{s}$ , we consider  
3027 a fully hierarchical model in which we regard  $\mathbf{s}$  as a latent variable and impose a  
3028 prior distribution on it.

3029 In this chapter we investigate the basic spatial capture-recapture model, which  
 3030 we refer to as “model SCR0”, and address some important considerations related  
 3031 to its analysis in **BUGS**. We demonstrate how to summarize posterior output for  
 3032 the purposes of producing density maps or spatial predictions of density. The key  
 3033 aspect of the SCR models considered in this chapter is the formulation of a model  
 3034 for encounter probability that is a function of distance between individual home  
 3035 range center and trap locations. We also discuss how encounter probability models  
 3036 are related to explicit models of space usage or “home range area.” Understanding  
 3037 this allows us to compute, for example, the area used by an individual during some  
 3038 prescribed time. While it is intuitive that SCR models should be related to some  
 3039 model of space usage, this has not been discussed much in the literature (but see  
 3040 Royle et al. (2012b) which we address further in Chapt. 13).

## 4.1 SAMPLING DESIGN AND DATA STRUCTURE

3041 In our development here, we will assume a standard sampling design in which  
 3042 an array of  $J$  traps is operated for  $K$  sample occasions (say, nights) producing  
 3043 encounters of  $n$  individuals. Because sampling occurs by traps and also over time,  
 3044 the most general data structure yields temporally *and* spatially indexed encounter  
 3045 histories for *each individual*. Thus a typical data set will include an encounter  
 3046 history *matrix* for each individual indicating which trap the individual was captured,  
 3047 during each sample occasion. For example, suppose we sample at 4 traps over 3  
 3048 nights. A plausible data set for a single individual captured one time in trap 1 on  
 3049 the first night and one time in trap 3 on the 3rd night is:

```
3050      night1 night2 night3
3051 trap1    1    0    0
3052 trap2    0    0    0
3053 trap3    0    0    1
3054 trap4    0    0    0
```

3055 This data structure would be obtained for *each* of the  $i = 1, 2, \dots, n$  captured  
 3056 individuals.

3057 We develop models in this chapter for passive detection devices such as “hair  
 3058 snares” or other DNA sampling methods (Kéry et al., 2010; Gardner et al., 2010b)  
 3059 and related types of sampling devices in which (i) devices (“traps”) may capture any  
 3060 number of individuals (i.e., they don’t fill up); (ii) an individual may be captured  
 3061 in more than one trap during each occasion but (iii) individuals can be encountered  
 3062 at most 1 time by each trap during any occasion. Hair snares for sampling DNA  
 3063 from bears and other species function according to these rules. An individual bear  
 3064 wandering about its territory might come into contact with  $> 1$  devices; a device  
 3065 may encounter multiple bears; however, in practice, it will often not be possible to  
 3066 attribute multiple visits of the same individual during a single occasion (e.g., night)  
 3067 to distinct encounter events. Thus, an individual may be captured at most 1 time

**Table 4.1.** 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

3068 in each trap during any occasion. While this model, which we refer to as SCR0, is  
 3069 most directly relevant to hair snares and other DNA sampling methods for which  
 3070 multiple detections of an individual are not distinguishable, we will also make use  
 3071 of the model for data that arise from camera-trapping studies. In practice, with  
 3072 camera trapping, individuals might be photographed several times in a night but  
 3073 it is common to distill such data into a single binary encounter event for reasons  
 3074 discussed later in Chapt. 9.

3075 The statistical assumptions we make to build a model for these data are that  
 3076 individual encounters within and among traps are independent, and this allows us  
 3077 to regard individual- and trap-specific encounters as *independent* Bernoulli trials  
 3078 (see next section). These basic (but admittedly at this point somewhat imprecise)  
 3079 assumptions define the basic spatial capture-recapture model, SCR0. We will make  
 3080 things more precise as we develop a formal statistical definition of the model shortly.

## 4.2 THE BINOMIAL OBSERVATION MODEL

3081 We begin by considering the simple model in which there are no time-varying co-  
 3082 variates that influence encounter, there are no explicit individual-specific covariates,  
 3083 and there are no covariates that influence density. In this case, we can aggregate  
 3084 the binary encounters over the  $K$  sample occasions and record the total number of  
 3085 encounters out of  $K$ . We will denote these individual- and trap-specific encounter  
 3086 frequencies by  $y_{ij}$  for  $i = 1, 2, \dots, n$  captured individuals and  $j = 1, 2, \dots, J$  traps.  
 3087 For example, suppose we observe 6 individuals in sampling at 4 traps over 3 nights  
 3088 of sampling then a plausible data set is the  $6 \times 4$  matrix of encounters (out of 3 sam-  
 3089 pling occasions) shown in Table 5.1. We assume that  $y_{ij}$  are mutually independent  
 3090 outcomes of a binomial random variable which we express as:

$$y_{ij} \sim \text{Binomial}(K, p_{ij}) \quad (4.2.1)$$

3091 This is the basic model underlying standard closed population models (Chapt. 4)  
 3092 except that, in the present case, the encounter frequencies are individual- *and* trap-  
 3093 specific, and encounter probability  $p_{ij}$  depends on both individual *and* trap.

3094 As we did in Sec. 4.5, we will make explicit the notion that  $p_{ij}$  is defined condi-  
 3095 tional on *where* individual  $i$  lives. Naturally, we think about defining an individual

3096 home range and then relating  $p_{ij}$  explicitly to a summary of its location relative to  
 3097 each trap. For example, the centroid of the individuals home range, or its center of  
 3098 activity (Efford, 2004; Borchers and Efford, 2008; Royle and Young, 2008). In what  
 3099 follows, we define  $\mathbf{s}_i$ , a two-dimensional spatial coordinate, to be the home range  
 3100 or activity center of individual  $i$ . Then, the SCR model postulates that encounter  
 3101 probability,  $p_{ij}$ , is a decreasing function of distance between  $\mathbf{s}_i$  and the location of  
 3102 trap  $j$ ,  $\mathbf{x}_j$  (also a two-dimensional spatial coordinate). A standard model for mod-  
 3103 eling binomial counts is the logistic regression, where we model the dependence of  
 3104  $p_{ij}$  on distance according to:

$$\text{logit}(p_{ij}) = \alpha_0 + \alpha_1 \|\mathbf{x}_j - \mathbf{s}_i\| \quad (4.2.2)$$

3105 where, here,  $\|\mathbf{x}_j - \mathbf{s}_i\|$  is the distance between  $\mathbf{s}_i$  and  $\mathbf{x}_j$ . We sometimes write  
 3106  $\|\mathbf{x}_j - \mathbf{s}_i\| = \text{dist}(\mathbf{x}_j, \mathbf{s}_i) = d_{ij}$ . Alternatively, a popular model is

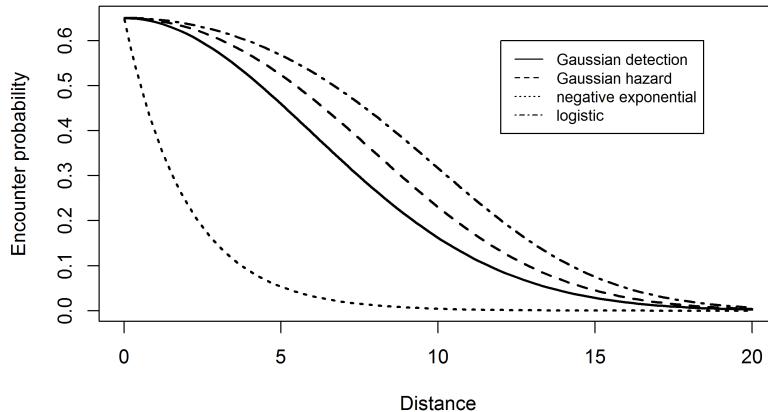
$$p_{ij} = p_0 \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{x}_j - \mathbf{s}_i\|^2\right) \quad (4.2.3)$$

3107 which is similar to the “half-normal” model in distance sampling, except with an  
 3108 intercept  $p_0 \leq 1$  which can be estimated in SCR studies. Because it is the kernel of a  
 3109 bivariate normal, or Gaussian, probability density function for the random variable  
 3110 “individual location” we will refer to it as the “(bivariate) normal” or “Gaussian”  
 3111 model although the distance sampling term “half-normal” is widely used. In the  
 3112 context of 2-dimensional space, the model is clearly interpretable as a primitive  
 3113 model of movement outcomes or space usage (we discuss this in Sec. 5.4).

3114 There are a large number of standard detection models commonly used (see  
 3115 Chapt. 7). All other standard models that relate encounter probability to  $\mathbf{s}$  will  
 3116 also have a parameter that multiplies distance in some non-linear function. To be  
 3117 consistent with parameter naming across models, we will sometimes parameterize  
 3118 any encounter probability model so that the coefficient on distance (or distance  
 3119 squared) is  $\alpha_1$ . So, for the Gaussian model,  $\alpha_1 = 1/(2\sigma^2)$ . A characteristic of  
 3120 the common parametric forms is they are monotone decreasing with distance, but  
 3121 vary in their characteristic behavior as they approach distance = 0. We show the  
 3122 standard Gaussian, Gaussian hazard, negative exponential and logistic models in  
 3123 Fig. 5.1. The negative exponential model has  $p_{ij} = p_0 \exp(-\alpha_1 \|\mathbf{x}_j - \mathbf{s}_i\|)$  and  
 3124 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  
 3125 Gaussian kernel. Whatever model we choose for encounter probability, we should  
 3126 always keep in mind that the activity center for individual  $i$ ,  $\mathbf{s}_i$ , is an unobserved  
 3127 random variable. To be precise about this in the model, we should express the  
 3128 observation model as

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

3129 but sometimes, for notational simplicity, we abbreviate this by omitting some of  
 3130 the arguments to  $p$ .



**Figure 4.1.** Some common encounter probability models showing the characteristic monotone decrease of encounter probability with distance between activity center and trap location.

#### 3131    4.2.1 Definition of home range center

3132    We define an individual's home range as *the area used by an organism during some time period* which has a clear meaning for most species regardless of their biology.  
 3133  
 3134    We therefore define the home range center (or activity center) to be the center of  
 3135    the space that individual was occupying (or using) during the period in which traps  
 3136    were active. Thinking about it in that way, it could even be observable (almost) as  
 3137    the centroid of a very large number of radio fixes over the course of a survey period  
 3138    or a season. Thus, this practical version of a home range center in terms of space  
 3139    usage is a well-defined construct regardless of whether one thinks the home range  
 3140    itself is a meaningful concept. We use the terms home range center and activity  
 3141    center interchangeably, and we recognize that this is a transient thing which applies  
 3142    only to a well-defined period of study.

#### 3143    4.2.2 Distance as a latent variable

3144    If we knew precisely every  $s_i$  in the population (and population size  $N$ ), then  
 3145    the model specified by Eqs. 5.2.1 and 5.2.2 would be just an ordinary logistic  
 3146    regression-type of a model (with covariate  $d_{ij}$ ) which we learned how to fit using  
 3147    **WinBUGS** previously (Chapt. 3). However, the activity centers are unobservable  
 3148    even in the best possible circumstances. In that case,  $d_{ij}$  is an unobserved variable,

3149 analogous to the situation in classical random effects models. We need to therefore  
 3150 extend the model to accommodate these random variables with an additional model  
 3151 component – the random effects distribution. The customary assumption is the  
 3152 so-called “uniformity assumption,” which is to assume that the  $\mathbf{s}_i$  are uniformly  
 3153 distributed over space (the obvious next question: “which space?” is addressed  
 3154 below). This uniformity assumption amounts to a uniform prior distribution on  $\mathbf{s}_i$ ,  
 3155 i.e., the pdf of  $\mathbf{s}_i$  is constant, which we may express

$$\Pr(\mathbf{s}_i) \propto \text{constant} \quad (4.2.4)$$

3156 As it turns out, this assumption is usually not precise enough to fit SCR models  
 3157 in practice for reasons we discuss shortly. We will give another way to represent  
 3158 this prior distribution that is more concrete, but depends on specifying the “state-  
 3159 space” of the random variable  $\mathbf{s}_i$ . The term state-space is a technical way of saying  
 3160 “the space of all possible outcomes” of the random variable.

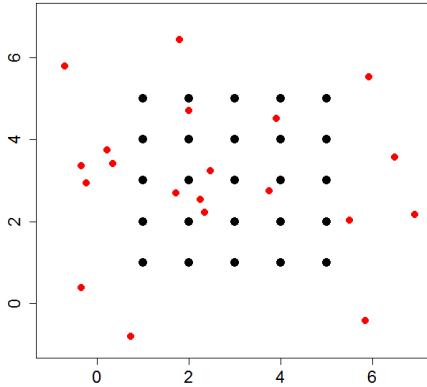
### 4.3 THE BINOMIAL POINT PROCESS MODEL

3161 In the SCR model, the individual activity centers are unobserved and thus we treat  
 3162 them as random effects. Specifically, the collection of individual activity centers  
 3163  $\mathbf{s}_1, \dots, \mathbf{s}_N$  represents a realization of a *binomial point process* (Illian et al., 2008, p.  
 3164 61). The binomial point process (BPP) is analogous to a Poisson point process in  
 3165 the sense that it represents a “random scatter” of points in space – except that the  
 3166 total number of points is *fixed*, whereas, in a Poisson point process, it is random  
 3167 (having a Poisson distribution). As an example, we show in Fig. 5.2 locations of  
 3168 20 individual activity centers (black dots) in relation to a grid of 25 traps. For  
 3169 a Poisson point process the number of such points in the prescribed state-space  
 3170 would be random whereas often we will simulate fixed numbers of points, e.g., for  
 3171 evaluating the performance of procedures, e.g., how well does our estimator perform  
 3172 when  $N = 50$ ?

3173 It is natural to consider a binomial point process in the context of capture-  
 3174 recapture models because it preserves  $N$  in the model and thus preserves the linkage  
 3175 directly with closed population models. In fact, under the binomial point process  
 3176 model, model  $M_0$  and other closed models are simple limiting cases of SCR models,  
 3177 i.e., they arise as the coefficient on distance ( $\alpha_1$  above) tends to 0.

3178 While we often will express SCR models “conditional-on- $N$ ”, it will sometimes  
 3179 be convenient to impose specific prior distributions on  $N$ . By assuming  $N$  has a  
 3180 binomial distribution, we can make use of data augmentation, our preferred tool, for  
 3181 Bayesian analysis of the models as in Chapt. 4, thus yielding a methodologically  
 3182 coherent approach to analyzing the different classes of models. We might also  
 3183 assume that  $N$  has a Poisson distribution in some cases (see Chapt. 14). Of  
 3184 course, the two assumptions are closely related in the usual limiting sense.

3185 One consequence of having fixed  $N$  in the BPP model is that the model is  
 3186 not strictly a model of “complete spatial randomness”. This is because, if one



**Figure 4.2.** Realization (small dots) of a binomial point process with  $N = 20$ . The large dots represent trap locations.

3187 forms counts  $n(A_1), \dots, n(A_k)$  in any set of disjoint regions of the state-space, say  
 3188  $A_1, \dots, A_k$ , then these counts are *not* independent. In fact, they have a multinomial  
 3189 distribution (see Illian et al., 2008, p. 61). Thus, the BPP model introduces a slight  
 3190 bit of dependence in the distribution of points. However, in most situations this will  
 3191 have no practical effect on any inference or analysis and, as a practical matter, we  
 3192 will usually regard the BPP model as one of spatial independence among individual  
 3193 activity centers because each activity center is distributed independently of each  
 3194 other activity center. Despite this independence we see in Fig. 5.2 that *realiza-*  
 3195 *tions* of randomly distributed points will typically exhibit distinct non-uniformity.  
 3196 Thus, independent, uniformly distributed points will almost never appear regu-  
 3197 larly, uniformly or systematically distributed. For this reason, the basic binomial  
 3198 (or Poisson) point process models are enormously useful in practical settings since  
 3199 they allow for a range of distribution patterns without violating the assumption of  
 3200 spatial randomness. More relevant for SCR models is that we actually have a little  
 3201 bit of data for some individuals and thus the resulting posterior point pattern can  
 3202 deviate strongly from uniformity, a point we come back to repeatedly in this book.  
 3203 The uniformity hypothesis is only a *prior* distribution which is directly affected by  
 3204 the quantity and quality of the observed data, to produce a posterior distribution  
 3205 which may appear distinctly non-uniform. In addition, we can build more flexible  
 3206 models for the point process, which we take up in Chapt. 11.

3207 **4.3.1 The state-space of the point process**

3208 Shortly we will focus on Bayesian analysis of model SCR0 with  $N$  known so that we  
 3209 can gain some basic experience with important elements of the model, and its anal-  
 3210 ysis. To do this, we note that the individual activity centers  $\mathbf{s}_i, \dots, \mathbf{s}_N$  are unknown  
 3211 quantities and we will need to be able to simulate each  $\mathbf{s}_i$  in the population from  
 3212 the posterior distribution. In order to simulate the  $\mathbf{s}_i$ , it is necessary to describe  
 3213 precisely the region over which they are distributed. This is the quantity referred  
 3214 to above as the state-space, which is sometimes called the *observation window* in  
 3215 the point process literature. We denote the state-space henceforth (throughout this  
 3216 book) by  $\mathcal{S}$ , which is a region or a set of points comprising the potential values (the  
 3217 support) of the random variable  $\mathbf{s}$ . Thus, an equivalent explicit statement of the  
 3218 “uniformity assumption” is

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

3219 where  $\mathcal{S}$  is a precisely defined region. e.g., in Fig. 5.2,  $\mathcal{S}$  is the square defined  
 3220 by  $[-1, 7] \times [-1, 7]$ . Thus each of the  $N = 20$  points were generated by randomly  
 3221 selecting each coordinate on the line  $[-1, 7]$ . When points are distributed uniformly  
 3222 over some region, the point process is usually called a *homogeneous point process*.

3223 **Prescribing the state-space**

3224 Evidently, to define the model, we need to define the state-space,  $\mathcal{S}$ . How can we  
 3225 possibly do this objectively? Prescribing any particular  $\mathcal{S}$  seems like the equivalent  
 3226 of specifying a “buffer” which we have criticized as being ad hoc. How is it, then,  
 3227 that the choice of a state-space is *not* ad hoc? As we observed in Chapt. 4, it is  
 3228 true that  $N$  increases with  $\mathcal{S}$ , but only at the same rate as the area of  $\mathcal{S}$  increases  
 3229 under the prior assumption of constant density. As a result, we say that density is  
 3230 invariant to  $\mathcal{S}$  as long as  $\mathcal{S}$  is sufficiently large. Thus, while choice of  $\mathcal{S}$  is (or can  
 3231 be) essentially arbitrary, once  $\mathcal{S}$  is chosen, it defines the population being exposed  
 3232 to sampling, which scales appropriately with the size of the state-space.

3233 For our simulated system developed previously in this chapter, we defined the  
 3234 state-space to be a square within which our trap array was centered. For many  
 3235 practical situations this might be an acceptable approach to defining the state-  
 3236 space, i.e., just a rectangle around the trap array. Although defining the state-space  
 3237 to be a regular polygon has computational advantages (e.g., we can implement this  
 3238 more efficiently in **BUGS** and cannot for irregular polygons), a regular polygon  
 3239 induces an apparent problem of admitting into the state-space regions that are  
 3240 distinctly non-habitat (e.g., oceans, large lakes, ice fields, etc.). It is difficult to  
 3241 describe complex regions in mathematical terms that can be used in **BUGS**. As  
 3242 an alternative, we can provide a representation of the state-space as a discrete set  
 3243 of points which the **R** package **secr** (Efford, 2011) permits (**secr** uses the term  
 3244 “mask” for what we call the state-space). Defining the state-space by a discrete set  
 3245 of points is handy because it allows specific points to be deleted or not, depending  
 3246 on whether they represent available or suitable habitat (see Sec. 5.10). We can

3247 also define the state-space as an arbitrary collection of polygons stored as a GIS  
 3248 shapefile which can be analyzed easily by MCMC in **R** (see Sec. 17.5), but not so  
 3249 easily in the **BUGS** engines. In Sec. 5.10, we provide an analysis of the wolverine  
 3250 camera trapping data, in which we define the state-space to be a regular continuous  
 3251 polygon (a rectangle).

### 3252 **Invariance to the state-space**

3253 We will assert for all models we consider in this book that density is invariant to  
 3254 the size and extent of  $\mathcal{S}$ , if  $\mathcal{S}$  is sufficiently large, and as long as our model relating  
 3255  $p_{ij}$  to  $\mathbf{s}_i$  is a decreasing function of distance. We can prove this easily by drawing  
 3256 an analogy with a 1-d case involving distance sampling. Let  $y_j$  be the number of  
 3257 individuals captured in some interval  $[d_{j-1}, d_j]$ , and define  $d_J = B$  for some large  
 3258 value of  $B$ . The observations from a survey are  $y_1, \dots, y_J$  and the likelihood is a  
 3259 multinomial likelihood, so the log-likelihood is of the form

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

3260 where  $\pi_j$  is the probability of detecting an individual in distance class  $j$ , which  
 3261 depends on parameters of the detection function (the manner of which is not rel-  
 3262 evant for the present discussion). Choosing  $B$  sufficiently large guarantees that  
 3263  $\mathbb{E}(y_J) = 0$  and therefore the observed frequency in the “last cell” contributes noth-  
 3264 ing to the likelihood, in regular situations in which the detection function decays  
 3265 monotonically with distance and prior density is constant. We can think of  $B$  as  
 3266 being related to the state-space in an SCR model, as the width of a rectangular  
 3267 state-space with area  $B \times L$ ,  $L$  being the length of the transect. Thus, if we choose  
 3268  $B$  large enough, then we ensure that the expected trap-frequencies beyond  $B$  will  
 3269 be 0, and thus contribute nothing to the likelihood.

3270 Sometimes our estimate of density can be affected by choosing  $\mathcal{S}$  too small.  
 3271 However, this might be sensible if  $\mathcal{S}$  is naturally well-defined. As we discussed in  
 3272 Chapt. 1,  $\mathcal{S}$  is part of the model, and thus it is sensible that estimates of density  
 3273 might be sensitive to its definition in problems where it is natural to restrict  $\mathcal{S}$ . One  
 3274 could imagine, however, in specific cases, e.g., a small population with well-defined  
 3275 habitat preferences, that a problem could arise because changing the state-space  
 3276 based on differing opinions, and GIS layers, might have substantial affects on the  
 3277 density estimate. But this is a real biological problem, and a natural consequence  
 3278 of the spatial formalization of capture-recapture models – a feature, not a bug  
 3279 or some statistical artifact – and it should be resolved with better information,  
 3280 research, and thinking. For situations where there is not a natural choice of  $\mathcal{S}$ ,  
 3281 we should default to choosing  $\mathcal{S}$  to be very large in order to achieve invariance or,  
 3282 otherwise, evaluate sensitivity of density estimates by trying a couple of different  
 3283 choices of  $\mathcal{S}$ . This is a standard “sensitivity to prior” argument that Bayesians  
 3284 always have to be conscious of. We demonstrate this in our analysis of Sec. 5.9

3285 below. As an additional practical consideration, we note that the area of the state-  
 3286 space  $\mathcal{S}$  affects data augmentation. If you increase the size of  $\mathcal{S}$ , then there are  
 3287 more individuals to account for and therefore the size of the augmented data set  
 3288  $M$  must increase. This has computational implications.

3289 **4.3.2 Connection to model  $M_h$  and distance sampling**

3290 SCR models are closely related to “model  $M_h$ ” and also distance sampling. In  
 3291 SCR models, heterogeneity in encounter probability is induced by both the effect  
 3292 of distance in the model for detection probability and also from specification of  
 3293 the state-space. Hence, the state-space is an explicit element of the model. To  
 3294 understand this, suppose activity centers have the uniform distribution:

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

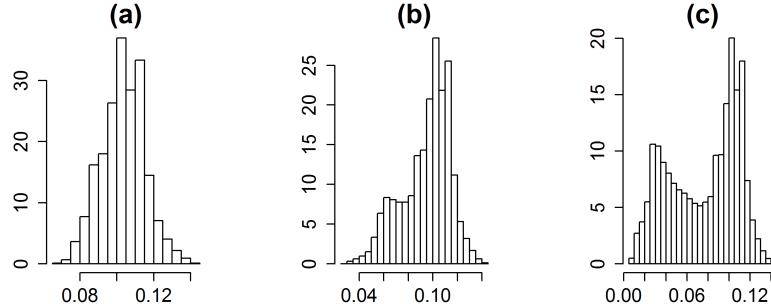
3295 and encounter probability is a function of  $\mathbf{s}$ , denoted by  $p(\mathbf{s}) = p(y = 1|\mathbf{s})$ . For  
 3296 example, under Eq. 5.2.2 we have that

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

3297 and we can work out, either analytically or empirically, what is the implied distri-  
 3298 bution of  $p$  for a population of individuals. Fig. 5.3 shows a histogram of  $p$  for a  
 3299 hypothetical population of 100000 individuals on a state-space enclosing our  $5 \times 5$   
 3300 trap array above, under the logistic model for distance given by Eq. 5.2.2 with  
 3301 buffers of 0.2, 0.5 and 1.0. We see the mass shifts to the left as the buffer increases,  
 3302 implying more individuals with lower encounter probabilities, as their home range  
 3303 centers increase in distance from the trap array.

3304 Another way to understand this is by representing  $\mathcal{S}$  as a set of discrete points  
 3305 on a grid. In the coarsest possible case where  $\mathcal{S}$  is a single arbitrary point, then  
 3306 every individual has exactly the same  $p$ . As we increase the number of points in  $\mathcal{S}$ ,  
 3307 more distinct values of  $p$  are possible. Indeed, when  $\mathcal{S}$  is characterized by discrete  
 3308 points, then SCR models are precisely a type of finite-mixture model (Norris and  
 3309 Pollock, 1996; Pledger, 2004), except, in the case of SCR models, we have some  
 3310 information about which group an individual belongs to (i.e., where their activity  
 3311 center is), as a result of which traps it is captured in.

3312 It is also worth re-emphasizing that the basic SCR encounter model is a binomial  
 3313 encounter model in which distance is a covariate. As such, it is strikingly similar  
 3314 to classical distance sampling models (Buckland et al., 2001). Both have distance  
 3315 as a covariate but, in classical distance sampling problems, the focus is on the  
 3316 distance between the observer and the animal at an instant in time, not the distance  
 3317 between a trap and an animal’s home range center. As a practical matter, in  
 3318 distance sampling, “distance” is *observed* for those individuals that appear in the  
 3319 sample. Conversely, in SCR problems, it is only imperfectly observed (we have  
 3320 partial information in the form of trap observations). Clearly, it is preferable to



**Figure 4.3.** Implied distribution of  $p_i$  for a population of individuals as a function of the size of the state-space buffer around the trap array. The state-space buffer is 0.2, 0.5 and 1.0 for panels (a), (b), (c), respectively. In each case, the trap array is fixed and centered within a square state-space.

3321 observe distance if possible, but distance sampling requires field methods that are  
 3322 not practical in many situations, e.g. when studying carnivores such as bears or  
 3323 large cats. Furthermore, SCR models allow us to relax many of the assumptions  
 3324 made in classical distance sampling, such as perfect detection at distance zero, and  
 3325 SCR models allow for estimates of quantities other than density, such as home range  
 3326 size, and space usage (see Chaps. 12 and 13).

#### 4.4 THE IMPLIED MODEL OF SPACE USAGE

3327 We developed the basic SCR model in terms of a latent variable,  $s$ , the home range  
 3328 center or activity center. Surely the encounter probability model, which relates  
 3329 encounter of individuals in specific traps to  $s$  must somehow imply a certain model  
 3330 for home range geometry and size. Here we explore the nature of that relationship  
 3331 and we argue that any given detection model implies a model of space usage – i.e.,  
 3332 the amount and extent of area used some prescribed percentage of the time. So we  
 3333 might say, for example, 95% of animal movements are within some distance from an  
 3334 individual’s activity center. While we have used the term “home range” or similar,  
 3335 what we really mean to imply is something that would be more clearly identified as  
 3336 resource selection or space usage (the latter term meaning resource selection, when  
 3337 the resource is only homogeneous space).

3338 Intuitively, the detection function of SCR models is related to space usage by  
 3339 individuals. Indeed, it is natural to interpret the detection model as the composite  
 3340 of two processes: movement of an individual about its home range i.e., how it uses  
 3341 space within its home range (“space usage”), and detection *conditional on use* in  
 3342 the vicinity of a trapping device. It is natural to decompose encounter probability

3343 according to:

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

3344 In practice it might make sense to think about the first component,  
 3345 i.e.,  $\Pr(\text{encounter}|\text{usage of } \mathbf{x}, \mathbf{s})$  as being a constant (e.g., if traps are located within  
 3346 arbitrarily small grid cells) and then, in that case, the encounter probability model  
 3347 is directly proportional to this model for individual movements about their home  
 3348 range center determining the use frequency of each  $\mathbf{x}$ . This is a sensible heuristic  
 3349 model for what ecologists would call a central place forager although, as we have  
 3350 stated previously, it may be meaningful as a description of transient space usage as  
 3351 well (that is, the space usage during the period of sampling).

3352 To motivate a specific model for space usage, imagine the area we are interested  
 3353 in consists of some large number of small pixels (i.e. we're looking at a discrete  
 3354 representation of space), and that we have some kind of perfect observation device  
 3355 (e.g., continuous telemetry) so that we observe every time an individual moves into  
 3356 a pixel. After a long period of time, we observe an enormous sample size of  $\mathbf{x}$   
 3357 values. We tally those up into each pixel, producing the frequency  $m(\mathbf{x}, \mathbf{s})$ , which  
 3358 is something like the "true" usage of pixel  $\mathbf{x}$  by individual with activity center  $\mathbf{s}$ . So,  
 3359 then, the usage model should be regarded as a probability mass function for these  
 3360 counts and, naturally, we regard the counts  $m(\mathbf{x}, \mathbf{s})$  as a multinomial observation  
 3361 with probabilities  $\pi(\mathbf{x}|\mathbf{s})$ , and prescribe a suitable model for  $\pi(\mathbf{x}|\mathbf{s})$  that describes  
 3362 how use events should accumulate in space. A natural null model for  $\pi(\mathbf{x}|\mathbf{s})$  has a  
 3363 decreasing probability of use as  $\mathbf{x}$  gets far away from  $\mathbf{s}$ ; i.e., animals spend more  
 3364 time close to their activity centers than far away. We can regard points used by  
 3365 the individual with activity center  $\mathbf{s}$  as the realization of a point process with  
 3366 conditional intensity:

$$\pi(\mathbf{x}|\mathbf{s}) = \frac{k(\mathbf{x}, \mathbf{s})}{\sum_x k(\mathbf{x}, \mathbf{s})} \quad (4.4.1)$$

3367 where  $k(\mathbf{x}, \mathbf{s})$  is any positive function. In continuous space, the equivalent repre-  
 3368 sentation would be:

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

3369 Clearly the space used by an individual will be proportional to whatever kernel,  
 3370  $k(\mathbf{x}, \mathbf{s})$ , we plug-in here. If we use a negative exponential function, then this pro-  
 3371 duces a standard resource selection function (RSF) model (e.g., Manly et al., 2002,  
 3372 Chapt. 8). But, here we use a Gaussian kernel, i.e.,

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

3373 so that contours of the probability of space usage resemble a bivariate normal or  
 3374 Gaussian probability distribution function.

3375 To apply this model of space-usage to SCR problems we allow for imperfect  
 3376 detection by introducing a non-uniform "thinning rate" of the true counts  $m(\mathbf{x}, \mathbf{s})$ .

3377 This yields, precisely, our Gaussian encounter probability model where the thinning  
 3378 rate is our baseline encounter probability  $p_0$  for each pixel where we place a trap,  
 3379 and  $p = 0$  in each pixel where we don't place a trap.

3380 The main take-away point here is that underlying most SCR models is some kind  
 3381 of model of space-usage, implied by the specific choice of  $k(\mathbf{x}, \mathbf{s})$ . Whether or not  
 3382 we have perfect sampling devices, the function we use in the encounter probability  
 3383 model equates to some conditional distribution of points, a utilization distribution,  
 3384 as in Eq. 5.4.1, from which we can compute effective home range area, i.e., the area  
 3385 that contains some percent of the mass of a probability distribution proportional  
 3386 to  $k(\mathbf{x}, \mathbf{s})$ ; e.g., 95% of all space used by an individual with activity center  $\mathbf{s}$ .

#### 3387 4.4.1 Bivariate normal case

3388 One encounter model that allows direct analytic computation of home range area  
 3389 is the Gaussian encounter probability model

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

3390 For this model, encounter probability is proportional to the kernel of a bivariate  
 3391 normal (Gaussian) pdf and so the natural interpretation is that in which movement  
 3392 outcomes (or successive locations of an individual) are draws from a bivariate nor-  
 3393 mal distribution with standard deviation  $\sigma$ . We say that use of this model implies  
 3394 a bivariate normal model of space usage. Under this model we can compute pre-  
 3395 cisely the effective home range area. In particular, if use outcomes are bivariate  
 3396 normal, then  $\|\mathbf{x} - \mathbf{s}\|^2$  has a chi-square distribution with 2 d.f. and the quantity  
 3397  $B(\alpha)$  that encloses  $(1 - \alpha)\%$  of all realized distances i.e.,  $\Pr(d \leq B(\alpha)) = 1 - \alpha$ ,  
 3398 is  $B(\alpha) = \sigma * \sqrt{q(\alpha, 2)}$  where  $q(\alpha, 2)$  is the 0.05 chi-square critical value on 2 df.  
 3399 For example, to compute  $q(.05, 2)$  in R we execute the command `qchisq(.95, 2)`  
 3400 which is  $q(2, \alpha) = 5.99$ . Then, for  $\sigma = 1$ ,  $B(\alpha) = 1 * \sqrt{5.99} = 2.447$ . Therefore  
 3401 95% of the points used will be within 2.447 (standard deviation) units of the home  
 3402 range center. So, in practice, we can estimate  $\sigma$  by fitting the bivariate normal  
 3403 encounter probability model to some SCR data, and then use the estimated  $\sigma$  to  
 3404 compute the “95% radius”, say  $r_{.95} = \sigma\sqrt{5.99}$ , and convert this to the 95% use area  
 3405 – the area around  $\mathbf{s}$  which contains 95% of the movement outcomes – according to  
 3406  $A_{.95} = \pi r_{.95}^2$ .

3407 An alternative bivariate normal model is the bivariate normal hazard rate model:

$$p(\mathbf{x}, \mathbf{s}) = 1 - \exp\left(-\lambda_0 * \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{x} - \mathbf{s}\|^2\right)\right) \quad (4.4.2)$$

3409 We use  $\lambda_0$  here because this parameter, the baseline encounter *rate*, can be  $> 1$ .  
 3410 This arises by assuming the latent “use frequency”  $m(\mathbf{x}, \mathbf{s})$  is a Poisson random  
 3411 variable with intensity  $\lambda_0 k(\mathbf{x}, \mathbf{s})$ . The model is distinct from our Gaussian encounter  
 3412 model  $p(\mathbf{x}, \mathbf{s}) = p_0 k(\mathbf{x}, \mathbf{s})$  used previously, although we find that they produce

3413 similar results in terms of estimates of density or 95% use area, as long as baseline  
 3414 encounter probability is low. We discuss these two formulations of the bivariate  
 3415 normal model further in Chapt. 9.

3416 **4.4.2 Empirical analysis**

3417 For any encounter model we can compute space usage quantiles empirically by  
 3418 taking a fine grid of points and either simulating movement outcomes with proba-  
 3419 bilities proportional to  $p(\mathbf{x}, \mathbf{s})$  and accumulating area around  $\mathbf{s}$ , or else we can do  
 3420 this precisely by varying  $B(\alpha)$  to find that value within which 95% of all move-  
 3421 ments are concentrated, i.e., the set of all  $\mathbf{x}$  such that  $\|\mathbf{x} - \mathbf{s}\| \leq B(q)$ . Under any  
 3422 detection model, movement outcomes will occur in proportion to  $p(\mathbf{x}, \mathbf{s})$ , as long as  
 3423 the probability of encounter is constant, *conditional on use*, and so we can define  
 3424 our space usage distribution according to:

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

3425 Given the probabilities  $\pi(\mathbf{x}, \mathbf{s})$  for all  $\mathbf{x}$  we can find the value of  $B(q)$ , for any  $q$ ,  
 3426 such that

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

3427 (here, we use  $\ni$  to mean “such that”). We have a function called `hra` in the  
 3428 `scrbook` package that computes the home range area for any encounter model and  
 3429 prescribed parameter values. The help file for `hra` has an example of simulating  
 3430 some data. The following commands illustrate this calculation for two different  
 3431 bivariate normal models of space usage:

```
3432 ##
3433 ## Define encounter probability model as R function
3434 ##
3435 > pGauss2 <- function(parms,Dmat){
3436   a0 <- parms[1]
3437   sigma <- parms[2]
3438   lp <- parms[1] -(1/(2*parms[2]*parms[2]))*Dmat*Dmat
3439   p <- 1-exp(-exp(lp))
3440   p
3441 }
3442
3443 > pGauss1 <- function(parms,Dmat){
3444   a0 <- parms[1]
3445   sigma <- parms[2]
3446   p <- plogis(parms[1])*exp( -(1/(2*parms[2]*parms[2]))*Dmat*Dmat )
3447   p
3448 }
```

```

3449
3450 ## Execute hra with sigma = .3993
3451 ## Execute hra with sigma = .3993
3452 ##
3453 > hra(pGauss1,parms=c(-2,.3993),plot=FALSE,xlim=c(0,6),ylim=c(0,6),
3454     ng=500,tol=.0005)
3455
3456 [1] 0.9784019
3457 radius to achieve 95% of area: 0.9784019
3458 home range area: 3.007353
3459 [1] 3.007353
3460
3461
3462 ## Analytic solution:
3463 ##      true sigma that produces area of 3
3464 > sqrt(3/pi)/sqrt(5.99)
3465 [1] 0.3992751

```

What this means is that  $B(q) = 0.978$  is the radius that encloses about 95% of all movements under the standard bivariate normal encounter model. Therefore, the area is about  $\pi * .978 * .978 = 3.007$  spatial units. You can change the intercept of the model and find that it has no effect. The true (analytic) value of  $\sigma$  that produces a home range area of 3.0 is 0.3993 which is the value we initially plugged in to the `hra` function. We can improve on the numerical approximation to home range area (get it closer to 3.0) by increasing the resolution of our spatial grid (increase the `ng` argument) along with the `tol` argument.

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

```

3481 > hra(pGauss1,parms=c(-2,.3993),plot=FALSE,xlim,ylim,ng=500,
3482     target.area=3.00735,tol=.0005)
3483
3484 Value of parm[2] to achieve 95% home range area of 3.00735: 0.3993674

```

#### 3485 4.4.3 Relevance of understanding space usage

3486 One important reason that we need to be able to deduce “home range area” from  
3487 a detection model is so that we can compare different models with respect to a  
3488 common biological currency. Many encounter probability models have some “scale  
3489 parameter”, which we might call  $\sigma$  no matter the model, but this relates to 95%  
3490 area in a different manner under each model. Therefore, we want to be able to

3491 convert different models to the same currency. Another reason to understand the  
3492 relationship between models of encounter probability and space usage is that it  
3493 opens the door to combining traditional resource selection data from telemetry  
3494 with spatial capture-recapture data. In Chapt. 13 we consider this problem, for  
3495 the case in which a sample of individuals produces encounter history data suitable  
3496 for SCR models and, in addition, we have telemetry relocations on a sample of  
3497 individuals. This is achieved by regarding the two sources of data as resulting from  
3498 the same underlying process of space usage but telemetry data produce “perfect”  
3499 observations, like always-on camera traps blanketing a landscape. We use this idea  
3500 to model the effect of a measured covariate at each pixel, say  $C(\mathbf{x})$ , on home range  
3501 size and geometry and, hence, the probability of encounter in traps.

3502 **4.4.4 Contamination due to behavioral response**

3503 Interpretation of encounter probability models as models of animal home range and  
3504 space usage can be complicated by a number of factors, including whether traps are  
3505 baited or not. In the case of baited traps, this might lead to a behavioral response  
3506 (Sec. 7.2.3) which could affect animal space usage. For example, if traps attract  
3507 animals from a long distance, it could make typical home ranges appear larger than  
3508 normal. More likely, in our view, it wouldn’t change the typical size of a range but  
3509 would change how individuals use their range e.g., by moving from baited trap to  
3510 baited trap, so that observed movement distances of individuals are typically larger  
3511 than normal.

3512 In other cases, the reliance on Euclidean distance in models for encounter prob-  
3513 ability might be unrealistic, and can lead to biased estimates of density (Royle  
3514 et al., 2012a). For example, animals might concentrate their movements along  
3515 trails, roads, or other landscape features. In this case, models that accommodate  
3516 other distance metrics can be considered. We present models based on least-cost  
3517 path in Chapt. 12.

## 4.5 SIMULATING SCR DATA

3518 It is always useful to simulate data because it allows you to understand the system  
3519 that you’re modeling and also calibrate your understanding with specific values of  
3520 the model parameters. That is, you can simulate data using different parameter  
3521 values until you obtain data that “look right” based on your knowledge of the spe-  
3522 cific situation that you’re interested in. Here we provide a simple script to illustrate  
3523 how to simulate spatial encounter history data. In this exercise we simulate data  
3524 for 100 individuals and a 25 trap array laid out in a  $5 \times 5$  grid of unit spacing.  
3525 The specific encounter model is the Gaussian model given above and we used this  
3526 code to simulate data used in subsequent analyses. The 100 activity centers were  
3527 simulated on a state-space defined by a  $8 \times 8$  square within which the trap array

3528 was centered (thus the trap array is buffered by 2 units). Therefore, the density of  
 3529 individuals in this system is fixed at 100/64.

```

3530 > set.seed(2013)
3531 # Create 5 x 5 grid of trap locations with unit spacing
3532 > traplocs <- cbind(sort(rep(1:5,5)),rep(1:5,5))
3533 > ntraps <- nrow(traplocs)
3534 # Compute distance matrix:
3535 > Dmat <- e2dist(traplocs,traplocs)
3536
3537
3538 # Define state-space of point process. (i.e., where animals live).
3539 # "buffer" just adds a fixed buffer to the outer extent of the traps.
3540 #
3541 > buffer <- 2
3542 > xlim <- c(min(traplocs[,1] - buffer),max(traplocs[,1] + buffer))
3543 > ylim <- c(min(traplocs[,2] - buffer),max(traplocs[,2] + buffer))
3544
3545 > N <- 100    # population size
3546 > K <- 20    # number nights of effort
3547
3548 > sx <- runif(N,xlim[1],xlim[2])    # simulate activity centers
3549 > sy <- runif(N,ylim[1],ylim[2])
3550 > S <- cbind(sx,sy)
3551 # Compute distance matrix:
3552 > D <- e2dist(S,traplocs) # distance of each individual from each trap
3553
3554 > alpha0 <- -2.5      # define parameters of encounter probability
3555 > sigma <- 0.5        # scale parameter of half-normal
3556 > alpha1 <- 1/(2*sigma*sigma) # convert to coefficient on distance
3557
3558 # Compute Probability of encounter:
3559 #
3560 > probcap <- plogis(-2.5)*exp( - alpha1*D*D)
3561
3562 # Generate the encounters of every individual in every trap
3563 > Y <- matrix(NA,nrow=N,ncol=ntraps)
3564 > for(i in 1:nrow(Y)){
3565   Y[i,] <- rbinom(ntraps,K,probcap[i,])
3566 }
```

3567 We remind the reader that, in presenting **R** or other code snippets throughout  
 3568 the book, we will deviate from our standard variable expressions for some quantities.  
 3569 In particular, we sometimes substitute words for integer variable designations: `nind`  
 3570 (for  $n$ ), `ntraps` (for  $J$ ), and `nocc` (for  $K$ ). In our opinion this leaves less to be  
 3571 inferred by the reader in trying to understand code snippets.

3572 Subsequently we will generate data using this code packaged in an **R** function  
 3573 called **simSCR0** in the package **scrbook** which takes a number of arguments includ-  
 3574 ing **discard0** which, if TRUE, will return only the encounter histories for captured in-  
 3575 dividuals. A second argument is **array3d** which, if TRUE, returns the 3-dimensional  
 3576 encounter history array instead of the aggregated **nind**  $\times$  **ntraps** encounter frequen-  
 3577 cies (see below). Finally we provide a random number seed, **rnd** = 2013 to ensure  
 3578 repeatability of the analysis here. We obtain a data set as above using the following  
 3579 command:

```
3580 > data <- simSCR0(discard0=TRUE, array3d=FALSE, rnd=2013)
```

3581 The **R** object **data** is a list, so let's take a look at what's in the list and then harvest  
 3582 some of its elements for further analysis below.

```
3583 > names(data)
3584 [1] "Y"      "traplocs" "xlim"      "ylim"      "N"       "alpha0"    "beta"
3585 [8] "sigma"   "K"
3586
3587 ## Grab encounter histories from simulated data list
3588 > Y <- data$Y
3589 ## Grab the trap locations
3590 > traplocs <- data$traplocs
```

### 3591 4.5.1 Formatting and manipulating real data sets

3592 Conventional capture-recapture data are easily stored and manipulated as a 2-  
 3593 dimensional array, an **nind**  $\times$  **K** (individuals by sample occasions) matrix, which is  
 3594 maximally informative for any conventional capture-recapture model, but not for  
 3595 spatial capture-recapture models. For SCR models we must preserve the spatial  
 3596 information in the encounter history information. We will routinely analyze data  
 3597 from 3 standard formats:

- 3598 (1) The basic 2-dimensional data format, which is an **nind**  $\times$  **ntraps** encounter  
 3599 frequency matrix such as that simulated previously. These are the total number  
 3600 of encounters in each trap, summed over the **K** sample occasions.
- 3601 (2) The maximally informative 3-dimensional array, for which we establish here  
 3602 the convention that it has dimensions **nind**  $\times$  **ntraps**  $\times$  **K**.
- 3603 (3) We use a compact format – the “encounter data file” – which we describe below  
 3604 in Sec. 5.9.

3605 To simulate data in the most informative format - the “3-d array” - we can use the  
 3606 **R** commands given previously but replace the last 4 lines with the following:

```
3607 > Y <- array(NA,dim=c(N,ntraps,K))
3608
3609 > for(i in 1:nrow(Y)){
3610     for(j in 1:ntraps){
```

---

```

3611     Y[i,j,1:K] <- rbinom(K,1,probcap[i,j])
3612   }
3613 }
```

3614     We see that a collection of  $K$  binary encounter events are generated for *each* individual and for *each* trap. The probabilities of those Bernoulli trials are computed  
 3615 based on the distance from each individual's home range center and the trap (see  
 3616 calculation above), and those are housed in the matrix `probcap`. Our data simula-  
 3617 tor function `simSCR0` will return the full 3-d array if `array3d=TRUE` is specified in  
 3618 the function call. To recover the 2-d matrix from the 3-d array, and subset the 3-d  
 3619 array to individuals that were captured, we do this:

```

3621 # Sum over the "replicates" dimension (3rd margin of the array)
3622 > Y2d <- apply(Y,c(1,2),sum)
3623
3624 # Compute how many times each individual was captured
3625 > ncaps <- apply(Y2d,1,sum)
3626
3627 # Keep those individuals that were captured
3628 > Y <- Y[ncaps>0,,]
```

## 4.6 FITTING MODEL SCR0 IN BUGS

3629 Clearly if we somehow knew the value of  $N$  then we could fit this model directly  
 3630 because, in that case, it is a special kind of logistic regression model, one with a  
 3631 random effect (`s`) that enters into the model in a peculiar fashion, and also with  
 3632 a distribution (uniform) which we don't usually think of as standard for random  
 3633 effects models. So our aim here is to analyze the known- $N$  problem, using our  
 3634 simulated data, as an incremental step in our progress toward fitting more generally  
 3635 useful models. To begin, we use our simulator to grab a data set and then harvest  
 3636 the elements of the resulting object for further analysis.

```

3637 > data <- simSCR0(discard0=FALSE,rnd=2013)
3638 > y <- data$Y
3639 > traplocs <- data$traplocs
3640
3641 # In this case nind=N because we're doing the known-N problem
3642 #
3643 > nind <- nrow(y)
3644 > X <- data$traplocs
3645 > J <- nrow(X) # number of traps
3646 > K <- data$K
3647 > xlim <- data$xlim
3648 > ylim <- data$ylim
```

3649     Note that we specify `discard0 = FALSE` so that we have a “complete” data set,  
 3650 i.e., one with the all-zero encounter histories corresponding to uncaptured individuals.  
 3651 Now, within an **R** session, we can create the **BUGS** model file and fit the  
 3652 model using the following commands.

```
3653 cat("
3654   model{
3655     alpha0 ~ dnorm(0,.1)
3656     logit(p0) <- alpha0
3657     alpha1 ~ dnorm(0,.1)
3658     sigma <- sqrt(1/(2*alpha1))
3659     for(i in 1:N){    # note N here -- N is KNOWN in this example
3660       s[i,1] ~ dunif(xlim[1],xlim[2])
3661       s[i,2] ~ dunif(ylim[1],ylim[2])
3662       for(j in 1:J){
3663         d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
3664         y[i,j] ~ dbin(p[i,j],K)
3665         p[i,j] <- p0*exp(- alpha1*d[i,j]*d[i,j])
3666       }
3667     }
3668   }
3669 ",file = "SCR0a.txt")
```

3670 This model describes the Gaussian encounter probability model, but it would be  
 3671 trivial to modify that to various others including the logistic described above. One  
 3672 consequence of using the half-normal is that we have to constrain the encounter  
 3673 probability to be in  $[0, 1]$  which we do here by defining `alpha0` to be the logit of the  
 3674 intercept parameter `p0`. Note that the distance covariate is computed within the  
 3675 **BUGS** model specification given the matrix of trap locations, `X`, which is provided  
 3676 to **WinBUGS** as data.

3677 Next we do a number of organizational activities including bundling the data for  
 3678 **WinBUGS**, defining some initial values, the parameters to monitor and some basic  
 3679 MCMC settings. We choose initial values for the activity centers `s` by generating  
 3680 uniform random numbers in the state-space but, for the observed individuals, we  
 3681 replace those values by each individual’s mean trap coordinate for all encounters

```
3682 ### Starting values for activity centers, s
3683 > sst <- cbind(runif(nind,xlim[1],xlim[2]),runif(nind,ylim[1],ylim[2]))
3684 > for(i in 1:nind){
3685   if(sum(y[i,])==0) next
3686   sst[i,1] <- mean( X[y[i,>0,1] )
3687   sst[i,2] <- mean( X[y[i,>0,2] )
3688 }
3689 > data <- list (y=y, X=X, K=K, N=nind, J=J, xlim=xlim, ylim=ylim)
3690 > inits <- function(){
3691   list (alpha0=rnorm(1,-4,.4), alpha1=runif(1,1,2), s=sst)
```

---

```

3693   }
3694
3695 > library(R2WinBUGS)
3696 > parameters <- c("alpha0","alpha1","sigma")
3697 > out <- bugs (data, inits, parameters, "SCR0a.txt", n.thin=1, n.chains=3,
3698           n.burnin=1000,n.iter=2000,debug=TRUE,working.dir=getwd())

```

3699 There is little to say about the preceding operations other than to suggest that  
 3700 you might explore the output and investigate additional analyses by running the  
 3701 `simSCR0` script provided in the **R** package `scrbook`.

3702 For purposes here, we ran 1000 burn-in and 1000 post-burn-in iterations, and  
 3703 3 chains, to obtain 3000 posterior samples. Because we know  $N$  for this particular  
 3704 data set we only have 2 parameters of the detection model to summarize (`alpha0`  
 3705 and `alpha1`), along with the derived parameter  $\sigma$ , the scale parameter of the Gaus-  
 3706 sian kernel, i.e.,  $\sigma = \sqrt{1/(2\alpha_1)}$ . When the object `out` is produced we print a  
 3707 summary of the results as follows:

```

3708 > print(out,digits=2)
3709 Inference for Bugs model at "SCR0a.txt", fit using WinBUGS,
3710   3 chains, each with 2000 iterations (first 1000 discarded)
3711   n.sims = 3000 iterations saved
3712     mean    sd   2.5%   25%   50%   75% 97.5% Rhat n.eff
3713 alpha0   -2.50  0.22  -2.95  -2.65  -2.48  -2.34  -2.09 1.01  190
3714 alpha1    2.44  0.42   1.64   2.15   2.44   2.72   3.30 1.00  530
3715 sigma     0.46  0.04   0.39   0.43   0.45   0.48   0.55 1.00  530
3716 deviance 292.80 21.16 255.60 277.50 291.90 306.00 339.30 1.01  380
3717
3718
3719 [...some output deleted...]
3720

```

3721 We know the data were generated with `alpha0 = -2.5` and `alpha1 = 2`. The  
 3722 estimates look reasonably close to those data-generating values and we probably feel  
 3723 pretty good about the performance of the Bayesian analysis and MCMC algorithm  
 3724 that **WinBUGS** cooked-up based on our sample size of 1 data set. It is worth  
 3725 noting that the `Rhat` statistics indicate reasonable convergence but, as a practical  
 3726 matter, we might choose to run the MCMC algorithm for additional time to bring  
 3727 these closer to 1.0 and to increase the effective posterior sample size (`n.eff`). Other  
 3728 summary output includes “deviance” and related things including the deviance  
 3729 information criterion (DIC). We discuss general issues of convergence and other  
 3730 MCMC considerations in Chapt. 17, and DIC and model selection in Chapt. 8.

## 4.7 UNKNOWN N

3731 In all real applications  $N$  is unknown. We handled this important issue in Chapt.  
 3732 4 using the method of data augmentation (DA) which we apply here to achieve

3733 a realistic analysis of model SCR0. As with the basic closed population models  
 3734 considered previously, we formulate the problem by augmenting our observed data  
 3735 set with a number of “all-zero” encounter histories - what we referred to in Chapt. 4  
 3736 as potential individuals. If  $n$  is the number of observed individuals, then let  $M - n$   
 3737 be the number of potential individuals in the data set. For the 2-dimensional  
 3738  $y_{ij}$  data structure ( $n$  individual  $\times$   $J$  traps encounter frequencies) we simply add  
 3739 additional rows of all-zero observations to that data set. Because such “individuals”  
 3740 are unobserved, they therefore necessarily have  $y_{ij} = 0$  for all  $j$ . A data set, say  
 3741 with 4 traps and 6 individuals, augmented with 4 pseudo-individuals therefore might  
 3742 look like this:

```
3743     trap1 trap2 trap3 trap4
3744 [1,]    1    0    0    0
3745 [2,]    0    2    0    0
3746 [3,]    0    0    0    1
3747 [4,]    0    1    0    0
3748 [5,]    0    0    1    1
3749 [6,]    1    0    1    0
3750 [7,]    0    0    0    0
3751 [8,]    0    0    0    0
3752 [9,]    0    0    0    0
3753 [10,]   0    0    0    0
```

3754 We typically have more than 4 traps and, if we’re fortunate, many more individuals  
 3755 in our data set.

3756 For the augmented data set, we introduce a set of binary latent variables (the  
 3757 data augmentation variables),  $z_i$ , and the model is extended to describe  $\Pr(z_i = 1)$   
 3758 which is, in the context of this problem, the probability that an individual in the  
 3759 augmented data set is a member of the population of size  $N$  that was exposed to  
 3760 sampling. In other words, if  $z_i = 1$  for one of the all-zero encounter histories, this is  
 3761 implied to be a sampling zero whereas observations for which  $z_i = 0$  are “structural  
 3762 zeros” under the model. Under DA, we also express the binomial observation model  
 3763 *conditional on  $z_i$*  as follows:

$$y_{ij}|z_i \sim \text{Binomial}(K, z_i p_{ij})$$

3764 where we see that the binomial probability evaluates to 0 if  $z_i = 0$  (so  $y_{ij}$  is a fixed  
 3765 0 in that case) and evaluates to  $p_{ij}$  if  $z_i = 1$ .

3766 How big does the augmented data set have to be? We discussed this issue  
 3767 in Chapt. 4 where we noted that the size of the data set is equivalent to the  
 3768 upper limit of a uniform prior distribution on  $N$ . Practically speaking, it should  
 3769 be sufficiently large so that the posterior distribution for  $N$  is not truncated. On  
 3770 the other hand, if it is too large then unnecessary calculations are being done. An  
 3771 approach to choosing  $M$  by trial-and-error is indicated. Do a short MCMC run  
 3772 and then consider whether you need to increase  $M$ . See Chapt. 17 for an example  
 3773 of this. Kéry and Schaub (2012, Chapt. 6) provide an assessment of choosing  $M$

3774 in closed population models. The useful thing about DA is that it removes  $N$  as  
 3775 an explicit parameter of the model. Instead,  $N$  is a derived parameter, computed  
 3776 by  $N = \sum_{i=1}^M z_i$ . Similarly, *density*,  $D$ , is also a derived parameter computed as  
 3777  $D = N/\text{area}(\mathcal{S})$ .

#### 3778 4.7.1 Analysis using data augmentation in WinBUGS

3779 We provide a complete **R** script for simulating and organizing a data set, and  
 3780 analyzing the data in **WinBUGS**. As before we begin by obtaining a data set  
 3781 using our **simSCR0** function and then harvesting the required data objects from the  
 3782 resulting data list. Note that we use the **discard0=TRUE** option this time so that  
 3783 we get a “real looking” data set with no all-zero encounter histories:

```
3784 ##  

3785 ## Simulate the data and extract the required objects  

3786 ##  

3787 > data <- simSCR0(discard0=TRUE,rnd=2013)  

3788 > y <- data$Y  

3789 > nind <- nrow(y)  

3790 > X <- data$traplocs  

3791 > K <- data$K  

3792 > J <- nrow(X)  

3793 > xlim <- data$xlim  

3794 > ylim <- data$ylim
```

3795 After harvesting the data we augment the data matrix  $y$  with  $M - n$  all-zero  
 3796 encounter histories, and create starting values for the variables  $z_i$  and also the  
 3797 activity centers  $s_i$  of which, for each, we require  $M$  values. One thing to take care  
 3798 of in using the **BUGS** engines is the starting values for the activity centers. It is  
 3799 usually helpful to start the  $s_i$  for each observed individual at or near the trap(s) it  
 3800 was captured. All of this happens as follows:

```
3801 ## Data augmentation  

3802 > M <- 200  

3803 > y <- rbind(y,matrix(0,nrow=M-nind,ncol=ncol(y)))  

3804 > z <- c(rep(1,nind),rep(0,M-nind))  

3805  

3806 ## Starting values for s  

3807 > sst <- cbind(runif(M,xlim[1],xlim[2]),runif(M,ylim[1],ylim[2]))  

3808 > for(i in 1:nind){  

3809   sst[i,1] <- mean( X[y[i,]>0,1] )  

3810   sst[i,2] <- mean( X[y[i,]>0,2] )  

3811 }
```

3812 Next, we write out the **BUGS** model specification and save it to an external  
 3813 file called **SCR0b.txt**. The model specification now includes  $M$  encounter histories  
 3814 including the augmented potential individuals, the data augmentation parameters  
 3815  $z_i$ , and the data augmentation parameter  $\psi$ :

```

3816 > cat("
3817 model{
3818   alpha0 ~ dnorm(0,.1)
3819   logit(p0) <- alpha0
3820   alphai ~ dnorm(0,.1)
3821   sigma <- sqrt(1/(2*alphai))
3822   psi ~ dunif(0,1)
3823
3824   for(i in 1:M){
3825     z[i] ~ dbern(psi)
3826     s[i,1] ~ dunif(xlim[1],xlim[2])
3827     s[i,2] ~ dunif(ylim[1],ylim[2])
3828     for(j in 1:J){
3829       d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
3830       y[i,j] ~ dbin(p[i,j],K)
3831       p[i,j] <- z[i]*p0*exp(- alphai*d[i,j]*d[i,j])
3832     }
3833   }
3834   N <- sum(z[])
3835   D <- N/64
3836 }
3837 ",file = "SCR0b.txt")

```

3838     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:

```

3841 > data <- list (y=y, X=X, K=K, M=M, J=J, xlim=xlim, ylim=ylim)
3842 > inits <- function(){
3843   list (alpha0=rnorm(1,-4,.4), alpha1=runif(1,1,2), s=sst, z=z)
3844 }
3845
3846 > library(R2WinBUGS)
3847 > parameters <- c("alpha0","alpha1","sigma","N","D")
3848 > out <- bugs (data, inits, parameters, "SCR0b.txt", n.thin=1,n.chains=3,
3849   n.burnin=1000,n.iter=2000,debug=TRUE,working.dir=getwd())

```

3850     Note the differences in this new **WinBUGS** model with that appearing in the  
3851 known- $N$  version – there are not many! The loop over individuals goes up to  $M$   
3852 now, and there is a model component for the DA variables  $z$ . We are also computing  
3853 some derived parameters: population size  $N(\mathcal{S})$  is computed by summing up all of  
3854 the data augmentation variables  $z_i$  (as we've done previously in Chapt. 4) and  
3855 density,  $D$ , is also a derived parameter, being a function of  $N$ . The input data has  
3856 changed slightly too, as the augmented data set has more rows to include excess  
3857 all-zero encounter histories. Previously we knew that  $N = 100$  but in this analysis  
3858 we pretend not to know  $N$ , but think that  $N = 200$  is a good upper bound. This

3859 analysis can be run directly using the `SCR0bayes` function once the `scrbook` package  
 3860 is loaded, by issuing the following commands:

```
3861 > library(scrbook)
3862 > data <- simSCR0(discard0=TRUE, rnd=2013)
3863 > out1 <- SCR0bayes(data, M=200, engine="winbugs", ni=2000, nb=1000)
```

3864 Summarizing the output from **WinBUGS** produces:

```
3865 > print(out1,digits=2)
3866 Inference for Bugs model at "SCR0b.txt", fit using WinBUGS,
3867 3 chains, each with 2000 iterations (first 1000 discarded)
3868 n.sims = 3000 iterations saved
3869      mean    sd   2.5%   25%   50%   75% 97.5% Rhat n.eff
3870 alpha0  -2.57  0.23 -3.04 -2.72 -2.56 -2.41 -2.15 1.01   320
3871 alpha1   2.46  0.42  1.63  2.16  2.46  2.73  3.33 1.02   120
3872 sigma    0.46  0.04  0.39  0.43  0.45  0.48  0.55 1.02   120
3873 N       113.62 15.73 86.00 102.00 113.00 124.00 147.00 1.01   260
3874 D        1.78  0.25  1.34  1.59  1.77  1.94  2.30 1.01   260
3875 deviance 302.60 23.67 261.19 285.47 301.50 317.90 354.91 1.00   1400
3876
3877 [...some output deleted...]
3878
```

3879 The `Rhat` statistic (discussed in Secs. 3.5.2 and 17.4.5) for this analysis indicates  
 3880 satisfactory convergence. We see that the estimated parameters ( $\alpha_0$  and  $\alpha_1$ ) are  
 3881 comparable to the previous results obtained for the known- $N$  case, and also not  
 3882 too different from the data-generating values. The posterior of  $N$  overlaps the  
 3883 data-generating value substantially.

#### 3884 Use of other BUGS engines: JAGS

3885 There are two other popular **BUGS** engines in widespread use: **OpenBUGS**  
 3886 (Thomas et al., 2006) and **JAGS** (Plummer, 2003). Both of these are easily called  
 3887 from **R**. **OpenBUGS** can be used instead of **WinBUGS** by changing the package  
 3888 option in the `bugs` call to `package='OpenBUGS'`. **JAGS** can be called using  
 3889 the function `jags()` in package `R2jags` which has nearly the same arguments as  
 3890 `bugs()`. Or, it can be executed from the **R** package `rjags` (Plummer, 2011) which  
 3891 has a slightly different implementation that we demonstrate here as we reanalyze  
 3892 the simulated data set in the previous section (note: the same **R** commands are used  
 3893 to generate the data and package the data, inits and parameters to monitor). The  
 3894 function `jags.model` is used to initialize the model and run the MCMC algorithm  
 3895 for an adaptive period during which tuning of the MCMC algorithm might take  
 3896 place. These samples cannot be used for inference. Then the Markov chains are  
 3897 updated using `coda.samples()` to obtain posterior samples for analysis, as follows:

```
3898 > jinit <- jags.model("SCR0b.txt", data=data, inits=inits,
3899                           n.chains=3, n.adapt=1000)
3900 > jout <- coda.samples(jinit, parameters, n.iter=1000, thin=1)
```

**Table 4.2.** 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

3901 These commands can be executed using the function `SCR0bayes` provided with  
 3902 the **R** package `scrbook`. Hobbs (2011) provides a good introduction to ecological  
 3903 modeling with **JAGS** which we recommend.

3904 **4.7.2 Implied home range area**

3905 Here we apply the method described in Sec. 5.4 to compute the effective home  
 3906 range area under different encounter probability models fit to simulated data. We  
 3907 simulated a data set from the Gaussian kernel model as in Sec. 5.7 and then we  
 3908 fitted 4 models to it: (1) the true data-generating Gaussian encounter probability  
 3909 model; (2) the “hazard” or complementary log-log link model (Eq. 5.4.2); (3) the  
 3910 negative exponential model and (4) the logit model (Eq. 5.2.2). We modified the  
 3911 function `SCR0bayes` for this purpose which you should be able to do with little  
 3912 difficulty. We fit each model to the same simulated data set using **WinBUGS**,  
 3913 based only on 1000 post-burn-in samples and 3 chains, which produced the posterior  
 3914 summaries given in Table 5.2. The main thing we see is that, while the implied home  
 3915 range area can vary substantially, there are smaller differences in the estimated  $N$   
 3916 and hence  $D$ .

3917 **4.7.3 Realized and expected density**

3918 In Bayesian analysis of the SCR model, we estimate a parameter  $N$  which is the  
 3919 size of the population for the prescribed state-space (presumably the state-space is  
 3920 defined so as to be relevant to where our traps were located, so  $N$  can be thought  
 3921 of as the size of the sampled population). In the context of Efford and Fewster  
 3922 (2012) this is the *realized* population size. Conversely, sometimes we see estimates  
 3923 of *expected* population size reported, which are estimates of  $\mathbb{E}(N)$ , the expected  
 3924 size of some hypothetical, unspecified population. Usually the distinction between  
 3925 realized and expected population size is not made in SCR models, because almost  
 3926 everyone only cares about actual populations – and their realized population size.

3927 If you do likelihood analysis of SCR models, then the distinction between re-  
 3928 alized and expected is often discussed by whether the estimator is “conditional on  
 3929  $N$ ” (realized) or not (expected). The naming arises because in obtaining the MLE

of  $N$ , its properties are evaluated *conditional* on  $N$  – in particular, if the estimator is unbiased then  $\mathbb{E}(\hat{N}|N) = N$  and  $\text{Var}(\hat{N}|N) = \tilde{\sigma}_{\hat{N}}^2$  is the sampling variance. This does not conform to any concept or quantity that is relevant to Bayesian inference. If we care about  $N$  for the population that we sampled it is understood to be a realization of a random variable, but the relevance of “conditional on  $N$ ” is hard to see. Bayesian analysis will provide a prediction of  $N$  that is based on the posterior  $[N|y, \theta]$  – which is certainly *not* conditional on  $N$ .

There is a third type of inference objective that is relevant in practice and that is prediction of  $N$  for a population that was not sampled – i.e., a “new” population. To elaborate on this, consider a situation in which we are concerned about the tiger population in 2 distinct reserves in India. We do a camera trapping study on one of the reserves to estimate  $N_1$  and we think the reserves are similar and homogeneous so we’re willing to apply a density estimate based on  $N_1$  to the 2nd reserve. For the 2nd reserve, do we want a prediction of the realized population size,  $N_2$ , or do we want an estimates of its expected value? We believe the former is the proper quantity for inference about the population size in the 2nd reserve. An estimate of  $N_2$  should include the uncertainty with which the mean is estimated (from reserve 1) and it should also include “process variation” for making the prediction of the latent variable  $N_2$ .

As a practical matter, to do a Bayesian analysis of this you could just define the state-space to be the union of the two state-spaces, increase  $M$  so that the posterior of the total population size is not truncated, and then have MCMC generate a posterior sample of individuals on the joint state-space. You can tally-up the ones that are on  $S_2$  as an estimate of  $N_2$ . Alternatively, we can define  $\mu = \psi M / A_1$  and then simulate posterior samples of  $N_s \sim \text{Binomial}(M, \mu A_2 / M)$  for the new state-space area,  $A_2$ .

To carry out a classical likelihood analysis of this 2nd type of problem, what should we do? The argument for making a prediction of a new value of  $N$  would go something like this: If you obtain an MLE of  $N$ , say  $\hat{N}$ , then the inference procedure tells us the variance of this *conditional* on  $N$ . i.e.,  $\text{Var}(\hat{N}|N)$ . This is fine, if we care about the specific value of  $N$  that generated our data set. However, if we don’t care about the specific one in question then we want to “uncondition” on  $N$  to introduce a new variance component. Law of total variance says:

$$\text{Var}(\hat{N}) = \mathbb{E}[\text{Var}(\hat{N}|N)] + \text{Var}[\mathbb{E}(\hat{N}|N)]$$

If  $\hat{N}$  is unbiased then we say the unconditional variance is

$$\text{Var}(\hat{N}) = \sigma_{\hat{N}}^2 + \text{Var}(N)$$

The first part is estimation error and the 2nd component is the “process variance.” If you do Bayesian analysis, then you don’t have to worry too much about how to compute variances properly. You decide if you care about  $N$ , or its expected value,

3967 or predictions of some “new”  $N$ , and you tabulate the correct posterior distribution  
 3968 from your MCMC output.

3969 The considerations for estimating density are the same. Density can be  $N/A$   
 3970 where  $N$  is the realized population, which we understand it to be unless we put an  
 3971 expectation operator around the  $N$  like  $\mathbb{E}(N)/A$ . Classically, density is thought of  
 3972 as being defined as the expected value of  $N$  but this might not always be meaningful  
 3973 because the context of whether we mean realized density, of an actual population,  
 3974 or expected density for some hypothetical unspecified population, should matter.  
 3975 The formula for obtaining “expected density” is slightly different depending on  
 3976 whether we assume  $N$  has a Poisson distribution or whether we assume a binomial  
 3977 distribution (under data augmentation). In the latter case  $\psi$  is related to the point  
 3978 process intensity (see Chapt. 11) in the sense that, under the binomial prior:

$$\mathbb{E}(N) = M \times \psi$$

3979 so, what we think of as “density”,  $D$ , is  $D = M\psi/A$ . Under the Poisson point  
 3980 process model we have:

$$\mathbb{E}(N) = D \times A.$$

3981 In summary, there are 3 basic inference problems that relate to estimating pop-  
 3982 ulation size (or density):

- 3983 (1) What is the value of  $N$  for some population that was sampled. This is what  
 3984 Efford and Fewster call “realized  $N$ ” In general, we want the uncertainty to reflect  
 3985 having to estimate  $n_0$ , the part of the population not seen.
- 3986 (2) We need to estimate  $N$  for some population that we didn’t sample but it is  
 3987 “similar” to the population that we have information on. In this case, we have to  
 3988 account for both variation in having to estimate parameters of the distribution of  
 3989  $N$  and we have to account for process variation in  $N$  (i.e., due to the stochastic  
 3990 model of  $N$ ).
- 3991 (3) In some extremely limited cases we might care about estimating the expected  
 3992 value of  $N$ ,  $\mathbb{E}(N)$ . This is only useful as a hypothetical statement that we might  
 3993 use, e.g., if we were to establish a new million ha refuge somewhere, then we  
 3994 might say its expected population size is 200 tigers.

## 4.8 THE CORE SCR ASSUMPTIONS

3995 It’s always a good idea to sit down and reflect on the meaning of any particular  
 3996 model, its various assumptions, and what they mean in a specific context. From  
 3997 the statistician’s point of view, the basic assumption, the omnibus assumption, as  
 3998 in all of statistics, and for every statistical model, is that “the model is correctly  
 3999 specified”. So, naturally, that precludes everything that isn’t explicitly addressed  
 4000 by the model. To point this out to someone seems to cause a lot of anxiety, so we  
 4001 enumerate here what we think are the most important statistical assumptions of  
 4002 the basic SCR0 model:

- 4003 • **Demographic closure.** The model does not allow for demographic processes.  
4004 There is no recruitment or entry into the sampled population. There is no mor-  
4005 tality or exit from the sampled population.
- 4006 • **Geographic closure.** We assume no permanent emigration or immigration  
4007 from the state-space. However, we allow for “temporary” movements around  
4008 the state-space and variable exposure to encounter as a result. The whole point  
4009 of SCR models is to accommodate this dynamic. In ordinary capture-recapture  
4010 models we have to assume geographic closure to interpret  $N$  in a meaningful way.
- 4011 • **Activity centers are randomly distributed.** That is, uniformity and inde-  
4012 pendence of the underlying point process  $\mathbf{s}_1, \dots, \mathbf{s}_N$  (see next section).
- 4013 • **Detection is a function of distance.** A detection model that describes how  
4014 encounter probability declines as a function of distance from an individual’s home  
4015 range center.
- 4016 • **Independence of encounters** among individuals. Encounter of any individual  
4017 is independent of encounter of each other individual.
- 4018 • **Independence of encounters** of the same individual. Encounter of an individ-  
4019 ual in any trap is independent of its encounter in any other trap, and subsequent  
4020 sample occasion.

4021 It’s easy to get worried and question the whole SCR enterprise just on the grounds  
4022 that these assumptions combine to form such a simplistic model, one that surely  
4023 can’t describe the complexity of real populations. On this sentiment, a few points  
4024 are worth making. First, you don’t have inherently fewer assumptions by using an  
4025 ordinary capture-recapture model but, rather, the SCR model relaxes a number of  
4026 important assumptions compared to the non-spatial counterpart. For one, here,  
4027 we’re not assuming that  $p$  is constant for all individuals but rather that  $p$  varies  
4028 substantially as a matter of the spatial juxtaposition of individuals with traps. So  
4029 maybe the manner in which  $p$  varies isn’t quite right, but that’s not an argument  
4030 that supports doing less modeling. Fundamentally a distance-based model for  $p$  has  
4031 some basic biological justification in virtually every capture-recapture study. Sec-  
4032 ondly, for some of these core assumptions such as uniformity, and independence of  
4033 individuals and of encounters, we expect a fair amount of robustness to departures.  
4034 They function primarily to allow us to build a model and an estimation scheme and  
4035 we don’t usually think they represent real populations (of course, no model does!).  
4036 Third, we can extend these assumptions in many different ways and we do that  
4037 to varying extents in this book, and more work remains to be done in this regard.  
4038 Forth, we can also evaluate the reasonableness of the assumptions formally in some  
4039 cases using standard methods of assessing model fit (Chapt. 8).

4040 Finally, we return back to our sentiment about the omnibus assumptions which  
4041 is that the model is properly specified. This precludes *everything* that isn’t in  
4042 the model. Sometimes you see in capture-recapture literature statements like “we  
4043 assume no marks are lost”, “marks are correctly identified” and similar things. We  
4044 might as well also assume that, a shopping mall is not built, or a meteor does not

4045 crash down into our study area, the sun does not go super-nova, and so forth. Our  
4046 point is that we should separate statistical assumptions about model parameters or  
4047 aspects of the probability model from what are essentially logistical or operational  
4048 assumptions about how we interpret our data, or based on our ability to conduct  
4049 the study. It is pointless to enumerate all of the possible explanations for apparent  
4050 departures, because there are an infinity of such cases.

## 4.9 WOLVERINE CAMERA TRAPPING STUDY

4051 We provide an illustration of some of the concepts we've introduced previously  
4052 in this chapter by analyzing data from a camera trapping data from a study of  
4053 wolverines *Gulo gulo* (Magoun et al., 2011; Royle et al., 2011b). The study took  
4054 place in SE Alaska (Fig. 5.4) where 37 cameras were operational for variable periods  
4055 of time (min = 5 days, max = 108 days, median = 45 days). A consequence of this  
4056 is that the number of sampling occasions,  $K$ , is variable for each camera. Thus,  
4057 we must provide a vector of sample sizes as data to **BUGS** and modify the model  
4058 specification in Sec. 5.7 accordingly.

### 4059 4.9.1 Practical data organization

4060 To carry out an analysis of these data, we require the matrix of trap coordinates  
4061 and the encounter history data. We usually store data in 2 distinct data files which  
4062 contain all the information needed for an analysis. These files are

- 4063 • The encounter data file (EDF) containing a record of which traps and when each  
4064 individual encounter occurred.
- 4065 • The trap deployment file (TDF) which contains the coordinates of each trap,  
4066 along with information indicating which sample occasions each trap was operat-  
4067 ing.

4068 **Encounter Data File (EDF)** – We store the encounter data in the an efficient  
4069 file format which is easily manipulated in **R** and easy to create in Excel and other  
4070 spreadsheets which are widely used for data management. The file structure is a  
4071 simple matrix with 4 columns, those being: (1) **session ID**: the trap *session* which  
4072 usually corresponds to a year or a primary period in the context of a Robust Design  
4073 situation, but it could also correspond to a distinct spatial unit (see Sec. 6.5.4 and  
4074 Chapt. 14). For a single-year study (as considered here) this should be an integer  
4075 that is the same for all records; (2) **individual ID**: the individual identity, being an  
4076 integer from 1 to  $n$  (repeated for multiple captures of the same individual) indicating  
4077 which individual the record (row) of the matrix belongs to; (3) **occasion ID**: The  
4078 integer sample occasion which generated the record, and (4) **trap ID**: the trap  
4079 identity, an integer from 1 to  $J$ , the number of traps. The structure of the EDF  
4080 is the same as used in the **secr** package (Efford, 2011) and similar to that used  
4081 in the **SPACECAP** (Gopalaswamy et al., 2012a), and **SCRbayes** (Russell et al.,



**Figure 4.4.** Wolverine camera trap locations (black dots) from a study that took place in SE Alaska. See Magoun et al. (2011) for details.

4082 2012) packages, both of which have a 3-column format (`trapID`, `indID`, `sampID`).  
 4083 We note that the naming of the columns is irrelevant as far as anything we do in  
 4084 this book, although `secr` and other software may have requirements on variable  
 4085 naming.

4086 To illustrate this format, the wolverine data are available in the package `scrbook`  
 4087 by typing:

4088 > `data(wolverine)`

4089 which contains a list having elements `wcaps` (the EDF) and `wtraps` (the TDF). We  
 4090 see that `wcaps` has 115 rows, each representing a unique encounter event including  
 4091 the trap identity, the individual identity and the sample occasion index (`sample`).  
 4092 The first 5 rows of `wcaps` are:

```

4093 > wolverine$wcaps[1:5,]
4094   year individual day trap
4095 [1,]    1          2 127  1
4096 [2,]    1          2 128  1
4097 [3,]    1          2 129  1
4098 [4,]    1         18 130  1
4099 [5,]    1          3 106  2

```

4100     The 1st column here, labeled `year`, is an integer indicating the year or session  
4101     of the encounter. All these data come from a single year (2008) and so `year` is set  
4102     to 1. Variable `individual` is an integer identity of each individual captured, `day`  
4103     is the sample occasion of capture (in this case, the sample occasions correspond  
4104     to days), and `trap` is the integer trap identity. The variable `trapid` will have to  
4105     correspond to the row of a matrix containing the trap coordinates - in this case the  
4106     TDF file `wtraps` which we describe further below.

4107     Note that the information provided in this encounter data file `wcaps` does not  
4108     represent a completely informative summary of the data. For example, if no indi-  
4109     viduals were captured in a certain trap or during a certain period, then this compact  
4110     data format will have no record. Thus we will need to know  $J$ , the number of traps,  
4111     and  $K$ , the number of sample occasions when reformatting this SCR data format  
4112     into a 2-d encounter frequency matrix or 3-d array. In addition, the encounter data  
4113     file does not provide information about which periods each trap was operated. This  
4114     additional information is also necessary as the trap-specific sample sizes must be  
4115     passed to **BUGS** as data. We provide this information along with trap coordinates,  
4116     in the “trap deployment file” (TDF) which is described below.

4117     For our purposes, we need to convert the `wcaps` file into the  $n \times J$  array of  
4118     binomial encounter frequencies, although more general models might require an  
4119     encounter-history formulation of the model which requires a full 3-d array. To  
4120     obtain our encounter frequency matrix, we do this the hard way by first converting  
4121     the encounter data file into a 3-d array and then summarize to trap totals. We have  
4122     a handy function `SCR23darray` which takes the compact encounter data file, and  
4123     converts it to a 3-d array, and then we use the **R** function `apply` to summarize over  
4124     the sample occasion dimension (by convention here, this is the 2nd dimension). To  
4125     apply this to the wolverine data in order to compute the 3-d array we do this:

```

4126 > y3d <- SCR23darray(wolverine$wcaps,wolverine$wtraps)
4127 > y <- apply(y3d,c(1,2),sum)

```

4128     See the help file for more information on `SCR23darray`. The 3-d array is necessary to  
4129     fit certain types of models (e.g., behavioral response) and this is why we sometimes  
4130     will require this maximally informative 3-d data format but, here, we analyze the  
4131     summarized data.

4132     **Trap Deployment File (TDF)** – The other important information needed to  
4133     fit SCR models is the “trap deployment file” (TDF) which provides additional  
4134     information not contained in the encounter data file. The traps file has  $K + 3$

4135 columns. The first column is assumed to be a trap identifier, columns 2 and 3  
 4136 are the easting and northing coordinates (assumed to be in a Euclidean coordinate  
 4137 system), and columns 4 to  $K + 3$  are binary indicators of whether each trap was  
 4138 operational during each sample occasion. The first 10 rows (out of 37) and 10  
 4139 columns (out of 167) of the trap deployment file for the wolverine data are shown  
 4140 as follows:

```
4141 > wolverine$wtraps[1:10,1:10]
4142
4143   Easting Northing 1 2 3 4 5 6 7 8
4144 1 632538 6316012 0 0 0 0 0 0 0 0
4145 2 634822 6316568 1 1 1 1 1 1 1 1
4146 3 638455 6309781 0 0 0 0 0 0 0 0
4147 4 634649 6320016 0 0 0 0 0 0 0 0
4148 5 637738 6313994 0 0 0 0 0 0 0 0
4149 6 625278 6318386 0 0 0 0 0 0 0 0
4150 7 631690 6325157 0 0 0 0 0 0 0 0
4151 8 632631 6316609 0 0 0 0 0 0 0 0
4152 9 631374 6331273 0 0 0 0 0 0 0 0
4153 10 634068 6328575 0 0 0 0 0 0 0 0
```

4154 This tells us that trap 2 was operated during occasions (days) 1-7 but the  
 4155 other traps were not operational during those periods. It is extremely important  
 4156 to recognize that each trap was operated for a variable period of time and thus  
 4157 the binomial “sample size” is different for each, and this needs to be accounted for  
 4158 in the **BUGS** model specification. To compute the vector of sample sizes  $K$ , and  
 4159 extract the trap locations, we do this:

```
4160 > traps <- wolverine$wtraps
4161 > traplocs <- traps[,1:2]
4162 > K <- apply(traps[,3:ncol(traps)],1,sum)
```

4163 This results in a matrix `traplocs` which contains the coordinates of each trap and  
 4164 a vector  $K$  containing the number of days that each trap was operational. We now  
 4165 have all the information required to fit a basic SCR model in **BUGS**.

4166 Summarizing the data for the wolverine study, we see that 21 unique individuals  
 4167 were captured a total of 115 times. Most individuals were captured 1-6 times, with  
 4168 4, 1, 4, 3, 1, and 2 individuals captured 1-6 times, respectively. In addition, 1  
 4169 individual was captured each 8 and 14 times and 2 individuals each were captured  
 4170 10 and 13 times. The number of unique traps that captured a particular individual  
 4171 ranged from 1-6, with 5, 10, 3, 1, 1, and 1 individual captured in each of 1 to  
 4172 6 different traps, respectively, for a total of 50 unique wolverine-trap encounters.  
 4173 These numbers might be hard to get your mind around whereas some tabular  
 4174 summary is often more convenient. For that it seems natural to tabulate individuals  
 4175 by trap and total encounter frequencies. The spatial information in SCR data is  
 4176 based on multi-trap captures, and so, it is informative to understand how many

4177 unique traps each individual is captured in, and the total number of encounters.  
4178 For the wolverine data, we reproduce Table 1 from Royle et al. (2011b) as Table  
4179 5.3.

**Table 4.3.** 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	0	1	0

#### 4180 4.9.2 Fitting the model in WinBUGS

4181 Here we fit the simplest SCR model with the Gaussian encounter probability model,  
4182 although we revisit these data and fit additional models in later chapters. Model  
4183 SCR0 is summarized by the following 4 elements:

- 4184 (1)  $y_{ij} | \mathbf{s}_i \sim \text{Binomial}(K, z_i p_{ij})$
- 4185 (2)  $p_{ij} = p_0 \exp(-\alpha_1 ||\mathbf{x}_j - \mathbf{s}_i||^2)$
- 4186 (3)  $\mathbf{s}_i \sim \text{Uniform}(\mathcal{S})$
- 4187 (4)  $z_i \sim \text{Bernoulli}(\psi)$

4188 We assume customary flat priors on the structural (hyper-) parameters of the model,  
4189  $\alpha_0 = \text{logit}(p_0)$ ,  $\alpha_1$  and  $\psi$ .

4190 It remains to define the state-space  $\mathcal{S}$ . For this, we nested the trap array  
4191 (Fig. 5.4) in a rectangular state-space extending 20 km beyond the traps in each  
4192 cardinal direction. We scaled the coordinate system so that a unit distance was  
4193 equal to 10 km, producing a rectangular state-space of dimension  $9.88 \times 10.5$  units  
4194 ( $\text{area} = 10374 \text{ km}^2$ ) within which the trap array was nested. As a general rule, we  
4195 recommend scaling the state-space so that it is defined near the origin  $(x, y) = (0, 0)$ .  
4196 While the scaling of the coordinate system is theoretically irrelevant, a poorly  
4197 scaled coordinate system can produce Markov chains that mix poorly. The buffer  
4198 of the state space should be large enough so that individuals beyond the state-  
4199 space boundary are not likely to be encountered (Sec. 5.3.1). To evaluate this, we  
4200 fit models for various choices of a rectangular state-space based on buffers from 1.0  
4201 to 5.0 units (10 km to 50 km). In the R package `scrbook` we provide a function  
4202 `wolvSCR0` which will fit model SCR0. For example, to fit the model in **WinBUGS**  
4203 using data augmentation with  $M = 300$  potential individuals, using 3 Markov

**Table 4.4.** Posterior summaries of SCR model parameters for the wolverine camera trapping data from SE Alaska, using state-space buffers from 10 up to 50 km. Each analysis was based on 3 chains, 12000 iterations, 2000 burn-in, for a total of 30000 posterior samples.

Buffer	$\sigma$		N			D			
	Mean	SD	n.eff	Mean	SD	n.eff	Mean	SD	n.eff
10	0.65	0.06	1800	39.63	6.70	7100	5.97	1.00	7100
15	0.64	0.06	510	48.77	9.19	3300	5.78	1.09	3300
20	0.64	0.06	1200	59.84	11.89	20000	5.77	1.15	20000
25	0.64	0.05	3600	72.40	14.72	2700	5.79	1.18	2700
30	0.63	0.05	5600	86.42	17.98	3900	5.82	1.21	3900
35	0.63	0.05	4500	101.79	21.54	30000	5.85	1.24	30000
40	0.64	0.05	410	118.05	26.17	410	5.87	1.30	450
45	0.64	0.05	10000	134.43	28.68	3300	5.83	1.24	3300
50	0.63	0.05	4700	151.61	31.65	3400	5.79	1.21	3400

4204 chains each of 12000 total iterations, discarding the first 2000 as burn-in, we execute  
 4205 the following **R** commands:

```
4206 > library(scrbook)
4207 > data(wolverine)
4208 > traps <- wolverine$wtraps
4209 > y3d <- SCR23darray(wolverine$wcaps,wolverine$wtraps)
4210 > wolv <- wolvSCR0(y3d,traps,nb=2000,ni=12000,buffer=1,M=300)
```

4211 The argument **buffer** determines the buffer size of the state-space in the scaled  
 4212 units (i.e., 10 km). Note that this analysis takes between 1-2 hours on many  
 4213 machines (in 2013) so we recommend testing it with lower values of  $M$  and fewer  
 4214 iterations. The posterior summaries are shown in Table 5.9.2.

#### 4215 4.9.3 Summary of the wolverine analysis

4216 We see that the estimated density is roughly consistent as we increase the state-  
 4217 space buffer from 15 to 55 km. We do note that the data augmentation parameter  
 4218  $\psi$  (and, correspondingly,  $N$ ) increase with the size of the state space in accordance  
 4219 with the deterministic relationship  $N = D * A$ . However, density is more or less  
 4220 constant as we increase the size of the state-space beyond a certain point. For the  
 4221 10 km state-space buffer, we see a slight effect on the posterior distribution of  $D$   
 4222 because the state-space is not sufficiently large. The full results from the analysis  
 4223 based on 20 km state-space buffer are given in Table 5.5.

4224 Our point estimate of wolverine density from this study, using the posterior  
 4225 mean from the state-space based on the 20 km buffer, is approximately 5.77 indi-  
 4226 viduals/1000 km<sup>2</sup> with a 95% posterior interval [3.86, 8.29]. Density is estimated

**Table 4.5.** 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

imprecisely which might not be surprising given the low sample size ( $n = 21$  individuals!). This seems to be a basic feature of carnivore studies although it should not (in our view) preclude the study of their populations by capture-recapture nor attempts to estimate density or vital rates.

It is worth thinking about this model, and these estimates, computed under a rectangular state space roughly centered over the trapping array (Fig. 5.4). Does it make sense to define the state-space to include, for example, ocean? What are the possible consequences of this? What can we do about it? There's no reason at all that the state space has to be a regular polygon – we defined it as such here strictly for convenience and for ease of implementation in **WinBUGS** where it enables us to specify the prior for the activity centers as uniform priors for each coordinate. While it would be possible to define a more realistic state-space using some general polygon GIS coverage, it might take some effort to implement that in the **BUGS** language but it is not difficult to devise custom MCMC algorithms to do that (see Chapt. 17). Alternatively, we recommend using a discrete representation of the state-space – i.e., approximate  $\mathcal{S}$  by a grid of  $G$  points. We discuss this in Sec. 5.10.

#### 4.9.4 Wolverine space usage

The parameter  $\alpha_1$  is related to the home range radius (Sec. 5.4). For the Gaussian model we interpret the scale parameter  $\sigma$ , related to  $\alpha_1$  by  $\alpha_1 = 1/(2\sigma^2)$ , as the radius of a bivariate normal model of space usage. In this case  $\sigma = 0.64$  standardized units (10 km), which corresponds to  $0.64 \times 10 = 6.4$  km. It can be argued then that 95% of space used by an individual is within  $6.4 \times \sqrt{5.99} = 15.66$  km of the home range center. The effective “home range area” is then the area of this circle, which is  $\pi \times 15.66^2 = 770.4$  km<sup>2</sup>. Using our handy function **hra** we do this:

```
hra(pGauss1,parms=c(-2,1/(2*.64*.64)),xlim=c(-1,7),ylim=c(-1,7))
[1] 7.731408
```

which is in units of  $100 \text{ km}^2$ , so 773.1. The difference in this case is due to numerical approximation of our all-purpose tool `hra`. This home range size is relatively huge for measured home ranges, which range between 100 and  $535 \text{ km}^2$  (Whitman et al., 1986).

Royle et al. (2011b) reported estimates for  $\sigma$  in the range  $6.3 - 9.8 \text{ km}$  depending on the model, which isn't too different than here<sup>1</sup>. However, these estimates are larger than the typical home range sizes suggested in the literature. One possible explanation is that if a wolverine is using traps as a way to get yummy chicken, so it's moving from trap to trap instead of adhering to "normal" space usage patterns, then the implied home range size might not be worth much biologically. Thus, interpretation of detection models in terms of home range area depends on some additional context or assumptions, such as that traps don't effect individual space usage patterns. As such, we caution against direct biological interpretations of home range area based on  $\sigma$ , although SCR models can be extended to handle more general, non-Euclidean, patterns of space usage. See Chaps. 12 and 13.

We can calibrate the desired size of the state-space by looking at the estimated home range radius of the species. We should target a buffer of width  $2$  to  $3 \times \sigma$  in order that the probability of encountering an individual is very close to 0 beyond the prescribed state-space. Essentially, by specifying a state-space, we're setting  $p = 0$  for individuals beyond the prescribed state-space. For the wolverine data, with  $\sigma$  in the range of 6-9 km, a state-space buffer of 20 km is sufficiently large.

## 4.10 USING A DISCRETE HABITAT MASK

The SCR model developed previously in this chapter assumes that individual activity centers are distributed uniformly over the prescribed state-space. Clearly this will not always be a reasonable assumption. In Chapt. 11, we develop models that allow explicitly for non-uniformity of the activity centers by modeling covariate effects on density. A simplistic method of affecting the distribution of activity centers, which we address here, is to modify the shape and organization of the state-space explicitly. For example, we might be able to classify the state-space into distinct blocks of habitat and non-habitat. In that case we can remove the non-habitat from the state-space and assume uniformity of the activity centers over the remaining portions judged to be suitable habitat. There are several ways to approach this: We can use a grid of points to represent the state-space, i.e., by the set of coordinates  $\mathbf{s}_1, \dots, \mathbf{s}_G$ , and assign equal probabilities to each possible value. Alternatively, we can retain the continuous formulation of the state-space but attempt to describe constraints analytically, or we can use polygon clipping methods to enforce

<sup>1</sup> Royle et al. (2011b) expressed the model as  $\text{cloglog}(p_{ij}) = \alpha_0 - (1/\sigma^2) * d_{ij}^2$ , but the estimates of  $\sigma$  reported in their Table 2 are actually based on the model according to  $\text{cloglog}(p_{ij}) = \alpha_0 - \frac{1}{2\sigma^2} * d_{ij}^2$ , and so the estimates of  $\sigma$  they report in units of km are consistent to what we report here except based on the complementary log-log (Gaussian hazard) model, instead of the Gaussian encounter probability model.

4290 constraints on the state-space in the MCMC analysis. We focus here on the formulation  
 4291 of the basic SCR model in terms of a discrete state-space but in Chapt. 17 we  
 4292 demonstrate the latter approach based on using polygon operations to define an irregular  
 4293 state-space. Use of a discrete state-space can be computationally expensive  
 4294 in **WinBUGS**. That said, it isn't too difficult to perform the MCMC calculations  
 4295 in **R** (discussed in Chapt. 17). The **R** package **SPACECAP** (Gopalaswamy et al.,  
 4296 2012a) arose from the **R** implementation of the SCR model in Royle et al. (2009a).

4297 While clipping out non-habitat seems like a good idea, we think investigators  
 4298 should go about this very cautiously. We might prefer to do it when non-habitat  
 4299 represents a clear-cut restriction on the state-space such as a reserve boundary or  
 4300 a lake, ocean or river. But, having the capability to do this also causes people  
 4301 to start defining "habitat" vs. "non-habitat" based on their understanding of the  
 4302 system whereas it can't be known whether the animal being studied has the same  
 4303 understanding. Moreover, differentiating the landscape by habitat or habitat quality  
 4304 must affect the geometry and morphology of home ranges (see Chapt. 13) much  
 4305 more so than the plausible locations of activity centers. That is, a home range  
 4306 centroid could, in actual fact, occur in a shopping mall parking lot if there is pretty  
 4307 good habitat around the shopping mall, so there is probably no sense preclude it  
 4308 as the location for an activity center. It would generally be better to include some  
 4309 definition of habitat quality in the model for the detection probability (Royle et al.,  
 4310 2012a) which we address in Chaps. 12 and 13.

#### 4311 4.10.1 Evaluation of coarseness of habitat mask

4312 The coarseness of the state-space should not really have much of an effect on es-  
 4313 timates if the grain is sufficiently fine relative to typical animal home range sizes.  
 4314 Why is this? We have two analogies that can help us understand. First is the  
 4315 relationship to model  $M_h$ . As noted in Sec. 5.3.2 above, we can think about SCR  
 4316 models as a type of finite mixture (Norris and Pollock, 1996; Pledger, 2004) where  
 4317 we are fortunate to be able to obtain direct information about which group indi-  
 4318 viduals belong to (group being location of activity center). In the standard finite  
 4319 mixture models we typically find that a small number of groups (e.g., 2 or 3 at  
 4320 the most) can explain high levels of heterogeneity and are adequate for most data  
 4321 sets of small to moderate sample sizes. We therefore expect a similar effect in SCR  
 4322 models when we discretize the state-space. We can also think about discretizing  
 4323 the state-space as being related to numerical integration where we find (see Chapt.  
 4324 6) that we don't need a very fine grid of support points to evaluate the integral to  
 4325 a reasonable level of accuracy. We demonstrate this here by reanalyzing simulated  
 4326 data using a state-space defined by a different number of support points. We pro-  
 4327 vide an **R** script called **SCR0bayesDss** in the **R** package **scrbook**. We note that for  
 4328 this comparison we generated the actual activity centers as a continuous random  
 4329 variable and thus the discrete state-space is, strictly speaking, an approximation  
 4330 to truth. That said, we regard all state-space specifications as approximations to

**Table 4.6.** 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

truth in the sense that they represent a component of the SCR model.

As with our **R** function **SCR0bayes**, the modification **SCR0bayesDss** will use either **WinBUGS** or **JAGS**. In addition, it requires a grid resolution argument (**ng**) which is the dimension of 1 side of a square state-space. To execute this function we do, for example:

```

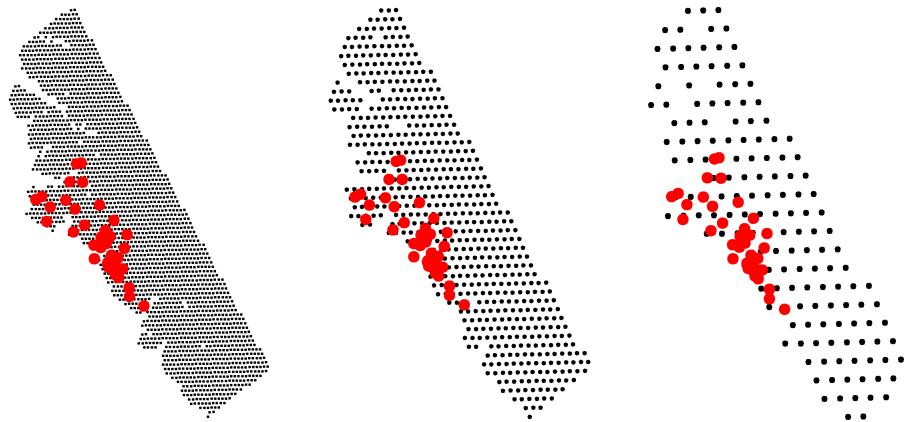
4336 > library(scrbook)
4337 > data <- simSCR0(discard0=TRUE,rnd=2013)      # generate data set
4338
4339 # run with JAGS
4340 > out1 <- SCR0bayesDss(data,ng=8,M=200,engine="jags",ni=2000,nb=1000)
4341
4342 # run with WinBUGS
4343 > out2 <- SCR0bayesDss(data,ng=8,M=200,engine="winbugs",ni=2000,nb=1000)

```

We fit this model to the same simulated data set for  $6 \times 6$ ,  $9 \times 9$ ,  $12 \times 12$ ,  $15 \times 15$  state-space grids. For **WinBUGS**, we used 3 chains of 5000 total length with 1000 burn-in, which yields 12000 total posterior samples. Summary results are shown in Table 5.6. The results are broadly consistent except for the  $6 \times 6$  case. We see that the run time increases with the size of the state-space grid (not unexpected), such that we imagine it would be impractical to run models with more than a few hundred state-space grid points. We found (not shown here) that the runtime of **JAGS** is much faster and, furthermore, relatively *constant* as we increase the grid size. We suspect that **WinBUGS** is evaluating the full-conditional for each activity center at all  $G$  possible values whereas it may be that **JAGS** is evaluating the full-conditional only at a subset of values or perhaps using previous calculations more effectively. While this might suggest that one should always use **JAGS** for this analysis, we found in our analysis of the wolverine (next section) that **JAGS** could be extremely sensitive to starting values, producing MCMC algorithms that often simply do not work for some problems, so be careful when using **JAGS**. To improve its performance, always start the latent activity centers at values near where individuals were captured. The performance of either should improve if we compute the full distance matrix outside of **BUGS** and pass it as data, although we haven't fully evaluated this approach.

**4.10.2 Analysis of the wolverine camera trapping data**

We reanalyzed the wolverine data using discrete state-space grids with points spaced by 2, 4 and 8 km (see Fig. 5.5). These were constructed from a 40 km buffered state-space, and deleting the points over water (see Royle et al., 2011b). Our interest in doing this was to evaluate the relative influence of grid resolution on estimated density because the coarser grids will be more efficient from a computational standpoint and so we would prefer to use them, but only if there is no strong influence on estimated density. The posterior summaries for the 3 habitat grids are given in Table 5.7. We see that the density estimates are quite a bit larger than obtained in our analysis (Table 5.9.2) based on a rectangular, continuous state-space. We also see that there are slight differences depending on the resolution of the state-space grid. Interestingly, the effectiveness of the MCMC algorithms, as measured by effective sample size (`n.eff`) is pretty remarkably different. Furthermore, the finest grid resolution (2 km spacing) took about 6 days to run and thus it would not be practical for large problems or with many models.



**Figure 4.5.** 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.

**4.11 SUMMARIZING DENSITY AND ACTIVITY CENTER LOCATIONS**

One of the most useful aspects of SCR models is that they are parameterized in terms of individual locations – i.e., *where* each individual lives – and, thus, we can

**Table 4.7.** Posterior summaries for the wolverine camera trapping data, using model SCR0, with a Gaussian hazard encounter probability model, and a discrete habitat mask of 3 different resolutions: 2, 4 and 8 km. Parameters are  $\lambda_0$  = baseline encounter rate,  $p_0 = 1 - \exp(-\lambda_0)$ ,  $\sigma$  is the scale parameter of the Gaussian kernel,  $\psi$  is the data augmentation parameter,  $N$  and  $D$  are population size and density, respectively. Models fitted using WinBUGS, 3 chains, each with 11000 iterations (first 1000 discarded) producing 30000 posterior samples.

2 km spacing										
	Mean	SD	2.5%	25%	50%	75%	97.5%	Rhat	n.eff	
$\sigma$	0.62	0.05	0.54	0.59	0.62	0.65	0.73	1.01	160	
$\lambda_0$	0.05	0.01	0.04	0.04	0.05	0.06	0.07	1.01	320	
$p_0$	0.05	0.01	0.03	0.04	0.05	0.05	0.06	1.01	320	
$\psi$	0.43	0.09	0.27	0.37	0.43	0.49	0.63	1.00	560	
$N$	86.56	16.94	57.00	75.00	85.00	97.00	124.00	1.00	510	
$D$	8.78	1.72	5.78	7.60	8.62	9.83	12.57	1.00	510	
4 km spacing										
	Mean	SD	2.5%	25%	50%	75%	97.5%	Rhat	n.eff	
$\sigma$	0.61	0.04	0.53	0.58	0.61	0.64	0.71	1	1600	
$\lambda_0$	0.05	0.01	0.04	0.05	0.05	0.06	0.07	1	2500	
$p_0$	0.05	0.01	0.03	0.04	0.05	0.05	0.07	1	2500	
$\psi$	0.45	0.09	0.28	0.38	0.44	0.50	0.64	1	1300	
$N$	89.25	17.44	59.00	77.00	88.00	100.00	127.00	1	1100	
$D$	9.01	1.76	5.96	7.77	8.88	10.10	12.82	1	1100	
8 km spacing										
	Mean	SD	2.5%	25%	50%	75%	97.5%	Rhat	n.eff	
$\sigma$	0.68	0.05	0.59	0.64	0.67	0.71	0.77	1.01	220	
$\lambda_0$	0.05	0.01	0.03	0.04	0.05	0.05	0.06	1.00	560	
$p_0$	0.05	0.01	0.03	0.04	0.04	0.05	0.06	1.00	560	
$\psi$	0.42	0.09	0.26	0.36	0.41	0.47	0.61	1.00	940	
$N$	83.18	16.14	56.00	72.00	82.00	93.00	119.00	1.00	700	
$D$	8.28	1.61	5.57	7.17	8.16	9.26	11.84	1.00	700	

4380 compute many useful and interesting summaries of the activity centers using output  
 4381 from an MCMC simulation, including maps of density (the number of activity  
 4382 centers per unit area), estimates of  $N$  for any well-defined polygon, or estimates of  
 4383 where the activity centers for specific individuals reside. In Bayesian analysis by  
 4384 MCMC, obtaining such summaries entails no added calculations, because we need  
 4385 only post-process the output for the individual activity centers to obtain the desired  
 4386 summaries. We demonstrate that in this section. Note that you have to be sure  
 4387 to retain the MCMC history for the  $s$  variables and also the data augmentation  
 4388 variables  $z$  in order to do the following analyses.

#### 4389 4.11.1 Constructing density maps

4390 Because SCR models are spatially-explicit, it is natural to want to summarize the  
 4391 results of fitting a model by producing a map of density. Using Bayesian analysis  
 4392 by MCMC, it is most easy to make a map of *realized* density. We can do this by

4393 tallying up the number of activity centers  $\mathbf{s}_i$  in pixels of arbitrary size and then  
 4394 producing a nice multi-color spatial plot of the result. Specifically, let  $B(\mathbf{x})$  indicate  
 4395 a pixel centered at  $\mathbf{x}$  then

$$N(\mathbf{x}) = \sum_{i=1}^M I(\mathbf{s}_i \in B(\mathbf{x}))$$

4396 (here,  $I(arg)$  is the indicator function which evaluates to 1 if  $arg$  is true, and 0  
 4397 otherwise) is the population size of pixel  $B(\mathbf{x})$ , and  $D(\mathbf{x}) = N(\mathbf{x})/\|B(\mathbf{x})\|$  is the  
 4398 local density. Note that these  $N(\mathbf{x})$  parameter are just “derived parameters” as we  
 4399 normally obtain from posterior output using the appropriate Monte Carlo average  
 4400 (see Chapt. 3).

4401 One thing to be careful about, in the context of models in which  $N$  is unknown,  
 4402 is that, for each MCMC iteration  $m$ , we only tabulate those activity centers which  
 4403 correspond to individuals in the sampled population, i.e., for which the data aug-  
 4404mentation variable  $z_i = 1$ . In this case, we take all of the output for MCMC  
 4405 iterations  $m = 1, 2, \dots, \text{niter}$  and compute this summary:

$$N(\mathbf{x}, m) = \sum_{i:z_{i,m}=1} I(\mathbf{s}_{i,m} \in B(\mathbf{x}))$$

4406 Thus,  $N(\mathbf{x}, 1), N(\mathbf{x}, 2), \dots$ , is the Markov chain for parameter  $N(\mathbf{x})$ . In what fol-  
 4407 lows we will provide a set of **R** commands for doing this calculation and making a  
 4408 basic image plot from the MCMC output.

4409 **Step 1:** Define the center points of each pixel  $B(\mathbf{x})$ , or point at which local density  
 4410 will be estimated:

```
4411 > xg <- seq(xlim[1], xlim[2], , 50)
4412 > yg <- seq(ylim[1], ylim[2], , 50)
```

4413 **Step 2:** Extract the MCMC histories for the activity centers and the data aug-  
 4414mentation variables. Note that these are each  $N \times \text{niter}$  matrices. Here we do this  
 4415 assuming that **WinBUGS** was run producing the **R** object named **out**:

```
4416 > Sxout <- out$sims.list$s[, , 1]
4417 > Syout <- out$sims.list$s[, , 2]
4418 > z <- out$sims.list$z
```

4419 **Step 3:** We associate each coordinate with the proper pixel using the **R** command  
 4420 **cut()**. Note that we keep only the activity centers for which  $z = 1$  (i.e., individuals  
 4421 that belong to the population of size  $N$ ):

```
4422 > Sxout <- cut(Sxout[z==1], breaks=xg, include.lowest=TRUE)
4423 > Syout <- cut(Syout[z==1], breaks=yg, include.lowest=TRUE)
```

4424 **Step 4:** Use the `table()` command to tally up how many activity centers are in  
 4425 each  $B(x)$ :

4426 > `Dn <- table(Sxout,Syout)`

4427 **Step 5:** Use the `image()` command to display the resulting matrix.

4428 > `image(xg, yg, Dn/nrow(z), col=terrain.colors(10))`

4429 It is worth emphasizing here that density maps will not usually appear uniform  
 4430 despite that we have assumed that activity centers are uniformly distributed. This is  
 4431 because the observed encounters of individuals provide direct information about the  
 4432 location of the  $i = 1, 2, \dots, n$  activity centers and thus their “estimated” locations  
 4433 will be affected by the observations. In a limiting sense, were we to sample space  
 4434 intensely enough, every individual would be captured a number of times and we  
 4435 would have considerable information about all  $N$  point locations. Consequently,  
 4436 the uniform prior would have almost no influence at all on the estimated density  
 4437 surface in this limiting situation. Thus, in practice, the influence of the uniformity  
 4438 assumption decreases as the fraction of the population encountered, and the total  
 4439 number of encounters per individual, increases.

4440 **On the non-intuitiveness of `image()`** – the R function `image()`, invoked for  
 4441 a matrix  $M$  by `image(M)`, might not be very intuitive to some – it plots  $M[1, 1]$  in  
 4442 the lower left corner. If you want  $M[]$  to be plotted “as you look at it” then  $M[1, 1]$   
 4443 should be in the upper left corner. We have a function `rot()` which does that. If  
 4444 you do `image(rot(M))` then it puts it on the monitor as if it was a map you were  
 4445 looking at. You can always specify the  $x$ - and  $y$ -labels explicitly as we did above.

4446 **Spatial dot plots** – A cruder version of the density map can be made using  
 4447 our “spatial dot map” function `spatial.plot` (in `scrbook`). This function requires,  
 4448 as input, point locations and the value to be displayed. A simplified version of this  
 4449 function is as follows:

```
4450 > spatial.plot <- function(x,y){
  4451   nc <- as.numeric(cut(y,20))
  4452   plot(x,pch=" ")
  4453   points(x,pch=20,col=topo.colors(20)[nc],cex=2)
  4454   image.scale(y,col=topo.colors(20))
  4455 }
  4456 #
  4457 # To execute the function do this:
  4458 #
  4459 > spatial.plot(cbind(xg,yg), Dn/nrow(z))
```

#### 4460 4.11.2 Example: Wolverine density map

4461 We return to the wolverine study which took place in 2008 in SE Alaska (Fig. 5.4)  
 4462 and we produce a density map of wolverines from that analysis. We include the

4463 function SCRdensity which requires a specific data structure as shown below. In  
 4464 particular, we have to package up the MCMC history for the activity centers and  
 4465 the data augmentation variables  $z$  into a list. This also requires that we add those  
 4466 variables to the parameters-to-be-monitored list when we pass things to **BUGS**.

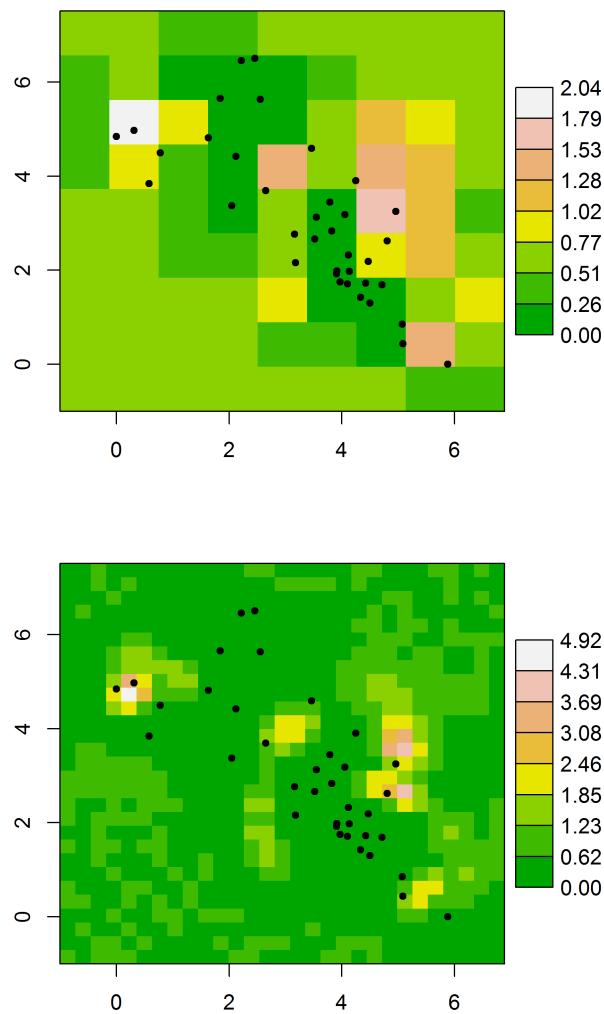
4467 We used the posterior output from the wolverine model fitted previously to  
 4468 compute a relatively coarse version of a density map, using 100 pixels in a  $10 \times 10$   
 4469 grid (Fig. 5.6 top panel) and using 900 pixels arranged in a  $30 \times 30$  grid (Fig. 5.6  
 4470 lower panel) for a fine-scale map. The **R** commands for producing such a plot (for  
 4471 a short MCMC run) are as follows:

```
4472 > library(scrbook)
4473 > data(wolverine)
4474 > traps <- wolverine$wtraps
4475 > y3d <- SCR23darray(wolverine$wcaps,wolverine$wtraps)
4476
4477 # this takes 341 seconds on a standard CPU circa 2011
4478 > out <- wolvSCR0(y3d,traps,nb=1000,ni=2000,buffer=1,M=100,keepz=TRUE)
4479
4480 > Sx <- out$sims.list$s[,,1]
4481 > Sy <- out$sims.list$s[,,2]
4482 > z <- out$sims.list$z
4483 > obj <- list(Sx=Sx,Sy=Sy,z=z)
4484 > tmp <- SCRdensity(obj,nx=10,ny=10,scalein=100,scaleout=100)
```

4485 In these figures density is expressed in units of individuals per  $100 \text{ km}^2$ , while the area of  
 4486 the pixels is about  $103.7 \text{ km}^2$  and  $11.5 \text{ km}^2$ , respectively. That calculation is based on:

```
4487 > total.area <- (ylim[2]-ylim[1])*(xlim[2]-xlim[1])*100
4488 > total.area/(10*10)
4489 [1] 103.7427
4490 > total.area/(30*30)
4491 [1] 11.52697
```

4492 A couple of things are worth noting: First is that as we move away from “where the  
 4493 data live” – away from the trap array – we see that the density approaches the mean  
 4494 density. This is a property of the estimator as long as the detection function decreases  
 4495 sufficiently rapidly as a function of distance. Relatedly, it is also a property of statistical  
 4496 smoothers such as splines, kernel smoothers, and regression smoothers – predictions tend  
 4497 toward the global mean as the influence of data diminishes. Another way to think of it is  
 4498 that it is a consequence of the prior, which imposes uniformity, and as you get far away  
 4499 from the data, the predictions tend to the expected constant density under the prior.  
 4500 Another thing to note about this map is that density is not 0 over water (although the  
 4501 coastline is not shown). This might be perplexing to some who are fairly certain that  
 4502 wolverines do not like water. However, there is nothing about the model that recognizes  
 4503 water from non-water and so the model predicts over water *as if* it were habitat similar to  
 4504 that within which the array is nested. But, all of this is OK as far as estimating density  
 4505 goes and, furthermore, we can compute valid estimates of  $N$  over any well-defined region  
 4506 which presumably wouldn’t include water if we so wished. Alternatively, areas covered by  
 4507 water could be masked out, which we discuss in the next section.



**Figure 4.6.** Density of wolverines (individuals per  $100 \text{ km}^2$ ) in SE Alaska in 2007 based on model SCR0. Map grid cells are about  $103.7 \text{ km}^2$  (top panel) and  $11.5 \text{ km}^2$  (bottom panel) in area. Dots are the trap locations.

**4.11.3 Predicting where an individual lives**

4509 The density maps in the previous section show the expected number of individuals per  
 4510 unit area. A closely related problem is that of producing a map of the probable location  
 4511 of a specific individual's activity center. For any observed encounter history, we can easily  
 4512 generate a posterior distribution of  $s_i$  for individual  $i$ . In addition, for an individual that  
 4513 is *not* captured, we can use the MCMC output to produce a corresponding plot of where  
 4514 such an individual might live, say  $s_{n+1}$ . Obviously, all such uncaptured individuals (for  
 4515  $i = n + 1, \dots, N$ ) should have the same posterior distribution. To illustrate, we show the  
 4516 posterior distribution of  $s_1$ , the activity center for the individual labeled 1 in the data  
 4517 set, in Fig. 5.7. This individual was captured a single time at trap 30 which is circled  
 4518 in Fig. 5.7. We see that the posterior distribution is affected by traps of capture *and*  
 4519 traps of non-capture in fairly intuitive ways. In particular, because there are other traps  
 4520 in close proximity to trap 30, in which individual 1 was *not* captured, the model pushes  
 4521 its activity center away from the trap array. The help file for **SCRdensity** shows how to  
 4522 calculate Fig. 5.7.

**4.12 EFFECTIVE SAMPLE AREA**

4523 One of the key issues in using ordinary capture recapture models which we've brought up  
 4524 over and over again is this issue that the area which is sampled by a trapping array is  
 4525 unknown – in other words, the  $N$  that is estimated by capture-recapture models does not  
 4526 have an explicit region of space associated with it. Classically this has been addressed in  
 4527 the ad hoc way of prescribing an area that contains the trap array, usually by adding a  
 4528 buffer of some width, which is not estimated as part of the capture-recapture model. In  
 4529 SCR models we avoid the problem of not having an explicit linkage between  $N$  and “area”,  
 4530 by prescribing explicitly the area within which the underlying point process is defined – the  
 4531 state-space of the point process. This state-space is *not* the effective sample (or sampled)  
 4532 area (ESA) – it is desirable that it be somewhat larger than the ESA, whatever that may  
 4533 be, in the sense that individuals at the edge of the state-space have no probability of being  
 4534 captured, but as part of the SCR model we don't need to try to estimate or otherwise  
 4535 characterize the ESA explicitly.

4536 However, it is possible to provide a characterization of effective sampled area under  
 4537 any SCR model. This is directly analogous to the calculation of “effective strip width” in  
 4538 distance sampling (Buckland et al., 2001; Borchers et al., 2002). The conceptual definition  
 4539 of ESA follows from equating density to “apparent density” – ESA is the magic number  
 4540 that satisfies that equivalence:

$$D = N/A = n/\text{ESA}$$

4541 In other words, the ratio of  $N$  to the area of the state-space should be equal to the ratio  
 4542 of the observed sample size  $n$  to this number ESA. Both of these should equal density.  
 4543 So, to compute ESA for a model, we substitute  $\mathbb{E}(n)$  for  $n$  into the above equation, and  
 4544 solve for  $\text{ESA}$ , to get:

$$\text{ESA} = \mathbb{E}(n)/D.$$

4545 Our following development assumes that  $D$  is constant, but these calculations can be  
 4546 generalized to allow for  $D$  to vary spatially. Imagine our habitat mask for the wolverine

4547 data, or the bins we just used to produce a density map, then we can write  $\mathbb{E}(n)$  according  
 4548 to

$$\mathbb{E}(n) = \sum_s \Pr(\text{encounter}|\mathbf{s}) \mathbb{E}(N(\mathbf{s}))$$

4549 where if we prefer to think of this more conceptually we could replace the summation with  
 4550 an integration (which, in practice, we would just replace with a summation, and so we  
 4551 just begin there). In this expression note that  $\mathbb{E}(N(\mathbf{s}))$  is the expected population size at  
 4552 pixel  $\mathbf{s}$  which is the density times the area of the pixel, i.e.,  $\mathbb{E}(N(\mathbf{s})) = D \times a$ . Therefore

$$\mathbb{E}(n) = D \times a \times \sum_s \Pr(\text{encounter}|\mathbf{s})$$

4553 and (plugging this into the expression above for ESA)

$$ESA = \frac{D \times a \times \sum_s \Pr(\text{encounter}|\mathbf{s})}{D}$$

4554 We see that  $D$  cancels and we have  $ESA = a \times \sum_s \Pr(\text{encounter}|\mathbf{s})$  So what you have to  
 4555 do here is substitute in  $\Pr(\text{encounter}|\mathbf{s})$  and just sum them up over all pixels. For the  
 4556 Bernoulli model of model SCR0

$$\Pr(\text{encounter}|\mathbf{s}) = 1 - (1 - p(\mathbf{s}))^K$$

4557 with slight modifications when encounter probability depends on covariates. Thus,

$$ESA = a \sum_s 1 - (1 - p(\mathbf{s}))^K \tag{4.12.1}$$

4558 Clearly the calculation of ESA is affected by the use of a habitat mask, because the  
 4559 summation in Eq. 5.12.1 only occurs over pixels that define the state-space.

4560 For the wolverine camera trapping data, we used the  $2 \times 2$  km habitat mask and the  
 4561 posterior means of  $p_0$  and  $\sigma$  (see Sec. 5.10.2) to compute the probability of encounter for  
 4562 each  $\mathbf{s}$  of the mask points. The result is shown graphically in Fig. 5.8. The ESA is the  
 4563 sum of the values plotted in that figure multiplied by 4, the area of each pixel. For the  
 4564 wolverine study, the result is 2507.152 km<sup>2</sup>. We note that the probability of encounter  
 4565 declines rapidly to 0 as we move away from the periphery of the camera traps, indicating  
 4566 the state-space constructed from a 40 km buffered trap array was indeed sufficient for the  
 4567 analysis of these data. An R script for producing this figure is in the `wolvESA` function of  
 4568 the `scrbook` package.

## 4.13 SUMMARY AND OUTLOOK

4569 In this chapter, we introduced the simplest SCR model – “model SCR0” – which is an ordinary  
 4570 capture-recapture model like model  $M_0$ , but augmented with a set of latent individual  
 4571 effects,  $\mathbf{s}_i$ , which relate encounter probability to some sense of individual location using a  
 4572 covariate, “distance”, from  $\mathbf{s}_i$  to each trap location. Thus, individuals in close proximity  
 4573 to a trap will have a higher probability of encounter, and *vice versa*. The explicit modeling  
 4574 of individual locations and distance in this fashion resolves classical problems related to

4575 estimating density: unknown sample area, and heterogeneous encounter probability due  
4576 to variable exposure to traps.

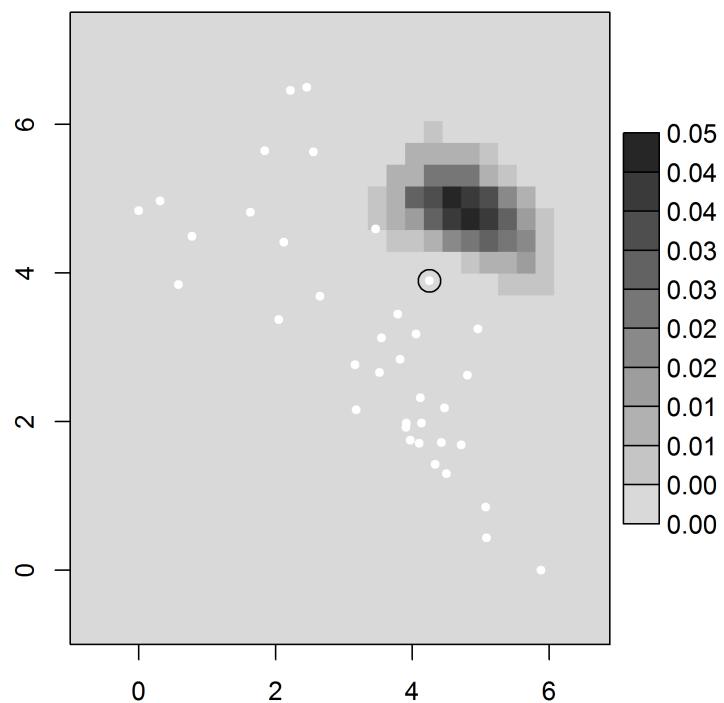
4577 SCR models are closely related to classical individual covariate models (“model  $M_x$ ”,  
4578 as introduced in Chapt. 4), but with imperfect information about the individual covari-  
4579 ate. Therefore, they are also not too dissimilar from standard GLMMs used throughout  
4580 statistics and, as a result, we find that they are easy to analyze using standard MCMC  
4581 methods encased in black boxes such as **WinBUGS** or **JAGS**. We will also see that they  
4582 are easy to analyze using likelihood methods, which we address in Chapt. 6.

4583 Formal consideration of the collection of individual locations ( $s_1, \dots, s_N$ ) is funda-  
4584 mental to all models considered in this book. In statistical terminology, we think of the  
4585 collection of points  $\{s_i\}$  as a realization of a point process. Because SCR models formally  
4586 link individual encounter history data to an underlying point process, we can obtain for-  
4587 mal inferences about the point process. For example, we showed how to produce a density  
4588 map (Fig. 5.6), or even a probability map for an individual’s home range center (Fig.  
4589 5.7). We can also use SCR models as the basis for doing more traditional point process  
4590 analyses, such as testing for “complete spatial randomness” (CSR) (see Chapt. 8), and  
4591 computing other point process summaries (Illian et al., 2008).

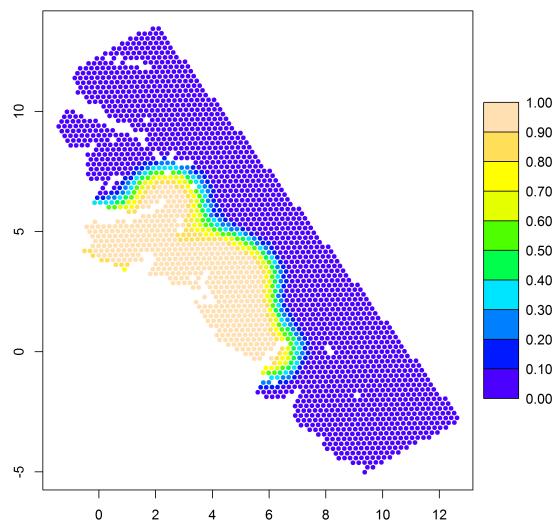
4592 Part of the promise, and ongoing challenge, of SCR models is to develop models that  
4593 reflect interesting biological processes, for example interactions among points or temporal  
4594 dynamics in point locations. In this chapter we considered the simplest possible point  
4595 process model in which points are independent and uniformly (“randomly”) distributed  
4596 over space. Despite the simplicity of this model, it should suffice in many applications of  
4597 SCR models, although we do address generalizations in later chapters. Moreover, even  
4598 though the *prior* distribution on the point locations is uniform, the realized pattern may  
4599 deviate markedly from uniformity as the observed encounter data provide information to  
4600 impart deviations from uniformity. Thus, estimated density maps will typically appear  
4601 distinctly non-uniform (as we saw in the wolverine example). In applications of the basic  
4602 SCR model, we find that this simple *a priori* model can effectively reflect or adapt to  
4603 complex realizations of the underlying point process. For example, if individuals are  
4604 highly territorial then the data should indicate this in the form of individuals not being  
4605 encountered in the same trap – the resulting posterior distribution of point locations should  
4606 therefore reflect non-independence. Obviously the complexity of posterior estimates of the  
4607 point pattern will depend on the quantity of data, both number of individuals and captures  
4608 per individual. Because the point process is such an integral component of SCR models,  
4609 the state-space of the point process plays an important role in developing SCR models.  
4610 As we emphasized in this chapter, the state-space is part of the model. It can have an  
4611 influence on parameter estimates and other inferences, such as model selection (see chapter  
4612 8).

4613 One concept we introduced in this chapter, which has not been discussed much in  
4614 the literature on SCR models, is the manner in which the encounter probability model  
4615 relates to a model of space usage by individuals. The standard SCR models of encounter  
4616 probability can all be motivated as simplistic models of space usage and movement, in  
4617 which individuals make random use decisions from a probability distribution proportional  
4618 to the encounter probability model. This both clarifies the simplicity of the underlying  
4619 model of space usage and also suggests a direct extension to produce more realistic models,  
4620 which we discuss in Chapt. 13. We consider some other important extensions of the basic

<sup>4621</sup> SCR model in later chapters. For example, we consider models that include covariates that  
<sup>4622</sup> vary by individual, trap, or over time (Chapt. 7), spatial covariates on density (Chapt.  
<sup>4623</sup> 11), open populations (Chapt. 16), and methods for model assessment and selection  
<sup>4624</sup> (Chapt. 8) among other topics. We also consider technical details of maximum likelihood  
<sup>4625</sup> (Chapt. 6) and Bayesian (Chapt. 17) estimation, so that the interested reader can develop  
<sup>4626</sup> or extend methods to suit their own needs.



**Figure 4.7.** Posterior probability distribution of  $s_1$ , the activity center for individual 1 in the wolverine data set. This individual was captured a single time in one trap (trap 30) which is circled. White dots are trap locations.



**Figure 4.8.** 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.



4627

4628

4629

4630

# 5

---

## LIKELIHOOD ANALYSIS OF SPATIAL CAPTURE-RECAPTURE MODELS

4631 We have so far mainly focused on Bayesian analysis of spatial capture-recapture models.  
4632 And, in the previous chapters we learned how to fit some basic spatial capture-recapture  
4633 models using a Bayesian formulation of the models analyzed in **BUGS** engines including  
4634 **WinBUGS** and **JAGS**. Despite our focus on Bayesian analysis, it is instructive to de-  
4635 velop the basic concepts and ideas behind classical analysis based on likelihood methods  
4636 and frequentist inference for SCR models. We recognized earlier (Chapt. 5) that SCR  
4637 models are versions of binomial (or other) GLMs, but with random effects (i.e., GLMMs).  
4638 Throughout statistics, such models are routinely analyzed by likelihood methods. In par-  
4639 ticular, likelihood analysis is based on the integrated or marginal likelihood in which the  
4640 random effects are removed, by integration, from the conditional-on-s likelihood ( $s$  being  
4641 the individual activity center). This has been the approach taken by Borchers and Ef-  
4642 ford (2008); Dawson and Efford (2009) and related papers. Therefore, in this chapter, we  
4643 provide some conceptual and technical foundation for likelihood-based analysis of spatial  
4644 capture-recapture models.

4645 We will show here that it is straightforward to compute the maximum likelihood esti-  
4646 mates (MLE) for SCR models by integrated likelihood. We develop the MLE framework  
4647 using **R**, and we also provide a basic introduction to the **R** package **secr** (Efford, 2011)  
4648 which does likelihood analysis of SCR models (see also the stand-alone program **DEN-**  
4649 **SITY** (Efford et al., 2004)). To set the context for likelihood analysis of SCR models,  
4650 we first analyze the SCR model when  $N$  is known because, in that case, analysis is no  
4651 different at all than a standard GLMM. We generalize the model to allow for unknown  $N$   
4652 using both conventional ideas based on the “full likelihood” (e.g., Borchers et al., 2002)  
4653 and also using a formulation based on data augmentation. We obtain the MLEs for the  
4654 SCR model from the wolverine camera trapping study (Magoun et al., 2011) analyzed in  
4655 previous chapters to compare/contrast the results.

## 5.1 MLE WITH KNOWN N

We noted in Chapt. 5 that, with  $N$  known, the basic SCR model is a type of binomial model with a random effect. For such models we can obtain maximum likelihood estimators of model parameters based on integrated likelihood. The integrated likelihood is based on the marginal distribution of the data  $y$  in which the random effects are removed by integration from the conditional-on-s distribution of the observations. See Chapt. 2 for a review of marginal, conditional and joint distributions. Conceptually, any SCR model begins with a specification of the conditional-on-s model  $[y|\mathbf{s}, \boldsymbol{\alpha}]$  and we have a “prior distribution” for  $\mathbf{s}$ , say  $[\mathbf{s}]$ . Then, the marginal distribution of the data  $y$  is

$$[y|\boldsymbol{\alpha}] = \int_{\mathcal{S}} [y|\mathbf{s}, \boldsymbol{\alpha}][\mathbf{s}]d\mathbf{s}.$$

When viewed as a function of  $\boldsymbol{\alpha}$  for purposes of estimation, the marginal distribution  $[y|\boldsymbol{\alpha}]$  is often referred to as the *integrated likelihood*.

It is worth analyzing the simplest SCR model with known- $N$  in order to understand the underlying mechanics and basic concepts. These are directly relevant to the manner in which many capture-recapture models are classically analyzed, such as model  $M_h$ , and individual covariate models (see Chapt. 4).

To develop the integrated likelihood for SCR models, we first identify the conditional-on-s likelihood. The observation model for each encounter observation  $y_{ij}$ , for individual  $i$  and trap  $j$ , specified conditional on  $\mathbf{s}_i$ , is

$$y_{ij}|\mathbf{s}_i \sim \text{Binomial}(K, p_{\boldsymbol{\alpha}}(\mathbf{x}_j, \mathbf{s}_i)) \quad (5.1.1)$$

where we have indicated the dependence of encounter probability,  $p_{ij}$ , on  $\mathbf{s}$  and parameters  $\boldsymbol{\alpha}$  explicitly. For example,  $p_{ij}$  might be the Gaussian model given by

$$p_{ij} = \text{logit}^{-1}(\alpha_0) \exp(-\alpha_1 \|\mathbf{x}_j - \mathbf{s}_i\|^2)$$

where  $\alpha_1 = 1/(2\sigma^2)$ . The joint distribution of the data for individual  $i$  is the product of  $J$  such terms (i.e., contributions from each of  $J$  traps).

$$[\mathbf{y}_i|\mathbf{s}_i, \boldsymbol{\alpha}] = \prod_{j=1}^J \text{Binomial}(K, p_{\boldsymbol{\alpha}}(\mathbf{x}_j, \mathbf{s}_i))$$

We note this assumes that encounter of individual  $i$  in each trap is independent of encounter in every other trap, conditional on  $\mathbf{s}_i$ . This is the fundamental property of the basic model SCR0. The marginal likelihood is computed by removing  $\mathbf{s}_i$ , by integration from the conditional-on-s likelihood, so we compute:

$$[\mathbf{y}_i|\boldsymbol{\alpha}] = \int_{\mathcal{S}} [\mathbf{y}_i|\mathbf{s}_i, \boldsymbol{\alpha}][\mathbf{s}_i]d\mathbf{s}_i$$

In most SCR models,  $[\mathbf{s}] = 1/A(\mathcal{S})$  where  $A(\mathcal{S})$  is the area of the prescribed state-space  $\mathcal{S}$  (but see Chapt. 11 for alternative specifications of  $[\mathbf{s}]$ ).

The joint likelihood for all  $N$  individuals, assuming independence of encounters among individuals, is the product of  $N$  such terms:

$$\mathcal{L}(\boldsymbol{\alpha}|\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N) = \prod_{i=1}^N [\mathbf{y}_i|\boldsymbol{\alpha}]$$

4685 We emphasize that two independence assumptions are explicit in this development: independence of trap-specific encounters within individuals and also independence among  
 4686 individuals. In particular, this would only be valid when individuals are not physically  
 4687 restrained or removed upon capture, and when traps do not “fill up.”

4688 The key operation for computing the likelihood is solving a 2-dimensional integration  
 4689 problem. There are some general purpose **R** packages that implement a number of multi-  
 4690 dimensional integration routines including **adapt** (Genz et al., 2007) and **R2cuba** (Hahn  
 4691 et al., 2010). In practice, we won’t rely on these extraneous **R** packages (except see Chapt.  
 4692 11 for an application of **R2cuba**) but instead will use perhaps less efficient methods in which  
 4693 we replace the integral with a summation over an equal area mesh of points on the state-  
 4694 space  $\mathcal{S}$  and explicitly evaluate the integrand at each point. We invoke the rectangular  
 4695 rule for integration here<sup>1</sup> in which we evaluate the integrand on a regular grid of points  
 4696 of equal area and compute the average of the integrand over that grid of points. Let  
 4697  $u = 1, 2, \dots, nG$  index a grid of  $nG$  points,  $\mathbf{s}_u$ , where the area of grid cells is constant, say  
 4698  $A$ . In this case, the integrand, i.e., the marginal pmf of  $\mathbf{y}_i$ , is approximated by

$$[\mathbf{y}_i | \boldsymbol{\alpha}] = \frac{1}{nG} \sum_{u=1}^{nG} [\mathbf{y}_i | \mathbf{s}_u, \boldsymbol{\alpha}] \quad (5.1.2)$$

4700 This is a specific case of the general expression that could be used for approximating  
 4701 the integral for any arbitrary distribution  $[\mathbf{s}]$ . The general case is

$$[\mathbf{y} | \boldsymbol{\alpha}] = \frac{A(\mathcal{S})}{nG} \sum_{u=1}^{nG} [y | \mathbf{s}_u, \boldsymbol{\alpha}] [\mathbf{s}_u]$$

4702 Under the uniformity assumption,  $[\mathbf{s}] = 1/A(\mathcal{S})$  and thus the grid-cell area cancels in the  
 4703 above expression to yield Eq. 6.1.2. The rectangular rule for integration can be seen as  
 4704 an application of the Law of Total Probability for a discrete random variable  $\mathbf{s}$ , having  
 4705  $nG$  unique values with equal probabilities  $1/nG$ .

### 4706 5.1.1 Implementation (simulated data)

4707 Here we will illustrate how to carry out this integration and optimization based on the  
 4708 integrated likelihood using simulated data (i.e., see Sec. 5.5). Using **simSCR0** we simulate  
 4709 data for 100 individuals and an array of 25 traps laid out in a  $5 \times 5$  grid of traps having unit  
 4710 spacing. The specific encounter model is the Gaussian model. The 100 activity centers  
 4711 were simulated on a state-space defined by an  $8 \times 8$  square within which the trap array was  
 4712 centered (thus the trap array is buffered by 2 units). Therefore, the density of individuals  
 4713 in this system is fixed at 100/64. In the following set of **R** commands we generate the  
 4714 data and then harvest the required data objects:

```
4715 ## simulate a complete data set (perfect detection)
4716 > data <- simSCR0(discard0=FALSE,rnd=2013)
4717     ## extract the objects that we need for analysis
4718 > y <- data$Y
```

---

<sup>1</sup>e.g., [http://en.wikipedia.org/wiki/Rectangle\\_method](http://en.wikipedia.org/wiki/Rectangle_method)

```

4719 > traplocs <- data$traplocs
4720 > nind <- nrow(y) ## in this case nind=N
4721 > J <- nrow(traplocs)
4722 > K <- data$K
4723 > xlim <- data$xlim
4724 > ylim <- data$ylim

```

4725 Now, we need to define the integration grid, say **G**, which we do with the following set of  
 4726 **R** commands (here, **delta** is the grid spacing):

```

4727 > delta <- .2
4728 > xg <- seq(xlim[1]+delta/2,xlim[2]-delta/2,by=delta)
4729 > yg <- seq(ylim[1]+delta/2,ylim[2]-delta/2,by=delta)
4730 > npix <- length(xg)           # valid for square state-space only
4731 > G <- cbind(rep(xg,npix),sort(rep(yg,npix)))
4732 > nG <- nrow(G)

```

4733 In this case, the integration grid is set up as a grid with spacing  $\delta = 0.2$  which produces,  
 4734 for our example, a  $40 \times 40$  grid of points for evaluating the integrand if the state-space  
 4735 buffer is set at 2. We note that the integration grid is set-up here to correspond exactly  
 4736 to the state-space used in simulating the data. However, in practice, we wouldn't know  
 4737 this, and our estimate of  $N$  (for the unknown case, see below) would be sensitive to choice  
 4738 of the extent of the integration grid. As we've discussed previously, density, which is  $N$   
 4739 standardized by the area of the state-space, will not be so sensitive in most cases.

4740 We are now ready to compute the conditional-on-s likelihood and carry out the  
 4741 marginalization described by Eq. 6.1.2. We need to do this by defining an **R** function  
 4742 that computes the likelihood for the integration grid, as a function of the data objects  
 4743 **y** and **traplocs** which were created above. However, it is a bit untidy to store the grid  
 4744 information in your workspace, and define the likelihood function in a way that depends  
 4745 on these things that exist in your workspace. Therefore, we build the **R** function so that  
 4746 it computes the integration grid *within* the function, thereby avoiding potential problems  
 4747 if our trapping grid locations change, or if we want to modify the state-space buffer easily.  
 4748 We therefore define the function, called **intlik1**, to which we pass the data objects and  
 4749 other information necessary to compute the marginal likelihood. This function is available  
 4750 in the **scrbook** package (use **?intlik1** at the **R** prompt). The code is reproduced here:

```

4751 intlik1 <- function(parm,y=y,X=traplocs, delta=.2, ssbuffer=2){
4752
4753   Xl <- min(X[,1]) - ssbuffer ## These lines of code are setting up the
4754   Xu <- max(X[,1]) + ssbuffer ## support for the integration which is
4755   Yu <- max(X[,2]) + ssbuffer ## the same as the state-space of "s"
4756   Yl <- min(X[,2]) - ssbuffer
4757   xg <- seq(Xl+delta/2,Xu-delta/2,,length=npix)
4758   yg <- seq(Yl+delta/2,Yu-delta/2,,length=npix)
4759   npix<- length(xg)
4760
4761   G <- cbind(rep(xg,npix),sort(rep(yg,npix)))

```

```

4762   nG <- nrow(G)
4763   D <- e2dist(X,G)
4764
4765   alpha0 <- parm[1]
4766   alpha1 <- exp(parm[2]) # alpha1 restricted to be positive here
4767
4768   probcap <- plogis(alpha0)*exp(-alpha1*D*D)
4769   Pm <- matrix(NA,nrow=nrow(probcap),ncol=ncol(probcap))
4770           # Frequency of all-zero encounter histories
4771   n0 <- sum(apply(y,1,sum)==0)
4772           # Encounter histories with at least 1 detection
4773   ymat <- y[apply(y,1,sum)>0,]
4774   ymat <- rbind(ymat,rep(0,ncol(ymat)))
4775   lik.marg <- rep(NA,nrow(ymat))
4776
4777   for(i in 1:nrow(ymat)){
4778     ## Next line: log conditional likelihood for ALL possible values of s
4779     Pm[1:length(Pm)] <- dbinom(rep(ymat[i,],nG),K,probcap[1:length(Pm)],
4780           log=TRUE)
4781     ## Next line: sum the log conditional likelihoods, exp() result
4782     ## same as taking the product
4783     lik.cond <- exp(colSums(Pm))
4784     ## Take the average value == computing marginal
4785     lik.marg[i] <- sum(lik.cond*(1/nG))
4786   }
4787   ## n0 = number of all-0 encounter histories
4788   nv <- c(rep(1,length(lik.marg)-1),n0)
4789   return( -1*(sum(nv*log(lik.marg)) ) )
4790 }
```

4791 We emphasize that this function (and subsequent) are not meant to be general-purpose  
 4792 routines for solving all of your SCR problems but, rather, they are meant for illustrative  
 4793 purposes – so you can see how the integrated likelihood is constructed and how we connect  
 4794 it to data and other information that is needed.

4795 The function `intlik1` accepts as input the encounter history matrix, `y`, the trap locations,  
 4796 `X`, and the state-space buffer. This allows us to vary the state-space buffer and easily  
 4797 evaluate the sensitivity of the MLE to the size of the state-space. Note that we have a  
 4798 peculiar handling of the encounter history matrix `y`. In particular, we remove the all-zero  
 4799 encounter histories from the matrix and tack-on a single all-zero encounter history as the  
 4800 last row which then gets weighted by the number of such encounter histories (`n0`). This is  
 4801 a bit long-winded and strictly unnecessary when  $N$  is known, but we did it this way be-  
 4802 cause the extension to the unknown- $N$  case is now transparent (as we demonstrate in the  
 4803 following section). The matrix `Pm` holds the log-likelihood contributions of each encounter  
 4804 frequency for each possible state-space location of the individual. The log contribu-  
 4805 tions are summed up and the result exponentiated on the next line, producing `lik.cond`, the  
 4806 conditional-on-s likelihood (Eq. 6.1.1 above). The marginal likelihood (`lik.marg`) sums  
 4807 up the conditional elements weighted by the probabilities [`s`] (Eq. 6.1.2 above).

4808 This is a fairly primitive function which doesn't allow much flexibility in the data  
 4809 structure. For example, it assumes that  $K$ , the number of replicates, is constant for each  
 4810 trap. Further, it assumes that the state-space is a square. We generalize this to some  
 4811 extent later in this chapter.

4812 Here is the **R** command for maximizing the likelihood using **nlm** (the function **optim**  
 4813 could also be used) and saving the results into an object called **frog**. The output is a list  
 4814 of the following structure and these specific estimates are produced using the simulated  
 4815 data set:

```
4816 # should take 15-30 seconds
4817
4818 > starts <- c(-2,2)
4819 > frog <- nlm(intlik1,starts,y=y,X=traplocs,delta=.1,ssbuffer=2,hessian=TRUE)
4820 > frog
4821
4822 $minimum
4823 [1] 297.1896
4824
4825 $estimate
4826 [1] -2.504824 2.373343
4827
4828 $gradient
4829 [1] -2.069654e-05 1.968754e-05
4830
4831 $hessian
4832 [,1]      [,2]
4833 [1,] 48.67898 -19.25750
4834 [2,] -19.25750 13.34114
4835
4836 $code
4837 [1] 1
4838
4839 $iterations
4840 [1] 11
```

4841 Details about this output can be found on the help page for **nlm**. We note briefly that  
 4842 **frog\$minimum** is the negative log-likelihood value at the MLEs, which are stored in the  
 4843 **frog\$estimate** component of the list. The order of the parameters is as they are defined  
 4844 in the likelihood function so, in this case, the first element (value =  $-2.504824$ ) is the  
 4845 logit transform of  $p_0$  and the second element (value =  $2.373343$ ) is the value of  $\alpha_1$  the  
 4846 “coefficient” on distance-squared. The Hessian is the observed Fisher information matrix,  
 4847 which can be inverted to obtain the variance-covariance matrix using the command:

```
4848 > solve(frog$hessian)
```

4849 It is worth drawing attention to the fact that the estimates are slightly different than  
 4850 the Bayesian estimates reported previously in Sec. 5.6. There are several reasons for this.  
 4851 First Bayesian inference is based on the posterior distribution and it is not generally the

4852 case that the MLE should correspond to any particular value of the posterior distribution.  
 4853 If the prior distributions in a Bayesian analysis are uniform, then the (multivariate) mode  
 4854 of the posterior is the MLE, but note Bayesians almost always report posterior *means*  
 4855 and so there will typically be a discrepancy there. Secondly, we have implemented an  
 4856 approximation to the integral here and there might be a slight bit of error induced by  
 4857 that. We will evaluate that shortly. Third, the Bayesian analysis by MCMC is itself  
 4858 subject to some amount of Monte Carlo error which the analyst should always be aware of  
 4859 in practical situations. All of these different explanations are likely responsible for some  
 4860 of the discrepancy. Accounting for these, we see general consistency between the two  
 4861 estimates.

4862 In summary, for the basic SCR model, computing the integrated likelihood is a simple  
 4863 task when  $N$  is known. Even for  $N$  unknown it is not too difficult, and we will do that  
 4864 shortly. However, if you can solve the known- $N$  problem then you should be able to do a  
 4865 real analysis, for example by considering different values of  $N$  and computing the results  
 4866 for each value and then making a plot of the log-likelihood or AIC and choosing the value  
 4867 of  $N$  that produces the best log-likelihood or AIC. As a homework problem we suggest  
 4868 that you can take the code given above and try to estimate  $N$  without modifying the  
 4869 code by just repeatedly applying it for different values of  $N$  in attempt to deduce the best  
 4870 value. We will formalize the unknown- $N$  problem next.

## 5.2 MLE WHEN N IS UNKNOWN

4871 Here we build on the previous introduction to integrated likelihood but we consider now  
 4872 the case in which  $N$  is unknown. We will see that adapting the analysis based on the  
 4873 known- $N$  model is straightforward for the more general problem. The main distinction is  
 4874 that we don't observe the all-zero encounter history so we have to make sure we compute  
 4875 the probability for that encounter history, which we do by tacking a row of zeros onto the  
 4876 encounter history matrix. In addition, we include the number of such all-zero encounter  
 4877 histories (that is, the number of individuals *not* encountered) as an unknown parameter of  
 4878 the model. Call that unknown quantity  $n_0$ , so that  $N = n_0 + n$  where  $n$  is the number of  
 4879 unique individuals encountered. We will usually parameterize the likelihood in terms of  $n_0$   
 4880 because optimization over a parameter space in which  $\log(n_0)$  is unconstrained is preferred  
 4881 to a parameter space in which  $N$  must be constrained  $N \geq n$ . With  $n_0$  unknown, we have  
 4882 to be sure to include a combinatorial term to account for the fact that, of the  $n$  observed  
 4883 individuals, there are  $\binom{N}{n}$  ways to realize a sample of size  $n$ . The combinatorial term  
 4884 involves the unknown  $n_0$  and thus it must be included in the likelihood. In evaluating the  
 4885 log-likelihood, we have to compute terms such as the log-factorial,  $\log(N!) = \log((n_0+n)!)$ .  
 4886 We do this in **R** by making use of the log-gamma function (`lgamma`) and the identity

$$\log(N!) = \text{lgamma}(N + 1).$$

4887 Therefore, to compute the likelihood, we require the following 3 components: (1) The  
 4888 marginal probability of each  $\mathbf{y}_i$  as before,

$$[\mathbf{y}_i | \boldsymbol{\alpha}] = \int_{\mathcal{S}} [\mathbf{y}_i | \mathbf{s}_i, \boldsymbol{\alpha}] [\mathbf{s}_i] d\mathbf{s}_i.$$

4889 (2) We compute the probability of an all-0 encounter history:

$$\pi_0 = [\mathbf{y} = \mathbf{0} | \boldsymbol{\alpha}] = \int_{\mathcal{S}} \text{Binomial}(\mathbf{0} | \mathbf{s}_i, \boldsymbol{\alpha}) [\mathbf{s}_i] d\mathbf{s}_i$$

4890 (3) The combinatorial term:  $\binom{N}{n}$ . Then, the marginal likelihood has this form:

$$\mathcal{L}(\boldsymbol{\alpha}, n_0 | \mathbf{y}) = \frac{N!}{n! n_0!} \left\{ \prod_{i=1}^n [\mathbf{y}_i | \boldsymbol{\alpha}] \right\} \pi_0^{n_0}. \quad (5.2.1)$$

4891 This is discussed in Borchers and Efford (2008, p. 379) as the conditional-on- $N$  form of the  
4892 likelihood – we also call it the “binomial form” of the likelihood because of its appearance.

4893 Operationally, things proceed much as before: We compute the marginal probability  
4894 of each observed  $\mathbf{y}_i$ , i.e., by removing the latent  $\mathbf{s}_i$  by integration. In addition, we com-  
4895 pute the marginal probability of the “all-zero” encounter history  $\mathbf{y}_{n+1}$ , and make sure to  
4896 weight it  $n_0$  times. We accomplish this by “padding” the data set with a single encounter  
4897 history having  $y_{n+1,j} = 0$  for all traps  $j = 1, 2, \dots, J$ . Then we be sure to include the  
4898 combinatorial term in the likelihood or log-likelihood computation. We demonstrate this  
4899 shortly. To analyze a specific case, we’ll simulate our fake data set (simulated using the  
4900 parameters given above). To set some things up in our workspace we do this:

```
4901 ## Obtain a simulated data set
4902 > data <- simSCRO(discard0=TRUE, rnd=2013)
4903
4904 ## Extract the items we need for analysis
4905 > y <- data$Y
4906 > nind <- nrow(y)
4907 > traplocs <- data$traplocs
4908 > J <- nrow(traplocs)
4909 > K <- data$K
```

4910 Recall that these data are simulated by default with  $N = 100$ , on an  $8 \times 8$  unit state-  
4911 space representing the trap locations buffered by 2 units, although you can modify the  
4912 simulation script easily.

4913 As before, the likelihood is defined in the **R** workspace as an **R** function, **intlik2**,  
4914 which takes an argument being the unknown parameters of the model and additional  
4915 arguments as prescribed. In particular, we provide the encounter history matrix **y**, the  
4916 trap locations **traplocs**, the spacing of the integration grid (argument **delta**) and the  
4917 state-space buffer. Here is the new likelihood function:

```
4918 intlik2 <- function(parm,y=y,X=traplocs,delta=.3,ssbuffer=2){
4919
4920   Xl <- min(X[,1]) - ssbuffer
4921   Xu <- max(X[,1]) + ssbuffer
4922   Yu <- max(X[,2]) + ssbuffer
4923   Yl <- min(X[,2]) - ssbuffer
4924
4925   xg <- seq(Xl+delta/2,Xu-delta/2,delta)
```

```

4926   yg <- seq(Yl+delta/2,Yu-delta/2,delta)
4927   npix.x <- length(xg)
4928   npix.y <- plength(yg)
4929   area <- (Xu-Xl)*(Yu-Yl)/((npix.x)*(npix.y))
4930   G <- cbind(rep(xg,npix.y),sort(rep(yg,npix.x)))
4931   nG <- nrow(G)
4932   D <- e2dist(X,G)
4933   # extract the parameters from the input vector
4934   alpha0 <- parm[1]
4935   alpha1 <- exp(parm[2])
4936   n0 <- exp(parm[3]) # note parm[3] lives on the real line
4937   probcap <- plogis(alpha0)*exp(-alpha1*D*D)
4938   Pm <- matrix(NA,nrow=nrow(probcap),ncol=ncol(probcap))
4939   ymat <- rbind(y,rep(0,ncol(y)))
4940
4941   lik.marg <- rep(NA,nrow(ymat))
4942   for(i in 1:nrow(ymat)){
4943     Pm[1:length(Pm)] <- (dbinom(rep(ymat[i,],nG),K,probcap[1:length(Pm)],
4944       log=TRUE))
4945     lik.cond <- exp(colSums(Pm))
4946     lik.marg[i] <- sum(lik.cond*(1/nG) )
4947   }
4948   nv <- c(rep(1,length(lik.marg)-1),n0)
4949   ## part1 here is the combinatorial term.
4950   ## math: log(factorial(N)) = lgamma(N+1)
4951   part1 <- lgamma(nrow(y)+n0+1) - lgamma(n0+1)
4952   part2 <- sum(nv*log(lik.marg))
4953   return( -1*(part1+ part2) )
4954 }
```

4955 To execute this function for the data that we created with `simSCR0`, we execute the  
 4956 following command (saving the result in our friend `frog`). This results in the usual output,  
 4957 including the parameter estimates, the gradient, and the numerical Hessian which is useful  
 4958 for obtaining asymptotic standard errors (see below):

```

4959 > starts <- c(-2.5,0,4)
4960 > frog <- nlm(intlik2,starts,hessian=TRUE,y=y,X=traplocs,delta=.2,ssbuffer=2)
4961
4962 Warning message:
4963 In nlm(intlik2, starts, hessian = TRUE, y = y, X = traplocs, delta = 0.2, :
4964 NA/Inf replaced by maximum positive value
4965
4966 > frog
4967 $minimum
4968 [1] 113.5004
4969
4970 $estimate
```

```

4971 [1] -2.538333 0.902807 4.232810
4972
4973 [... additional output deleted ...]

```

4974 Executing `nls` here usually produces one or more **R** warnings due to numerical calculations  
4975 happening on extremely small or large numbers (calculation of  $p$  near the edge of the  
4976 state-space), and they also happen if a poor parameterization is used which produces  
4977 evaluations of the objective function beyond the boundary of the parameter space (e.g.,  
4978  $n_0 < 0$ ). Such numerical warnings can often be minimized or avoided altogether by picking  
4979 judicious starting values of parameters or properly transforming or scaling the parameters  
4980 but, in general, they can be ignored. You will see from the `nls` output that the algorithm  
4981 performed satisfactory in minimizing the objective function. The estimate of population  
4982 size,  $\hat{N}$ , for the state-space (using the default state-space buffer) is

```

4983 > Nhat <- nrow(y) + exp(4.2328) #### This is n + MLE of n0
4984 > Nhat
4985 [1] 110.9099

```

4986 Which differs from the data-generating value ( $N = 100$ ), as we might expect for a single  
4987 realization. We usually will present an estimate of uncertainty associated with this MLE  
4988 which we can obtain by inverting the Hessian. Note that  $\text{Var}(\hat{N}) = n + \text{Var}(\hat{n}_0)$ . Since  
4989 we have parameterized the model in terms of  $\log(n_0)$  we use the delta method<sup>2</sup> described  
4990 in Williams et al. (2002, Appendix F4) (see also Ver Hoef, 2012) to obtain the variance  
4991 on the scale of  $n_0$  as follows:

```

4992 > (exp(4.2328)^2)*solve(frog$hessian)[3,3]
4993 [1] 260.2033
4994
4995 > sqrt(260)
4996 [1] 16.12452

```

4997 Therefore, the asymptotic “Wald-type” confidence interval for  $N$  is  $110.91 \pm 1.96 \times 16.125 =$   
4998  $(79.305, 142.515)$ . To report this in terms of density, we scale appropriately by the area  
4999 of the prescribed state-space which is 64 units of area (i.e., an  $8 \times 8$  square). Our MLE  
5000 of  $D$  is  $\hat{D} = 110.91/64 = 1.733$  individuals per square unit. To get the standard error  
5001 for  $\hat{D}$  we need to divide the SE for  $\hat{N}$  by the area of the state-space, and so  $\text{SE}(\hat{D}) =$   
5002  $(1/64) * 16.12452 = 0.252$ .

### 5003 5.2.1 Integrated likelihood under data augmentation

5004 The likelihood analysis developed in the previous sections is based on the likelihood in  
5005 which  $N$  (or  $n_0$ ) is an explicit parameter. This is usually called the “full likelihood” or  
5006 sometimes “unconditional likelihood” (Borchers et al., 2002) because it is the likelihood  
5007 for all individuals in the population, not just those which have been captured, i.e., not that  
5008 which is *conditional on capture*. It is also possible to express an alternative unconditional

<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>

5009 likelihood using data augmentation, replacing the parameter  $N$  with  $\psi$  (e.g., see Sec. 7.1.6  
 5010 Royle and Dorazio, 2008, for an example). We don't go into detail here, but we note that  
 5011 the likelihood under data augmentation is a zero-inflated binomial mixture – precisely an  
 5012 occupancy type model (Royle, 2006). Thus, while it is possible to carry out likelihood  
 5013 analysis of models under data augmentation, we primarily advocate data augmentation  
 5014 for Bayesian analysis.

### 5015 5.2.2 Extensions

5016 We have only considered basic SCR models with no additional covariates. However,  
 5017 in practice, we are interested in covariate effects including “behavioral response”, sex-  
 5018 specificity of parameters, and potentially others. Some of these can be added directly to  
 5019 the likelihood if the covariate is fixed and known for all individuals captured or not. An  
 5020 example is a behavioral response, which amounts to having a covariate  $x_{ik} = 1$  if individ-  
 5021 ual  $i$  was captured prior to occasion  $k$  and  $x_{ik} = 0$  otherwise. For uncaptured individuals,  
 5022  $x_{ik} = 0$  for all  $k$ . Royle et al. (2011b) called this a global behavioral response because the  
 5023 covariate is defined for all traps, no matter the trap in which an individual was captured.  
 5024 We could also define a *local* behavioral response which occurs at the level of the trap, i.e.,  
 5025  $x_{ijk} = 1$  if individual  $i$  was captured in trap  $j$  prior to occasion  $k$ , etc... Trap-specific  
 5026 covariates such as trap type or status, or time-specific covariates such as date, are eas-  
 5027 ily accommodated as well. As an example, Kéry et al. (2010) develop a model for the  
 5028 European wildcat *Felis silvestris* in which traps are either baited or not (a trap-specific  
 5029 covariate with only 2 values), and also encounter probability varies over time in the form  
 5030 of a quadratic seasonal response. We consider models with behavioral response or fixed  
 5031 covariates in Chapt. 7. The integrated likelihood routines we provided above can be  
 5032 modified directly for such cases, which we leave to the interested reader to investigate.

5033 Sex-specificity is more difficult to deal with since sex is not known for uncaptured  
 5034 individuals (and sometimes not even for all captured individuals). To analyze such models,  
 5035 we do Bayesian analysis of the joint likelihood using data augmentation (Gardner et al.,  
 5036 2010b; Russell et al., 2012), discussed further in Chapt. 7. For such covariates (i.e., that  
 5037 are not fixed and known for all individuals), it is somewhat more challenging to do MLE  
 5038 based on the joint likelihood as we have developed above. Instead it is more conventional  
 5039 to use what is colloquially referred to as the “Huggins-Alho” type model which is one of  
 5040 the approaches taken in the software package **secr** (Efford, 2011). We introduce the **secr**  
 5041 package in Sec. 6.5 below.

## 5.3 CLASSICAL MODEL SELECTION AND ASSESSMENT

5042 In most analyses, one is interested in choosing from among various potential models, or  
 5043 ranking models, or something else to do with assessing the relative merits of a set of  
 5044 models. A good thing about classical analysis based on likelihood is we can apply Akaike  
 5045 Information Criterion (AIC) methods (Burnham and Anderson, 2002) without difficulty.  
 5046 AIC is convenient for assessing the relative merits of these different models although if  
 5047 there are only a few models it is not objectionable to use hypothesis tests or confidence  
 5048 intervals to determine importance of effects. A second model selection context has to  
 5049 do with choosing among various detection models, although, as a general rule, we don't

5050 recommend this application of model selection. This is because there is hardly ever (if at  
 5051 all) a rational subject-matter based reason motivating specific distance functions. As a  
 5052 result, we believe that doing too much model selection will invariably lead to over-fitting  
 5053 and thus over-statement of precision. This is the main reason that we haven't loaded you  
 5054 down with a basket of models for detection probability so far, although we discuss many  
 5055 possibilities in Chapt. 7.

5056 **Goodness-of-fit or model-checking** – For many standard capture-recapture mod-  
 5057 els, it is possible to identify goodness-of-fit statistics based on the multinomial likelihood,  
 5058 (Cooch and White, 2006, Chapt. 5), and evaluate model adequacy using formal statistical  
 5059 tests. Similar strategies can be applied to SCR models using expected cell-frequencies  
 5060 based on the marginal distribution of the observations. Also, because computing MLEs  
 5061 is somewhat more efficient in many cases compared to Bayesian analysis, it is sometimes  
 5062 feasible to use bootstrap methods. At the present time, we don't know of any applica-  
 5063 tions of goodness-of-fit testing for SCR models based on likelihood inference, although we  
 5064 discuss the use of Bayesian p-values for assessing model fit in Chapt. 8. An important  
 5065 practical problem in trying to evaluate goodness-of-fit is that, in realistic sample sizes, fit  
 5066 tests often lack the power to detect departures from the model under consideration and  
 5067 so they may not be generally useful in practice.

## 5.4 LIKELIHOOD ANALYSIS OF THE WOLVERINE CAMERA TRAPPING DATA

5068 Here we compute the MLEs for the wolverine data using an expanded version of the  
 5069 function we developed in the previous section. To accommodate that each trap might  
 5070 be operational a variable number of nights, we provided an additional argument to the  
 5071 likelihood function (allowing for a vector  $\mathbf{K} = (K_1, \dots, K_J)$ ), which requires also a modifi-  
 5072 cation to the construction of the likelihood. In addition, we accommodate the state-space  
 5073 is a general rectangle, and we included a line in the code to compute the state-space  
 5074 area which we apply below for computing density. The more general function (`intlik3`)  
 5075 is given in the **R** package `scrbook`. Incidentally, this function also returns the area of  
 5076 the state-space for a given set of parameter values, as an attribute to the function value,  
 5077 which will be used in converting  $\hat{N}$  to  $\hat{D}$ . To use this function to obtain the MLEs for the  
 5078 wolverine camera trap study, we execute the following commands (note: these are in the  
 5079 help file and will execute if you type `example(intlik3)`):

```
5080 > library(scrbook)
5081 > data(wolverine)
5082
5083 > traps <- wolverine$wtraps
5084 > traplocs <- traps[,2:3]/10000
5085 > K.wolv <- apply(traps[,4:ncol(traps)],1,sum)
5086
5087 > y3d <- SCR23darray(wolverine$wcaps,traps)
5088 > y2d <- apply(y3d,c(1,2),sum)
5089
5090 > starts <- c(-1.5,0,3)
5091
```

```

5092 > wolv <- nlm(intlik3,starts,hessian=TRUE,y=y2d,K=K.wolv,X=traplocs,
5093           delta=.2,ssbuffer=2)
5094
5095 > wolv
5096 $minimum
5097 [1] 220.4313
5098
5099 $estimate
5100 [1] -2.8176120 0.2269395 3.5836875
5101
5102 [.... output deleted ....]

```

5103 Of course we're interested in obtaining an estimate of population size for the prescribed  
 5104 state-space, or density, and associated measures of uncertainty which we do using the delta  
 5105 method (Williams et al., 2002, Appendix F4). To do all of that we need to manipulate the  
 5106 output of `nlm` since we have our estimate in terms of  $\log(n_0)$ . We execute the following  
 5107 commands:

```

5108 > wolv <- nlm(intlik3,starts,hessian=TRUE,y=y2d,K=K.wolv,X=traplocs,delta=.2,
5109           ssbuffer=2)
5110 > Nhat <- nrow(y2d)+exp(wolv$estimate[3])
5111 > area <- attr(intlik3(starts,y=y2d,K=K.wolv,X=traplocs,delta=.2,ssbuffer=2),
5112           "SSarea")
5113 > Dhat <- Nhat/area
5114
5115 > Dhat
5116 [1] 0.5494947
5117
5118 > SE <- (1/area)*exp(wolv$estimate[3])*sqrt(solve(wolv$hessian)[3,3])
5119
5120 > SE
5121 [1] 0.1087073

```

5122 Our estimate of density is 0.55 individuals per “standardized unit” which is  $100 \text{ km}^2$ ,  
 5123 because we divided UTM coordinates by 10000. So this is about 5.5 individuals per 1000  
 5124  $\text{km}^2$ , with a SE of around 1.09 individuals. This compares closely with 5.77 reported in  
 5125 Sec. 5.9 based on Bayesian analysis of the model.

#### 5126 5.4.1 Sensitivity to integration grid and state-space buffer

5127 The effect of approximating the integral by a discrete mesh of points is that it induces  
 5128 some numerical error in evaluation of the integral and, further, that error increases as the  
 5129 coarseness of the mesh increases. To evaluate the effect (or sensitivity) of the integration  
 5130 grid spacing, we obtained the MLEs for a state-space buffer of 2 (standardized units) and  
 5131 for integration grid with spacing  $\delta = .3, .2, .1, .05$ . The MLEs for these 4 cases including  
 5132 the relative runtime are given in Table 6.1. We see the results change only slightly as the  
 5133 integration grid changes. Conversely, the runtime on the platform of the day for the 4 cases

5134 increases rapidly. These runtimes could be regarded in relative terms, across platforms,  
 5135 for gaging the decrease in speed as the fineness of the integration grid increases.

**Table 5.1.** Runtime and MLEs for different integration grid resolutions for the wolverine camera trapping data.

$\delta$	Estimates			
	runtime (sec)	$\hat{\alpha}_0$	$\hat{\alpha}_1$	$\log(n_0)$
0.30	9.9	-2.819786	1.258468	3.569731
0.20	32.3	-2.817610	1.254757	3.583690
0.10	115.1	-2.817570	1.255112	3.599040
0.05	407.3	-2.817559	1.255281	3.607158

5136 We studied the effect of the state-space buffer on the MLEs, using a fixed  $\delta = .2$  for  
 5137 all analyses. We used state-space buffers of 1 to 4 units stepped by .5. As we can see  
 5138 (Table 6.2), the estimates of  $D$  stabilize rapidly and the incremental difference is within  
 5139 the numerical error associated with approximating the integral.

**Table 5.2.** Results of the effect of the state-space buffer on the MLE. Given here are the state-space buffer, area of the state-space (area), the MLE of  $N$  ( $\hat{N}$ ) for the prescribed state-space and the corresponding MLE of density ( $\hat{D}$ ).

Buffer	Area	$\hat{N}$	$\hat{D}$
1.0	66.98212	37.73338	0.5633352
1.5	84.36242	46.21008	0.5477567
2.0	103.74272	57.00617	0.5494956
2.5	125.12302	69.03616	0.5517463
3.0	148.50332	82.17550	0.5533580
3.5	173.88362	96.44018	0.5546249
4.0	201.26392	111.83524	0.5556646

#### 5140 5.4.2 Using a habitat mask (Restricted state-space)

5141 In Sec. 5.10 we used a discrete representation of the state-space in order to have control  
 5142 over its extent and shape. This makes it easy to do things like clip out non-habitat, or  
 5143 create a *habitat mask* which defines suitable habitat. Clearly that formulation of the model  
 5144 is relevant to the calculation of the marginal likelihood in the sense that the discrete state-  
 5145 space is equivalent to the integration grid. Thus, for example, we could easily compute  
 5146 the MLE of parameters under some model with a restricted state-space merely by creating  
 5147 the required state-space at whatever grid resolution is desired, and then inputting that  
 5148 state-space into the likelihood function above, instead of computing it within the function.  
 5149 We can easily create an explicit state-space grid for integration from arbitrary polygons or  
 5150 GIS shapefiles which we demonstrate here. Our approach is to create the integration grid  
 5151 (or state-space grid) outside of the likelihood evaluation, and then determine which points  
 5152 of the grid lie in the polygon defined by the shapefile using functions in the **R** packages **sp**  
 5153 and **maptools**. For each point in the state-space grid (object **G** in the code below which is

assumed to exist), we determine whether it is inside the polygon<sup>3</sup>, identifying such points with a value of `mask=1` and `mask=0` for points that are *not* in the polygon. We load the shapefile which originates by an application of the `readShapeSpatial` function. We have saved the result into an **R** data object called `SSp` which is in the `scrbook` package. Here are the **R** commands for doing this (see the helpfile `?intlik4`):

```
5154 > library(maptools)
5155 > library(sp)
5156 > library(scrbook)
5157
5158 #### If we have the .shp file in place, we would use this command:
5159 ##### SSp <- readShapeSpatial('Sim_Polygon.shp')
5160 ##### The object SSp is in data(fakeshapefile)
5161 > data(fakeshapefile)
5162 > Pcoord <- SpatialPoints(G)
5163 > PinPoly <- over(Pcoord,SSp)  ### determine if each point is in polygon
5164 > mask <- as.numeric(!is.na(PinPoly[,1])) ## convert to binary 0/1
5165 > 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, bays, 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 using maximum likelihood we execute the following commands:

```
5171 > library(scrbook)
5172 > data(wolverine)
5173
5174 > traps <- wolverine$wtraps
5175 > traplocs <- traps[,2:3]/10000
5176 > K.wolv <- apply(traps[,4:ncol(traps)],1,sum)
5177
5178 > y3d <- SCR23darray(wolverine$wcaps,traps)
5179 > y2d <- apply(y3d,c(1,2),sum)
5180 > G <- wolverine$grid2/10000
5181
5182 > starts <- c(-1.5,0,3)
5183 > wolv <- nlm(intlik4, starts, y=y2d, K=K.wolv, X=traplocs, G=G)
```

---

<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. 17.5 in Chapt. 17). If you use the `over` function, make sure you know the class of your second argument so that when processing the function output you index it correctly.

**Table 5.3.** 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

```

5191
5192 > wolv
5193
5194 $minimum
5195 [1] 225.8355
5196
5197 $estimate
5198 [1] -2.9955424 0.2350885 4.1104757
5199
5200 [... some output deleted ...]

```

5201 Next we convert the parameter estimates to estimates of total population size for the  
 5202 prescribed state-space, and then obtain an estimate of density (per 1000 km<sup>2</sup>) using the  
 5203 area computed as the number of pixels in the state-space grid, G, multiplied by the area  
 5204 per grid cell. In the present case (the calculation above) we used a state-space grid with 2  
 5205 km × 2 km pixels. Finally, we compute a standard errors using the delta approximation:

```

5206 > area <- nrow(G)*4
5207 # Nhat = n (observed) + MLE of n0 (not observed)
5208 > Nhat <- 21 + exp(wolv$estimate[3])
5209 > SE <- exp(wolv$estimate[3])*sqrt(solve(wolv$hessian)[3,3])
5210 > D <- (Nhat/(nrow(G)*area))*1000
5211 > SE.D <- (SE/(nrow(G)*area))*1000

```

5212 We did this for each the 2 km, 4 km and 8 km state-space grids which produced the  
 5213 estimates summarized in Table 6.3. These estimates compare with the 8.6 (2 km grid)  
 5214 and 8.2 (8 km grid) reported in Royle et al. (2011b) based on a clipped state-space as  
 5215 described in Sec. 5.10.

## 5.5 DENSITY AND THE R PACKAGE SECR

5216 **DENSITY** is a software program developed by Efford (2004) for fitting spatial capture-  
 5217 recapture models based mostly on classical maximum likelihood estimation and related  
 5218 inference methods. Efford (2011) has also released an **R** package called **secr**, that contains  
 5219 much of the functionality of **DENSITY** but also incorporates new models and features.  
 5220 Here, we briefly introduce the **secr** package which we prefer to use over **DENSITY**, be-  
 5221 cause it allows us to remain in the **R** environment for data processing and summarization.  
 5222 We provide a brief introduction to **secr** and some of its capabilities here, and we also use

5223 it for doing some analysis in other parts of this book. We believe that **secr** will be sufficient  
 5224 for many (if not most) of the SCR problems that one might encounter. It provides  
 5225 a flexible analysis platform, with a large number of summary features, and “publication  
 5226 ready” output. Its user-interface is clean and intuitive to **R** users, and it has been stable,  
 5227 efficient and reliable in the (fairly extensive) evaluations that we have done.

5228 To install and run models in **secr**, you must download the package and load it in **R**.

```
5229 > install.packages("secr")
5230 > library(secr)
```

5231 **secr** allows the user to simulate data and fit a suite of models with various detection functions  
 5232 and covariate responses. It also contains a number of helpful constructor functions  
 5233 for creating objects of the proper class that are recognized by other **secr** functions. We  
 5234 provide a brief overview of the capabilities here, but the **secr** help manual can be accessed  
 5235 with the command:

```
5236 > RShowDoc("secr-manual", package = "secr")
```

5237 We note that **secr** has many capabilities that we will not cover or do so only sparingly.  
 5238 We encourage you to read through the manual, the extensive documentation, and the  
 5239 vignettes, in order to get a better understanding of what the package is capable of. We  
 5240 also cover certain capabilities of **secr** in other chapters.

5241 The main model-fitting function in **secr** is called **secr.fit**, which makes use of the  
 5242 standard **R** model specification framework with tildes. As an example, the equivalent of  
 5243 the basic model SCR0 is fitted as follows:

```
5244 > secr.fit(capturedata, model = list(D ~ 1, g0 ~ 1, sigma ~ 1),
5245           buffer = 20000)
```

5246 where **capturedata** is the object created by **secr** containing the encounter history data  
 5247 and the trap information, and the model expression  $g0 \sim 1$  indicates the intercept-only (i.e.,  
 5248 constant) model. Note that we use  $p_0$  for the baseline encounter probability parameter,  
 5249 which is  $g_0$  in **secr** notation. A number of possible models for encounter probability can  
 5250 be fitted including both pre-defined variables (e.g., **t** and **b** corresponding to “time” and  
 5251 “behavior”), and user-defined covariates of several kinds. For example, to include a global  
 5252 behavioral response, this would be written as  $g0 \sim b$ . The discussion of this (global versus  
 5253 local trap-specific behavioral response) and other covariates is developed more in Chapt.  
 5254 7. We can also model covariates on density in **secr**, which we discuss in Chapt. 11. It  
 5255 is important to note that **secr** requires the buffer distance to be defined in meters and  
 5256 density will be returned as number of animals per hectare. Thus to make comparisons  
 5257 between **secr** and output from other programs, we will often have to convert the density  
 5258 to the same units.

5259 Before we can fit the models, the data must first be packaged properly for **secr**.  
 5260 We require data files that contain two types of information: trap layout (location and  
 5261 identification information for each trap), which is equivalent to the trap deployment file  
 5262 (TDF) described in Sec. 5.9 and the capture data file containing sampling *session*, animal  
 5263 identification, trap occasion, and trap location, equivalent in information content to the  
 5264 encounter data file (EDF). Sample session can be thought of as primary period identifier

5265 in a robust design like framework – it could represent a yearly sample or multiple sample  
 5266 periods within a year, each of them producing data on a closed population. We discuss  
 5267 “multi-session” models in more detail below, in Sec. 6.5.4 and Chapt. 14.

5268 There are three important constructor functions that help package-up your data for  
 5269 use in **secr**: **read.traps**, **make.capthist** and **read.mask**. We provide a brief description  
 5270 of each here, but apply them to our wolverine camera trapping data in the next section:

5271 (1) **read.traps**: This function points to an external file *or* **R** data object containing the  
 5272 trap coordinates, and other information, and also requires specification of the type of  
 5273 encounter devices (described in the next section). A typical application of this function  
 5274 looks like the following, invoking the **data=** option when there is an existing **R** object  
 5275 containing the trap information:

```
5276 > trapfile <- read.traps(data=traps, detector="proximity")
```

5277 (2) **make.capthist**: This function takes the EDF and combines it with trap information,  
 5278 and the number of sampling occasions. A typical application looks like this:

```
5279 > capturedata <- make.capthist(enc.data, trapfile, fmt="trapID",  

  5280   noccasions=165)
```

5281 See **?make.capthist** for definition of distinct file formats. Specifying **fmt = trapID** is  
 5282 equivalent to our EDF format.

5283 (3) **read.mask**: If there is a habitat mask available (as described in sec. 6.4.2), then this  
 5284 function will organize it so that **secr.fit** knows what to do with it. The function  
 5285 accepts either an external file name (see **?read.mask** for details of the structure) or a  
 5286  $nG \times 2$  **R** object, say **mask.coords**, containing the coordinates of the mask. A typical  
 5287 application looks like the following:

```
5288 > grid <- read.mask(data=mask.coords)
```

5289 These constructor functions produce output that can then be used in the fitting of models  
 5290 using **secr.fit**.

### 5291 5.5.1 Encounter device types and detection models

5292 The **secr** package requires that you specify the type of encounter device. Instead of  
 5293 describing models by their statistical distribution (Bernoulli, Poisson, etc..), **secr** uses  
 5294 certain operational classifications of detector types including ‘proximity’, ‘multi’, ‘single’,  
 5295 ‘polygon’ and ‘signal’. For camera trapping/hair snares we might consider ‘proximity’  
 5296 detectors or ‘count’ detectors. The ‘proximity’ detector type allows, at most, one detection  
 5297 of each individual at a particular detector on any occasion (i.e., it is equivalent to what  
 5298 we call the Bernoulli or binomial encounter process model, or model SCR0). The ‘count’  
 5299 detector designation allows repeat encounters of each individual at a particular detector  
 5300 on any occasion. There are other detector types that one can select such as: ‘polygon’  
 5301 detector type which allows for a trap to be a sampled polygon (Royle and Young, 2008)  
 5302 which we discuss further in Chapt. 15, and ‘signal’ detector which allows for traps that  
 5303 have a strength indicator, e.g., acoustic arrays (Dawson and Efford, 2009). The detector  
 5304 types ‘single’ and ‘multi’ refer to traps that retain individuals, thus precluding the ability  
 5305 for animals to be captured in other traps during the sampling occasion. The ‘single’ type

5306 indicates trap that can only catch one animal at a time (single-catch traps), while 'multi'  
 5307 indicates traps that may catch more than one animal at a time (multi-catch). These are  
 5308 both variations of the multinomial encounter models described in Chapt. 9.

5309 As with all SCR models, **secr** fits an encounter probability model ("detection function"  
 5310 "in **secr** terminology relating the probability of encounter to the distance of a  
 5311 detector from an individual activity center. **secr** allows the user to specify one of  
 5312 a variety of detection functions including the commonly used half-normal ("Gaus-  
 5313 sian"), hazard rate ("Gaussian hazard"), and (negative) exponential models. There  
 5314 are 12 different functions as of version 2.3.1 (see Table 7.1 in Chapt. 7), but some  
 5315 are only available for simulating data. The different detection functions are defined  
 5316 in the **secr** manual and can be found by calling the help function for the detection  
 5317 function:

5318 > ?detectfn

5319 Most of the detection functions available in **secr** contain some kind of a scale  
 5320 parameter which is usually labeled  $\sigma$ . The units of this parameter default to meters  
 5321 in the **secr** output. We caution that the meaning of this parameter depends on  
 5322 the specific detection model being used, and it should not be directly compared as  
 5323 a measure of home-range size across models. Instead, as we noted in Sec. 5.4 most  
 5324 encounter probability models imply a model of space-usage and fitted encounter  
 5325 models should be converted to a common currency such as "area used."

### 5326 5.5.2 Analysis using the **secr** package

5327 To demonstrate the use of the **secr** package, we will show how to do the same  
 5328 analysis on the wolverine study as shown in Sec. 5.9. To use the **secr** package, the  
 5329 data need to be formatted in a similar but slightly different manner than we use in  
**WinBUGS**.

5330 For example, in Sec. 5.9 we introduced a standard data format for the encounter  
 5331 data file (EDF) and trap deployment file (TDF). The EDF shares the same format  
 5332 as that used by the **secr** package with 1 row for every encounter observation and  
 5333 4 columns representing trap session ('Session'), individual identity ('ID'), sample  
 5334 occasion ('Occasion'), and trap identity ('trapID'). For a standard closed population  
 5335 study that takes place during a single season, the 'Session' column in our case is  
 5336 all 1's, to indicate a single primary sampling occasion. In addition to providing the  
 5337 encounter data file (EDF), we must tell **secr** information about the traps, which  
 5338 is formated as a matrix with column labels 'trapID', 'x' and 'y', the last two being  
 5339 the coordinates of each trap, with additional columns representing the operational  
 5340 state of each trap during each occasion (1=operational, 0=not).

5341 We demonstrate these differences now by walking through an analysis of the  
 5342 wolverine camera trapping data using **secr**. To read in the trap locations and other  
 5343 related information, we make use of the constructor function **read.traps** which also  
 5344 requires that we specify the detector type. The detector type is important because

5346 it will determine the likelihood that **secr** will use to fit the model. Here, we have  
 5347 selected “proximity” which corresponds to the Bernoulli encounter model in which  
 5348 individuals are captured at most once in each trap during each sampling occasion:

```
5349 > library(secr)
5350 > library(scrbook)
5351 > data(wolverine)
5352
5353 > traps <- as.matrix(wolverine$wtraps)
5354 > dimnames(traps) <- list(NULL,c("trapID","x","y",paste("day",1:165,sep="")))
5355 > traps1 <- as.data.frame(traps[,1:3])
5356 > trapfile1 <- read.traps(data=traps1,detector="proximity")
```

5357 Here we note that trap coordinates are extracted from the wolverine data but  
 5358 we do *not* scale them. This is because **secr** defaults to coordinate scaling of meters  
 5359 which is the extant scaling of the wolverine trap coordinates. Note that we add  
 5360 a ‘trapID’ column to the trap coordinates and provide appropriate column labels  
 5361 to the ‘traps’ matrix. An important aspect of the wolverine study is that while  
 5362 the camera traps were operated over a 165 day period, each trap was operational  
 5363 during only a portion of that period. We need to provide the trap operation infor-  
 5364 mation which is contained in the columns to the right of the trap coordinates in  
 5365 our standard trap deployment file (TDF). Unfortunately, this is less easy to do in  
 5366 **secr**<sup>4</sup>, which requires an external file with a single long string of 1’s and 0’s indi-  
 5367 cating the days in which each trap was operational (1) or not (0). The **read.traps**  
 5368 function will not allow for this information on trap operation if the data exists as  
 5369 an **R** object – instead, we can create this external file and then read it back in with  
 5370 **read.traps** using these commands:

```
5371 > hold <- rep(NA,nrow(traps))
5372 > for(i in 1:nrow(traps)){
5373 >   hold[i] <- paste(traps[i,4:ncol(traps)],collapse="")
5374 > }
5375 > traps1 <- cbind(traps[,1:3],"usage"=hold)
5376
5377 > write.table(traps1, "traps.txt", row.names=FALSE, col.names=FALSE)
5378 > trapfile2 <- read.traps("traps.txt",detector="proximity")
```

5379 These operations can be accomplished using the function **scr2secr** which is pro-  
 5380 vided in the **R** package **scrbook**.

5381 After reading in the trap data, we now need to create the encounter matrix or  
 5382 array using the **make.capthist** command, where we provide the capture histories  
 5383 in EDF format, which is the existing format of the data input file **wcaps**. In  
 5384 creating the capture history, we provide also the trapfile created previously, the  
 5385 format (e.g., here EDF format is **fmt= ‘trapID’**), and finally, we provide the  
 5386 number of occasions.

---

<sup>4</sup>as of v. 2.3.1

```

5387 #
5388 # Grab the encounter data file and format it:
5389 #
5390 wolv.dat <- wolverine$wcaps
5391 dimnames(wolv.dat) <- list(NULL,c("Session","ID","Occasion","trapID"))
5392 wolv.dat <- as.data.frame(wolv.dat)
5393 wolvcapt2 <- make.captist(wolv.dat,trapfile2,fmt="trapID",noccasions=165)

```

5394 We also set up a habitat mask using the  $2 \times 2$  km grid which we used previously  
 5395 in the analysis of the wolverine data and then pass the relevant objects to `secr.fit`  
 5396 as follows:

```

5397 #
5398 # Grab the habitat mask (2 x 2 km) and format it:
5399 #
5400 gr2 <- (as.matrix(wolverine$grid2))
5401 dimnames(gr2) <- list(NULL,c("x","y"))
5402 gr2 <- read.mask(data=gr2)
5403 #
5404 # To fit the model we use secr.fit:
5405 #
5406 wolv.secr2 <- secr.fit(wolvcapt2,model=list(D ~ 1, g0 ~ 1, sigma ~ 1),
5407                           buffer=20000,mask=gr2)

```

5408 We are using the “proximity detector” model (SCR0), so we do not need to  
 5409 make any specifications in the command line because we have specified the detec-  
 5410 tor type using the constructor function `read.traps`, except to provide the buffer  
 5411 size (in meters). To specify different models, you can change the default model  
 5412  $D \sim 1$ ,  $g_0 \sim 1$ ,  $\sigma \sim 1$ . We provide all of these commands and additional analyses  
 5413 in the `scrbook` package with the function called `secr_wolverine`. Printing the  
 5414 output object produces the following (slightly edited):

```

5415 > wolv.secr2
5416
5417 secr 2.3.1, 15:52:45 29 Aug 2012
5418
5419 Detector type      proximity
5420 Detector number     37
5421 Average spacing    4415.693 m
5422 x-range            593498 652294 m
5423 y-range            6296796 6361803 m
5424 N animals          : 21
5425 N detections       : 115
5426 N occasions        : 165
5427 Mask area          : 987828.1 ha
5428
5429 Model              : D ~ 1 g0 ~ 1 sigma ~ 1

```

```

5430 Fixed (real)      : none
5431 Detection fn     : halfnormal
5432 Distribution      : poisson
5433 N parameters     : 3
5434 Log likelihood    : -602.9207
5435 AIC              : 1211.841
5436 AICc             : 1213.253
5437
5438 Beta parameters (coefficients)
5439          beta   SE.beta      lcl      ucl
5440 D      -9.390124 0.22636698 -9.833795 -8.946452
5441 g0     -2.995611 0.16891982 -3.326688 -2.664535
5442 sigma  8.745547 0.07664648  8.595323  8.895772
5443
5444 Variance-covariance matrix of beta parameters
5445          D        g0        sigma
5446 D      0.0512420110 -0.0004113326 -0.003945371
5447 g0     -0.0004113326  0.0285339045 -0.006269477
5448 sigma -0.0039453711 -0.0062694767  0.005874683
5449
5450 Fitted (real) parameters evaluated at base levels of covariates
5451      link   estimate  SE.estimate      lcl      ucl
5452 D      log 8.354513e-05 1.915674e-05 5.360894e-05 1.301982e-04
5453 g0     logit 4.762453e-02 7.661601e-03 3.466689e-02 6.509881e-02
5454 sigma  log 6.282651e+03 4.822512e+02 5.406315e+03 7.301037e+03

```

5455 The object returned by `secr.fit` provides extensive default output when printed.  
5456 Much of this is basic descriptive information about the model, the traps, or the  
5457 encounter data. We focus here on the parameter estimates. Under the fitted  
5458 (real) parameters, we find  $D$ , the density, given in units of individuals/hectare  
5459 (1 hectare = 10000  $m^2$ ). To convert this into individuals/1000 km $^2$ , we multiply  
5460 by 100000, thus our density estimate is 8.35 individuals/1000 km $^2$ . The parameter  
5461  $\sigma$  is given in units of meters, and so this corresponds to 6.283 km. Both of these  
5462 estimates are very similar to those obtained in our likelihood analysis summarized  
5463 in Table 6.3 which, for the  $2 \times 2$  km grid, we obtained  $\hat{D} = 8.31$  with a SE of  
5464  $100000 \times 1.915674e - 05 = 1.9156$  and, accounting for the scale difference (1 unit  
5465 = 10000 m in the previous analysis),  $\hat{\sigma} = \sqrt{1/(2\hat{\alpha}_1)} * 10000 = 6.289$  km. The  
5466 difference in the MLE between Table 6.3 and those produced by `secr` could be due  
5467 to subtle differences in internal tuning of optimization algorithms, starting values  
5468 or other numerical settings. In addition, the likelihood is based on a Poisson prior  
5469 for  $N$  (see the next section). On the other hand, the SE is slightly larger based on  
5470 `secr` which is due to a subtle difference in the interpretation of  $D$  under the `secr`  
5471 model (See below).

---

### 5472 5.5.3 Likelihood analysis in the `secr` package

5473 The `secr` package does likelihood analysis of SCR models for most classes of models  
 5474 as developed by Borchers and Efford (2008). Their formulation deviates slightly  
 5475 from the binomial form we presented in Sec. 6.2 above (though Borchers and  
 5476 Efford (2008) also mention the binomial form). Specifically, the likelihood that  
 5477 `secr` implements is that based on removing  $N$  from the likelihood by integrating  
 5478 the binomial likelihood (Eq. 6.2.1 above) over a Poisson prior for  $N$  – what we will  
 5479 call the *Poisson-integrated likelihood* as opposed to the conditional-on- $N$  (*binomial-*  
 5480 *form*) considered previously.

5481 To develop the Poisson-integrated likelihood we compute the marginal proba-  
 5482 bility of each  $\mathbf{y}_i$  and the probability of an all-0 encounter history,  $\pi_0$ , as before, to  
 5483 arrive at the marginal likelihood in the binomial-form:

$$\mathcal{L}(\boldsymbol{\alpha}, n_0 | \mathbf{y}) = \frac{N!}{n! n_0!} \left\{ \prod_i [\mathbf{y}_i | \boldsymbol{\alpha}] \right\} \pi_0^{n_0}$$

5484 Now, what Borchers and Efford (2008) do is assume that  $N \sim \text{Poisson}(\Lambda)$  and they  
 5485 do a further level of marginalization over this prior distribution:

$$\sum_{n_0=0}^{\infty} \frac{N!}{n! n_0!} \left\{ \prod_i [\mathbf{y}_i | \boldsymbol{\alpha}] \right\} \pi_0^{n_0} \frac{\exp(-\Lambda) \Lambda^{n_0}}{N!}$$

5486 In Chapt. 11 we write  $\Lambda = \mu ||\mathcal{S}||$  where  $||\mathcal{S}||$  is the area of the state-space, and  $\mu$  is  
 5487 the density (“intensity”) of the point process. Carrying out the summation above  
 5488 produces exactly this marginal likelihood:

$$\mathcal{L}_2(\boldsymbol{\alpha}, \Lambda | \mathbf{y}) = \left\{ \prod_i [\mathbf{y}_i | \boldsymbol{\alpha}] \right\} \Lambda^n \exp(-\Lambda(1 - \pi_0))$$

5489 which is Eq. 2 of Borchers and Efford (2008) except for notational differences. It  
 5490 also resembles the binomial-form of the likelihood in Eq. 6.2.1 with  $\Lambda^n \exp(-\Lambda \pi_0)$   
 5491 replacing the combinatorial term and the  $\pi_0^{n_0}$  term. We emphasize there are two  
 5492 marginalizations going on here: (1) the integration to remove the latent variables  
 5493  $\mathbf{s}$ ; and, (2) summation to remove the parameter  $N$ . We provide a function for  
 5494 computing this in the `scrbook` package called `intlik3Poisson`. The help file for  
 5495 that function shows how to conduct a small simulation study to compare the MLE  
 5496 under the Poisson-integrated likelihood with that from the binomial form.

5497 The essential distinction between our MLE and Borchers and Efford as imple-  
 5498 mented in `secr` is whether you keep  $N$  in the model or remove it by integration over  
 5499 a Poisson prior. If you have prescribed a state-space explicitly with a sufficiently  
 5500 large buffer, then we imagine there should be hardly any difference at all between  
 5501 the MLEs obtained by either the Poisson-integrated likelihood or the binomial-form

of the likelihood which retains  $N$ . There is a subtle distinction in the sense that under the binomial form, we estimate the realized population size  $N$  for the state-space whereas, for the Poisson-integrated form we estimate the *prior* expected value which would apply to a hypothetical new study of a similar population (see Sec. 5.7.3).

Both models (likelihoods) assume  $s$  is uniformly distributed over space, but for the binomial model we make no additional assumption about  $N$  whereas we assume  $N$  is Poisson using the formulation in **secr** from (Borchers and Efford, 2008). Using data augmentation we could do a similar kind of integration but integrate  $N$  over a binomial  $(M, \psi)$  prior – which we referred to as the binomial-integrated likelihood in Sec. 4.2.4. So obviously the two approaches (data augmentation and Poisson-integrated likelihood) are approximately the same as  $M$  gets large. However, doing a Bayesian analysis by MCMC, we obtain an estimate of both  $N$ , the *realized population size*, and the parameter controlling its expected value  $\psi$  which are, in fact, both identifiable from the data even using likelihood analysis (Royle et al., 2007). That said we can integrate  $N$  out completely and just estimate  $\psi$  as we noted in Sec. 6.2.1 above.

#### 5.5.4 Multi-session models in **secr**

In practice we will often deal with SCR data that have some meaningful stratification or group structure. For example, we might conduct mist-netting of birds on  $K$  consecutive days, repeated, say,  $T$  times during a year, or perhaps over  $T$  years. Or we might collect data from  $R$  distinct trapping grids. In these cases, we have  $T$  or  $R$  groups which we might reasonably regard as being samples of independent populations. While the groups might be distinct sites, year, or periods within years, they could also be other biological groups such as sex or age. Conveniently, **secr** fits a specific model for stratified populations – referred to as *multi-session* models. These models build on the Poisson assumption which underlies the integrated likelihood used in **secr** (as described in the previous section). To understand the technical framework, let  $N_g$  be the population size of group  $g$  and *assume*

$$N_g \sim \text{Poisson}(\Lambda_g).$$

Naturally, we model group-specific covariates on  $\Lambda_g$ :

$$\log(\Lambda_g) = \beta_0 + \beta_1 z_g$$

where  $z_g$  is some group-specific covariate such as a categorical index to the group, or a trend variable, or a spatial covariate, such as treatment effect or habitat structure, if the groups represent spatial units. Under this model, we can marginalize *all*  $N_g$  parameters out of the likelihood to concentrate the likelihood on the parameters  $\beta_0$  and  $\beta_1$  precisely as discussed in the previous section. This Poisson hierarchical model is the basis of the multi-session models in **secr**.

5538 To implement a multi-session model (or stratified population model) in **secr**, we  
 5539 provide the relevant stratification information in the ‘Session’ variable of the input  
 5540 encounter data file (EDF). If ‘Session’ has multiple values then a “multi-session”  
 5541 object is created by default and session-specific variables can be described in the  
 5542 model. For example, if the session has 2 values for males and females then we have  
 5543 sex-specific densities , and baseline encounter probability  $p_0$  ( $g_0$  in **secr**) by just  
 5544 doing this (see Chapt. 8 for the **R** code to set this up):

```
5545 > out <- secr.fit(capdata, model=list(D ~ session, g0 ~ session, sigma~ 1),  

  5546   buffer=20000)
```

5547 More detailed analysis is given in Sec. 8.1 where we fit a number of different  
 5548 models and apply methods of model selection to obtain model-averaged estimates  
 5549 of density.

5550 We can also easily implement stratified population models in the various **BUGS**  
 5551 engines using data augmentation (Converse and Royle, 2012; Royle and Converse,  
 5552 in review) which we discuss, with examples, in Chapt. 14.

### 5553 **5.5.5 Some additional capabilities of secr**

5554 The **secr** package has capabilities to do a complete analysis of SCR data sets,  
 5555 including model fitting, selection, and many summary analyses. In the previous  
 5556 sections, we’ve given a basic overview, and we do more in later chapters of this  
 5557 book. Here we mention a few of these other capabilities that you should know about  
 5558 as you use **secr**. Of course, you should skim through the associated documentation  
 5559 (**?secr**) to see more of what’s available.

#### 5560 **Alternative observation models**

5561 **secr** fits a wide range of alternative observation models besides the Bernoulli en-  
 5562 counter model, including multinomial encounter models for “multi-catch” and “single  
 5563 catch” traps, models for sound attenuation from acoustic detection devices, and  
 5564 many others. We discuss many of these other methods in Chapt. 9 and elsewhere  
 5565 in the book.

#### 5566 **Summary statistics**

5567 **secr** provides a useful default summary of the data, but it also has summary  
 5568 statistics about animal movement including mean-maximum distance moved (the  
 5569 function **MMDM**). For example, see the help page **?MMDM** which lists a number of other  
 5570 summary functions which take a **capthist** object:

```
5571 > moves(capthist)  

  5572 > dbar(capthist)  

  5573 > RPSV(capthist)  

  5574 > MMDM(capthist, min.recapt = 1, full = FALSE)  

  5575 > ARL(capthist, min.recapt = 1, plt = FALSE, full = FALSE)
```

5576 The function `moves` returns the observed distances moved, `dbar` returns the average distance moved, `RPSV` produces a measure of dispersion about the home-range center, and `ARL` gives the *Asymptotic Range Length* which is the asymptote of an exponential model fit to the observed range length vs. the number of detections of each individual (Jett and Nichols, 1987).

#### 5581 **State-space buffer**

5582 `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:

```
5585 Warning message:  
5586 In secr.fit(wolvcapt2, model=list(D ~ 1, g0 ~ 1, sigma ~ 1), buffer=1000):  
5587 predicted relative bias exceeds 0.01 with buffer = 1000
```

5588 This should cause you to contemplate modifying the state-space buffer if that is a reasonable thing to do in the specific application.

#### 5590 **Model selection and averaging**

5591 `secr` does likelihood ratio tests to compare nested models using the function `LR.test`.  
5592 You can create model selection tables based on AIC or AICc, using the function `AIC`,  
5593 and obtain model-averaged parameter estimates using the function `model.average`  
5594 (See Chapt. 8 for examples).

#### 5595 **Population closure test**

5596 `secr` has a population closure test with the function `closure.test` which implements the tests of Stanley and Burnham (1999) or Otis et al. (1978). The function is used like this:

5599 `closure.test(object, SB = FALSE)`. Here `object` is a capthist object and `SB` is  
5600 a logical variable that, if TRUE, produces the Stanley and Burnham (1999) test.

#### 5601 **Density mapping and effective sample area**

5602 `secr` produces likelihood versions of the various summaries of posterior density  
5603 and effective sample area that we discussed in Chapt. 5. For example, while `secr`  
5604 reports estimates of the expected value of  $N$  or density directly in the summary  
5605 output from fitting a model, you can use the function `region.N` to produce estimates  
5606 of  $N$  for any given region. In addition, `secr` has functions for creating maps  
5607 of detection contours for individuals traps, or for the entire trap array. See the  
5608 function `pdot.contour`, and also `fxi.contour` for computing the 2-dimensional  
5609 pdf of the locations of one or more individual activity centers (as in Sec. 5.11.3).  
5610 In the context of likelihood analysis, estimation of a random effect  $s$  is based on a  
5611 plug-in application of Bayes' Rule. When  $s$  has a uniform distribution, and we use  
5612 a discrete evaluation of the integral, it can be computed simply by renormalizing  
5613 the likelihood:

$$[s|y, \theta] = \frac{[y|s, \theta]}{\sum_s [y|s, \theta]}.$$

5614 Any of the `intlik` functions given previously in this chapter can be easily modified  
5615 to return the posterior distribution of `s` for any, or all, individuals, or an individual  
5616 that is not encountered.

5617 Effective sample area (see Sec. 5.12) can be calculated in `secr` using the func-  
5618 tions `esa` and `esa.plot`.

### 5619 Covariate models

5620 `secr` has many capabilities for modeling covariates. It has a number of built-in  
5621 models that allow certain covariates on encounter probability, which we cover to a  
5622 large extent in Chapt. 7, and also see Chapt. 8 for more examples. `secr` also allows  
5623 covariates to be built into the density model (see Chapt. 11). It has some built  
5624 in response surface models, allowing for the fitting of linear or quadratic response  
5625 surfaces. This is done by modifying the density model in `secr.fit`. For example,  
5626  $D \sim 1$  is a constant density surface, and  $D \sim x + y$  fits a linear response surface,  
5627 etc.. See the manual `secr-densitysurfaces.pdf` for the details.

5628 There are a number of ways to model your own “custom” covariates (as op-  
5629 posed to pre-specified models). One way is to use the `addCovariates` function  
5630 and supply it a `mask` or `traps` object along with some “spatialdata.” Or, if you  
5631 have covariates at each trap location then it will extrapolate to all points on the  
5632 habitat mask. There’s also a method by which the user can create a function of  
5633 geographic coordinates, `userDfn`, which seems to provide additional flexibility, al-  
5634 though we haven’t used this method. There is a handy function `predictDsurface`  
5635 for producing density maps under the specified model for density.

## 5.6 SUMMARY AND OUTLOOK

5636 In this chapter, we discussed basic concepts related to classical analysis of SCR  
5637 models based on likelihood methods. Analysis is based on the so-called integrated  
5638 or marginal likelihood in which the individual activity centers (random effects) are  
5639 removed from the conditional-on-`s` likelihood by integration. We showed how to  
5640 construct the integrated likelihood and fit some simple models in the **R** pro-  
5641 gramming language. In addition, likelihood analysis for some broad classes of SCR  
5642 models can be accomplished using the **R** library `secr` (Efford, 2011) which we pro-  
5643 vided a brief introduction to. In later chapters we provide more detailed analyses  
5644 of SCR data using likelihood methods and the `secr` package.

5645 Why or why not use likelihood inference exclusively? For certain specific models,  
5646 it is may be more computationally efficient to produce MLEs (for an example see  
5647 Chapt. 12). And, likelihood analysis makes it easy to do model-selection by AIC  
5648 and compute standard errors or confidence intervals. However, **BUGS** is extremely  
5649 flexible in terms of describing models and we can devise models in the **BUGS**  
5650 language easily that we cannot fit in `secr`. For example, in Chapt 16 we consider  
5651 open population models which are straightforward to develop in **BUGS** but, so  
5652 far, there is no available platform for doing MLE of such models. We can also fit

5653 models in **BUGS** that accommodate missing covariates in complete generality (e.g.,  
5654 unobserved sex of individuals), and we can adopt SCR models to include auxiliary  
5655 data types. For example, we might have camera trapping and genetic data and we  
5656 can describe the models directly in **BUGS** and fit a joint model (Gopalaswamy  
5657 et al., 2012b). To do maximum likelihood estimation, we have to write a custom  
5658 new piece of code for each model<sup>5</sup> or hope someone has done it for us. You should  
5659 have some capability to develop your own MLE routines with the tools we provided  
5660 in this chapter.

---

<sup>5</sup>Although we may be able to handle multiple survey methods together in **secr** using the multi-session models.

5661  
5662

5663

# 6

---

## MODELING ENCOUNTER PROBABILITY

5664 In previous chapters we showed how to fit basic spatial capture-recapture models  
5665 using Bayesian analysis (in **WinBUGS** or **JAGS**; Chapt. 5) or by classical likeli-  
5666 hood methods (Chapt. 6 or using **secr**). We covered a suite of possible encounter  
5667 models (e.g., the Binomial, Poisson, and Multinomial) for dealing with different  
5668 types of sampling. We have not, however, described a general framework for mod-  
5669eling covariates that might influence encounter probability of individuals, traps or  
5670 over time. In practice, investigators are invariably concerned with explicit factors  
5671 or covariates that might influence variation in parameters. Such covariates include  
5672 time (e.g., day of year, or season), behavior (e.g., is there an effect of trapping  
5673 on subsequent capture probabilities), sex of the individual, and trap type (e.g.,  
5674 various camera types, or different constructions for hair snares). Traditionally, in  
5675 the non-spatial capture recapture literature, such models were called “model  $M_t$ ”,  
5676 “model  $M_h$ ”, or “model  $M_b$ ”, identifying models that account for variation in de-  
5677tection probability as a function of time, “individual heterogeneity” or “behavior”,  
5678 where behavior describes whether or not an individual had been previously cap-  
5679 tured. In SCR models, more complex covariate models are possible because we  
5680 might also have trap-specific covariates, or covariates that vary spatially over the  
5681 landscape, and because we generally have more than one parameter describing the  
5682 detection function: Most detection functions include a baseline encounter rate ( $\lambda_0$ )  
5683 or probability ( $p_0$ ) parameter, and a shape parameter ( $\sigma$ ), which takes on differ-  
5684 ent interpretations depending on the specific encounter probability function under  
5685 consideration.

5686 In this chapter, we generalize the basic SCR model to accommodate both alter-  
5687 native detection functions as well as many different kinds of covariates. We focus on  
5688 the binomial observation model used throughout Chaps. 5 and 6 and the Gaussian  
5689 encounter model (also called the “half-normal” model in the distance sampling liter-

ature), but the extension to other observation models is straightforward (and other encounter probability models with different functions of distance are considered in Sec. 7.1). Specifically, we consider three distinct types of covariates – those which are fixed, partially observed or 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 (Sec. 7.4 at the end of the chapter). 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. 11. Alternatively, these landscape covariates might affect the way individuals use space. There are probably very few circumstances under which animals use all space uniformly and we develop more realistic models of encounter probability in which covariates affect space usage in Chapt. 12.

## 6.1 ENCOUNTER PROBABILITY MODELS

In Chapt. 5, we developed a basic spatial capture recapture model using a standard detection probability function based on the kernel of a normal (Gaussian) probability distribution:

$$p_{ij} = p_0 \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

where  $||\mathbf{x}_j - \mathbf{s}_i||$  is the distance between  $\mathbf{x}_j$  and  $\mathbf{s}_i$  and  $\alpha_1 = 1/(2 * \sigma^2)$ . We argued (see Sec. 5.4) 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:

$$\text{logit}(p_{ij}) = \alpha_0 + \alpha_1 ||\mathbf{x}_j - \mathbf{s}_i||. \quad (6.1.1)$$

5725 However, there's nothing preventing us from constructing a myriad of other models  
 5726 for detection probability as a function of distance. The most commonly used detec-  
 5727 tion probability models are also those used in the distance sampling literature: the  
 5728 half-normal (Gaussian), the hazard, and the negative exponential. The negative  
 5729 exponential model is:

$$p_{ij} = p_0 * \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||)$$

5730 where we define  $\alpha_1 = 1/\sigma$ . We could use the general power model (Russell et al.,  
 5731 2012):

$$p_{ij} = p_0 * \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^\theta)$$

5732 of which the Gaussian and exponential models are special cases. Another model  
 5733 that could be considered is the Gaussian hazard rate model (Hayes and Buckland,  
 5734 1983):

$$p_{ij} = 1 - \exp(-\lambda_0 * \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2))$$

5735 which was previously discussed in Sec. .

5736 In each of the cases, the relationship of  $\alpha_1$  to  $\sigma$  varies and must be properly  
 5737 specified. The **R** package **secr** allows the user to access 12 different detection  
 5738 models, of which some are only used for simulating data (see Table 7.1). These  
 5739 detection functions can also be implemented in **R**, **WinBUGS**, **JAGS** etc..

5740 Insofar as all these encounter probability models are symmetric and stationary,  
 5741 they are pretty crude descriptions of space usage by real animals. This is not to  
 5742 say they are inadequate descriptions of the data and, as we discuss in Chaps. 13  
 5743 and 12, we can use them as the basis for producing more realistic models of space  
 5744 usage.

5745 By changing the encounter probability model and the specification of  $\alpha_1$ , we  
 5746 can basically create any function of distance for the data. It is important to note  
 5747 that  $\sigma$  is not comparable under these different encounter probability models and  
 5748 should not be regarded as "home range radius" in general. While there is generally  
 5749 a relationship between  $\sigma$  and home range size, that relationship varies depending  
 5750 on the model under consideration. We demonstrate how to fit different encounter  
 5751 probability models in the Bayesian framework here, and then provide a section on  
 5752 the likelihood analysis (in **secr**) in a separate section below.

### 5753 6.1.1 Bayesian analysis with bear.JAGS

5754 To demonstrate how to incorporate various types of covariates into models for  
 5755 encounter probability using **JAGS**, we return to the data collected during the Fort  
 5756 Drum bear study. This data set was first introduced in Chapt. 4, but, to refresh  
 5757 your memory, there were 38 baited hair snares that were operated between June  
 5758 and July 2006. The snares were checked each week for a total of  $K = 8$  sample

**Table 6.1.** Basic encounter probability models (“distance functions”) available in `secr`. (Table taken from the `secr` help files). Notation deviates from that used in the text. In this table  $g_0$  is the baseline encounter rate or probability parameter used in `secr` 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`).

	Name	Params	Function
0	half-normal	$g_0, \sigma$	$g(d) = g_0 e^{-d^2/(2\sigma^2)}$
1	hazard rate	$g_0, \sigma, z$	$g(d) = g_0(1 - e^{-(d/\sigma)^{-z}})$
2	exponential	$g_0, \sigma$	$g(d) = g_0 e^{-d/\sigma}$
3	compound half-normal	$g_0, \sigma, z$	$g(d) = g_0[1 - \{1 - e^{-d^2/(2\sigma^2)}\}^z]$
4	uniform	$g_0, \sigma$	$g(d) = g_0, d \leq \sigma;$ $g(d) = 0, \text{otherwise}$
5	w exponential	$g_0, \sigma, w$	$g(d) = g_0, d < w;$ $g(d) = g_0 e^{(-(d-w)/\sigma)}, \text{otherwise}$
6	annular normal	$g_0, \sigma, w$	$g(d) = g_0 e^{(-(d-w)^2/(2\sigma^2))}$
7	cumulative lognormal	$g_0, \sigma, z$	$g(d) = g_0[1 - F(d - \mu)/s)]$
8	cumulative gamma	$g_0, \sigma, z$	$g(d) = g_0\{1 - G(d; k, \theta)\}$
9	binary signal strength	$b_0, b_1$	$g(d) = 1 - F\{-(b_0 + b_1 d)\}$
10	signal strength	$\beta_0, \beta_1, S$	$g(d) = 1 - F[\{c - (\beta_0 + \beta_1 d)\}/S]$
11	signal strength spherical	$\beta_0, \beta_1, S$	$g(d) = 1 - F[\{c - (\beta_0 + \beta_1(d-1) - 10 * \log_{10}(d^2))\}/S]$

5759 occasions and  $n = 47$  individual bears were encountered at least once. The data  
 5760 are provided in the **R** package `scrbook` and an **R** function called `bear.JAGS` allows  
 5761 the user to easily pick which model to analyze. The function `bear.JAGS` will set  
 5762 up the data, write the model, define the MCMC specifications (e.g., initial values,  
 5763 etc.) and, finally, run the selected model in **JAGS**. In addition to choosing which  
 5764 model to run, the user can also specify the number of chains, iterations and length  
 5765 of the burn-in phase. Calling the function will provide all the code to implement  
 5766 the models independently as well. In the following sections we will present the  
 5767 model code and output for the most commonly employed models; for all analyses  
 5768 we ran 3 chains with a burn-in of 500 iterations and 20000 saved iterations.

### 5769 6.1.2 Bayesian analysis of encounter probability models

5770 In Panel 7.1, we present the basic SCR model and show how to specify the negative  
 5771 exponential encounter probability model. To call each of these from the function  
 5772 `bear.JAGS` set `model='SCR0'` or `model='SCRexp'` in the function call, respectively.  
 5773 To reduce repetition of the R coding, we include the basic code here and then only  
 5774 show modifications when necessary throughout the chapter. All of the R coding can  
 5775 be found within the `bear.JAGS` function as well. To begin, the required R libraries  
 5776 are installed and then we attached the Ft. Drum bear data set. The bear data set

5777 includes a 3-d data array (called `bearArray` in our code), with dimensions `nind` ×  
 5778 `ntraps` × `nreps` representing the capture histories of `nind` captured individuals  
 5779 at `ntraps` trap locations. In the Bayesian analysis, data augmentation is used to  
 5780 estimate  $N$  and therefore the `bearArray` data must be augmented with  $M - nind$   
 5781 all zero encounter histories. In models without time dependence, the augmented  
 5782 `bearArray` (called `Yaug` in the code) will be reduced to a 2 dimensional array  
 5783 (denoted `y` in the code) that has dimensions `M` × `ntraps`.

```

5784 > library(rjags) #load the necessary libraries
5785 > library(scrbook)
5786
5787 > data(beardata) #attach the bear data for Ft. Drum
5788 > ymat <- beardata$bearArray
5789 > trapmat <- beardata$trapmat
5790 > nind <- dim(beardata$bearArray)[1]
5791 > K <- dim(beardata$bearArray)[3]
5792 > ntraps <- dim(beardata$bearArray)[2]
5793 > M <- 650
5794 > nz <- M-nind
5795
5796 #create augmented array
5797 > Yaug <- array(0, dim=c(M,ntraps,K))
5798 > Yaug[1:nind,,] <- ymat
5799 > y <- apply(Yaug,1:2, sum)
```

5800 The function `bear.JAGS` also establishes the upper and lower limits on the  
 5801 state space by centering the trap array coordinates (which are imported with the  
 5802 `beardata` and saved in the code above as `trapmat`) and then buffering by 20km.

5803 Applying the SCR model with Gaussian encounter probability model provides  
 5804 an estimate (posterior mean) of  $D = 0.167$  bears per  $km^2$  and with the negative  
 5805 exponential encounter probability model the posterior mean is virtually the same  
 5806  $D = 0.167$ . In distance sampling, the use of different encounter probability  
 5807 models often results in very different estimates of density (especially when using the  
 5808 negative exponential model). There are two main reasons why the different models  
 5809 may have less of an impact on the density estimates under the SCR models.  
 5810 First, we can estimate the baseline encounter probability parameter ( $p_0$ ). In most  
 5811 distance sampling models, detection at distance 0 is set to 1. In Table 7.2, the  
 5812 posterior mean of  $p_0$  is 0.11 under the Gaussian model and 0.34 under the negative  
 5813 exponential model. The larger baseline encounter probability under the negative  
 5814 exponential model reduces the impact of the having “no shoulder”. Secondly, the  
 5815 detection probability function here is governing ‘movement’ of individuals (which  
 5816 we have more information on than in distance sampling), not the whole detection  
 5817 process, so the shape of the detection probability function does not impact the  
 5818 density estimation as much.

---

```

model{
alpha0 ~ dnorm(0,.1)                                # Prior distributions
logit(p0) <- alpha0
alpha1 <- 1/(2*sigma*sigma)
sigma ~ dunif(0, 15)
psi ~ dunif(0,1)

for(i in 1:M){
z[i] ~ dbern(psi)
s[i,1] ~ dunif(xlim[1],xlim[2])
s[i,2] ~ dunif(ylim[1],ylim[2])
for(j in 1:J){
d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
y[i,j] ~ dbin(p[i,j],K)
p[i,j] <- z[i]*p0*exp(- alpha1*d[i,j]*d[i,j])  # Gaussian model
#p[i,j] <- z[i]*p0*exp(- alpha1*d[i,j])          # exponential model
}
}
N <- sum(z[])
D <- N/area
}

```

---

Panel 6.1: **JAGS** model specification for a basic SCR model with Gaussian encounter probability function and the alternative exponential encounter probability function.

5819 In all analyses it is important to check that the size of the augmented data set  
 5820 ( $M$ ) is sufficiently large and does not impact the estimate of  $N$ . Here, the 97.5%  
 5821 percentile for  $N$  is 628 (Table 7.2), thus not reaching our  $M = 650$  value. We could  
 5822 also increase  $M$  and compare the posterior of  $N$  under the different scenarios as  
 5823 another check that the data augmentation is sufficient.

**Table 6.2.** Posterior summaries of SCR model parameters having different encounter probability models, for the Fort Drum black bear data.

Parameter	Mean	SD	2.5	97.5
<b>Gaussian</b>				
$D$	0.17	0.022	0.122	0.207
$N$	500.63	66.652	371.000	628.000
$p_0$	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
<b>Exponential</b>				
$D$	0.17	0.022	0.130	0.210
$N$	512.06	65.771	382.000	634.000
$p_0$	0.34	0.056	0.246	0.465
$\psi$	0.79	0.102	0.584	0.974
$\sigma$	1.12	0.095	0.951	1.323

5824 A very important consideration when using different detection probability functions  
 5825 is the interpretation of  $\sigma$ . The estimate of  $\sigma$  under the negative exponential  
 5826 model is 1.12, which is distinct from our estimate of  $\sigma$  under the Gaussian model,  
 5827  $\sigma = 1.996$ . The interpretation of  $\sigma$  in the two models is really quite distinct. In  
 5828 the normal model it can be interpreted as the standard deviation of a bivariate  
 5829 normal movement model whereas the manner in which  $\sigma$  relates to “area used” for  
 5830 the negative exponential model has nothing to do with a bivariate normal model of  
 5831 movement. This highlights that it is important for the user to know what detection  
 5832 probability function is used and what the interpretation of  $\sigma$  might be in relation  
 5833 to the home range size. This relationship was discussed in Sec. 5.4.

5834 We now move onto incorporating covariates into the model using the **JAGS**  
 5835 language. For this part, we will stick with the Gaussian encounter probability  
 5836 model shown in the Panel 7.1 above.

## 6.2 MODELING COVARIATE EFFECTS

5837 The basic strategy for modeling covariate effects is to include them on the baseline  
 5838 encounter rate or probability parameter,  $p_0$  (or  $\lambda_0$ ), or the scale parameter of the  
 5839 encounter model,  $\sigma$ , or in some cases, both parameters.

5840 Broadly speaking, we recognize (here) 3 types of covariates. Fixed covariates  
 5841 that are fully observable and might vary by trap alone (e.g., type of trap, baited  
 5842 or not, disturbance regime, even habitat), sample occasion (e.g., day of season or

weather conditions), or both (e.g., behavior, weather - if over a large region). Another class of covariates are those which vary at the level of the individual (and possibly also over time). As a technical matter, and as noted before, these are different from fixed covariates because we cannot see all of the individuals and the covariates are almost always incompletely observed (if at all). The lone exception is the behavioral response to capture which is known for all individuals, captured or not (an animal never captured/observed has never been captured before). We noted many times before that space itself (i.e., the activity centers) is a type of individual covariate and this notion actually helped us derive the fully spatial capture-recapture model from the traditional, non-spatial model (Chapt. 4). We do not get to observe the activity center for any individuals, but for individuals that are encountered we get to observe some information about it in the form of which traps the individual was encountered in. And finally, we have completely unobserved covariates such as heterogeneity in home range size. We consider heterogeneity in a separate section below since alone there are a suite of models for describing latent heterogeneity.

**Table 6.3.** Examples of different types of covariates in SCR models.

Covariate type	Examples
individual	sex, age, home range
trap	baited/not, habitat (see also Chapter 13)
time	season, shedding, weather
individual x time	global behavioral response
trap x time	trap failures
individual x trap x time	local behavioral response

To develop covariate models, we assume a standard sampling design in which an array of  $J$  traps is operated for  $K$  sample occasions, which produces encounter histories for  $n$  individuals. For the null model, there are no time-varying covariates that influence encounter, there are no explicit individual-specific covariates, and there are no covariates that influence density. For fixed effects, those which we observe fully, we can easily incorporate these into the encounter probability model, just as we would do in any standard GLM or GLMM, on some suitable scale for the encounter probability,  $p_{ijk}$ . For example,

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * C_{ijk}$$

$$p_{ijk} = p_{0,ijk} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

where  $C_{ijk}$  is some covariate that varies (potentially) by individual ( $i$ ), trap ( $j$ ) and occasions ( $k$ ), and  $\alpha_2$  is the coefficient to be estimated. How we define specific covariates (e.g., trap specific versus individual specific) will influence exactly how we include them in the model. Table 7.3 shows examples of covariates by type – trap, individual, and time – and also gives examples of some combined types. These

5872 are the types of covariates we will specifically address in this chapter demonstrating  
 5873 how to analyze the different types in the following sections.

5874 **6.2.1 Date and time**

5875 Often, researchers are interested in modeling the effect of date or chronological  
 5876 time on encounter probability. For example, in a long term hair snare study, we  
 5877 may expect that seasonal shedding (Wegan et al., 2012) will influence encounter  
 5878 probabilities directly. Or, we may expect behaviors such as denning, mating, etc.,  
 5879 to influence the encounter of certain species at certain times of year (Kéry et al.,  
 5880 2011). There are two common ways to incorporate date or time information into  
 5881 a model for encounter probability. For cases with a small number of sampling  
 5882 occasions we can fit a time-specific intercept (analogous to “model  $M_t$ ” in classical  
 5883 capture-recapture (Otis et al., 1978)). In this model, there are  $K$  sampling occasion-  
 5884 specific parameters to reflect potential variation in sampling effort or other factors  
 5885 that might vary across samples. Alternatively, we can model parametric functions  
 5886 of date or time such as polynomial or sinusoidal functions.

5887 In the first case, we allow each sampling occasion,  $k$ , to have its own baseline  
 5888 encounter probability, e.g.,

$$\text{logit}(p_{0,k}) = \alpha_{0,k}$$

5889 so that

$$p_{ijk} = p_{0,k} \exp(-\alpha_1 * \|\mathbf{x}_j - \mathbf{s}_i\|^2).$$

5890 This description of the model includes  $k$  occasion-specific baseline encounter proba-  
 5891 bilities. Thus, if there were 4 sampling occasions, then there are 4 different baseline  
 5892 encounter probabilities. We imagine that complete time-specificity of  $p_0$  (i.e., one  
 5893 distinct value for each sample occasion) would be most useful in situations where  
 5894 there are just a few sampling occasions (if there are many, this formulation will dra-  
 5895 matically increase the number of parameters to be estimated) or we do not expect  
 5896 systematic patterns over time (e.g., explainable by a polynomial function).

5897 To implement this in **JAGS**,  $\alpha_0$  has to be estimated for each time period  $k$   
 5898 either using an index vector or dummy variables (as described in Chapt. 2 and Sec.  
 5899 4.3) and this can be done by only changing only a few lines in Panel 7.1:

```
5900 alpha0[k] ~ dnorm(0,.1)
5901 logit(p0[k]) <- alpha0[k]
5902 .....
5903 .....
5904 y[i,j,k] ~ dbin(p[i,j,k],K)
5905 p[i,j,k] <- z[i]*p0[k]*exp(- alpha1*d[i,j]*d[i,j])
```

5906 Since the model contains a parameter for each time period, the encounter his-  
 5907 tories must be time-dependent. Thus, a 3-d data array (called **bearArray** in our  
 5908 code), with dimensions **nind** × **ntraps** × **nreps** is required (recall that we use the 3-d

augmented array called `Yaug` with dimensions `M × ntraps × nreps` for the Bayesian analysis). In addition to using the 3-d data array, the initial values must be updated so that there are  $K$  values generated for  $\alpha_0$ . And finally, this means that another nested for loop is needed in the code to account for the  $K$  sample occasions. A side note: the computation time will increase quite a bit (this model for the bear data may take up to 15 hours or more on your machine to obtain a sufficient posterior sample).

Running this model with the function `bear.JAGS` by setting `model=SCRt`, returns estimates of density similar to those from the model without covariates (see Table 7.4), but now we have a characterization of variation in encounter probability over time. Encounter probability seems to increase for the first few time periods before stabilizing around 0.14, dropping off again at the end of the study. The differences in encounter probability from the first time periods to the others might actually be due to something like a behavioral response (see below) or possibly seasonal differences in the efficiency of the sampling technique. Researchers have found that hair snares are more effective at different times of the year (even within season) due to shedding (Wegan et al., 2012). In this particular example, our density estimates are similar to the base model, likely because the differences in encounter probability between occasion were not that large. In a longer term study or in one with greater variation in the encounter probability, the implication of such differences might have a bigger impact on the estimates of density and  $\sigma$ .

**Table 6.4.** Posterior summaries of parameter estimates from a SCR model with time-dependent baseline encounter probability for the Ft. Drum black bear data set.

Parameter	Mean	SD	2.5	97.5
$D$	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

The occasion specific intercepts (baseline encounter probability) model might not be the most appropriate for all scenarios (and could require the estimation of many parameters if we had many sampling occasions, take the wolverine example from Chapt. 5.9 where there were 165 daily sampling occasions). Particularly in such a case, variation in the encounter process over time is to be expected. For example, if a camera trap study is conducted for an entire year, it is expected that

5936 there would be behavioral patterns in individuals due to mating or denning. Instead  
 5937 of fitting a model with  $K$  baseline encounter probabilities, we can include date as  
 5938 a linear (or quadratic, ...) effect. An example can be found in Kéry et al. (2011)  
 5939 who incorporated a day-of-year covariate, both as a linear and a quadratic effect,  
 5940 into their SCR model of European wildcats; the data had been collected over a  
 5941 year-long period and cat behavior was expected to vary seasonally thus influencing  
 5942 the probability of encounter. In these cases, we would specifically incorporate day  
 5943 of year (variable “Date”) as a numeric covariate as:

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * \text{Date}_k \\ p_{ijk} = p_{0,ijk} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

5944 or a quadratic effect of day-of-year:

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * \text{Date}_k + \alpha_3 * \text{Date}_k^2 \\ p_{ijk} = p_{0,ijk} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

5945 where the variable Date is an integer coding of day-of-year, indexed to some arbitrary  
 5946 start point in time.

### 5947 6.2.2 Trap-specific covariates

5948 In some studies it makes sense to model encounter probability as a function of local  
 5949 or trap-specific covariates. These can be one of two types: genuine trap covariates  
 5950 that describe the trap or encounter site, such as whether a trap is baited or not,  
 5951 or how many traps were set at a sampling location, or what kind of bait was used,  
 5952 etc., or local covariates that describe the likelihood that an animal would use the  
 5953 habitat in the vicinity of the trap (see Chapt. 13 for more on this situation).  
 5954 We imagine that these covariates, of either type, should affect baseline encounter  
 5955 probability. For example, Sollmann et al. (2011) found a large difference in the  
 5956 encounter probability of jaguars due to traps being located on roads, which the  
 5957 animals were using to travel along, as opposed to traps placed off of roads. In this  
 5958 case, the trap type is a binary variable – on/off road, (another binary variable could  
 5959 be baited/non-baited). We can write this as:

$$\text{logit}(p_{0,j}) = \alpha_{0,type_j} \\ p_{ijk} = p_{0,j} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2).$$

5960 Here, we use an index variable, “type”, an integer value for the trap-specific co-  
 5961 variate. Thus for our example of on/off road, we would have  $type_j = 1$  if trap  
 5962  $j$  is on a road and  $type_j = 2$  otherwise, and we would estimate two separate  $\alpha_0$   
 5963 parameters – one for on-road and one for off-road cameras. This general set up also  
 5964 allows for more than 2 categories, say if 4 different camera models were used in a

study, we would use a set of 3 binary dummy variables to allow for estimation of the different encounter rates (i.e., the intercept). To express the model in terms of dummy variables using the 2-category example above, we would specify our “type” vector as  $Type_j = 0$  if trap  $j$  is on a road and  $Type_j = 1$  otherwise, and write our model as

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * Type_j$$

Now,  $\alpha_0$  is the baseline encounter probability (on the logit scale) for traps on a road ( $Type_j = 0$ ) and  $\alpha_2$  is the effect on baseline encounter probability of a trap being of Type = 1. While these models are equivalent, and should yield identical results, sometimes one parameterization might work better than the other in **WinBUGS** or **JAGS** (Kéry, 2010).

### 6.2.3 Behavior or trap response by individual

One of the most basic of encounter models is that which accommodates a change in encounter probability as a result of initial encounter. This is colloquially referred to as “trap happiness” or “trap shyness”, or in other words, a behavioral response of individuals to being captured (Otis et al., 1978). If a trap is baited with a food source, an individual might come back for more. On the other hand, if being captured is traumatic then an individual might learn to avoid traps. Both of these types of responses can occur in most species depending on the type of encounter mechanisms being employed. Moreover, behavioral response can be either global (Gardner et al., 2010b) or local (Royle et al., 2011b). The local response is a trap-specific response while a global response suggests that initial capture provides a net increase or decrease in subsequent probabilities of capture (across all traps). A behavioral response does not need to be enduring (i.e., persist for the entire study after the individual has been captured/observed for the first time) but can also be ephemeral, if, for example, an animal only avoids a trap on the occasion immediately after it was captured (Yang and Chao, 2005; Royle, 2008). While we will focus the examples in this chapter on enduring behavioral effects, extending such a model to the case of an ephemeral response should not pose any difficulties.

To describe these behavioral models we need to create a binary matrix that indicates if an individual has been captured previously. For the global behavioral response, define the  $n \times K$  matrix,  $\mathbf{C}$ , where  $C_{ik} = 1$  if individual  $i$  was captured at least once prior to session  $k$ , otherwise  $C_{ik} = 0$ .

$$\begin{aligned}\text{logit}(p_{0,ik}) &= \alpha_0 + \alpha_2 * C_{ik} \\ p_{ijk} &= p_{0,ik} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2)\end{aligned}$$

For the local behavioral response, which is trap specific, we create an array,  $C_{ijk}$ , that indicates if an individual  $i$  has been previously captured in trap  $j$  at time  $k$ . We then include this in the model in the exact same form as above (with the sole

6000 difference that both  $C$  and  $p$  are now also indexed by  $k$ ):

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * C_{i,j,k}$$

$$p_{ijk} = p_{0,ijk} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

6001 Since the behavioral response is occasion specific, to implement either the local  
 6002 or global response model in **JAGS**, we will have to use the 3-d array of the  
 6003 augmented capture histories ( $M \times ntraps \times nreps$ ) as we did for the time-varying  
 6004 encounter probability model above. The code must loop over each sampling occa-  
 6005 sion, but otherwise, the model varies only a little from the basic SCR model shown  
 6006 in Panel 7.1. Here is the specification of the the occasion specific ( $k$ ) loop:

```
6007 for(k in 1:K){
  6008   logit(p0[i,j,k]) <- alpha0 + alpha2*C[i,j,k]
  6009   y[i,j,k] ~ dbin(p[i,j,k],1)
  6010   p[i,j,k] <- z[i]*p0[i,j,k]*exp(- alpha1*d[i,j]*d[i,j])
  6011 }
```

6012 Despite the minor changes to the **BUGS** code, this model can require quite a bit  
 6013 of time and computational effort to carry out the behavior response models. Imple-  
 6014 menting the behavioral models with the function **bear.JAGS** by setting **model=SCRb**  
 6015 or **model=SCRb** for the local or global model respectively, returns the results, shown  
 6016 in Table 7.5. There is a strong global behavior response suggested by the posterior  
 6017 mean of  $\alpha_2 = 0.90$ . The estimate of  $N$  and subsequently  $D$  are larger than under  
 6018 the model wihout a behavioral response, here we estimate  $N = 577.56$  and in the  
 6019 SCR0 model, we estimated  $N = 500$ . This makes sense given the large estimate of  
 6020  $\alpha_2$ , which suggests that bears are trap happy. In situations where animals are trap  
 6021 happy, the model tends to over estimate encounter probability (i.e., the bears that  
 6022 are never observed have a lower encounter probability than those that have been  
 6023 captured in the study) and thereby reduce the estimate of  $N$ . We do not include  
 6024 the results here, but the estimates were similar under the local behavioral response  
 6025 model.

**Table 6.5.** Posterior summaries of parameter estimates from the SCR model with a global behavioral response of encounter for the Fort 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

---

 6026 **6.2.4 Individual covariates**

6027 Individual covariates are those which are measured (or measurable) on individuals,  
 6028 so we get to observe them only for the captured individuals. Sex is a simple  
 6029 example of an individual covariate, but one of the most commonly used in capture-  
 6030 recapture studies. The sex of an individual can influence many aspects of its ecology  
 6031 and behavior, including, for example, its home range size, frequency of movement,  
 6032 and seasonal behavior. This is common in studies of carnivores where females  
 6033 often have smaller home ranges than males (Gardner et al., 2010b; Sollmann et al.,  
 6034 2011). Additionally, we may find differences in the baseline encounter probability  
 6035 between males and females because females may move around less frequently, or  
 6036 possibly because they are less likely to use landscape structures that researchers  
 6037 may target with sampling devices in order to increase sample size, such as roads  
 6038 (e.g. Salom-Pérez et al., 2007). Therefore, we can imagine that sex may impact  
 6039 both the baseline encounter probability  $\alpha_0$  and the typical home range size, so that  
 6040  $\alpha_1$  might also be sex-specific also. The fully sex-specific model is:

$$\text{logit}(p_{0,i}) = \alpha_{0,\text{sex}_i}$$

$$p_{ijk} = p_{0,i} \exp(-\alpha_{1,\text{sex}_i} * \|\mathbf{x}_j - \mathbf{s}_i\|^2)$$

6041 where  $\text{sex}_i$  is a vector indicating the sex of each individual (1 = male, 2 = female).  
 6042 While we might know the sex of all individuals observed in the study, we will  
 6043 never know the sex of individuals that are not observed, resulting in missing values  
 6044 (Gardner et al., 2010b). It is also possible that we may not be able to determine the  
 6045 sex of individuals that are observed during the study. For example photographic  
 6046 captures do not necessarily result in pictures that allow the sex to be absolutely  
 6047 determined, thus sometimes resulting in missing values of this covariate for animals  
 6048 captured in the study. We deal with this slightly differently depending on the  
 6049 inference framework that we adopt (Bayesian or likelihood). Here we demonstrate  
 6050 the Bayesian implementation and we discuss the likelihood approach using **secr** in  
 6051 detail below in Sec. 7.4.2. Before proceeding with that, we note that it would be  
 6052 possible also to model covariates directly on the parameter  $\sigma$  (or its logarithm),  
 6053 e.g.,  $\log(\sigma_i) = \theta_1 + \theta_2 \text{sex}_i$  (see Sec. 8.1). One or the other (or perhaps *some* other)  
 6054 parameterization may yield a better performing MCMC algorithm or provide a  
 6055 more natural or preferred interpretation. In the context of Bayesian analysis, given  
 6056 that priors are not invariant to transformation of the parameters, this may be a  
 6057 consideration in choosing the particular parameterization.

6058 Specifying a fully sex-specific model for **JAGS** is similar to the time-specific  
 6059 model shown above. We need to use an index or dummy variable to let  $\alpha_0$  and/or  
 6060  $\alpha_1$  be defined separately for males and females. The main difference in this specification  
 6061 is that we do not observe sex for the augmented individuals. Therefore,  
 6062 we have missing observations of the covariate for those individuals. As a result, sex  
 6063 is regarded as a random variable and so the missing values can be estimated along  
 6064 with the other structural parameters of the model.

Because we are regarding sex as a random variable, we have to specify a distribution for it. With only two possible outcomes, it is natural to suppose that  $\text{Sex}_i \sim \text{Bernoulli}(\psi_{\text{sex}})$  where the parameter  $\psi_{\text{sex}}$  is the sex ratio of the population. We assume our default non-informative prior for this parameter:  $\psi_{\text{sex}} \sim \text{Uniform}(0, 1)$ . The model specification in Panel 7.2 demonstrates how to incorporate a partially observed covariate (i.e., “sex”). It is important to note that in the previous equation,  $\text{sex}_i$  is a vector with two categories indicating the sex of each individual (e.g., 1 = male, 2 = female). This corresponds directly to having a binary indicator of sex (e.g.,  $\text{Sex}_i = 1$  if individual  $i$  is female, and 0 otherwise). In the Bayesian formulation of the model, we use both the binary indicator (`Sex`) and a categorical indicator (`Sex2 = Sex + 1`). The former (termed `Sex` in Panel 7.2) allows us to specify the Bernoulli distribution for the random variable, and the latter (termed `Sex2`) allows us to use the dummy or indicator variable specification in the model.

In both **JAGS** or **BUGS** missing data are indicated by `NA` in the data objects passed to the program through the `bugs` or `jags` functions in **R**. To set up the data, we need to create a vector of length  $M$  with the first  $n$  elements being 0 if individual  $i$  is a female, or 1 if  $i$  is a male (for the Fort Drum black bear data the function `bear.JAGS` extracts this information automatically from the `beardata` object), and the subsequent  $M - n$  elements being `NA`. It is generally a good idea to provide starting values for the missing data, but we cannot provide starting values for observed data; in this case where one vector (or other object) contains both observed and missing data, initial values for the observed data have to be specified as `NA`. The code snippet below shows you how to set up the data including the `Sex` vector and the initial values function (the remainder of the code is identical to what we've shown before).

```
6090 > sex <- beardata$sex #the sex data for captured individual
6091 > Sex <- c(sex-1, rep(NA, nz)) #sex enters as 1/2, this recodes it to 0/1
6092                                     #so we can use Bernoulli distribution
6093
6094 > data <- list(y=y, Sex=Sex, M=M, K=K, J=ntraps, xlim=xlim, ylim=ylim, area=areaX)
6095 > params <- c('psi', 'p0', 'N', 'D', 'sigma', 'psi.sex')
6096 > inits <- function() { list(z=c(rep(1,nind), rbinom(nz,1,0.5)), psi=runif(1),
6097                           s=cbind(runif(M, xlim[1],xlim[2]), runif(M,ylim[1],ylim[2])),
6098                           psi.sex=runif(1), Sex=c(rep(NA, nind), rbinom(nz,1,0.5)),
6099                           sigma=runif(2,2,3), alpha0=runif(2)) }
```

6100 The **BUGS** model specification is shown in Panel ??.

6101 Our estimate of density under the fully sex-specific model is still very similar  
 6102 to the previous models (Table 7.6), and while the baseline detection was not very  
 6103 different between males and females, we can see that they had very different  $\sigma$   
 6104 estimates (note that the BCIs do not overlap). As usual, you can reproduce this  
 6105 analysis by calling the function `bear.JAGS` and set `model='SCRsex'`.

---

```

model{

psi ~ dunif(0,1)                                # Prior distributions
psi.sex ~ dunif(0,1)
for(t in 1:2){
  alpha0[t] ~ dnorm(0,.1)
  logit(p0[t]) <- alpha0[t]
  alphai[t] <- 1/(2*sigma[t]*sigma[t])
  sigma[t] ~ dunif(0, 15)
}

for(i in 1:M){
  z[i] ~ dbern(psi)
  Sex[i] ~ dbern(psi.sex)                      # Sex is binary
  Sex2[i] <- Sex[i] + 1                         # Convert to categorical
  s[i,1] ~ dunif(xlim[1],xlim[2])
  s[i,2] ~ dunif(ylim[1],ylim[2])

  for(j in 1:J){
    d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
    y[i,j] ~ dbin(p[i,j],K)
    p[i,j] <- z[i]*p0[Sex2[i]]*exp(-alphai[Sex2[i]]*d[i,j]*d[i,j])
  }
}
N <- sum(z[])
D <- N/area
}

```

---

Panel 6.2: **JAGS** model specification for an SCR model with sex-specific encounter probability parameters.

**Table 6.6.** Posterior summaries of parameter estimates from sex-specific SCR models for the Fort Drum black bear data set.

Parameter	Mean	SD	2.5	97.5
$D$	0.168	0.022	0.12	0.21
$N$	509.982	66.355	376	631
$p_{0,female}$	0.136	0.025	0.09	0.19
$p_{0,male}$	0.092	0.017	0.06	0.13
$\psi_{sex}$	0.310	0.068	0.19	0.45
$\psi$	0.784	0.103	0.58	0.97
$\sigma_{female}$	1.542	0.132	1.31	1.83
$\sigma_{male}$	2.682	0.389	2.09	3.62

### 6.3 INDIVIDUAL HETEROGENEITY

6106 Here we consider SCR models with individual heterogeneity. Capture-recapture  
 6107 models with individual heterogeneity in detection probability, so-called model  $M_h$ ,  
 6108 have a long history in classical capture recapture models and they have special  
 6109 relevance to SCR (Sec. 4.4). While the advent of SCR models may appear to  
 6110 have rendered the use of classical model  $M_h$  obsolete (because the heterogeneity is  
 6111 being accounted for explicitly) we may still wish to consider heterogeneity models  
 6112 for other biological reasons. It is reasonable to expect in real populations that  
 6113 there exists heterogeneity in home range size and so we think that  $\alpha_1$  could exhibit  
 6114 heterogeneity among individuals. As we noted previously, it may be advantageous  
 6115 or desirable in some cases to model heterogeneity directly in terms of the scale  
 6116 parameter of the distance function  $\sigma$  or some other transformation of the “distance  
 6117 coefficient”, perhaps even 95% home range area.

6118 In this section, we describe a class of spatial capture-recapture models to allow  
 6119 for individual heterogeneity in encounter probability. In particular, one class of  
 6120 models we propose explicitly admits individual heterogeneity in home range *size*.  
 6121 In addition, we consider a standard representation for heterogeneity in which an  
 6122 additive individual-specific random effect is included in the linear predictor for  
 6123 baseline encounter probability.

#### 6124 6.3.1 Models of heterogeneity

6125 An obvious extension to the SCR model is to include an additive individual effect,  
 6126 analogous to classical “model  $M_h$ ”. We’ll call this model “SCR+Mh”:

$$\text{logit}(p_{0,i}) = \alpha_0 + \eta_i \\ p_{ijk} = p_{0,i} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

6127 where  $\eta_i$  is an individual random effect having distribution  $[\eta|\sigma_p]$ . A popular class  
 6128 of models arises by assuming  $\eta_i \sim \text{Normal}(0, \sigma_p^2)$  (Coull and Agresti, 1999; Dorazio  
 6129 and Royle, 2003). We show how to implement this specific SCR + Mh model in

6130 Panel 7.3, although many other random effects distributions are possible. A popular  
 6131 one is the finite-mixture of point masses (Norris and Pollock, 1996; Pledger, 2004)  
 6132 which we demonstrate how to fit using `secr` in Sec. 7.4.3.

---

```
model{

  alpha0 ~ dnorm(0,.1)                                # Prior distributions
  alpha1 <- 1/(2*sigma*sigma)
  sigma ~ dunif(0, 15)
  psi ~ dunif(0,1)
  tau_p ~ dgamma(.001,.001)

  for(i in 1:M){
    eta[i] ~ dnorm(0, tau_p)                         # Individual level variables
    z[i] ~ dbern(psi)
    s[i,1] ~ dunif(xlim[1],xlim[2])
    s[i,2] ~ dunif(ylim[1],ylim[2])

    for(j in 1:J){                                    # The "likelihood" etc..
      d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
      y[i,j] ~ dbin(p[i,j],K)
      logit(p0[i,j]) <- alpha0 + eta[i]
      p[i,j] <- z[i]*p0[i,j]*exp(-alpha1*d[i,j]*d[i,j])
    }
  }
  N <- sum(z[])                                     # N, D are derived
  D <- N/area
}
```

---

Panel 6.3: **JAGS** model specification for the SCR + Mh model with Gaussian encounter probability model and additive normal random effect.

### 6133 6.3.2 Heterogeneity induced by variation in home range size

6134 An alternative heterogeneity model, one that has more of a direct biological moti-  
 6135 vation and interpretation, describes heterogeneity in home range size among indi-  
 6136 viduals. To model heterogeneity in home range area, we can assume a distribution  
 6137 for a transformation of the scale parameter of the encounter probability model such  
 6138 as  $\sigma^2$ , or  $\log(\sigma^2)$ , etc.. We call this “model SCR + Ah” (Ah here for area-induced  
 6139 heterogeneity).

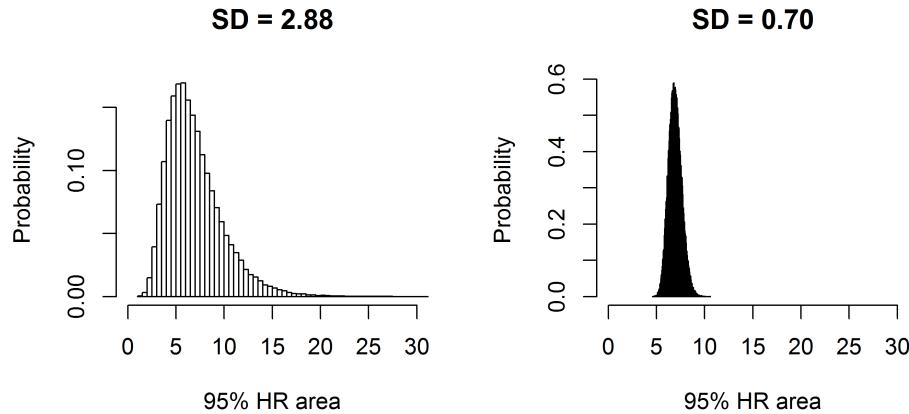
6140 Consider the following log-normal model for individual scale parameter of the  
 6141 Gaussian encounter probability model,  $\sigma_i^2$ :

$$\log(\sigma_i^2) \sim \text{Normal}(\mu_{hra}, \tau_{hra}^2)$$

6142 then the 95% home range area has a scaled log-normal distribution with mean

$$6\pi \exp(\mu_{hra} + \tau_{hra}^2/2).$$

6143 The variance is slightly more complicated, but you can look-up the variance of a  
 6144 log-normal distribution and combine it with the 95% home range area calculation  
 6145 in Sec. 5.4 to work out the implied variance of home range area under this model.  
 6146 We show two examples of the implied *population* distribution of home range area  
 6147 under this log-normal model that implies a mean home range area of about 6.9 area  
 6148 units (Figure 7.1). The left panel shows a standard deviation in home range area  
 6149 of 2.88 units and the right panel shows a standard deviation in home range area of  
 6150 0.70 units. The two cases were generated by tweaking the  $\mu_{hra}$  and  $\tau_{hra}^2$  parameters  
 6151 of the log-normal distribution to achieve a constant expected value of home range  
 6152 area, but modify the standard deviation.



**Figure 6.1.** Population distribution of home range area for a model in which  $\log(\sigma^2)$  has a normal distribution with mean  $\mu_{hra}$  and variance  $\tau_{hra}^2$ . The parameters were chosen to yield a constant expected value of about 6.9 units of area, but to produce two different levels of heterogeneity: A population standard deviation of 2.88 units (left panel) and 0.70 units (right panel).

## 6.4 LIKELIHOOD ANALYSIS IN SECR

6153 Previously, in Chapt. 6, we introduced the **R** package **secr** and described the  
 6154 likelihood based inference approach taken by that package (see Sec. 6.5.3). Here  
 6155 we discuss how to implement some standard covariate models in **secr** and provide  
 6156 an example of model selection using AIC. As we saw in Chapt. 6, **secr** uses the  
 6157 standard **R** model specification syntax, defining the dependent and independent  
 6158 variable relationship using tildes (e.g.,  $y \sim x$ ). Thus, in **secr** we might have  
 6159  $g_0 \sim \text{behavior}$  or  $\sigma \sim \text{time}$ ; when left unspecified or set to 1 (e.g.,  $g_0 \sim 1$ ),  
 6160 this will default to a model with no covariates (i.e., constant parameter values).  
 6161 A number of default model formulas for the baseline and scale parameter of the  
 6162 encounter probability model are available in **secr**. Additionally, **secr** allows us  
 6163 to specify covariates on density (we cover this in Chapt. 11), which are set for  
 6164 example as  $D \sim \text{habitat}$ .

6165 To demonstrate models with various types of covariates using **secr**, we continue  
 6166 using the Fort Drum black bear data. We include in the **scrbook** package a function  
 6167 called **secr.bear** that will format the data (see Chapt. 6 for the **secredata** format)  
 6168 and then fit and compare 8 models (details shown in Panel 7.4). We have described  
 6169 all of these models in the previous sections, so we only briefly comment here on how  
 6170 to fit certain models in **secr** and compare them using AIC, and give a few helpful  
 6171 notes.

### 6172 6.4.1 Notes for fitting standard models

6173 In the **secr** package, the encounter probability model is called the “detection func-  
 6174 tion” and it is specified by changing the “**detectfn**” option (an integer code) within  
 6175 the **secr.fit** command. Table 7.1 shows the possible encounter probability models  
 6176 that **secr** allows; the default is that based on the kernel of a bivariate normal proba-  
 6177 bility distribution function (hence we call this the Gaussian model, but it is referred  
 6178 to as “half-normal” in **secr**) and the (negative) exponential is **detectfn** = 2. See  
 6179 model 2 in Panel 7.4 for how to fit the exponential model to the Fort Drum bear  
 6180 data set.

6181 The **secr** package easily fits a range of SCR equivalents of standard capture-  
 6182 recapture models. The package has pre-defined versions of the classic model  $M_t$   
 6183 where each occasion has its own encounter probability, as well as a linear trend  
 6184 in baseline encounter probability over occasions (in a spatial modeling framework  
 6185  $\sigma$  could also be an occasion specific parameter, but having encounter probability  
 6186 change with time seems like the more common case). For the classical time-effects  
 6187 type of model with  $K$  distinct parameters **secr** uses ‘t’ to denote this in the model  
 6188 specification formula (see model 3 in panel 7.4); whereas, for a linear trend over  
 6189 occasions **secr** uses ‘T’.

6190 The global trap response model (what we called model  $M_B$ ), or a local trap-  
 6191 specific behavioral response (model  $M_b$ ) can be fitted in **secr** using formulae with

6192 “b” for the global response model and “bk” for the local trap response model (see  
 6193 models 4 and 5 in Panel 7.4; note that to fit the trap specific behavioral response  
 6194 model you need version 2.3.1 or newer of **secr**).  


---

1. null model with a bivariate normal encounter probability model  
`bear_0=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ 1, sigma ~ 1))`
2. null model with an exponential encounter probability model  
`bear_0exp=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ 1, sigma ~ 1), detectfn=2)`
3. model with fixed time effects  
`bear_t=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ t, sigma ~ 1))`
4. global behavioral model  
`bear_B=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ b, sigma ~ 1))`
5. trap specific behavioral response  
`bear_b=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ bk, sigma ~ 1))`
6. global behavior model with fixed time effects  
`bear_bt=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ b+t, sigma ~ 1))`
7. sex-specific model  
`bear_sex=secr.fit(bear.cap, model=list(D ~ session, g0 ~ session, sigma ~ session))`
8. heterogeneity model  
`bear_h2=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ h2, sigma ~ h2))`

---

Panel 6.4: Models called from **secr.bear** function. All models use **buffer = 20000**

#### 6195 6.4.2 Sex effects

6196 Incorporating sex effects into models with **secr** can be done a few different ways,  
 6197 but there are not pre-defined models for this. A limitation of fitting models with sex  
 6198 effects in **secr** is that it does not accommodate missing values of the sex variable.  
 6199 Thus, in all cases, individuals that are of unknown sex must be removed from the  
 6200 data set (recall that in a Bayesian framework we can keep these individuals in the  
 6201 data set by specifying a distribution for the individual covariate “sex”). In **secr**,  
 6202 the easiest way to include sex effects is to code sex as a “session” variable using the

6203 multi-session models (see Sec. 6.5.4 for a description of the multi-session models),  
 6204 providing two sessions, one representing males and one for females (see model 7 in  
 6205 Panel 7.4). This method provides two separate density estimates, which can then  
 6206 be combined into a total density.

6207 **6.4.3 Individual heterogeneity**

6208 To incorporate heterogeneity, **secr** fits a set of finite mixture models (Norris and  
 6209 Pollock, 1996; Pledger, 2004). These are expensive in terms of parameters but  
 6210 they have been widely adopted because they are easy to analyze using likelihood  
 6211 methods, as the marginal distribution of the data is just a sum of a small number  
 6212 of components. Using **secr**, individual heterogeneity can be incorporated into the  
 6213 encounter probability model using default models for either a 2- or 3-component  
 6214 finite mixture model using the “**h2**” or “**h3**” model terms. The 2-part mixture is  
 6215 shown in model 8 of panel 7.4 and the 3-part mixture can easily be fit by substituting  
 6216 **h3** for **h2**. The finite-mixture model can be fit in **JAGS** or **BUGS**, but we only  
 6217 showed the SCR + Mh logit-normal mixture in the version above (see Sec. 7.3.1).

6218 **6.4.4 Model selection in secr using AIC**

6219 One practical advantage to using the **secr** package, or likelihood inference in  
 6220 general, is the convenience of automatic model selection using AIC (Burnham and  
 6221 Anderson, 2002). The **secr** package has a number of convenient functions for  
 6222 computing AIC and producing model selection tables, or doing model-averaging  
 6223 (as described in Chapt. 8). Running the function **secr.bear**, which calls all of  
 6224 the models we have described, will return, in addition to all model results, an AIC  
 6225 table with all of the summarized results including the AIC values, delta AIC, and  
 6226 model weights (see Table 7.7 or reproduce results in R using **out<- secr.bear();**  
**out\$AIC.tab**).

6227 It is important to note that AIC is not comparable between a multi-session  
 6228 model and a model that is not a multi-session model. Therefore, to compare the  
 6229 sex-specific model (which uses “sessions”) with all the other models including the  
 6230 null, time, and behavioral models, we coded the data set as a multi-session design  
 6231 when first loading it to **secr**. This results in all the model outputs listing sepa-  
 6232 rate parameter estimates for each session, even the null model with no covariates;  
 6233 however, the estimates are the same for both “sessions” in all but the sex-specific  
 6235 model.

6236 The results from this AIC analysis are straightforward to interpret; the model  
 6237 with a local trap response of encounter probability, “**bk**”, has a model weight of 1  
 6238 and thus, according to AIC, 100% support. The 2-part finite mixture model for  $g_0$   
 6239 and  $\sigma$  has the second lowest AIC, but considering the large dAICc compared to the  
 6240 local trap response model we would probably not consider it any further.

**Table 6.7.** 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

## 6.5 SUMMARY AND OUTLOOK

6241 There are endless covariates and encounter probability models that can be defined  
 6242 and our goal in this chapter was to introduce basic types of covariate models and  
 6243 demonstrate how to implement them in **BUGS** and `secr`. Essentially, SCR's are  
 6244 GLMMs and therefore we develop covariate models in much the same way, using  
 6245 a suitable transformation (link function) of the parameter(s). In SCR models, we  
 6246 typically have 2 parameters of the encounter probability model for which we might  
 6247 specify covariate models – the baseline encounter probability (or rate) parameter,  
 6248 and a scale parameter that is related in many cases to the home range size of the  
 6249 species. A few examples of different covariate models are given in Table 7.3. We  
 6250 can also consider covariates by their classification as fixed, partially observed, or  
 6251 unobserved (see Table 7.8). This classification of covariate types can be important  
 6252 because the MLE and Bayesian approaches to dealing with partially and unobserved  
 6253 covariates is often different. This was seen above in how the covariate `Sex` was  
 6254 handled in the two frameworks.

**Table 6.8.** Examples of different covariate classifications.

Covariate class	Examples
Fixed	baited, weather, habitat
Partially observed	sex, age,
Unobserved	home range size, ind. effects

6255 While the move to spatially explicit models in capture-recapture studies has  
 6256 largely rendered the basic CR models (Otis et al., 1978) obsolete, we continue to  
 6257 find this classification useful for categorizing the *spatial* extensions of these standard  
 6258 CR models. The extended models include the standard  $M_0$ ,  $M_t$ ,  $M_b$ , and  $M_h$ , but  
 6259 also new models that allow for trap-specific information such as "baited/not-baited"  
 6260 or "on/off road". In addition, in Chaps. 12, 13 and 11, we explore additional mod-  
 6261 els for explaining variation in encounter probability and density based on spatial  
 6262 covariates that describe variation in landscape or habitat conditions.



6263  
6264

# 7

6265

## MODEL SELECTION AND ASSESSMENT

6266 Our purpose in life is to analyze models. By that, we mean one or more of the  
6267 following basic 4 tasks: (1) estimate parameters, (2) make predictions of unobserved  
6268 random variables, (3) evaluate the relative merits of different models or choosing a  
6269 best model (model selection), and (4) checking whether a specific model appears to  
6270 provide a reasonable description of the data or not (model checking, assessment, or  
6271 “goodness-of-fit”). In previous chapters we addressed the problems of estimation  
6272 of model parameters, and also making predictions of latent variables,  $s$  or  $z$ , or  
6273 functions of these variables such as density or population size. In this chapter, we  
6274 focus on the last two of these basic inference tasks: model selection (which model  
6275 or models should be favored), and model assessment (do the data appear to be  
6276 consistent with a particular model).

6277 In this chapter we review basic strategies of model selection using both likelihood  
6278 methods (as implemented in the `secr` package) and Bayesian analysis. Specifically,  
6279 we review a number of standard methods of model selection that apply to “variable  
6280 selection” problems, when our set of models consists of distinct covariate effects  
6281 and they represent constraints of some larger model. For classical analysis based on  
6282 likelihood, model selection by Akaike Information Criterion (AIC) is the standard  
6283 approach (Burnham and Anderson, 2002). For Bayesian analysis we rely on a  
6284 number of different methods. We demonstrate the use of the deviance information  
6285 criterion (DIC) (Spiegelhalter et al., 2002) for variable selection problems although  
6286 it has deficiencies when applied to hierarchical models in some cases (Millar, 2009).  
6287 We use the Kuo and Mallick indicator variable selection approach (Kuo and Mallick,  
6288 1998) which produces direct statements of posterior model probabilities which we  
6289 think are the most useful, and leads directly to model-averaged estimates of density.  
6290 There is a good review paper recently by O’Hara and Sillanpää (2009) that discusses  
6291 these and many other related ideas for variable selection. In addition to O’Hara

6292 and Sillanpää (2009) we also recommend Link and Barker (2010, Chapt. 7) for  
 6293 general information on model selection and assessment.

6294 To check model adequacy in a Bayesian framework, or whether a specific model  
 6295 provides a satisfactory description of our data set, we rely exclusively on the  
 6296 Bayesian p-value framework (Gelman et al., 1996). For assessing fit of SCR mod-  
 6297 els, part of the challenge is coming up with good measures of model fit, and there  
 6298 does not appear much definitive guidance in the literature on this point. Following  
 6299 Royle et al. (2011a), we break the problem up into 2 components which we attack  
 6300 separately: (1) Conditional on the underlying point process, does the encounter  
 6301 model fit? (2) Do the uniformity and independence assumptions appear adequate  
 6302 for the point process model of activity centers? The latter component of model fit  
 6303 has a considerable precedence in the ecological literature as it is analogous to the  
 6304 classical problem of testing “complete spatial randomness” (Cressie, 1991; Illian  
 6305 et al., 2008).

6306 We apply some of these methods to the wolverine camera trapping data first  
 6307 introduced in Chapt. 5 to investigate sex specificity of model parameters and  
 6308 whether there is a behavioral response to encounter. We note that individuals are  
 6309 drawn to the camera trap devices by bait and therefore it stands to reason that  
 6310 once an individual discovers a trap, it might be more likely to return subsequently,  
 6311 a response termed “trap happiness”. We evaluate whether certain models for en-  
 6312 counter probability appear to be adequate descriptions of the data, and we evaluate  
 6313 the uniformity assumption for the underlying point process.

## 7.1 MODEL SELECTION BY AIC

6314 Using classical analysis based on likelihood, model selection is easily accomplished  
 6315 using AIC (Burnham and Anderson, 2002) which we demonstrate below. The  
 6316 AIC of a model is simply twice the negative log-likelihood evaluated at the MLE,  
 6317 penalized by the number of parameters ( $np$ ) in the model:

$$\text{AIC} = -2\log L(\hat{\theta}|\mathbf{y}) + 2np$$

6318 Models with small values of AIC are preferred. It is common to use a modified  
 6319 (“corrected”) AIC referred to as  $AIC_c$  for small sample sizes which is

$$AIC_c = -2\log L(\hat{\theta}|\mathbf{y}) + \frac{2np(np+1)}{n-np-1}$$

6320 where  $n$  is the sample size. Two important problems with the use of AIC and  
 6321  $AIC_c$  are that they don’t apply directly to hierarchical models that contain random  
 6322 effects, unless they are computed directly from the marginal likelihood (for SCR  
 6323 models we can do this, see Chapt. 6). Moreover, it is not clear what should be the  
 6324 effective sample size  $n$  in calculation of  $AIC_c$ , as there can be covariates that affect

6325 individuals, that vary over time, or space. We do not offer strict guidelines as to  
 6326 when to use a small sample size adjustment.

6327 The **R** package **secr** computes and outputs AIC automatically for each model  
 6328 fitted and it provides some capabilities for producing a model selection table (func-  
 6329 tion **AIC**) and also doing model-averaging (function **model.average**), which we  
 6330 recommend for obtaining estimates of density from multiple models.

### 6331 7.1.1 AIC analysis of the wolverine data

6332 We provide an example of model selection for the wolverine camera trapping data  
 6333 using **secr**. We consider a model set with distinct models to accommodate various  
 6334 types of sex specificity of model parameters:

- 6335    Model 0: model SCR0 with constant density and constant encounter model pa-  
   6336    rameters;
- 6337    Model 1: model SCR0 with constant parameter values for both male and female  
   6338    wolverines but with sex-specific density only;
- 6339    Model 2: Sex-specific density, sex-specific  $p_0$  but constant  $\sigma$ ;
- 6340    Model 3: Sex-specific density, sex-specific  $\sigma$  but constant  $p_0$ ;
- 6341    Model 4: Sex-specific density, sex-specific  $p_0$  and sex-specific  $\sigma$ .

6342 To model sex-specific abundance (density), we use the multi-session models  
 6343 provided by **secr** (introduced in Sec. 6.5.4), which allow one to model session-  
 6344 specific effects on density, baseline encounter probability,  $p_0$  (labeled  $g_0$  in **secr**),  
 6345 and also the scale parameter  $\sigma$  of the encounter probability model. Using this  
 6346 formulation, we define the “Session” variable to be a *categorical* sex code having  
 6347 value 1 or 2 (demonstrated below) and thus *session*-specific parameters represent  
 6348 sex-specific parameters. For example, if we model session-specific density,  $D$ , then  
 6349 this corresponds to Model 1 in our list above. We note that “Model 0” in our list  
 6350 corresponds to a model where all of the encounter histories have the same session  
 6351 ID. This model is one of constant density, which implies that the population sex  
 6352 ratio is fixed at 0.5, i.e.,  $\psi_{\text{sex}} = 0.5$ .

6353 Although **secr** also uses the logit/log linear predictors as the default for mod-  
 6354 eling covariates on baseline encounter probability and the scale parameter, respec-  
 6355 tively, **secr** does something different with the multi-session models. It reports  
 6356 estimates in a *session mean* parameterization (equivalent to, in **BUGS**, using an  
 6357 index variable instead of a set of dummy variables), and not the *session effect* (i.e.,  
 6358 deviation from the intercept) which arises from the use of dummy variables. We  
 6359 show this **BUGS** model description in Sec. 8.2.2.

6360 To fit these models using **secr**, we load the wolverine data and do a slight bit  
 6361 of formatting to prepare the data objects for analysis by **secr**. The key differ-  
 6362 ence from our analysis in Chapt. 6 is, here, we use the wolverine sex information  
 6363 (**wolverine\$wsex**) which is a binary 0/1 variable (1=male) and we add 1 so that  
 6364 we can define a categorical “Session” variable (having values 1 or 2). We also have

6365 a function `scr2secr` which converts a standard trap-deployment file (TDF) matrix  
 6366 into a `secr` object of class “traps.” The **R** commands are as follows (contained in  
 6367 the help file `?secr_wolverine`):

```

6368
6369 > library(secr)
6370 > library(scrbook)
6371 > data(wolverine)
6372 > traps <- as.matrix(wolverine$wtraps)
6373
6374 ## Name variables as required by secr
6375 > dimnames(traps) <- list(NULL,c("trapID","x","y",paste("day",1:165,sep="")))
6376 ## Convert trap information to a secr "traps" object
6377 > trapfile <- scr2secr(scrtraps=traps,type="proximity")
6378
6379 ## Grab the wolverine state-space grid (2km here)
6380 > gr <- as.matrix(wolverine$grid2)
6381 > dimnames(gr) <- list(NULL,c("x","y"))
6382 > gr2 <- read.mask(data=gr)
6383
6384 ## Grab the encounter data, and re-name variables
6385 > wolv.dat <- wolverine$wcaps
6386 > dimnames(wolv.dat) <- list(NULL,c("Session","ID","Occasion","trapID"))
6387
6388 ## Convert binary 0/1 sex variable to categorical 1/2 for "session"
6389 > wolv.dat[,1] <- wolverine$wsex[wolv.dat[,2]]+1
6390 > wolv.dat <- as.data.frame(wolv.dat)
6391
6392 ## Convert to capthist object
6393 > wolvcapt <- make.capthist(wolv.dat,trapfile,fmt="trapID",noccasions=165)
```

6394 Once the data have been prepared in this way, we use the `secr` model fitting  
 6395 function `secr.fit` to fit the different models, and then the function `AIC` to pack-  
 6396 age the models together and summarize them in the form of an AIC table, with  
 6397 rows of the table ordered from best to worst. The function `model.average` per-  
 6398 forms AIC-based model-averaging of the parameters specified by the `realnames`  
 6399 variable (below this is demonstrated for the parameter density,  $D$ ). Because this  
 6400 function defaults to averaging by  $AIC_c$ , we slightly modified this function (called  
 6401 `model.average2`) to do model averaging by either  $AIC$  or  $AIC_c$  as specified by the  
 6402 user. The model fitting commands look like this (for Model 0 and Model 1):

```

6403 > model0 <- secr.fit(wolvcapt, model=list(D~1, g0~1, sigma~1),
6404           buffer=20000)
6405 > model1 <- secr.fit(wolvcapt, model=list(D~session, g0~1, sigma~1),
6406           buffer=20000)
```

6407 Next we use the function `AIC`, passing the fit objects from all 5 models, and  
 6408 that produces the following output (abbreviated horizontally to fit on the page):

---

```

6409 > AIC (model0,model1,model2,model3,model4)
6410      model      ... npar logLik   AIC    AICc dAICc  AICwt
6411 model0  D~1 g0~1 sigma~1 ... 3 -627.2603 1260.521 1261.932 0.000 0.5831
6412 model2      ..      ... 5 -624.9051 1259.810 1263.810 1.878 0.2280
6413 model1      ..      ... 4 -627.2365 1262.473 1264.973 3.041 0.1275
6414 model4      ..      ... 6 -624.6632 1261.326 1267.326 5.394 0.0393
6415 model3      ..      ... 5 -627.2358 1264.472 1268.472 6.540 0.0222

```

6416 Model averaging the results is done as follows:

```

6417 > model.average (model0,model1,model2,model3,model4,realnames="D")
6418      estimate SE.estimate      lcl      ucl
6419 session=1 2.707190e-05 7.913577e-06 1.544474e-05 4.745224e-05
6420 session=2 2.927423e-05 8.270402e-06 1.700631e-05 5.039193e-05

```

6421 As usual, estimates and standard errors of the individual model parameters can  
 6422 be obtained from the **secr.fit** summary output of any of the **modelX** objects shown  
 6423 above. The default output of estimated density is in individuals per ha, so we have  
 6424 to scale this up to something more reasonable. To get into units of per 1000 km<sup>2</sup>,  
 6425 we need to first multiply by 100 to get to units of km<sup>2</sup> and then multiply by 1000.  
 6426 This produces an estimated density of about 2.71 for **session=1** (females) and 2.93  
 6427 for **session=2** (males). We can use the generic **R** function **predict** applied to the  
 6428 **secr.fit** output to obtain specific information about the MLEs on the natural  
 6429 scale.

6430 We don't necessarily agree with the use of AIC<sub>c</sub> here and think its better to  
 6431 use AIC, in general. This is because, as noted previously, it is not clear what the  
 6432 effective sample size is for most capture-recapture problems. While we have 21  
 6433 individuals in the data set, most of the model structure has to do with encounter  
 6434 probability samples and for that there are hundreds of observations. We do note  
 6435 that the AIC and AIC<sub>c</sub> results are not entirely consistent. By looking at the best  
 6436 model by AIC (Table 8.1), we find that the model with sex specific density and  
 6437 sex-specific baseline encounter probability,  $p_0$ , is preferred (Model 2). This is just  
 6438 slightly better than the null model (Model 0) with no sex effects at all and hence  
 6439 an implied fixed sex ratio of  $\psi_{sex} = 0.50$ .

6440 We fit the same models but now using a modified state-space which excludes  
 6441 the ocean (this is a habitat mask in **secr**). Results are shown in Table 8.1 along  
 6442 with the previous models without a mask. We see AIC values are smaller for the  
 6443 model without the mask. It is probably acceptable to compare these different fits  
 6444 (with and without habitat mask) by AIC because we recognize the mask as having  
 6445 the effect of modifying the random effects distribution (i.e., of the activity centers,  
 6446 **s**) and the results should be sensitive to choice of the distribution for **s**. That said,  
 6447 we tend to prefer the mask model because it makes sense to exclude the areas of  
 6448 open water from the state-space of **s**. For females the model-averaged density estimate  
 6449 is 3.88 individuals per 1000 km<sup>2</sup> and for males the model-averaged density estimate  
 6450 is 4.46 individuals per 1000 km<sup>2</sup> as we see here:

**Table 7.1.** Model selection results for the wolverine models of sex specificity, with/without habitat mask. Fitting was done using `secr` with a half-normal (Gaussian) encounter probability model. Models are ordered by *AIC*. Density, *D*, is reported in units of individuals per 1000 km<sup>2</sup>. Model abbreviations indicate which parameters are sex-specific in order  $D/p_0/\sigma$ .

NO HABITAT MASK									
model	npar	AIC	AICc	D	Female		Male		
					$p_0$	$\sigma$	D	$p_0$	$\sigma$
2: sex/sex/1	5	1259.8	1263.8	2.45	0.08	6435.51	3.16	0.04	6435.51
0: 1/1/1	3	1260.5	1261.9	2.83	0.06	6298.66	2.83	0.06	6298.66
4: sex/sex/sex	6	1261.3	1267.3	2.59	0.08	6080.70	2.99	0.04	6833.16
1: sex/1/1	4	1262.5	1265.0	2.69	0.06	6298.69	2.96	0.06	6298.69
3: sex/1/sex	5	1264.5	1268.5	2.70	0.06	6280.49	2.95	0.06	6319.03

WITH HABITAT MASK									
model	npar	AIC	AICc	D	Female		Male		
					$p_0$	$\sigma$	D	$p_0$	$\sigma$
2: sex/sex/1	5	1268.1	1272.1	3.64	0.07	6382.88	4.73	0.03	6382.88
4: sex/sex/sex	6	1268.7	1274.7	3.87	0.07	5859.40	4.41	0.03	7039.09
0: 1/1/1	3	1271.2	1272.6	4.18	0.05	6282.62	4.18	0.05	6282.62
1: sex/1/1	4	1273.1	1275.6	3.98	0.05	6282.65	4.38	0.05	6282.65
3: sex/1/sex	5	1275.1	1279.1	3.93	0.05	6357.26	4.41	0.05	6220.22

```

6451 > model.average (model0b,model1b,model2b,model3b,model4b,realnames="D")
6452
6453      estimate   SE.estimate        lcl        ucl
6454 session=1 3.876615e-05 1.189102e-05 2.153795e-05 6.977518e-05
6455 session=2 4.459658e-05 1.323696e-05 2.523280e-05 7.882022e-05

```

6456 This is quite a bit higher than that based on the rectangular state-space (i.e., not  
6457 specifying a habitat mask). This is not surprising given that **the state-space is**  
6458 **part of the model** and the specific state-space modification we made here, which  
6459 reduces the area from the rectangular state-space, should be extremely important  
6460 from a biological standpoint (i.e., wolverines are not actively using open ocean).

## 7.2 BAYESIAN MODEL SELECTION

6461 Model selection is somewhat less straightforward as a Bayesian, and there is no  
6462 canned all-purpose method like AIC. As such we recommend a pragmatic approach,  
6463 in general, for all problems, based on a number of basic considerations:

- 6464 (1) For a small number of fixed effects we think it is reasonable to adopt a con-  
6465 ventional “hypothesis testing” approach – i.e., if the posterior for a parameter  
6466 overlaps zero substantially, then it is probably reasonable to discard that effect  
6467 from the model.
- 6468 (2) Calculation of posterior model probabilities: In some cases we can implement  
6469 methods which allow calculation of posterior model probabilities. One such idea

6470 is the indicator variable selection method from Kuo and Mallick (1998). For this,  
 6471 we introduce a latent variable  $w \sim \text{Bern}(.5)$  and expand the model to include the  
 6472 variable  $w$  as follows:

$$\text{logit}(p_{ijk}) = \alpha_0 + w * \alpha_1 * C_{ijk}.$$

6473 The importance of the covariate  $C$  is then measured by the posterior probability  
 6474 that  $w = 1$ .

- 6475 (3) The Deviance Information Criterion (DIC): Bayesian model selection is now  
 6476 routinely carried out using DIC ((Spiegelhalter et al., 2002)), although its effec-  
 6477 tiveness in hierarchical models depends very much on the manner in which it is  
 6478 constructed (Millar, 2009). We recommend using it if it leads to sensible results,  
 6479 but we think it should be calibrated to the extent possible for specific classes of  
 6480 models. This has not yet been done in the literature for SCR models, to our  
 6481 knowledge.
- 6482 (4) Logical argument: For something like sex specificity of certain parameters,  
 6483 it seems to make sense to leave an extra parameter in the model no matter  
 6484 what because, biologically, we might expect a difference (e.g., home range size).  
 6485 In some cases failure to apply logical argument leads to meaningless tests of  
 6486 gratuitous hypotheses (Johnson, 1999).

6487 In all modeling activities, as in life itself, the use of logical argument should not be  
 6488 under-utilized.

### 6489 7.2.1 Model selection by DIC

6490 The availability of AIC makes the use of likelihood methods convenient for problems  
 6491 where likelihood estimation is achievable. For Bayesian analysis, DIC seemed like  
 6492 a general-purpose equivalent, at least for a brief period of time after its invention.  
 6493 However, there seem to be many variations of DIC, and a consistent version is not  
 6494 always reported across computing platforms. Even statisticians don't have general  
 6495 agreement on practical issues related to the use of DIC (Millar, 2009). Despite  
 6496 this, it is still widely reported. We think DIC is probably reasonable for certain  
 6497 classes of models that contain only fixed effects, or for which the latent variable  
 6498 structure is the same across models so that only the fixed effects are varied (this  
 6499 covers many SCR model selection problems). However, it would be useful to see  
 6500 some calibration of DIC for some standardized model selection problems.

6501 Model deviance is defined as negative twice the log-likelihood; i.e., for a given  
 6502 model with parameters  $\theta$ :  $\text{Dev}(\theta) = -2 * \text{logL}(\theta|\mathbf{y})$ . The DIC is defined as the  
 6503 posterior mean of the deviance,  $\bar{\text{Dev}}(\theta)$ , plus a measure of model complexity,  $p_D$ :

$$\text{DIC} = \bar{\text{Dev}}(\theta) + p_D$$

6504 The standard definition of  $p_D$  is

$$p_D = \bar{\text{Dev}}(\theta) - \text{Dev}(\bar{\theta})$$

6505 where the 2nd term is the deviance evaluated at the posterior mean of the model  
 6506 parameter(s),  $\bar{\theta}$ . The  $p_D$  here is interpreted as the effective number of parameters in  
 6507 the model. Gelman et al. (2004) suggest a different version of  $p_D$  based on one-half  
 6508 the posterior variance of the deviance:

$$p_V = \text{Var}(\text{Dev}(\theta)|\mathbf{y})/2.$$

6509 This is what is produced from **WinBUGS** and **JAGS** if they are run from **R2WinBUGS**  
 6510 or **R2jags**, respectively. It is less easy to get DIC summaries from **rjags**, so we  
 6511 used **R2jags** in our analyses below.

### 6512 7.2.2 DIC analysis of the wolverine data

6513 We repeated the analysis of the wolverine models with sex specificity, but this time  
 6514 doing a Bayesian analysis paralleling the likelihood analysis we did above in **secr**,  
 6515 using the logit/log parameterization of the model parameters. To do so in **BUGS**,  
 6516 we used dummy variables. Thus, we can express models allowing for sex specificity  
 6517 using a dummy variable **Sex** and new parameters  $(\alpha_{sex}, \beta_{sex})$  which represent the  
 6518 effect of **Sex** at level 1:

$$\text{logit}(p_{0,i}) = \alpha_0 + \alpha_{sex}\mathbf{Sex}_i$$

6519 and

$$\log(\sigma_i) = \log(\sigma_0) + \beta_{sex}\mathbf{Sex}_i.$$

6520 In these expressions, the sex variable  $\mathbf{Sex}_i$  is a binary variable where  $\mathbf{Sex}_i = 0$   
 6521 corresponds to female, and  $\mathbf{Sex}_i = 1$  corresponds to male.

6522 Unlike the multi-session model in **secr**, we carry out the analysis of the sex-  
 6523 specific model here by putting all of the data into a single data set, and explicitly  
 6524 accounting for the covariate 'sex' in the model by assigning it a Bernoulli prior  
 6525 distribution with  $\psi_{sex}$  being the proportion of males in the population. In this  
 6526 case, we produce "Model 0" above, the model with no sex effect on density, by  
 6527 setting the population proportion of males at one-half:  $\psi_{sex} = 0.5$  (see also Sec.  
 6528 7.2.4). As usual, handling of missing values of the sex variable is done seamlessly  
 6529 which might be a practical advantage of Bayesian analysis in situations where sex  
 6530 is difficult to record in the field which may lead to individuals of unknown sex (i.e.,  
 6531 missing values).

6532 The **BUGS** model specification for the most complex model, Model 4, is shown  
 6533 in Panel 8.1. This model has sex-specific intercept, scale parameter,  $\sigma$ , and density.  
 6534 We provide an **R** script named **wolvSCROms** in the **scrbook** package which will fit  
 6535 each model. The function uses **JAGS** by default for the fitting, using the **R2jags**  
 6536 package. The kernel of this function is the model specification in Panel 8.1, which  
 6537 gets modified depending on the model we wish to fit using a command line option  
 6538 **model**. For example, **model = 1** fits the model with constant parameter values for  
 6539 males and females, but sex-specific population sizes (**model = 0** constrains the male  
 6540 probability parameter,  $\psi_{sex}$ , to be 0.5). The **R** function fits each of the 5 models

---

```

alpha.sex ~ dunif(-3,3)                      ## Prior distributions
beta.sex ~ dunif(-3,3)
sigma0 ~ dunif(0,50)
alpha0 ~ dnorm(0,.1)
psi ~ dunif(0,1)                             ## Data augmentation parameter
psi.sex ~ dunif(0,1)                          ## Probability of 'male'

for(i in 1:M){                                ## DA loop
  wsex[i] ~ dbern(psi.sex)                   ## Latent sex state (male = 1)
  z[i] ~ dbern(psi)                         ## DA variables
  s[i,1] ~ dunif(Xl,Xu)
  s[i,2] ~ dunif(Yl,Yu)
  logit(p0[i]) <- alpha0 + alpha.sex*wsex[i]
  log(sigma.vec[i]) <- log(sigma0) + beta.sex*wsex[i]
  alpha1[i] <- 1/(2*sigma.vec[i]*sigma.vec[i])
  for(j in 1:ntraps){
    mu[i,j] <- z[i]*p[i,j]
    y[i,j] ~ dbin(mu[i,j],K[j])
    dd[i,j] <- pow(s[i,1] - traplocs[j,1],2) + pow(s[i,2] - traplocs[j,2],2)
    p[i,j] <- p0[i]*exp( - alpha1[i]*dd[i,j] )
  }
}

```

---

Panel 7.1: Part of the **BUGS** specification for a complete sex specificity of model parameters. This is a simplified version of the model contained in the `wolvSCR0ms` script, because it does not contain the on/off switches for creating the various sub-models.

6541 using a binary indicator variable to turn ‘on’ or ‘off’ each effect. Here is how we  
 6542 obtain the MCMC output for each of the 5 models:

```

6543 > wolv0 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=0)
6544 > wolv1 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=1)
6545 > wolv2 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=2)
6546 > wolv3 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=3)
6547 > wolv4 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=4)

```

6548 We fitted the 5 models to the wolverine data and summarize the DIC computa-  
 6549 tion results in Table 8.2. The model rank has model 0, model 2, model 1, model 4,  
 6550 model 3. Interestingly, this is the same order as the models based on  $AIC_c$  which  
 6551 we found above (see Table 8.1). The posterior mean and SD of model parameters  
 6552 under the 5 models are given in Table 8.3.

**Table 7.2.** DIC results for the 5 models of sex specificity fitted to the wolverine camera trapping data, using the function `wolvSCR0ms`. Results are based on 3 chains of length 61000 yielding 180000 posterior samples.

	meandev	pd	DIC	rank
Model 0	441.01	77.09	518.10	1
Model 1	441.78	77.504	519.28	3
Model 2	440.12	78.440	518.56	2
Model 3	443.31	79.478	522.79	5
Model 4	441.24	80.078	521.32	4

**Table 7.3.** Posterior summaries of model parameters for models with varying sex specificity of model parameters. Model 0 = no sex specificity, model 4 = fully sex-specific (see text). Models are based on the Gaussian encounter probability model, each with 21000 iterations, 1000 burn-in, 3 chains for a total of 60000 posterior samples.

Parameter	model 0		model 1		model 2		model 3		model 4	
	Mean	SD								
$N$	60.02	11.91	60.24	11.93	59.37	11.97	59.67	11.97	58.77	11.75
$D$	5.79	1.15	5.81	1.15	5.72	1.15	5.75	1.15	5.66	1.13
$\alpha_0$	-2.81	0.18	-2.82	0.17	-2.44	0.25	-2.82	0.18	-2.43	0.25
$\alpha_{sex}$	0.00	1.73	0.00	1.73	-0.75	0.34	0.00	1.73	-0.79	0.36
$\sigma_0$	0.64	0.06	0.64	0.05	0.66	0.06	0.65	0.08	0.63	0.09
$\beta_{sex}$	0.00	1.73	-0.01	1.73	0.01	1.74	-0.01	0.17	0.10	0.18
$\psi$	0.30	0.07	0.30	0.07	0.30	0.07	0.30	0.07	0.30	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	441.01	12.42	441.78	12.45	440.12	12.53	443.31	12.61	441.24	12.66
	pD = 77.1		pD = 77.5		pD = 78.4		pD = 79.5		pD = 80.1	
	DIC = 518.1		DIC = 519.3		DIC = 518.6		DIC = 522.8		DIC = 521.3	

### 6553 7.2.3 Bayesian model averaging with indicator variables

6554 A convenient way to deal with model selection and averaging problems in Bayesian  
 6555 analysis by MCMC is to use the method of model indicator variables (Kuo and  
 6556 Mallick, 1998). Using this approach, we expand the model to include a set of pre-  
 6557 scribed models as specific reductions of a larger model. This has been demonstrated  
 6558 in some specific capture-recapture models in Royle and Dorazio (2008, Sec. 3.4.3),  
 6559 and Royle (2009) and in the context of SCR by Tobler et al. (2012). A useful aspect  
 6560 of this method is that model-averaged parameters are produced by default. We em-  
 6561 phasize the need to be careful of reporting model-averaged parameters that don't  
 6562 have a common interpretation in the different models because they are meaningless  
 6563 (averaging apples and oranges....). For example, if a regression parameter is in a  
 6564 specific model then the posterior is informed by the data and a specific MCMC draw  
 6565 is from the appropriate posterior distribution. On the other hand, if the regression  
 6566 parameter is not in the model then the MCMC draw is obtained directly from the  
 6567 prior distribution, and so we need to think carefully about whether it makes sense  
 6568 to report an average of such a thing (in the vast majority of cases the answer is  
 6569 no). But some parameters like  $N$  or density,  $D$ , do have a consistent interpretation  
 6570 and we support producing model-averaged results of those parameters.

6571 To implement the Kuo and Mallick approach, we expand the model to include

6572 the latent indicator variables, say  $w_m$ , for variable  $m$  in the model, such that

$$w_m = \begin{cases} 1 & \text{linear predictor includes covariate } m \\ 0 & \text{linear predictor does not include covariate } m \end{cases}$$

6573 We assume that the indicator variables  $w_m$  are mutually independent with

$$w_m \sim \text{Bernoulli}(0.5)$$

6574 for each variable  $m = 1, 2, \dots$ , in the model. For example, with 2 variables, the  
6575 expanded model has the linear predictor:

$$\text{logit}(p_{ijk}) = \alpha_0 + \alpha_1 w_1 C_{1,i} + \alpha_2 w_2 C_{2,ijk}$$

6576 where, let's suppose,  $C_{1,i}$  is an individual covariate such as sex, and  $C_{2,ijk}$  is a  
6577 behavioral response covariate which is individual-, trap-, and occasion-specific. We  
6578 can assume a parallel model specification on the parameter  $\sigma$  which is liable to vary  
6579 by individual level covariates such as sex:

$$\log(\sigma_i) = \beta_0 + \beta_1 w_3 C_{1,i}.$$

6580 Using this indicator variable formulation of the model selection problem we  
6581 can characterize unique models by the sequence of  $w$  variables. In this case, each  
6582 unique sequence  $(w_1, w_2, w_3)$  represents a model, and we can tabulate the posterior  
6583 frequencies of each model by post-processing the MCMC histories of  $(w_1, w_2, w_3)$ ,  
6584 as we demonstrate shortly. This method then evaluates all possible combinations  
6585 of covariates or  $2^m$  models.

6586 Conceptually, analysis of this expanded model within the data augmentation  
6587 framework does not pose any additional difficulty. One broader, technical consider-  
6588 ation is that posterior model probabilities are well known to be sensitive to priors  
6589 on parameters (Aitkin, 1991; Link and Barker, 2006). See also Royle and Dorazio  
6590 (2008, Sec. 3.4.3) and Link and Barker (2010, Sec. 7.2.5). What might normally  
6591 be viewed as vague or non-informative priors, are not usually innocuous or unin-  
6592 formative when evaluating posterior model probabilities. The use of AIC seems  
6593 to avoid this problem largely by imposing a specific and perhaps undesirable prior  
6594 that is a function of the sample size (Kadane and Lazar, 2004). One solution is to  
6595 compute posterior model probabilities under a model in which the prior for param-  
6596 eters is fixed at the posterior distribution under the full model (Aitkin, 1991). At  
6597 a minimum, one should evaluate the sensitivity of posterior model probabilities to  
6598 different prior specifications.

#### 6599 **Analysis of the wolverine data**

6600 The R script `wolvSCR0ms` in the package `scrbook` provides the model indicator  
6601 variable implementation for the fully sex-specific SCR model. It is run by setting  
6602 `model=5` in the function call. We note again that it is not very useful to report

most parameter estimates from this model because their marginal posterior is a mixture from the prior (when a value of the indicator variable of 0 is sampled) and draws informed by the data (i.e., from the posterior, when a 1 is drawn for the indicator variable  $w$ ). On the other hand, the parameters  $N$  and density  $D$  should be reported and they represent marginal posteriors over all models in the model set. In effect, model averaging is done as part of the MCMC sampling. The variable ‘mod’ contains the two binary indicator variables ( $w$  above) which pre-multiply the ‘sex’ term in each of the  $p_0$  and  $\sigma$  model components, like this:

$$\text{logit}(p_{0,i}) = \alpha_0 + \text{mod}[1]\alpha_{\text{sex}}\text{sex}_i$$

and

$$\log(\sigma_i) = \log(\sigma_0) + \text{mod}[2]\beta_{\text{sex}}\text{sex}_i$$

The third element of `mod` determines whether the  $\psi_{\text{sex}}$  parameter is estimated or fixed at  $\psi_{\text{sex}} = 0.5$  which is accomplished with the line of **BUGS** code as follows:

`sex.ratio <- psi.sex*mod[3] + .5*(1-mod[3]).`

The MCMC output for ‘mod’ was post-processed to obtain the model-weights using the following **R** commands:

```
6617 > mod <- wolv5$BUGSoutput$sims.list$mod
6618 > mod <- paste(mod[,1],mod[,2],mod[,3],sep="")
6619 >
6620 > table(mod)
6621 mod
6622   000   001   010   011   100   101   110   111
6623 17181  4935  1057   296 25211  8337  2275   708
6624
6625 > round( table(mod)/length(mod) , 3)
6626 mod
6627   000   001   010   011   100   101   110   111
6628 0.286  0.082  0.018  0.005  0.420  0.139  0.038  0.012
```

This results in a comparison of all 8 possible models (based on  $m = 3$  covariates) instead of just the 5 models we originally proposed. We see that the best model is that labeled 100 which, according to our construction above, has `mod[1]=1`, `mod[2]=0` and `mod[3]=0`. This is the model having sex-specific baseline encounter probability  $p_0$ , and  $\psi_{\text{sex}} = 0.5$ . This model has posterior model probability 0.420. The model with no sex specificity at all (the model with label 000) has posterior probability 0.286 and the remaining posterior mass is distributed over the other six models. We could arrive at a qualitatively similar conclusion using a more ad hoc approach based on looking at the posterior mass for each parameter under the full model (model 4; see Table 8.3, in part). Considering the sex-specific intercept, it appears to be very important as its posterior mass is mostly away from 0. On the other hand, the coefficient on log-sigma is concentrated around 0, and the estimated  $\psi_{\text{sex}}$  (probability that an individual is a male) is 0.54 with a large posterior standard

deviation. We might therefore be inclined to discard the sex effect on  $\log(\sigma)$  based on classical thinking-like-a-hypothesis-testing-person and settle for the model with a sex-specific intercept in the encounter probability model. This is consistent with our indicator variable approach which found that model (1,0,0) has posterior probability of 0.420. Looking at the posteriors for each parameter to thin the model down is consistent with these results. We can obtain model-averaged estimates from the indicator variable approach, which produces direct model-averaged estimates of  $N$  and  $D$ :

```
6642   mu.vect sd.vect    2.5%    25%    50%    75% 97.5% Rhat n.eff
6643 D      5.695  1.133  3.759  4.916  5.591  6.362 8.193 1.002 3600
6644 N     59.077 11.758 39.000 51.000 58.000 66.000 85.000 1.002 3600
```

We obtain a model-averaged estimate (posterior mean) for density of  $D = 5.695$  which is hardly any different from our model specific estimates (Table 8.3) and, in particular, from model 2 which has only a sex-specific intercept.

#### 6656 7.2.4 Choosing among detection functions

Another approach to implementing model indicator variables is to introduce a categorical “model identity” variable which is itself a parameter of the model. Using this approach, then each distinct model is associated with a unique set of covariates or other set of model features. This is convenient especially when we cannot specify the linear predictor as some general model that reduces to various alternative sub-models simply by switching binary variables on or off. In the context of SCR models, choosing among different encounter probability models would be an example. For this case we do something like this `mod ~ dcat(probs[])` where `probs` is a vector with elements  $1/(\#models)$ , and the encounter probability matrix is filled in depending on the value of `mod`. In particular, instead of a 2-dimensional array `p[i,j]`, we build `p[i,j,m]` for each of  $m = 1, 2, \dots, M$  models. An example with 3 distinct models is:

```
6669 mod ~ dcat(probs[])
6670 ##
6671 ## Using a double loop construction fill-in p[,] for each model:
6672 ##
6673 p[i,j,1] <- p0[1]*exp( - alpha1[1]*dist2[i,j] )
6674 p[i,j,2] <- 1-exp(-p0[2]*exp( - alpha1[2]*dist2[i,j] ) )
6675 logit(p[i,j,3]) <- p0[3] - alpha1[3]*dist2[i,j]
6676
6677 mu[i,j] <- z[i]*p[i,j,mod]
6678 y[i,j] ~ dbin(mu[i,j],K[j])
```

As before the posterior probabilities can be highly sensitive to priors on the different model parameters and sometimes mixing is really poor and, in general, we’ve experienced mixed success trying to carry out model selection using this construction. We do provide a template **R/JAGS** script (`wolvSCR0ms2`) in the

6683 **scrbook** package which has an example of choosing among 3 different encounter  
 6684 probability models: The Gaussian encounter probability, Gaussian hazard, and  
 6685 logistic model with the square of distance (defined in Sec. 7.1). The key things  
 6686 to note are that there are 3 intercepts and 3 different ‘*alpha1*’ parameters (the  
 6687 coefficient on distance). The parameters should not be regarded as equivalent across  
 6688 the models, so it is important to have them separately defined (and estimated) for  
 6689 each model. In our analysis we used a vague normal prior (precision = 0.1) for  
 6690 the intercept parameter (either log or logit-scale of baseline encounter probability  
 6691  $p_0$ ) and a *Uniform*(0,5) prior for one-half the inverse of the coefficient on distance-  
 6692 squared. In the **BUGS** model specification the priors look like this:

```
6693 for(i in 1:3){  

  6694   alpha0[i] ~ dnorm(0,.1)  

  6695   sigma[i] ~ dunif(0,5)  

  6696   alpha1[i] <- 1/(2*sigma[i]*sigma[i])  

  6697 }
```

6698 Then, we create a probability of encounter for each individual, trap *and* model  
 6699 so that the holder object “*p*” in the model description is a 3-dimensional array  
 6700 (sometimes this would have to be a 4 or 5-d array in more complex models with  
 6701 time effects, etc..), so that construction of the encounter probability models look  
 6702 like this:

```
6703 p[i,j,1] <- p0[1]*exp( - alpha1[1]*dist2[i,j] )  

  6704 p[i,j,2] <- 1-exp(-p0[2]*exp( - alpha1[2]*dist2[i,j] ) )  

  6705 logit(p[i,j,3]) <- p0[3] - alpha1[3]*dist2[i,j]
```

6706 where

```
6707 logit(p0[1]) <- alpha0[1]  

  6708 log(p0[2]) <- alpha0[2]  

  6709 p0[3] <- alpha0[3]
```

6710 You can experiment with the **wolvSCR0ms2** script to investigate the importance of  
 6711 different models of encounter probability and whether they have an affect on the  
 6712 inferences.

### 7.3 EVALUATING GOODNESS-OF-FIT

6713 In practical settings, we estimate parameters of a desirable model, or maybe fit a  
 6714 bunch of models and report estimates from all of them or a model-averaged sum-  
 6715 mary of density. An important question is: Is our model worth anything? In other  
 6716 words, does the model appear to be an adequate description of our data? For-  
 6717 mal assessment of model adequacy or goodness-of-fit is a challenging problem and  
 6718 there are no all-purpose algorithms for doing this in either frequentist or Bayesian

6719 paradigms. Moreover, there are some philosophical challenges to evaluating model  
6720 fit, such as, if we do model averaging then should all of the models have to fit? Or  
6721 should the averaged model have to fit? What if none of the models fit? We don't  
6722 know the answers to these questions and we won't try to answer them. Instead,  
6723 we will provide what guidance we can on taking the first steps to evaluating fit, of  
6724 a single model, as if it were a cherished family heirloom of great importance. We  
6725 suggest that if you have a model that you really like, a single model, then it is a  
6726 sensible thing to check that the model is a good fit to your data. If it is not, we  
6727 do not imagine that the model is useless but just that some thought should be put  
6728 into why the model doesn't fit so that, perhaps, some remediation might happen  
6729 as future data are collected. After all, you may have spent 2, 3 or many more years  
6730 of your life collecting that data set, perhaps thousands of hours, and therefore it  
6731 seems a reasonable proposition to expect to do some estimation and analysis of the  
6732 model regardless of model fit. You can still learn something from a model that does  
6733 not pass some technical litmus test of model fit.

6734 Conceptually, we can think of evaluation of model fit as follows: if we simulate  
6735 data under the model in question, do the simulated realizations resemble the data  
6736 set that we actually have? For either Bayesian or classical inference, the basic  
6737 strategy to assessing model fit is to come up with a fit statistic that depends on the  
6738 parameters and the data set, which we denote by  $T(\mathbf{y}, \theta)$ , and then we compute  
6739 this for the observed data set, and compare its value to that computed for perfect  
6740 data sets simulated under the correct model. In the case of classical inference, we  
6741 will often rely on the standard practice of parametric bootstrapping (Dixon, 2002),  
6742 where we simulate data sets conditional on the MLE  $\hat{\theta}$  and compare realizations  
6743 with what we've observed. The **R** package **unmarked** (Fiske and Chandler, 2011)  
6744 contains generic bootstrapping methods for certain hierarchical models, including  
6745 distance sampling (e.g., see Sillett et al., 2012, for an application). In simple cases,  
6746 using classical inference methods, it is sometimes possible to identify a test statistic  
6747 of theoretical merit, perhaps with a known asymptotic distribution. For examples  
6748 from capture-recapture see Burnham et al. (1987), Lebreton et al. (1992), and  
6749 Chapt. 5 of Cooch and White (2006). For Bayesian analysis we use the Bayesian  
6750 p-value method (Gelman et al., 1996) (we introduced the Bayesian p-value in sec.  
6751 3.9.1). Using this approach, data sets are simulated based on a posterior sample of  
6752 the model parameters  $\theta$  and some fit statistic for the simulated data sets, usually  
6753 based on the discrepancy of the observed data from its expected values, is compared  
6754 to that for the actual data. In most cases, whether Bayesian or frequentist, the main  
6755 idea for assessing model fit is the same: We compare data sets from the model we're  
6756 interested in with the data set we have in hand. If they appear to be consistent  
6757 with one another, then our faith in the model increases, at least to some extent,  
6758 and we say "the model fits."

6759 To date, we are unaware of any goodness-of-fit applications based on likelihood  
6760 analysis of SCR models. For Bayesian analysis of SCR models, there has not been

6761 a definitive or general proposal for a fit statistic or even a class of fit statistics,  
 6762 although a few specialized implementations of Bayesian p-values have been pro-  
 6763 vided (Royle, 2009; Gardner et al., 2010a; Royle et al., 2011a; Gopalaswamy et al.,  
 6764 2012a,b; Russell et al., 2012). While we universally adopt the Bayesian p-value  
 6765 approach, and suggest some fit statistics in the following text, we caution that  
 6766 there is no general expectation to support how well they should do. As such, one  
 6767 might consider doing some kind of custom evaluation or calibration when using  
 6768 such methods, if the power of the test (ability to reject under specific departures  
 6769 from the model) is of paramount interest. We note that this uncertain power or  
 6770 performance of the Bayesian p-value is not a weakness of the Bayesian approach  
 6771 because the same issue applies in using bootstrap approaches applied to classical  
 6772 analysis of models, if we were to devise such methods.

## 7.4 THE TWO COMPONENTS OF MODEL FIT

6773 For most SCR models, there are at least two distinct components of model fit,  
 6774 and we propose to evaluate these two distinct components individually. First, we  
 6775 can ask, are the data consistent with the *observation* model, conditional on the  
 6776 underlying point process? We can evaluate this based on the encounter frequencies  
 6777 of individuals *conditional* on (posterior samples of) the underlying point process  
 6778  $s_1, \dots, s_N$ . We discuss some potential fit statistics for addressing this in the next  
 6779 section. Second, we can evaluate whether the data appear consistent with the *state*  
 6780 process model (i.e., the “uniformity” assumption of the point process). For the  
 6781 simple model of independence and uniformity, this is similar to the assumption of  
 6782 *complete spatial randomness* (CSR) which we consider in Sec. 8.4.1 below. Actually,  
 6783 this is not strictly the assumption of CSR because of the binomial assumption on  
 6784  $N$  under data augmentation, so we instead use the term *spatial randomness*.

### 6785 7.4.1 Testing uniformity or spatial randomness

6786 Historically, especially in ecology, there has been an extraordinary amount of inter-  
 6787 est in whether a realization of a point process indicates “complete spatial random-  
 6788 ness,” i.e., that the points are distributed uniformly and independently in space.  
 6789 Two good references for such things are Cressie (1991, Ch. 8) and Illian et al.  
 6790 (2008)<sup>1</sup>. In the context of animal capture-recapture studies, the spatial random-  
 6791 ness hypothesis is manifestly false, purely on biological grounds. Typically individ-  
 6792 uals will be clustered, or more regular (for territorial species), than expected under  
 6793 spatial randomness and heterogeneous habitat will generate the appearance of clus-  
 6794 tering even if individuals are distributed independently of one another. While we  
 6795 recommend modeling spatial structure explicitly when possible (Chaps. 11, 12,

<sup>1</sup>We also like Tony Smith’s lecture notes (Univ. of Penn. ESE 502), which can be found at [http://www.seas.upenn.edu/~ese502/NOTEBOOK/Part\\_I/3\\_Testing\\_Spatial\\_Randomness.pdf](http://www.seas.upenn.edu/~ese502/NOTEBOOK/Part_I/3_Testing_Spatial_Randomness.pdf), accessed January 24, 2013.

6796    13), the uniformity assumption may be an adequate description of data sets in  
 6797    some situations. Further, we find that it is generally flexible enough to reflect non-  
 6798    uniform patterns in the data, because we do observe some direct information about  
 6799    some of the point locations.

6800    The basic technical framework for evaluating the spatial randomness hypothesis  
 6801    is based on counts of activity centers in cells or bins. For that we use any standard  
 6802    goodness-of-fit test statistic, based on gridding (i.e., binning) the state-space of the  
 6803    point process into  $g = 1, 2, \dots, G$  cells or bins, and we tabulate  $N_g \equiv N(\mathbf{x}_g)$  the  
 6804    number of activity centers in bin  $g$ , centered at coordinate  $\mathbf{x}_g$ . Specifically, let  
 6805     $B(\mathbf{x})$  indicate a bin centered at coordinate  $\mathbf{x}$ , then<sup>2</sup>  $N(\mathbf{x}) = \sum_{i=1}^N I(\mathbf{s}_i \in B(\mathbf{x}))$  is  
 6806    the population size of bin  $B(\mathbf{x})$ . In Sec. 5.11.1, we used the summaries  $N(\mathbf{x})$  for  
 6807    producing density maps from MCMC output. Here, we use them for constructing  
 6808    a fit statistic. We have used the Freeman-Tukey statistic of this form:

$$T(\mathbf{N}, \theta) = \sum_g (\sqrt{N_g} - \sqrt{\mathbb{E}(N_g)})^2$$

6809    where  $\mathbb{E}(N_g)$  is estimated by the mean bin count. An alternative conventional  
 6810    assessment of fit is based on the following statistic: Conditional on  $N$ , the total  
 6811    number of activity centers in the state-space  $\mathcal{S}$ , the bin counts  $N_g$  should have a  
 6812    binomial distribution. It will usually suffice to approximate the binomial cell counts  
 6813    by Poisson cell counts, in which case we can use the classical “index-of-dispersion”  
 6814    test (Illian et al., 2008, p. 87), based on the variance-to-mean ratio:

$$I = (G - 1) * s^2 / \bar{N}$$

6815    where  $s^2$  is the sample variance of the bin counts and  $\bar{N}$  is the sample mean. When  
 6816    the point process realization is *observed*, as in classical point pattern modeling  
 6817    (but not in SCR), this statistic has approximately a Chi-square distribution on  
 6818     $(G - 1)$  degrees-of-freedom under the spatial randomness hypothesis. If  $s^2 / \bar{N} > 1$ ,  
 6819    clustering is suggested whereas,  $s^2 / \bar{N} < 1$  suggests the point process is too regular.

6820    Whatever statistic we choose as our basis for assessing spatial randomness, *the*  
 6821    important technical issue is that we don’t observe the point process and so the  
 6822    standard statistics for evaluating spatial randomness cannot be computed directly.  
 6823    However, using Bayesian analysis, we do have a posterior sample of the underlying  
 6824    point process and so we suggest computing the posterior distribution of any  
 6825    statistic in a Bayesian p-value framework. For a given posterior draw of all model  
 6826    parameters,  $N$  is known, based on the value of the data augmentation variables  $z_i$ ,  
 6827    and so we can obtain a posterior sample of  $N(\mathbf{x})$  by taking all of the output for  
 6828    MCMC iterations  $m = 1, 2, \dots$ , and doing this:

$$N(\mathbf{x})^{(m)} = \sum_{z_i^{(m)}=1} I(\mathbf{s}_i^{(m)} \in B(\mathbf{x}))$$

---

<sup>2</sup> $I(arg)$  is the indicator function which evaluates to 1 if  $arg$  is true, otherwise 0

6829 Thus,  $N(\mathbf{x})^{(1)}, N(\mathbf{x})^{(2)}, \dots$ , is the Markov chain for the derived parameter  $N(\mathbf{x})$ .

6830 In addition to computing the bin counts for each iteration of the MCMC algorithm,  
 6831 at the same time we generate a realization of the activity centers  $\mathbf{s}_i$  under the  
 6832 spatial randomness model, and we obtain bin counts for these “new” data,  $\tilde{N}(\mathbf{x})$ .  
 6833 For each of the posterior samples – that of the real data, and that of the posterior  
 6834 simulated data, we compute the fit-statistic. The fit statistic based on the actual  
 6835 data is:

$$T(\mathbf{N}, \theta) = \sum_x (\sqrt{N(\mathbf{x})} - \sqrt{\bar{N}(\mathbf{x})})^2$$

6836 whereas the fit statistic based on a simulated realization of points under the spatial  
 6837 randomness hypothesis is:

$$T(\tilde{\mathbf{N}}, \theta) = \sum_x (\sqrt{\tilde{N}(\mathbf{x})} - \sqrt{\bar{N}(\mathbf{x})})^2$$

6838 And we compute the Bayesian p-value by tallying up the proportion of times that  
 6839  $T(\tilde{\mathbf{N}}, \theta)$  is larger than  $T(\mathbf{N}, \theta)$ , as an estimate of:  $p = \Pr(T(\tilde{\mathbf{N}}, \theta) > T(\mathbf{N}, \theta))$ . The  
 6840 R function `SCRgof` in our package `scrbook` will do this, given the output from  
 6841 **JAGS** (see below).

#### 6842 Sensitivity to bin size

6843 Evaluating fit based on bin counts in point process models are sensitive to the  
 6844 number of bins (Illian et al., 2008, p. 87-88). This is related to the classical problem  
 6845 of fit testing for binary regression because in a point process model, as the number of  
 6846 grid cells gets small, the grid cell counts go to 0 or 1 and standard fit statistics (e.g.,  
 6847 based on deviance or Pearson residuals) are known not to be very useful. There is  
 6848 some good discussion of this in McCullagh and Nelder (1989, Sec. 4.4.5). What it  
 6849 boils down to is, using the example of the Pearson residual statistic considered by  
 6850 McCullagh and Nelder (1989), the fit statistic is exactly a deterministic function of  
 6851 the sample size only, which clearly should not be regarded as useful for model fit.  
 6852 This is why, in order to do a check of model fit when you have a binary response, one  
 6853 must always aggregate the data in some fashion. In the context of testing spatial  
 6854 randomness, computing the test statistic we described above has us chop up the  
 6855 region  $\mathcal{S}$  into bins, and tally up  $N_g$ , the frequency of activity centers in each bin  $g$ .  
 6856 Suppose that we choose the bin size to be extremely small such that  $\mathbb{E}(N_g)$  tends  
 6857 to  $N/G$  ( $N$  being the number of activity centers). Further,  $N_g$  tends to a binary  
 6858 outcome. Therefore the fit statistic has  $N$  components that have value  $N_g = 1$ ,  
 6859 and it has  $G - N$  components that have value  $N_g = 0$ . Therefore, the fit statistic  
 6860 resembles:

$$T(\mathbf{N}, \theta) = \sum_{g \ni N_g=1}^N (1 - \sqrt{N/G})^2 + \sum_{g \ni N_g=0}^{G-N} (N/G)^2 = N(1 + (G - N)/G)$$

(here  $\exists$  means “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 typical SCR applications this will therefore result, usually, in very large (and few) bins, and presumably not much power.

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 on which inference is focused 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.

#### **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 observed data that live in the vicinity of the trap array. Therefore we recommend computing these goodness-of-fit statistics in the vicinity of the trap array only. Perhaps, as an ad hoc rule-of-thumb, less than the average trap spacing from the rectangle enclosing the trap array. For example, if the average 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. This should be a matter of future research.

#### **7.4.2 Assessing fit of the observation model**

In evaluating the spatial randomness hypothesis, we could draw on well-established ideas from point process modeling. On the other hand, 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,  $y_{ijk}$  for individual  $i$ , trap  $j$  and sample occasion  $k$ . 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 too sparse to have much power, unless the data are 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 we 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  $y_{ijk}$  aggregated over all sample occasions. Using standard “dot notation” to represent summed quantities, we express that as:  $y_{ij\cdot} = \sum_{k=1}^K y_{ijk}$ . Conditional on  $\mathbf{s}_i$ , the expected value under any encounter model is:

$$\mathbb{E}(y_{ij\cdot}) = p_{ij}K$$

(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  $\mathbb{E}(y_{ij\cdot}) = \sum_k p_{ijk}$ . Then we can define a fit statistic from the Freeman-Tukey residuals according to:

$$T_1(\mathbf{y}, \theta) = \sum_i \sum_j (\sqrt{y_{ij\cdot}} - \sqrt{\mathbb{E}(y_{ij\cdot})})^2$$

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  $n_{ind} \times K$  matrix of count statistics. We imagine that either summary of the data will probably be too disaggregated (have mostly values of 0) in most practical settings to have much power.

**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 part of 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:  $\mathbb{E}(y_{i..}) = \sum_j \sum_k p_{ijk}$ . Then, we define statistic  $T_2$  according to:

$$T_2(\mathbf{y}, \theta) = \sum_i (\sqrt{y_{i..}} - \sqrt{\mathbb{E}(y_{i..})})^2$$

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:  $y_{j\cdot} = \sum_i \sum_k y_{ijk}$  (Gopalswamy et al., 2012b) and the expected value is a similar summary over individuals and occasions:  $\mathbb{E}(y_{j\cdot}) = \sum_i \sum_k p_{ijk}$ . Then statistic  $T_3$  is:

$$T_3(\mathbf{y}, \theta) = \sum_j (\sqrt{y_{j\cdot}} - \sqrt{\mathbb{E}(y_{j\cdot})})^2$$

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?

We use the ideas described in the previous section to evaluate goodness-of-fit of the SCR model to the wolverine camera trapping data.

We consider first whether the simple model of spatial randomness of the activity centers is adequate. We think that the encounter model shouldn't have a large effect on whether the spatial randomness assumption is adequate or not, so we fit "Model 0" (in which parameters are *not* sex-specific) using an **R** script provided in the function `wolvSCR0gof` which will default to fitting the model in **JAGS**. This is the same script as `wolvSCR0ms` except that it saves the MCMC output for the activity centers `s` and the data augmentation variables `z`, which are required in order to compute the Bayesian p-value test of spatial randomness.

The MCMC output is processed with the **R** function `SCRgof` which computes the test of spatial randomness based on bin counts, using the Bayesian p-value calculation. The function `SCRgof` requires a few things as inputs: (1) the output from a **BUGS** run (in particular, the activity center coordinates and the data augmentation variables); (2) the number of bins to create for computing spatial frequencies of activity centers; (3) the trap locations and, (4) the buffer around the trap array to use in computing the bin counts. This buffer could be that used in defining the state-space for the model fitting, but we think it should be relatively tighter to the trap array than the state-space used in model-fitting. For the wolverine analysis, where we're using 10-km grid cells (1 unit = 10 km) and a 20 km buffer for model fitting, we'll use a state-space buffer of 0.4 units (4 km) for computing the fit statistic. The **R** code to fit the model and obtain the goodness-of-fit result is as follows:

```

6975 > wolv1 <- wolvSCR0gof(nb=1000,ni=6000,buffer=2,M=200,model=0)
6976
6977 > bugsout <- wolv1$BUGSoutput$sims.list
6978
6979 > traplocs <- wolverine$wtraps[,2:3]
6980 > traplocs[,1] <- traplocs[,1] - min(traplocs[,1])
6981 > traplocs[,2] <- traplocs[,2] - min(traplocs[,2])
6982 > traplocs <- traplocs/10000
6983
6984 > set.seed(2013) # set seed so Bayesian p-value is the same each time
6985
6986 > SCRgof(bugsout,5,5,traplocs=traplocs,buffer=.4)
6987
6988 Cluster index observed: 1.099822
6989 Cluster index simulated: 1.000453
6990 P-value index of dispersion: 0.408
6991 P-value2 freeman-tukey: 0.6842667

```

6992 The output produced by **SCRgof** is the cluster index based on the ratio of the  
 6993 variance to the mean (see above), which is computed as the posterior mean index  
 6994 of dispersion for the latent point process, and also the average value for simulated  
 6995 data. If this value is  $> 1$  then clustering is suggested, which we see a (very) minor  
 6996 amount of evidence for here. Two Bayesian p-values are produced: the first is  
 6997 based on the cluster index, and the 2nd is based on the Freeman-Tukey statistic  
 6998 calculated as described in Sec. 8.4.1. Because our p-values aren't close to 0 or  
 6999 1, we judge that the model of spatial randomness provides an adequate fit to the  
 7000 data. You can verify that a similar result is obtained if we use the model with fully  
 7001 sex-specific parameters (Model 4).

7002 Next, we did a Bayesian p-value analysis of the observation component of the  
 7003 model, using the 3 fit statistics described in Sec. 8.4.2. These statistics can be  
 7004 calculated as part of the **BUGS** model specification or by post-processing the  
 7005 MCMC output returned from a **BUGS** run. The **R** script **wolvSCR0gof** contains  
 7006 the relevant calculations. For example, to compute fit statistic 1, we have to add  
 7007 some commands to the **BUGS** model specification such as this (note: this is only  
 7008 a fraction of the model specification):

```

7009 .....
7010 for(j in 1:ntraps){
7011   mu[i,j] <- w[i]*p[i,j]
7012
7013   y[i,j] ~ dbin(mu[i,j],K[j])
7014   ynew[i,j] ~ dbin(mu[i,j],K[j])
7015
7016   err[i,j] <- pow(pow(y[i,j],.5) - pow(K[j]*mu[i,j],.5),2)
7017   errnew[i,j] <- pow(pow(ynew[i,j],.5) - pow(K[j]*mu[i,j],.5),2)

```

---

```

7018 }
7019
7020 T1obs <- sum(err[,])
7021 T1new <- sum(errnew[,])
7022 .....
7023 Similar calculations are carried out to obtain the posterior samples of test statistics
7024 2 (individual totals) and 3 (trap totals). For the wolverine data, the Bayesian
7025 p-value calculations produce:
7026 > mean(wolv1$BUGSoutput$sims.list$T1new>wolv1$BUGSoutput$sims.list$T1obs)
7027 [1] 0
7028
7029 > mean(wolv1$BUGSoutput$sims.list$T2new>wolv1$BUGSoutput$sims.list$T2obs)
7030 [1] 0.17
7031
7032 > mean(wolv1$BUGSoutput$sims.list$T3new>wolv1$BUGSoutput$sims.list$T3obs)
7033 [1] 0.02066667

```

7034 Based on statistic  $T_2$ , we might conclude that the model is adequate for explaining individual heterogeneity although the other two statistics suggest a general lack of fit of the observation model. A similar result is obtained using the fully sex-specific model. We note that one individual was captured 8 times in one trap, which is pretty extreme under a model which assumes independent Bernoulli trials. We summarize that the trap-counts simply are not well-explained by this model.

7040 In attempt to resolve this problem, we extended the model to include a local  
 7041 (trap-specific) behavioral response (following Royle et al. (2011b)) which can be  
 7042 fitted using the sample **R** script **wolvSCRMb**. To fit a model using **WinBUGS**, and  
 7043 then compute the Bayesian p-values we do this:

```

7044 > wolv.Mb <- wolvSCRMb(nb=1000,ni=6000,buffer=2,M=200)
7045
7046 > mean(wolv.Mb$sims.list$T1new>wolv.Mb$sims.list$T1obs)
7047 [1] 0.9666667
7048
7049 > mean(wolv.Mb$sims.list$T2new>wolv.Mb$sims.list$T2obs)
7050 [1] 0.3644667
7051
7052 > mean(wolv.Mb$sims.list$T3new>wolv.Mb$sims.list$T3obs)
7053 [1] 0.4990667

```

7054 Given that this model seems to fit better, we might prefer reporting estimates  
 7055 under this model, which we do in Table 8.4. (the behavioral response parameter is  
 7056 labeled  $\alpha_2$  in the table). Estimated density is about 1 individual higher per 1000  
 7057  $\text{km}^2$  compared with the various models that lack a behavioral response. It might be  
 7058 useful to try these fit assessment exercises using the habitat mask as described in

7059 Sec. 5.10. That takes an extremely long time to run in **BUGS** though, especially  
 7060 for the behavioral response model.

**Table 7.4.** Posterior summary statistics for local (trap-specific) behavioral response model  $M_b$  fitted to the wolverine camera trapping data using **WinBUGS**. The parameter  $\alpha_2$  is the local (trap-specific) behavioral response parameter.  $T_x()$  are the posterior summaries of fit statistics  $x = 1, 2, 3$  used in the Bayesian p-value analysis (See text for definitions). Results are based on 3 chains, each with 6000 iterations (first 1000 discarded) for a total of 15000 posterior samples.

Parameter	Mean	SD	2.5%	50%	97.5%	Rhat	n.eff
$N$	71.32	19.07	42.00	69.00	114.02	1.00	2100
$D$	6.87	1.84	4.05	6.65	10.99	1.00	2100
$\sigma$	0.88	0.13	0.68	0.86	1.17	1.00	730
$p_0$	0.01	0.00	0.01	0.01	0.02	1.01	530
$\alpha_1$	0.69	0.19	0.37	0.67	1.10	1.00	730
$\alpha_2$	2.50	0.27	1.99	2.50	3.04	1.00	700
$\psi$	0.36	0.10	0.20	0.35	0.58	1.00	2600
$T_1^{obs}$	54.71	6.12	43.69	54.39	67.47	1.00	3900
$T_1^{new}$	64.73	7.62	50.93	64.39	80.96	1.00	3900
$T_2^{obs}$	13.93	4.07	7.25	13.53	23.04	1.00	5700
$T_2^{new}$	12.65	3.35	6.93	12.36	20.07	1.00	2000
$T_3^{obs}$	12.80	1.74	9.80	12.64	16.61	1.00	2400
$T_3^{new}$	12.94	3.05	7.77	12.67	19.58	1.00	15000

## 7.5 QUANTIFYING LACK-OF-FIT AND REMEDIATION

7061 Molinari-Jobin et al. (2013) used a strategy for assessing model fit in dynamic  
 7062 occupancy models (Royle and Kéry, 2007) similar to that which we suggested above.  
 7063 They constructed a fit statistic based on aggregating the data over replicate samples  
 7064 ( $k$ ), to obtain the total detections per site  $i$  and year  $j$ . They used a Bayesian p-  
 7065 value analysis based on a Chi-squared test statistic (also see Kéry and Schaub,  
 7066 2012, Chapt. 12). Their analysis suggested a model that didn't fit, and, so they  
 7067 computed the "lack-of-fit ratio" (see Kéry and Schaub, 2012, Sec. 12.3) – the  
 7068 ratio of the fit statistic computed for the actual data to that of the replicate data  
 7069 sets. They interpret this analogous to the over-dispersion coefficient in generalized  
 7070 linear models (McCullagh and Nelder, 1989), usually called the c-hat statistic in  
 7071 capture-recapture literature (see Cooch and White, 2006, Chapt. 5). Molinari-  
 7072 Jobin et al. (2013) reported the lack-of-fit ratio for their model to be 1.14 which  
 7073 suggests a minor lack-of-fit, compared to perfect data having a value of 1, because  
 7074 the posterior standard deviations will be too small by a factor of  $\sqrt{1.14} = 1.07$ .  
 7075 In classical capture-recapture applications of goodness-of-fit assessment, inference  
 7076 for non-fitting models is dealt with by inflating the resulting SEs (of the non-  
 7077 fitting model), by the square-root of c-hat. We believe that these ideas related to

7078 quantifying lack-of-fit and understanding its effect could also be applied to SCR  
7079 models, although we have not yet explored this.

## 7.6 SUMMARY AND OUTLOOK

7080 In this chapter, we offered some general strategies for model selection and model  
7081 checking, or assessment of model fit. We think the strategies we outlined for model  
7082 selection are fairly standard and can be effectively applied to many SCR modeling  
7083 problems. Some technical issues of Bayesian analysis need to be addressed (in  
7084 general) before Bayesian methods are more generally useful and accessible. For  
7085 one thing, Bayesian model selection based on the indicator variable approach of  
7086 Kuo and Mallick (1998) can be tediously slow even for small data sets, and so  
7087 improved computation will improve our ability to do Bayesian model selection in  
7088 practical situations. Also, and most importantly, sensitivity to prior distributions  
7089 is an important issue. Further research and practice might identify preferred prior  
7090 configurations for SCR that provide a good calibration in relevant model selection  
7091 problems. Finally, we believe that cross-validation should prove to be a useful  
7092 method in model assessment and selection, as SCR models are a form of spatial  
7093 model of counts, and so it is natural to pick models that predict the observable  
7094 spatial counts (i.e., at trap locations) well.

7095 For Bayesian model assessment, or goodness-of-fit checking, we suggested a  
7096 framework based on independent testing of the spatial model of independence and  
7097 uniformity, and testing fit of the observation model conditional on the underlying  
7098 point process. These ideas are based on mostly *ad hoc* attempts in a number  
7099 of published applications (Royle et al., 2009a, 2011a; Gopalaswamy et al., 2012b;  
7100 Russell et al., 2012, e.g.). While we think this general strategy should be fruitful,  
7101 we know of no studies on the power to detect various model departures, and so  
7102 the ideas should be viewed as experimental. We have not discussed assessment  
7103 of model fit for SCR models using likelihood methods, although we imagine that  
7104 standard bootstrapping ideas should be effective, perhaps based on the fit statistics  
7105 (or similar ones) we suggested here for computing Bayesian p-values.

7106 Clearly there is much research to be done on assessment of model fit in SCR  
7107 models. For testing the spatial randomness hypothesis, we used a classical ap-  
7108 proach based on count frequencies, in which point locations are put into spatial  
7109 bins. Other approaches from spatial point process modeling should be pursued in-  
7110 cluding nearest-neighbor methods or distance-based methods. In addition, studies  
7111 to evaluate the power to detect relevant departures from the standard assumptions,  
7112 and the robustness of inferences about  $N$  or density, need to be conducted. If the  
7113 spatial randomness model appears inadequate, it is possible to fit models that al-  
7114 low for a non-uniform distribution of points (see Chapt. 11) and even point process  
7115 models that allow for interactions among points (Reich et al., 2012). On the other  
7116 hand, we expect that most of these Bayesian p-value tests will have low power in  
7117 typical data sets consisting of a few to a few dozen individuals. As such, failure to

7118 detect a lack of fit may not be that meaningful. But, on the other hand, it may not  
7119 make a difference in terms of density estimates either. We think inference about  
7120 density should be relatively insensitive to departures from spatial randomness, be-  
7121 cause we get to observe direct information on some component of the population,  
7122 component of density is *observed*. For those activity centers, the assumed model  
7123 of the point process should exert little influence on the placement of the activity  
7124 centers. Conversely, as is the case with classical closed population models (Otis  
7125 et al., 1978; Dorazio and Royle, 2003; Link, 2003), inferences may be somewhat  
7126 more sensitive to bad-fitting models for the observation process.

7127  
7128

---

7129

# 8

## ALTERNATIVE OBSERVATION MODELS

7130 In previous chapters we considered various models of *encounter probability*, both  
7131 in terms of parametric functions of distance and also a myriad of covariate models  
7132 (Chapt. 7 and elsewhere). However, we have so far only considered a specific  
7133 probability model for the observations (we'll call this the "encounter process") –  
7134 the Bernoulli encounter process model which, in `secr`, is the *proximity detector*  
7135 model. This assumes that individual and trap-specific encounters are independent  
7136 Bernoulli trials. Here, we focus on developing additional models for the encounter  
7137 process. The encounter process could be thought of as being determined by the  
7138 type of device – or the type of "detector" using the terminology of `secr` (Efford,  
7139 2011).

7140 In this chapter, we consider alternative observation models that accommodate  
7141 observations that are not binary, and do not require independence of the observations.  
7142 In particular, we consider models for encounter *frequencies*, and encounter  
7143 process models based on the multinomial distribution. For example, if sampling de-  
7144 vices can detect an individual some arbitrary number of times during an interval,  
7145 then it is natural to consider observation models for encounter frequencies, such as  
7146 the Poisson model. Another type of encounter device is the "multi-catch" device  
7147 (Efford et al., 2009a) which is a physical device that can capture and hold an ar-  
7148bitrary number of individuals. A typical example is a mist-net for birds (Borchers  
7149 and Efford, 2008). It is natural to regard observations from these kinds of studies as  
7150 independent multinomial observations. A related type of device that produces *de-  
7151 pendent* multinomial observations are the so-called *single-catch* traps (Efford, 2004;  
7152 Efford et al., 2009a). The canonical example are small-mammal live traps which  
7153 catch and hold a single individual. Competition among individuals for traps in-  
7154 duces a complex dependence structure among individual encounters. To date, no  
7155 formal inference framework has been devised for this method although it stands to

7156 reason that the independent multinomial model should be a good approximation in  
 7157 some situations (Efford et al., 2009a). We analyze a number of examples of these  
 7158 different observation models using **JAGS** and also the **R** package **secr** (Efford,  
 7159 2011).

## 8.1 POISSON OBSERVATION MODEL

7160 The models we analyze in Chapt. 5 assumed binary observations – i.e., standard  
 7161 encounter history data – so that individuals are captured at most one time in a trap  
 7162 on any given sample occasion. This makes sense for many types of DNA sampling  
 7163 (e.g., based on hair snares) because distinct visits to sampled locations or devices  
 7164 cannot be differentiated. However, for some encounter devices, or methods, the  
 7165 potential number of encounters is *not* fixed, and so it is possible to encounter an  
 7166 individual some arbitrary number of times during any particular sampling episode.  
 7167 That is, we might observe encounter frequencies  $y_{ijk} > 1$  for individual  $i$ , trap  $j$   
 7168 and sampling interval  $k$ . As an example, if a camera device is functioning properly  
 7169 it may be programmed to take photos every few seconds if triggered. For a second  
 7170 example, suppose we are searching a quadrat or length of trail for scat, we may  
 7171 find multiple samples from the same individual. Therefore, we seek observation  
 7172 models that accommodate such encounter frequency data. In general, any discrete  
 7173 probability mass function could be used for this purpose, including the standard  
 7174 models for count data used throughout ecology, the Poisson and negative binomial.  
 7175 Here we focus on using the Poisson model only although other count frequency  
 7176 models are possible for SCR models (Efford et al., 2009b).

7177 Let  $y_{ijk}$  be the frequency of encounter for individual  $i$ , in trap  $j$ , during occasion  
 7178  $k$ , then assume:

$$y_{ijk} \sim \text{Poisson}(\lambda_{ij})$$

7179 where the expected encounter frequency  $\lambda_{ij}$  depends on both individual and trap.  
 7180 As we did in the binary model of Chapt. 5, we now seek to model the expected  
 7181 value of the observation (which was  $p_{ij}$  in Chapt 5) as a function of the individual  
 7182 activity center  $\mathbf{s}_i$ . We propose

$$\lambda_{ij} = \lambda_0 k(\mathbf{x}_j, \mathbf{s}_i)$$

7183 Where  $k(\mathbf{x}, \mathbf{s})$  is any positive valued function, such as the negative exponential or  
 7184 the bivariate Gaussian kernel, and  $\lambda_0$  is the baseline encounter rate – the expected  
 7185 number of encounters if a trap is placed precisely at an individuals home range  
 7186 center (note: in **secr** the notation for this is  $g_0$ ). Then,  $\lambda_0 k(\mathbf{x}_j, \mathbf{s}_i)$  is the expected  
 7187 encounter rate in trap  $\mathbf{x}_j$  for an individual having activity center  $\mathbf{s}_i$ . Note that

$$\log(\lambda_{ij}) = \log(\lambda_0) + \log(k(\mathbf{x}_j, \mathbf{s}_i)).$$

7188 Equating  $\alpha_0 \equiv \log(\lambda_0)$ , and, if  $k(\mathbf{x}, \mathbf{s}) \equiv \exp(-d(\mathbf{x}, \mathbf{s})^2/(2\sigma^2))$  (i.e., the Gaussian  
 7189 model), then:

$$\log(\lambda_{ij}) = \alpha_0 - \alpha_1 d(\mathbf{x}_j, \mathbf{s}_i)^2 \quad (8.1.1)$$

where  $\alpha_1 = 1/(2\sigma^2)$ , which is the same linear predictor as we have seen for the Bernoulli model in Chapt. 5. This Poisson SCR model is therefore a type of Poisson generalized linear mixed model (GLMM).

We can accommodate covariates at the level of individual-, trap- or sample occasion by including them on the baseline encounter rate parameter  $\lambda_0$ . For example, if  $C_j$  is some covariate that depends on trap only, then we express the relationship between  $\lambda_0$  and  $C_j$  as:

$$\log(\lambda_{0,ijk}) = \alpha_0 + \alpha_2 C_j$$

and therefore covariates on the logarithm of baseline encounter probability appear also as linear effects on  $\lambda_{ij}$ . In general, covariates might also affect the coefficient on the distance term ( $\alpha_1$ ) (e.g., sex of individual). We don't get into too much discussion of general covariate models here, but we covered them in some detail in both Chaps. 7 and 8.

For models in which we do not have covariates that vary across the sample occasions  $k$ , we can aggregate the observed data by the property of compound additivity of the Poisson distribution (if  $x$  and  $y$  are iid Poisson with mean  $\lambda$  then  $x + y$  is Poisson with mean  $2\lambda$ ). Therefore,

$$y_{ij} = \left( \sum_{k=1}^K y_{ijk} \right) = \text{Poisson}(K\lambda_0 k(\mathbf{x}_j, \mathbf{s}_i))$$

We see that  $K$  and  $\lambda_0$  serve the same role as affecting the base encounter rate. Since the observation model is the same, probabilistically speaking, for all values of  $K$ , evidently we need only  $K = 1$  "survey" from which to estimate model parameters (Efford et al., 2009b). We know this intuitively, as sampling by multiple traps serves as replication in SCR models. This has great practical relevance to the conduct of capture-recapture studies and the use of SCR models. For example, if individuality is obtained by genetic information from scat sampling, one should only have to carry out a single spatial sampling of the study area. However, one must be certain that sufficient spatial recaptures will be obtained so that effective estimation is possible.

### 8.1.1 Poisson model of space usage

It is natural to interpret the Poisson encounter model as a model of space usage resulting from movement of individuals about their home range (Sec. 5.4). Imagine we have perfect samplers in every pixel of the landscape so that whenever an individual moves from one pixel to another, we can record it. Let  $m_{ij}$  be the number of times individual  $i$  was recorded in pixel  $j$  (i.e., it selected or used pixel  $j$ ). Then, we might think of the Poisson model for the observed use frequencies:

$$m_{ij} \sim \text{Poisson}(\lambda_0 k(\mathbf{x}_j, \mathbf{s}_i))$$

7223 where  $\lambda_0$  is related to the baseline movement rate of the animal (how often it  
 7224 moves). This model of space usage gives rise to the standard resource selection  
 7225 function (RSF) models (see Chapt. 13). But now suppose our samplers are not  
 7226 perfect but, rather, record only a fraction of the resulting visits. A sensible model  
 7227 is

$$y_{ij}|m_{ij} \sim \text{Binomial}(m_{ij}, p).$$

7228 The marginal distribution of  $y_{ij}$  is:

$$y_{ij} \sim \text{Poisson}(p_0 k(\mathbf{x}_j, \mathbf{s}_i)).$$

7229 where  $p_0$  is a composite of the movement rate and conditional detection probability  
 7230  $p$ . Therefore, we see that encounters accumulate in proportion to the frequency of  
 7231 outcomes of an individual using space (or “selecting resources”).

7232 We introduced an interpretation of SCR models in terms of movement and space  
 7233 usage in Sec. 5.4, and it is one of the main underlying concepts of SCR models  
 7234 that is not present in ordinary capture-recapture models. As we noted there, the  
 7235 underlying model of space usage is only as complex as the encounter probability  
 7236 model which has been, so far in this book, only symmetric and stationary (does not  
 7237 vary in space). We generalize this model of space usage substantially in Chapt. 13.

### 7238 8.1.2 Poisson relationship to the Bernoulli model

7239 There is a sense in which the Poisson and Bernoulli models can be viewed as consistent  
 7240 with one another. Note that under the Poisson model, the relationship between  
 7241 the expected count and the probability of counting “at least 1”, is given by

$$\Pr(y > 0) = 1 - \exp(-\lambda) \tag{8.1.2}$$

7242 where  $\mathbb{E}(y) = \lambda$ . Therefore, if we equate the event “encountered” with the event  
 7243 that the individual was captured at least 1 time under the Poisson model, i.e.,  
 7244  $y > 0$ , then it would be natural to set  $p_{ij} = \Pr(y > 0)$  according to Eq. 9.1.2.  
 7245 That is, we can use Eq. 9.1.2 as the model for encounter probability for binary  
 7246 observations. This is the “hazard rate” model in distance sampling.

7247 In fact, as  $\lambda$  gets small, the Poisson model is a close approximation to the  
 7248 Bernoulli model in the sense that outcomes concentrate on  $\{0, 1\}$ , i.e.,  $\Pr(y \in$   
 7249  $\{0, 1\}) \rightarrow 1$  as  $\lambda \rightarrow 0$ . Indeed, under the Poisson model,  $\Pr(y > 0) \rightarrow \lambda$  for small  
 7250 values of  $\lambda$ . This phenomenon is shown in Fig. 9.1 where the left panel shows  
 7251 a plot of  $\lambda_{ij} = \lambda_0 k(\mathbf{x}_j, \mathbf{s}_i)$  vs. distance and superimposed on that is a plot of  
 7252  $p_{ij} = 1 - \exp(-\lambda_{ij})$  vs. distance, for values  $\lambda_0 = 0.1$  and  $\sigma = 1$ , and the right  
 7253 panel shows a plot of  $\Pr(y > 0)$  vs.  $\mathbb{E}(y)$ . We see that the two quantities are  
 7254 practically indistinguishable. This is convenient in some cases because the Poisson  
 7255 model might be more tractable to fit (or even vice versa). For an example, see  
 7256 the models described in Chapt. 18, and we also consider another case in Sec. 9.3  
 7257 below. To evaluate the closeness of the approximation, you can use the following  
 7258 **R** commands which we used to produce Fig. 9.1:

```

7259 > x <- seq(0.001,5,,200)
7260 > lam0 <- .1
7261 > sigma <- 1
7262 > lam <- lam0*exp(-x*x/(2*sigma*sigma))
7263
7264 > par(mfrow=c(1,2))
7265 > p1 <- 1-exp(-lam)
7266 > plot(x, lam, ylab="E[y] or Pr(y>0)", xlab="distance", type="l", lwd=2)
7267 > lines(x,p1,lwd=2,col="red")
7268 > plot(lam, p1, xlab="E[y]", ylab="Pr(y>0)", type="l", lwd=2)
7269 > abline(0,1,col="red")

```

7270 To summarize, if  $y$  is Poisson then, as  $\lambda$  gets small,

$$\begin{aligned} \Pr(y > 0) &\approx \mathbb{E}(y) \\ 1 - \exp(-\lambda_0 k(\mathbf{x}, \mathbf{s})) &\approx \lambda_0 k(\mathbf{x}, \mathbf{s}) \end{aligned} \quad (8.1.3)$$

7271 What all of this suggests it that if we have very few observations  $> 1$  in our SCR  
 7272 data set, then we won't lose much information by using the Bernoulli model. On the  
 7273 other hand, the Poisson model may have some advantages in terms of analytic or  
 7274 numerical tractability in some cases. Further, this approximation explains the close  
 7275 correspondence we have found between these two versions of the Gaussian encounter  
 7276 probability model (Sec. 5.4). Namely, the Gaussian hazard model and the Gaussian  
 7277 encounter probability model are close approximations because  $1 - \exp(-\lambda) \approx \lambda$  if  
 7278  $\lambda$  is small.

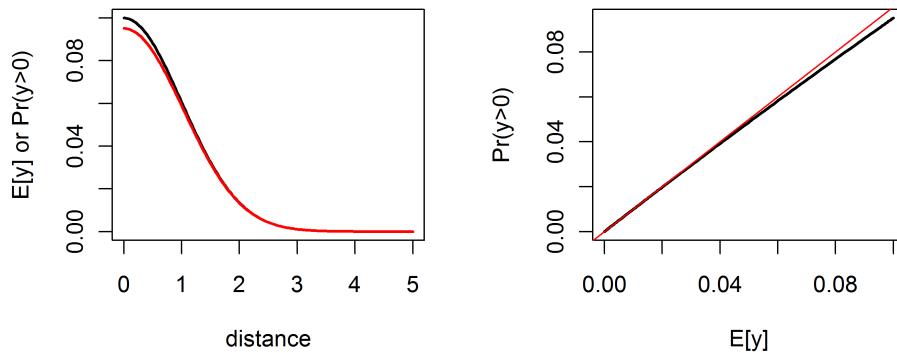
7279 Even in such cases where the Poisson and Bernoulli models are not quite equiv-  
 7280 alent, we might choose to truncate individual encounter frequencies to binary ob-  
 7281 servations anyhow (transforming counts to 0/1 is called "quantizing"). We might  
 7282 do this intentionally in some cases, such as when the distinct encounter events are  
 7283 highly dependent as often happens in camera trap studies when the same individual  
 7284 moves back-and-forth in front of a camera during a short period of time. But some-  
 7285 times, truncation is a feature of the sampling. For example, in the case of bear hair  
 7286 snares, the number of encounters might be well approximated by a Poisson distri-  
 7287 bution but we cannot determine unique visits and so only get to observe the binary  
 7288 event " $y > 0$ ". In this case, we might choose to model the encounter probability  
 7289 for the binary encounter using Eq. 9.1.4. This is equivalent to the complementary  
 7290 log-log link model, or the "Gaussian hazard" as we called it in Chapt. 5:

$$\text{cloglog}(p_{ij}) = \log(\lambda_0) + \log(k(\mathbf{x}, \mathbf{s}))$$

7291 where  $\text{cloglog}(u) = \log(-\log(1 - u))$ .

### 7292 8.1.3 A cautionary note on modeling encounter frequencies

7293 Other models for counts might be appropriate. For example, ecologists are espe-  
 7294 cially fond of negative binomial models for count data (Ver Hoef and Boveng, 2007;



**Figure 8.1.** 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  $\lambda$ .

7295 White and Bennetts, 1996; Kéry et al., 2005) but other models for excess-Poisson  
 7296 variation are possible. For example, we might add a normally distributed random  
 7297 effect to the linear predictor (Coull and Agresti, 1999).

7298 As a general rule we favor the Bernoulli observation model even if our sampling  
 7299 scheme produces encounter frequencies. The main reason is that, with frequency  
 7300 data, we are forced to confront a model choice problem (i.e., Poisson, negative  
 7301 binomial, log-normal mixture) that is wholly unrelated to the fundamental space  
 7302 usage process that underlies the genesis of many types of SCR data. Repeated  
 7303 encounters over short time intervals are not likely to be the result of independent  
 7304 encounter events. E.g., an individual moving back and forth in front of a camera  
 7305 yields a cluster of observations that is not informative about the underlying spatial  
 7306 structure of the population. Similarly in scat surveys dogs are used to locate scats  
 7307 which are processed in the lab for individuality (Kohn et al., 1999; MacKay et al.,  
 7308 2008; Thompson et al., 2012). The process of local scat deposition is not strictly the  
 7309 outcome of movement or space usage but rather the outcome of complex behavioral  
 7310 considerations as well as dependence in detection of scat by dogs. For example, dogs  
 7311 find (or smell) one scat and then are more likely to find one or more nearby ones, if  
 7312 present, or they get into a den or latrine area and find many scats. The additional  
 7313 assumption required to model variation in observed frequencies (i.e., conditional on  
 7314 location) provides relatively no information about space usage and density, and we  
 7315 feel that the model selection issue should therefore be avoided.

To elaborate on this, we suppose that an individual with activity center  $\mathbf{s}$  visits a particular pixel  $\mathbf{x}$  with some probability  $p(\mathbf{x}, \mathbf{s})$ , and then, once there, deposits a number of scat, or visits a camera some number of times with frequency  $y(\mathbf{x}, \mathbf{s}) \geq 0$ . We describe the outcome of this movement/usage process with a two-level hierarchical model of the form:  $[y|w][w|p(\mathbf{x}, \mathbf{s})]$  where  $w(\mathbf{x}, \mathbf{s})$  is a binary variable that indicates whether the individual with activity center  $\mathbf{s}$  used pixel  $\mathbf{x}$  during some interval, and let  $w(\mathbf{x}, \mathbf{s}) \sim \text{Bernoulli}(p(\mathbf{x}, \mathbf{s}))$ . If we suppose encounter frequency  $y$  is independent of  $\mathbf{x}$  and  $\mathbf{s}$  conditional on the use variable  $w$ , then we see that the model for  $y$  (amount of use) does not depend on  $\mathbf{s}$ .

#### 8.1.4 Analysis of the Poisson SCR model in BUGS

We consider the simplest possible model here in which we have no covariates that vary over sample occasions  $k = 1, 2, \dots, K$  so that we work with the aggregated individual- and trap-specific encounters:

$$y_{ij} = \left( \sum_{k=1}^K y_{ijk} \right) = \text{Poisson}(K\lambda_0 k(\mathbf{x}_j, \mathbf{s}_i))$$

and we consider the bivariate normal form of  $k(\mathbf{x}, \mathbf{s})$ :

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

so that

$$\log(\lambda_{ij}) = \alpha_0 - \alpha_1 d(\mathbf{x}_j, \mathbf{s}_i)^2$$

where  $\alpha_0 = \log(\lambda_0)$  and  $\alpha_1 = 1/(2\sigma^2)$ .

As usual, we approach Bayesian analysis of these models using data augmentation (Sec. 4.2). Under data augmentation, we introduce a collection of all-zero encounter histories to bring the total size of the data set up to  $M$ , and a corresponding set of data augmentation variables  $z_i \sim \text{Bern}(\psi)$ . Then the observation model is specified conditional on  $z$  according to:

$$y_{ij} \sim \text{Poisson}(z_i K \lambda_{ij})$$

which evaluates to a point mass at  $y = 0$  if  $z = 0$ . In other words, the observation model under data augmentation is a zero-inflated Poisson model which is easily analyzed by Bayesian methods, e.g., in one of the **BUGS** dialects or, alternatively, using likelihood methods, which we neglect here although the same principles as in Chapt. 6 apply.

#### 8.1.5 Simulating data and fitting the model

Simulating a sample SCR data set under the Poisson model requires only a couple minor modifications to the procedure we used in Chapt. 5 (see the function

7345 **simSCRO**). In particular, we modify the block of code which defines the model to  
 7346 be that of  $\mathbb{E}(y)$  and not  $\Pr(y = 1)$ , and we change the random variable generator  
 7347 from `rbinom` to `rpois`:

```
7348 ##  

7349 ## S =activity centers and traplocs defined as in simSCRO()  

7350 ##  

7351 ## Compute distance between activity centers and traps:  

7352 > D <- e2dist(S,traplocs)  

7353  

7354 ## Define parameter values:  

7355 > alpha0 <- -2.5  

7356 > sigma <- 0.5  

7357 > alpha1 <- 1/(2*sigma*sigma)  

7358  

7359 ## Encounter probability model:  

7360 > muy <- exp(alpha0)*exp(-alpha1*D*D)  

7361  

7362 ## Now generate the encounters of every individual in every trap  

7363 > Y <-matrix(NA,nrow=N,ncol=ntraps)  

7364 > for(i in 1:nrow(Y)){  

7365   Y[i,] <- rpois(ntraps,K*muy[i,])  

7366 }  

7367
```

7367 We modified our simulation code from Chapt. 5 to simulate Poisson encounter  
 7368 frequencies for each trap and then we analyze an ideal data set using **BUGS**. This  
 7369 Poisson simulator function `simPoissonSCR` is available in the `scrbook` package (it  
 7370 can produce 3-d encounter history data too, although we don't do that here). Here  
 7371 is an example of simulating a data set and harvesting the required data objects,  
 7372 and doing the data augmentation:

```
7373 ## Simulate data and extract data elements  

7374 ##  

7375 > data <- simPoissonSCR(discard0=TRUE,rnd=2013)  

7376 > y <- data$Y  

7377 > nind <- nrow(y)  

7378 > X <- data$traplocs  

7379 > K <- data$K  

7380 > J <- nrow(X)  

7381 > xlim <- data$xlim  

7382 > ylim <- data$ylim  

7383  

7384 ## Data augmentation  

7385 > M <- 200  

7386 > y <- rbind(y,matrix(0,nrow=M-nind,ncol=ncol(y)))  

7387 > z <- c(rep(1,nind),rep(0,M-nind))
```

7388 The process for fitting the model in **WinBUGS** or **JAGS** is identical to what  
 7389 we've done previously in Chapt. 5. In particular, we set up some starting values,

7390 package the data and inits, identify the parameters to be monitored, and then send  
 7391 everything off to our MCMC engine. Here it all is for fitting the Poisson observation  
 7392 model (these commands are shown in the help file for `simPoissonSCR`):

```

7393 ## Starting values for activity centers
7394 ##
7395 > sst <- X[sample(1:J,M,replace=TRUE),]
7396 > for(i in 1:nind){
7397   if(sum(y[i,])==0) next
7398   sst[i,1] <- mean( X[y[i,]>0,1] )
7399   sst[i,2] <- mean( X[y[i,]>0,2] )
7400 }
7401 ## Dithered a little bit from trap locations
7402 > sst <- sst + runif(nrow(sst)*2,0,1)/8
7403 > data <- list (y=y,X=X,K=K,M=M,J=J,xlim=xlim,ylim=ylim)
7404 > inits <- function(){
7405   list (alpha0=rnorm(1,-2,.4),alpha1=runif(1,1,2),s=sst,z=z,psi=.5)
7406 }
7407 > parameters <- c("alpha0","alpha1","N","D")

```

7408 Next, we write the **BUGS** model to an external file:

```

7409 > cat("
7410 model{
7411   alpha0 ~ dnorm(0,.1)
7412   alpha1 ~ dnorm(0,.1)
7413   psi ~ dunif(0,1)
7414
7415   for(i in 1:M){
7416     z[i] ~ dbern(psi)
7417     s[i,1] ~ dunif(xlim[1],xlim[2])
7418     s[i,2] ~ dunif(ylim[1],ylim[2])
7419     for(j in 1:J){
7420       d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
7421       y[i,j] ~ dpois(lam[i,j])
7422       lam[i,j] <- z[i]*K*exp(alpha0)*exp(- alpha1*d[i,j]*d[i,j])
7423     }
7424   }
7425   N <- sum(z[])
7426   D <- N/64
7427 }
7428 ",file = "SCR-Poisson.txt")

```

7429 To fit the model we execute `bugs` in the usual way:

```

7430 > library(R2WinBUGS)
7431 > out1 <- bugs (data, inits, parameters, "SCR-Poisson.txt", n.thin=1,
7432   n.chains=3,n.burnin=1000,n.iter=2000,working.dir=getwd(),
7433   debug=TRUE)

```

**Table 8.1.** 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

7434 Or, using **JAGS** via `rjags` we would do something like this:

```
7435 > library(rjags)
7436 > jm <- jags.model("SCR-Poisson.txt", data=data, inits=inits,
7437           n.chains=3, n.adapt=1000)
7438 > out2 <- coda.samples(jm, parameters, n.iter=1000, thin=1)
```

7439 Summarizing the output from the **WinBUGS** run produces the following:

```
7440 > print(out1,digits=2)
7441 Inference for Bugs model at "SCR-Poisson.txt", fit using WinBUGS,
7442 3 chains, each with 2000 iterations (first 1000 discarded)
7443 n.sims = 3000 iterations saved
7444      mean   sd  2.5%   25%   50%   75% 97.5% Rhat n.eff
7445 alpha0  -2.57 0.19 -2.95 -2.69 -2.57 -2.44 -2.19 1.00 2600
7446 alpha1   2.34 0.36  1.69  2.08  2.32  2.57  3.12 1.00 3000
7447 N       114.13 15.25 87.97 103.00 113.00 124.00 147.00 1.01 370
7448 D       1.78 0.24  1.37  1.61  1.77  1.94  2.30 1.01 370
7449 deviance 329.95 21.92 290.00 314.20 329.50 344.40 375.80 1.00 1700
7450 ...
7451 [..some output deleted..]
7452 ...
```

### 7453 8.1.6 Analysis of the wolverine study data

7454 We reanalyzed the data from the wolverine camera trapping study that were first  
 7455 introduced in Sec. 5.9. We modified the **R** script from the function `wolvSCRO` to  
 7456 fit the Poisson model (see the help file for `wolvSCROpois`). Executing this function  
 7457 produces the results shown in Table 9.1. The results are almost indistinguishable  
 7458 from the Bernoulli model fitted previously, where we had a posterior mean for  $N$   
 7459 of 59.84 and  $\sigma$  was 0.64. You can edit the script `wolvSCROpois` to obtain more  
 7460 posterior samples, or modify the model in some way.

---

### 8.1.7 Count detector models in the secr package

7462 The **R** package **secr** will fit Poisson or negative binomial encounter frequency  
 7463 models. The formatting of data and structure of the analysis proceeds in a similar  
 7464 fashion to the Bernoulli model described in Sec. 6.5, except that we specify the  
 7465 **detector**=‘‘count’’ option when the traps object is created. The set-up proceeds  
 7466 as follows:

```
7467 > library(secr)
7468 > library(scrbook)
7469 > data(wolverine)
7470
7471 > traps <- as.matrix(wolverine$wtraps)
7472 > dimnames(traps) <- list(NULL,c("trapID","x","y",paste("day",1:165,sep="")))
7473 > traps1 <- as.data.frame(traps[,1:3])
7474 > trapfile1 <- read.traps(data=traps1,detector="count")
```

7475 You can proceed with analysis of these data and compare/contrast with the  
 7476 Bayesian analysis given above, or the results of the Bernoulli model fitted in Chapt.  
 7477 6.

## 8.2 INDEPENDENT MULTINOMIAL OBSERVATIONS

7478 Several types of encounter devices yield multinomial observations in which an in-  
 7479 dividual can be caught in a single trap during a particular encounter occasion, but  
 7480 traps might catch any number of individuals. Mist netting is the canonical example  
 7481 of such a “multi-catch” device (Efford et al., 2009a). Also some kinds of bird or  
 7482 mammal cage-traps hold multiple animals, as do pit-fall traps which are commonly  
 7483 used for many species of herptiles. Another type of sample method that might be  
 7484 viewed (in some cases) as a multi-catch device are area-searches of, for example,  
 7485 reptiles where we think of a small polygon as the “trap” – we could get multiple  
 7486 individuals (turtles, lizards) in the same plot but not, in the same sample occasion,  
 7487 at different plots. The key features of this independent multinomial or multi-catch  
 7488 model are: (1) capture of an individual in a trap is *not* independent of its capture  
 7489 in other traps, because initial capture precludes capture in any other trap and (2)  
 7490 individuals behave independently of one another, so whether a trap captures some  
 7491 individual doesn’t have an affect on whether it captures another. A type of model  
 7492 in which the 2nd assumption is violated are the “single catch” trap systems which  
 7493 we address in Sec. 9.3 below.

7494 In this case we assume the observation  $\mathbf{y}_{ik}$  for individual  $i$  during sample occa-  
 7495 sion  $k$  is a multinomial observation which consists of a sequence of 0’s and a single  
 7496 1 indicating the trap of capture, or “not captured”. For the “not captured” event  
 7497 we define an additional outcome, by convention element  $J + 1$  of the vector. As  
 7498 an example, if we capture an individual in trap 2 during some occasion of a study  
 7499 involving  $J = 6$  traps. 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	trap						"not captured"
	1	2	3	4	5	6	
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

$$\mathbf{y}_{ik} | \mathbf{s}_i \sim \text{Multinomial}(1, \boldsymbol{\pi}(\mathbf{s}_i)) \quad (8.2.1)$$

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. 5.4).

It is convenient to express these multinomial models short-hand as follows, e.g., for the Gaussian encounter probability model:

$$\text{mlogit}(\pi_{ij}) = \alpha_0 - \alpha_1 d_{ij}^2$$

7529 In this way we can refer to models with covariates in a more concise way. For  
 7530 example, a model with a trap-specific covariate, say  $C_j$ , is:

$$\text{mlogit}(\pi_{ij}) = \alpha_0 - \alpha_1 d_{ij}^2 + \alpha_2 C_j$$

7531 or we could include occasion-specific covariates too, such as behavioral response.

7532 A statistically equivalent distribution to the multinomial is the *categorical* dis-  
 7533 tribution. If  $\mathbf{y}$  is a multinomial trial with probabilities  $\boldsymbol{\pi}$  than the *position* of the  
 7534 non-zero element of  $\mathbf{y}$  is a categorical random variable with probabilities  $\boldsymbol{\pi}$ . We  
 7535 express this for SCR models as

$$\mathbf{y}|\mathbf{s} \sim \text{Categorical}(\boldsymbol{\pi}(\mathbf{s}))$$

7536 In the SCR context, the categorical version of the multinomial trial corresponds to  
 7537 the *trap of capture*. Using our example above with 6 traps then we could as well say  
 7538  $y_{ik}$  is a categorical random variable with possible outcomes (1, 2, 3, 4, 5, 6, 7) where  
 7539 outcome  $y = 7$  corresponds to “not captured.” Obviously, how this is organized or  
 7540 labeled is completely irrelevant, although it is convenient to use the integers 1 to  
 7541 ( $J+1$ ) where  $J+1$  is the event not captured. Therefore, for our illustration in the  
 7542 previous table,  $y_{i1} = 2$ ,  $y_{i2} = 7$ ,  $y_{i3} = 4$  and so on.

7543 For simulating and fitting data in the **BUGS** engines we will typically use the  
 7544 categorical representation of the model because it is somewhat more convenient.  
 7545 We have found that fitting multinomial models in **WinBUGS** is less efficient than  
 7546 **JAGS** (Royle and Converse, in review), which we use in the subsequent examples  
 7547 involving multinomial observation models.

### 7548 8.2.1 Multinomial resource selection models

7549 The multinomial probabilities in Eq. 9.2.2 look similar to the multinomial resource  
 7550 selection function (RSF) model for telemetry data (Manly et al., 2002; Lele and  
 7551 Keim, 2006). This suggests how we might model landscape or habitat covariates  
 7552 using such methods – i.e., by including them as explicit covariates in a larger  
 7553 multinomial model for “use” – which, if we take the product of use with encounter,  
 7554 produces a model for the observable encounter data. This leads naturally to the  
 7555 development of models that integrate RSF data from telemetry studies with SCR  
 7556 data (Royle et al., 2012b), which is the topic of Chapt. 13.

### 7557 8.2.2 Simulating data and analysis using JAGS

7558 We’re going to show the nugget of a simulation function which is used in the function  
 7559 `simMnSCR` found in the **R** package `scrbook`. The first lines of the following **R** code  
 7560 make use of some things that you need to define, but we omit them here (e.g., `xlim`,  
 7561 `ylim` are the boundaries of the state-space, `N` is the population size, etc.):

```

7562 ##
7563 ## Simulate random activity centers:
7564 ##      (first define N, xlim, ylim, etc..)
7565 ##
7566 > S <- cbind(runif(N,xlim[1],xlim[2]),runif(N,ylim[1],ylim[2]))
7567
7568 ## Distance from each individual to each trap
7569 > D <- e2dist(S,traplocs)
7570
7571 ## Set parameter values
7572 > sigma <- 0.5
7573 > alpha0 <- -1
7574 > alpha1 <- -1/(2*sigma*sigma)
7575
7576 ## make an empty data matrix and fill it up with data
7577 > Ycat <- matrix(NA,nrow=N,ncol=K)
7578 > for(i in 1:N){
7579   for(k in 1:K){
7580     lp <- alpha0 + alpha1*D[i,]*D[i,]
7581     cp <- exp(c(lp,0))
7582     cp <- cp/sum(cp)
7583     Ycat[i,k] <- sample(1:(ntraps+1),1,prob=cp)
7584   }
7585 }
```

7586 We save the data in the matrix `Ycat` to clarify that it is the categorical observation representing “trap of capture”. The matrix `Ycat` here has the maximal dimension  $N$  and so, to do an analysis that mimics a real situation, we would have to discard the uncaptured individuals. The function `simMnSCR` in the package `scrbook` will also simulate data that includes a behavioral response which will be the typical situation in small-mammal trapping problems (see Converse and Royle, 2012, for details).

7587 Here we use our function `simMnSCR` to simulate a data set with  $K = 7$  occasions.  
7588 We'll run the model using JAGS which we have found is much more effective for this  
7589 class of models. We get the data set-up for analysis by augmenting the size of  
7590 the data set to  $M = 200$ . In addition we choose starting values for `s` and the  
7591 data augmentation variables `z`. For starting values of `s` we cheat a little bit here  
7592 and use the true values for the observed individuals and then augment the  $M \times 2$   
7593 matrix `S` with  $M - n$  randomly selected activity centers. Our function `spiderplot`  
7594 returns the mean observed location of individuals for use as starting values for the  
7595 `nind` encountered individuals. The parameters input to `simMnSCR` are the intercept  
7596  $\alpha_0$ ,  $\sigma = \sqrt{1/(2\alpha_1)}$  for the Gaussian encounter probability model, and  $\alpha_2$  is the  
7597 behavioral response parameter. The data simulation and set-up proceeds as follows:

```

7604 > set.seed(2013)
7605 > parms <- list(N=100,alpha0= -.40, sigma=0.5, alpha2= 0)
7606 > data <- simMnSCR(parms, K=7, ssbuff=2)
```

```

7607 > nind <- nrow(data$Ycat)
7608
7609 > M <- 200
7610 > Ycat <- rbind(data$Ycat,matrix(nrow(data$X)+1,nrow=(M-nind),ncol=data$K))
7611 > Sst <- rbind(data$S,cbind(runif(M-nind,data$xlim[1],data$xlim[2]),
7612                               runif(M-nind,data$ylim[1],data$ylim[2])))
7613 > zst <- c(rep(1,160),rep(0,40))

```

7614     The model specification is not much more complicated than the binomial or  
7615     Poisson models given previously. The main consideration is that we define the cell  
7616     probabilities for each trap  $j = 1, 2, \dots, J$  and then define the last cell probability,  
7617      $J + 1$ , for “not captured”, to be the complement of the sum of the others. The code  
7618     is shown in Panel 9.1. In the last lines of code here we specify  $N$  and density,  $D$ ,  
7619     as derived parameters.

7620     To fit the model, we need to package everything up (inits, parameters, data)  
7621     and send it off to **JAGS** to build an MCMC simulator for us (these commands are  
7622     executed in the help file for `simMnSCR`). In addition to the usual data objects, we  
7623     also pass the limits of the assumed rectangular state-space (`ylim`, `xlim`, both  $1 \times 2$   
7624     vectors) and the scale of the standardized units, called `trap.space` here because  
7625     we typically will define the trap coordinates to be an integer grid. If the trap  
7626     spacing is 10 m and we want units of density computed in terms of individuals  
7627     per meter-squared, then we input `trap.space=10`. The analysis is carried out as  
7628     follows:

```

7629 > inits <- function(){ list (z=zst,sigma=runif(1,.5,1) ,S=Sst) }
7630
7631 # Parameters to monitor
7632 > parameters <- c("psi","alpha0","alpha1","sigma","N","D")
7633
7634 # Bundle the data. Note this reuses "data"
7635 > data <- list (X=data$X,K=data$K, trap.space=1,Ycat=Ycat,M=M,
7636   ntraps=nrow(data$X),ylim=data$ylim,xlim=data$xlim)
7637
7638 > library(R2jags)
7639 > out <- jags (data, inits, parameters, "model.txt", n.thin=1,
7640   n.chains=3, n.burnin=1000, n.iter=2000)

```

7641     The posterior summaries are provided in the following **R** output (recall that  
7642      $N = 100$ ,  $\alpha_0 = -.40$ , and  $\sigma = 0.5$ ):

```

7643 > out
7644 Inference for Bugs model at "model.txt", fit using jags,
7645   3 chains, each with 2000 iterations (first 1000 discarded)
7646   n.sims = 3000 iterations saved
7647   mu.vect sd.vect    2.5%     25%     50%     75%   97.5% Rhat n.eff
7648 D       1.873  0.189   1.531   1.750   1.859   2.000   2.250 1.006  1300
7649 N      119.867 12.107  98.000 112.000 119.000 128.000 144.000 1.006  1300

```

---

```

model{
psi ~ dunif(0,1)
alpha0 ~ dnorm(0,10)
sigma ~ dunif(0,10)
alpha1 <- 1/(2*sigma*sigma)

for(i in 1:M){
z[i] ~ dbern(psi)
S[i,1] ~ dunif(xlim[1],xlim[2])
S[i,2] ~ dunif(ylim[1],ylim[2])
for(j in 1:ntraps){
#distance from capture to the center of the home range
d[i,j] <- pow(pow(S[i,1]-X[j,1],2) + pow(S[i,2]-X[j,2],2),1)
}
for(k in 1:K){
for(j in 1:ntraps){
lp[i,k,j] <- exp(alpha0 - alpha1*d[i,j])*z[i]
cp[i,k,j] <- lp[i,k,j]/(1+sum(lp[i,k,]))
}
cp[i,k,ntraps+1] <- 1-sum(cp[i,k,1:ntraps]) # last cell = not captured
Ycat[i,k] ~ dcat(cp[i,k,])
}
}

N <- sum(z[1:M])
A <- ((xlim[2]-xlim[1])*trap.space)*((ylim[2]-ylim[1])*trap.space)
D <- N/A
}

```

---

Panel 8.1: **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`.

---

```

7650 alpha0    -0.435   0.151  -0.738  -0.535  -0.439  -0.331  -0.146  1.004   580
7651 alpha1     2.195   0.286   1.658   2.004   2.180   2.372   2.785  1.003   2400
7652 psi        0.599   0.069   0.465   0.552   0.599   0.645   0.739  1.006   1400
7653 sigma      0.480   0.032   0.424   0.459   0.479   0.500   0.549  1.003   2400
7654 deviance  892.164 21.988  850.922  877.417  891.561  906.246  937.728 1.003   950
7655
7656 [... output deleted ...]

```

### 7657 8.2.3 Multinomial relationship to the Poisson

7658 The multinomial is related to the Poisson encounter rate model by a conditioning  
 7659 argument. Let  $y_{ij}$  be the number of encounters for individual  $i$  in trap  $j$ . If  
 7660  $y_{ij} \sim \text{Poisson}(\lambda_{ij})$ , then, conditional on the *total* number of captures (i.e., across  
 7661 all traps),  $y_i = \sum_j y_{ij}$ , the trap encounter frequencies are multinomial with prob-  
 7662 abilities

$$\pi_{ij} = \frac{\lambda_{ij}}{\sum_j \lambda_{ij}}$$

7663 for  $j = 1, 2, \dots, J$ . Or equivalently the *trap of capture* is categorical with probabili-  
 7664 ties  $\pi_{ij}$  as given above. Under the Gaussian kernel model, these probabilities are:

$$\pi_{ij} = \frac{\exp(-\alpha_1 d(\mathbf{x}_i, \mathbf{s}_i)^2)}{\sum_j \exp(-\alpha_1 d(\mathbf{x}_i, \mathbf{s}_i)^2)} \quad (8.2.2)$$

7665 where, we note, the intercept  $\alpha_0$  has canceled from both the numerator and denom-  
 7666 inator. This makes sense because, here, these probabilities describe the trap-specific  
 7667 capture probabilities *conditional on capture*. Therefore, the model is not completely  
 7668 specified, absent a model for the “overall” probability of encounter or the expected  
 7669 frequency of captures, say  $\phi_i$ . Depending on how we specify a model for this quan-  
 7670 tity  $\phi_i$ , we can reconcile it directly with the Poisson model. Let  $y_i$  be the total  
 7671 number of encounters for individual  $i$  and suppose  $y_i$  has a Poisson distribution  
 7672 with mean  $\phi_i$ . Then, marginalizing Eq. 9.2.1 over the Poisson distribution for  $y_i$   
 7673 produces the original set of *iid* Poisson frequencies with probabilities:

$$\lambda_{ij} = \phi_i \pi_{ij}$$

7674 for  $j = 1, 2, \dots, J$ . In particular, if we suppose that  $\phi_i = \sum_j \exp(\alpha_0 - \alpha_1 d(\mathbf{x}, \mathbf{s})^2)$   
 7675 then the marginal distribution of  $y_{ij}$  is Poisson with mean  $\exp(\alpha_0 - \alpha_1 d(\mathbf{x}, \mathbf{s})^2)$ ,  
 7676 equivalent to Eq. 9.1.1.

7677 In summary, the Poisson and multinomial models are equivalent in how they  
 7678 model the distribution of captures among traps. It stands to reason that, if the  
 7679 encounter rate of individuals is low, we could use the Poisson and multinomial  
 7680 models interchangeably. In fact, based on our discussion in Sec. 9.1.2 above we  
 7681 could use any of the binomial/Poisson/multinomial models with little ill-effect when  
 7682 encounter rate is low.

**8.2.4 Avian mist-netting example**

We analyze data from a mist-netting study of ovenbirds, conducted at the Patuxent Wildlife Research Center, Laurel MD, by D.K. Dawson and M.G. Efford. The data from this study are available in the **secr** package, and have been analyzed previously by Efford et al. (2004), see also Borchers and Efford (2008). Forty-four mist nets spaced 30 m apart on the perimeter of a 600-m x 100-m rectangle were operated on 9 or 10 non-consecutive days in late May and June for 5 years from 2005-2009. The ovenbird data can be loaded as follows:

```
> library(secr)  
> data(ovenbird)
```

The data set consists of adult ovenbirds caught during sampling in each of 5 years, 2005-2009. (one ovenbird was killed in 2009, indicated by a negative net number in the encounter data file). As with most mist-netting studies, nets are checked multiple times during a 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. 16.

**Multiple sample sessions**

Up to this point we have only dealt with a basic closed population sampling situation consisting of repeated sample occasions on a single population of individuals using a single array of traps. In practice, many studies produce repeated samples over longer periods of time over which demographic closure isn’t valid, or at different locations where the populations are completely distinct. 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. 14). 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. 16 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.

When the sessions represent explicit time periods, the multi-session model of **secr** can be thought of as a type of open population model. In particular, 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 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. 14 (Converse and Royle, 2012; Royle and Converse, in review).

A 3rd way to develop models for stratified or grouped populations, not based on multi-session models, but that is convenient in **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 to distinguish it from the multi-session model which assumes the  $N_t$  are related to one another by having been generated from a common Poisson distribution. 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  $N_t$ <sup>1</sup>

We demonstrate these 3 approaches to analyzing grouped/stratified data using the ovenbird data: (1) In the following section, we provide the nonparametric multi-session model with unconstrained  $N_t$ ; (2) we demonstrate the Poisson model-based multi-session models from **secr** both here (following section) and in Chapt. 14 from a Bayesian standpoint; (3) later, in Chapt. 16, we provide a fully dynamic “spatial Jolly-Seber” model and apply it to the ovenbird data.

### Analysis in **JAGS**

The ovenbird data are provided as a multi-session **capthist** object **ovenCH** which, by regarding years as independent strata, or sessions, allows for the fitting of the multi-session model. For doing a Bayesian analysis in one of the **BUGS** engines (we use **JAGS** here) there are a number of ways to structure the data and describe the model. We can analyze either a 2-d data set with all years (data augmented) “stacked” into a data set of dimension  $(5 * M) \times 10$  (5 years,  $M$  = size of the augmented data set,  $K = 10$  replicate sample occasions). Or, we could produce a 3-d array  $(M \times J \times K)$ . We adopted the former approach, analyzing the data as a 2-d array and creating an additional categorical variable for “year” to indicate which stratum (year) each record goes with.

Data on individual sex is included with **secr**, but we provide an analysis of a

---

<sup>1</sup>We do not know of **secr** documentation that states this (or contradicts it). We think this is what is being done, based partially on conversations or emails with M.G. Efford, D.L. Borchers, the various publications on **secr**, and our own thinking about it.

**Table 8.2.** Posterior summary statistics for the ovenbird mist-netting data based on the independent multinomial (“multi-catch”) encounter process model. Parameters  $\psi$ ,  $N$  and  $D$  are indexed by year. MCMC was done using jags with 3 chains, each with 5000 iterations, discarding the first 1000, for a total of 12000 posterior samples.

parameter	Mean	SD	2.5%	50%	97.5%	Rhat	n.eff
D[1]	1.000	0.202	0.682	0.971	1.446	1.009	250
D[2]	1.041	0.192	0.723	1.033	1.467	1.010	230
D[3]	1.212	0.208	0.868	1.198	1.653	1.004	580
D[4]	0.895	0.173	0.599	0.868	1.281	1.012	190
D[5]	0.760	0.155	0.496	0.744	1.095	1.002	1700
$\alpha_0$	-3.488	0.159	-3.801	-3.490	-3.171	1.005	450
$\alpha_1$	0.000	0.000	0.000	0.000	0.000	1.018	120
$\sigma$	76.106	6.061	65.620	75.709	89.205	1.018	120
$\psi[1]$	0.484	0.108	0.299	0.475	0.715	1.008	300
$\psi[2]$	0.505	0.103	0.325	0.497	0.725	1.008	270
$\psi[3]$	0.585	0.110	0.388	0.578	0.813	1.004	710
$\psi[4]$	0.434	0.095	0.264	0.428	0.639	1.010	210
$\psi[5]$	0.370	0.087	0.215	0.364	0.556	1.002	2000
N[1]	48.396	9.762	33.000	47.000	70.000	1.009	250
N[2]	50.403	9.290	35.000	50.000	71.000	1.010	230
N[3]	58.687	10.049	42.000	58.000	80.000	1.004	580
N[4]	43.341	8.366	29.000	42.000	62.000	1.012	190
N[5]	36.792	7.502	24.000	36.000	53.000	1.002	1700

single model for all adults, constant  $\sigma$  across years, constant  $p_0$ , and year-specific values of  $N_t$  (and hence  $D_t$ ). There is a habitat mask provided with the data but the mask appears to just be a modified rectangle around the net locations, clipped to have rounded corners, and so we don’t use it here. Instead, we used a rectangular state-space buffer of 200 meters for our analysis. There was a single loss-on-capture which we accounted for by fixing  $p = 0$  for all subsequent encounters of that individual (indicated by the binary variable `dead`, as shown in Panel 9.2). We have an **R** script in `scrbook` package called `SCRovenbird`, so you can see how to set-up the data and run the model. Executing the script `SCRovenbird` produces the posterior summaries given in Table 9.2. Here, density is in units of birds per ha. The posterior mean of  $\sigma$  is about 76 meters, and there is considerable variability in density over the 5 year period with density peaking at 1.2 birds/ha in year 3, although there is considerable posterior uncertainty. The R-hat’s look a little bit peaked and so we might consider running the MCMC analysis longer.

#### Analysis in `secr`

Included with the ovenbird data are a number of models fitted as examples. Those include:

```

model{
  alpha0 ~ dnorm(0,.1)
  sigma ~ dunif(0,200)
  alpha1 <- 1/(2*sigma*sigma)

  A <- ((xlim[2]-xlim[1]))*((ylim[2]-ylim[1]))
  for(t in 1:5){
    N[t] <- inprod(z[1:bigM],yrdummy[,t])
    D[t] <- (N[t]/A)*10000 # Put in units of per ha
    psi[t] ~ dunif(0,1)
  }

  for(i in 1:bigM){ # bigM = total size of jointly augmented data set
    z[i] ~ dbern(psi[year[i]])
    S[i,1] ~ dunif(xlim[1],xlim[2])
    S[i,2] ~ dunif(ylim[1],ylim[2])

    for(j in 1:ntraps){ # X = trap locations, S = activity centers
      d2[i,j] <- pow(pow(S[i,1]-X[j,1],2) + pow(S[i,2]-X[j,2],2),1)
    }
    for(k in 1:K){
      Ycat[i,k] ~ dcat(cp[i,k,])
      for(j in 1:ntraps){
        lp[i,k,j] <- exp(alpha0 - alpha1*d2[i,j])*z[i]*(1-dead[i,k])
        cp[i,k,j] <- lp[i,k,j]/(1+sum(lp[i,k,1:ntraps]))
      }
      cp[i,k,ntraps+1] <- 1-sum(cp[i,k,1:ntraps]) # Last cell = not captured
    }
  }
}

```

Panel 8.2: **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`.

```

7781 ovenbird.model.1    fitted secr model -- null
7782 ovenbird.model.1b   fitted secr model -- g0 net shyness
7783 ovenbird.model.1T   fitted secr model -- g0 time trend within years
7784 ovenbird.model.h2   fitted secr model -- g0 finite mixture
7785 ovenbird.model.D    fitted secr model -- trend in density across years

```

7786 The model fit objects provided in **secr** are based on the use of the habitat  
 7787 mask. To make the analyses consistent with our previous analysis in **JAGS**, we  
 7788 refit all of the models here without the habitat mask. The re-analysis proceeds as  
 7789 follows, changing the “trend in density across years” model to allow for year-specific  
 7790 density:

```

7791 ## Fit constant-density model
7792 > ovenbird.model.1 <- secr.fit(ovenCH)
7793 ## Fit net avoidance model
7794 > ovenbird.model.1b <- secr.fit(ovenCH, model = list(g0 ~ b))
7795 ## Fit model with time trend in detection
7796 > ovenbird.model.1T <- secr.fit(ovenCH, model = list(g0 ~ T))
7797 ## Fit model with 2-class mixture for g0
7798 > ovenbird.model.h2 <- secr.fit(ovenCH, model = list(g0 ~ h2))
7799 ## Fit a model with session (year)-specific Density
7800 > ovenbird.model.DT <- secr.fit(ovenCH, model = list(D ~ session))

```

7801 All of these can be fitted easily in **JAGS** but the model we fitted previously  
 7802 is roughly equivalent to the last model, **ovenbird.model.DT**, because we allowed  
 7803 for year-specific population sizes (and hence density). So, we’ll compare our results  
 7804 from **JAGS** to that model. The **secr** output is extensive and so we do not repro-  
 7805 duce it completely here. By default, it summarizes the trap information for each  
 7806 year, encounter information, and then output for each year. Here is an abbreviated  
 7807 version for **ovenbird.model.DT**:

```

7808 > print(ovenbird.model.DT,digits=2)
7809
7810 secr.fit( capthist = ovenCH, model = list(D ~ session), buffer = 300 )
7811 secr 2.3.1, 14:46:52 23 Jan 2013
7812
7813 $‘2005‘
7814 Object class      traps
7815 Detector type    multi
7816 Detector number   44
7817 Average spacing   30.27273 m
7818 x-range           -50 49 m
7819 y-range           -285 285 m
7820
7821 [... deleted ...]
7822
7823          2005 2006 2007 2008 2009
7824 Occasions       9   10   10   10   10

```

```

7825 Detections   35   42   52   30   33
7826 Animals      20   22   26   19   16
7827 Detectors    44   44   44   44   44
7828
7829 Model          : D~session g0~1 sigma~1
7830 Fixed (real)   : none
7831 Detection fn   : halfnormal
7832 Distribution    : poisson
7833 N parameters   : 7
7834 Log likelihood  : -1119.845
7835 AIC            : 2253.689
7836 AICc           : 2254.868
7837
7838 [... deleted ...]

```

7839 To do model selection we use the handy helper-function `AIC` as follows (output  
 7840 edited to fit on the page):

```

7841 AIC (ovenbird.model.1, ovenbird.model.1b, ovenbird.model.1T,
7842           ovenbird.model.h2, ovenbird.model.DT)
7843
7844             model detectfn npar logLik     AIC     AICc     dAICc
7845 ovenbird.model.1T [edited output] 4 -1111.850 2231.700 2232.109 0.000
7846 ovenbird.model.1b       ....      4 -1117.615 2243.229 2243.637 11.528
7847 ovenbird.model.h2       ....      3 -1121.164 2248.327 2248.570 16.461
7848 ovenbird.model.1        ....      5 -1119.762 2249.524 2250.143 18.034
7849 ovenbird.model.DT       ....      7 -1119.845 2253.689 2254.868 22.759

```

7850 We see that our DT model is way down at the bottom of the list. Instead,  
 7851 the model with a time-trend (within-season) in detection probability is preferred,  
 7852 followed by a behavioral response. We encourage you to adapt the **JAGS** model  
 7853 specification for such models which is easily done (see Chapt. 7 for many examples).  
 7854 We provide the summary results for the model having  $D \sim \text{session}$  as follows:

```

7855 > print(ovenbird.model.DT,digits=2)
7856
7857 secr.fit( capthist = ovenCH, model = list(D ~ session), buffer = 300 )
7858 secr 2.3.1, 14:46:52 23 Jan 2013
7859
7860 [...deleted....]
7861
7862 Fitted (real) parameters evaluated at base levels of covariates
7863
7864 session = 2005
7865       link estimate SE.estimate     lcl     ucl
7866 D      log      0.920      0.228  0.571  1.484
7867 g0     logit     0.028      0.004  0.021  0.037
7868 sigma   log     78.566      6.379 67.025 92.095

```

```

7869
7870 session = 2006
7871      link estimate SE.estimate    lcl    ucl
7872 D      log     0.963      0.238  0.598  1.553
7873 g0     logit    0.028      0.004  0.021  0.037
7874 sigma   log    78.566      6.379 67.025 92.095
7875
7876 session = 2007
7877      link estimate SE.estimate    lcl    ucl
7878 D      log     1.139      0.282  0.706  1.836
7879 g0     logit    0.028      0.004  0.021  0.037
7880 sigma   log    78.566      6.379 67.025 92.095
7881
7882 session = 2008
7883      link estimate SE.estimate    lcl    ucl
7884 D      log     0.832      0.206  0.516  1.341
7885 g0     logit    0.028      0.004  0.021  0.037
7886 sigma   log    78.566      6.379 67.025 92.095
7887
7888 session = 2009
7889      link estimate SE.estimate    lcl    ucl
7890 D      log     0.701      0.173  0.435  1.130
7891 g0     logit    0.028      0.004  0.021  0.037
7892 sigma   log    78.566      6.379 67.025 92.095

```

7893     The point estimates (MLEs) of density are uniformly lower than the Bayesian  
7894 estimates (posterior means) shown in Table 9.2. We expect some difference in this  
7895 direction due to small-sample skew of the posterior. In addition, there may be  
7896 slight differences due to the fact that **secr** multi-session model assumes that the  
7897  $N_t$  have a Poisson prior, but the implementation in **JAGS** using data augmentation  
7898 is based on a binomial prior. The estimated  $\sigma$  is very similar between the **JAGS**  
7899 analysis and **secr**.

### 8.3 SINGLE-CATCH TRAPS

7900     The classical animal trapping experiment is based on a physical trap which captures  
7901 a single animal and holds that individual until subsequent molestation by a biol-  
7902 ogist. This type of observation model – the “single-catch” trap – was the original  
7903 situation considered in the context of spatial capture-recapture by Efford (2004).  
7904 Nowadays, capture-recapture data are more often obtained by other methods (DNA  
7905 from hair snares, or scat sampling, camera traps etc...) but nevertheless the single-  
7906 catch traps are still widely used in small mammal studies (Converse et al., 2006;  
7907 Converse and Royle, 2012) and other situations.

7908     The single-catch model is basically a multinomial model but one in which the  
7909 number of available traps is reduced as each individual is captured. As such, the

7910 constraints on the joint likelihood for the sample of  $n$  encounter histories are very  
 7911 complicated. As a result, at the time of this writing, there has not been a formal  
 7912 development of either likelihood or Bayesian analysis of this model and applications  
 7913 of SCR models to single-catch systems have used the independent multinomial  
 7914 model as an approximation (see below).

7915 Nevertheless, we can make some progress to describing the basic observation  
 7916 model formally. In particular, if we imagine that all of the individuals captured  
 7917 queued up at the beginning of the capture session to draw a number indicating  
 7918 their order of capture, then there is a nice conditional structure resulting from a  
 7919 “removal process” operating on the traps. The first individual captured has the  
 7920 multinomial observation model:

$$\mathbf{y}_1 \sim \text{Multinomial}(\boldsymbol{\pi}_1)$$

7921 whereas the 2nd individual captured also has a multinomial encounter probability  
 7922 model but with the trap which captured the first individual removed. We might  
 7923 express this as:

$$\mathbf{y}_2 \sim \text{Multinomial}(\boldsymbol{\pi}_2)$$

7924 where

$$\pi_{2j} = \frac{(1 - y_{1j}) * \exp(\alpha_0 - \alpha_1 d_{ij}^2)}{\sum_j (1 - y_{1j}) * \exp(\alpha_0 - \alpha_1 d_{ij}^2)}$$

7925 and so on for  $i = 3, 4, \dots, n$ . In a certain way, this model is a type of local behav-  
 7926 ior response model but where the response is to other individuals being captured.  
 7927 Evidently, the **order of capture** is relevant to the construction of these multi-  
 7928 nomial cell probabilities. More generally, the *time* of capture of an individual in  
 7929 any trapping interval will affect the encounter probability of subsequently captured  
 7930 individuals, but we think that order of capture might lead to a practical approxi-  
 7931 mation to the single-catch process (this is how we simulate the data in our function  
 7932 `simScSCR`). In the simulation of single catch data, we randomly ordered the popula-  
 7933 tion of individuals for each sample occasion, and then cycled through them, turning  
 7934 off each trap if an individual was captured in it.

### 7935 8.3.1 Inference for single-catch systems

7936 For the single-catch model, we argued that the observations have a multinomial  
 7937 type of observation model, but the multinomial observations have a unique condi-  
 7938 tional dependence structure among them owing to the “removal” of traps as they  
 7939 fill-up with individuals. Thus, competition for single-catch traps renders the inde-  
 7940 pendence assumptions for the independent multinomial model invalid. However, as  
 7941 Efford et al. (2009a) noted, we expect “bias to be small when trap saturation (the  
 7942 proportion of traps occupied) is low. Trap saturation will be higher when popula-  
 7943 tion density is high...” relative to trap density, or when net encounter probability

7944 is high. Efford et al. (2009a) did a limited simulation study and found essentially  
 7945 no effective bias and concluded that estimators of density from the misspecified  
 7946 independent multinomial model are robust to the mild dependence induced when  
 7947 trap saturation is low. Naturally then, we expect that the Poisson model could also  
 7948 be an effective approximation under the same set of circumstances.

7949 In the **R** package **scrbook** we provide a function for simulating data from a  
 7950 single-catch system (function **simScSCR**) and fitting the misspecified model (**example(simScSCR)**)  
 7951 in **JAGS** so that you can evaluate the effectiveness of this misspecified model for  
 7952 situations that interest you.

### 7953 8.3.2 Analysis of Efford's possum trapping data

7954 We provide an analysis here of data from a study of brushtail possums in New  
 7955 Zealand. The data are available with the **R** package **secr** (Efford et al., 2009a); see  
 7956 the help file **?possum** after loading the **secr** package. Originally the data were an-  
 7957 alyzed by Efford et al. (2005), and a detailed description of the data set is available  
 7958 in the help file, from which we summarize:

7959     *Brushtail possums (*Trichosurus vulpecula*) are an unwanted invasive species in New  
 7960     Zealand. Although most abundant in forests, where they occasionally exceed densities  
 7961     of 15/ha, possums live wherever there are palatable food plants and shelter.*

7962 To load the possum data, execute the following commands:

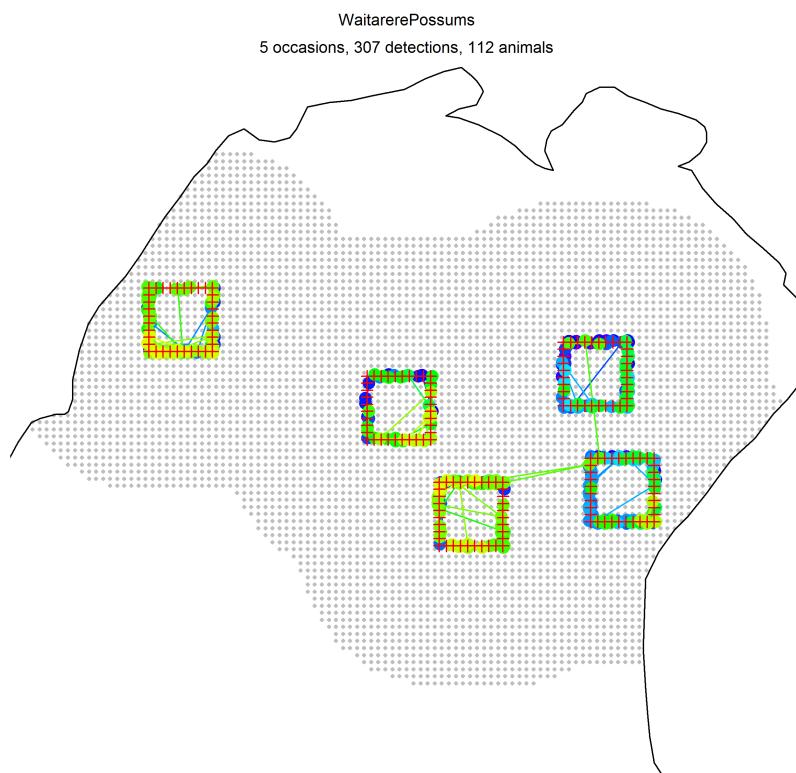
```
7963 > library(secr)
7964 > data(possum)
```

7965 The study area encompasses approximately 300 ha, and 180 live traps were orga-  
 7966 nized in 5 distinct grids, shown in Fig. 9.2. Each square arrangement of traps  
 7967 consisted of 36 traps with a spacing of 20 m. Thus the squares are 180 m on a side.  
 7968 Individuals were captured, tagged, and released over 5 days during April, 2002. A  
 7969 noteworthy aspect of this study is that it involves replicated grids selected in some  
 7970 fashion from within a prescribed region. From an analysis standpoint, we could  
 7971 adopt the use of the multi-session models which we used previously to analyze the  
 7972 ovenbird data. This would be useful if we had covariates at the trapping grid level  
 7973 that we wanted to model. Alternatively, we could pool the data from all of the grids  
 7974 and analyze them jointly as if they were based on a single trapping grid (with 180  
 7975 traps) which is clearly a reasonable view in this case. In doing this sort of pooling,  
 7976 there is an implicit assumption that  $N_t$  ( $t$  indexing trapping grid in this case) is  
 7977 Poisson distributed, with constant mean (Royle, 2004a; Royle et al., 2012c) which  
 7978 we also address in Chapt. 14.

7979 The data file **possumCH** contains 112 encounter histories, and we analyze those  
 7980 here although the last 8 of those are recaptures treated as new individuals<sup>2</sup>. The en-  
 7981 counter process is not strictly a single-catch multinomial process because, as noted

---

<sup>2</sup>M. Efford, personal communication



**Figure 8.2.** 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.

in the `possum` help file “One female possum was twice captured at two sites on one day, having entered a second trap after being released; one record in each pair was selected arbitrarily and discarded.” which is a similar situation to what might happen in bird mist net studies, as a bird might fly into a net upon release from another. By discarding the two extra-capture events, we can satisfactorily view these data as single-catch data, for which `secr` uses the independent multinomial likelihood (M. Efford, pers. comm.). If multiple, same-session captures were common, then it might be worth developing a model for  $n_{ik}$  = the number of captures of individual  $i$  during sample occasion  $k$ , in order to make use of all captures.

For our Bayesian analysis here, we used a rectangular state-space which doesn’t account for any geographic boundaries of the survey region, but we note that a habitat mask is included in `secr` and it could be used in a Bayesian analysis. Whether or not we use the mask is probably immaterial as long as we understand the predictions of  $N$  or  $D$  over the water don’t mean anything biological and we probably wouldn’t report such predictions. The **JAGS** model specification is based on that of the ovenbird analysis given previously, and so we don’t reproduce the model here. The **R/JAGS** script is called `SCRpossum`, which is in the `scrbook` package. The results are summarized in Table 9.3.

**Table 8.3.** Results of fitting the independent multinomial observation model to the possum trapping data. Strictly speaking, the trapping device is a “single-catch” trap, and the model represents an intentional misspecification. Density is reported in individuals per ha ( $Dha$ ). Posterior summaries were obtained using **JAGS** with 3 chains, each with 2000 iterations, discarding the first 1000 as burn-in, to yield a total of 3000 posterior samples.

Parameter	Mean	SD	2.5%	50%	97.5%	Rhat	n.eff
N	235.407	17.435	204.000	235.000	270.000	1.009	340
Dha	1.549	0.115	1.343	1.547	1.777	1.009	340
$\alpha_0$	-0.935	0.167	-1.270	-0.934	-0.605	1.007	870
$\alpha_1$	0.000	0.000	0.000	0.000	0.000	1.001	2800
$\psi$	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

The estimated density (posterior mean) is about 1.53 possums/ha. To obtain the `secr` results for the equivalent null model, we execute the following command

```
8002 > secr.fit( capthist = possumCH, trace = F )
8003 which produces (edited) summary output:
8004 [... some output deleted ...]
8005
8006 Fitted (real) parameters evaluated at base levels of covariates
8007      link   estimate SE.estimate      lcl       ucl
8008 D      log    1.6988930  0.17352645  1.3913904  2.0743547
```

---

```

8009 g0    logit  0.1968542  0.02256272  0.1563319  0.2448321
8010 sigma  log 51.4689114  2.59981905  46.6204139  56.8216500
8011
8012 [... some output deleted ...]

```

8013 As we've discussed previously, there are many reasons for why there might be  
8014 differences between Bayesian and likelihood estimates. But even among likelihood  
8015 estimates – any time you run a model there is some numerical integration going  
8016 on which requires some specific choices of how to do the integration (see Chapt.  
8017 6). For now we just observe that the estimated density is certainly in the ballpark  
8018 (compared to those in Table. 9.3), and so too is the estimated  $\sigma$ .

## 8.4 ACOUSTIC SAMPLING

8019 The last decade has seen an explosion of technology that benefits the study of animal  
8020 populations. This includes DNA sampling methods that allow for identification  
8021 from hair or scat, camera trapping and identification software that allow efficient  
8022 sampling of many mammals, and the resulting statistical technology that helps us  
8023 to make sense of such data (Borchers and Efford, 2008; Royle and Young, 2008;  
8024 Efford et al., 2009b; Gopalaswamy et al., 2012b; Sollmann et al., 2012; Chandler  
8025 and Royle, In press). One other extremely promising technology area is that of  
8026 acoustic sampling using microphones or recording devices. That is, instead of having  
8027 cameras record encounters, or humans pick up scat, we can establish an array of  
8028 (usually) electronic recording devices which, instead of establishing a visual identity  
8029 of individuals, record a vocal expression of each individual. In this context, Efford  
8030 et al. (2009b) referred to audio recorders as “signal strength proximity detectors”  
8031 to distinguish them from other types of proximity detections, including camera  
8032 traps, which are *visual* proximity detector. Using audio records, the spatial pattern  
8033 of the *signal strength* at the different audio recorders or microphones can be used  
8034 for inference about density (Dawson and Efford, 2009; Efford et al., 2009b) in the  
8035 same way as the spatial pattern of detections is used in the types of SCR models  
8036 we have discussed so far. The basic technical formulation of these models comes  
8037 from Efford et al. (2009b), and it was applied to field study of birds by Dawson  
8038 and Efford (2009). In that study, recording devices were organized in groups of 4  
8039 (in a square pattern), with an array of  $5 \times 15$  such clusters of 4, separated by 100  
8040 m (300 total recorder locations). This data set, called `signalCH`, is provided with  
8041 the `secr` package along with some sample analyses and help files. See Efford and  
8042 Dawson (2010), a version of the document `secr-sound.pdf` (that also comes with  
8043 the `secr` package) which you can access directly from the main help file (`?secr`).

8044 Our development here mostly follows Efford et al. (2009b), but we change some  
8045 notation to be consistent with our previous material. Let  $S(\mathbf{x}, \mathbf{u})$  be the strength  
8046 of a signal emanating from signal location  $\mathbf{u}$ , as recorded by a device at location  
8047  $\mathbf{x}$ . Just as ordinary SCR models represent a model of *encounter frequency* as a

8048 function of distance, in acoustic models, the acoustic SCR model is a model of  
 8049 sound attenuation as a function of distance. In particular, the acoustic models  
 8050 assumes that  $S$  (or a suitable transformation) declines with distance  $d$  from the  
 8051 origin of the sound, to the recording device. In the context of spatial sampling  
 8052 of animals, the origin is the actual location of some individual animal, and the  
 8053 recording device is something we nailed to a tree, or mounted on a post. For  
 8054 example, a model of sound attenuation used by Dawson and Efford (2009) is the  
 8055 following:

$$S(\mathbf{x}, \mathbf{u}) = \alpha_0 + \alpha_1 d(\mathbf{x}, \mathbf{u}) + \epsilon \quad (8.4.1)$$

8056 where  $\epsilon \sim \text{Normal}(0, \sigma_s^2)$ . In many standard situations,  $S$  will be measured in  
 8057 decibels, which can be any value on the real line. In the conduct of acoustic sampling  
 8058 and the development of custom models for your own situation, it would probably  
 8059 be helpful to know something about sound dynamics and signal processing. In this  
 8060 model, the parameters  $\alpha_0$ ,  $\alpha_1$  and  $\sigma_s^2$  are to be estimated. We abbreviate the set  
 8061 of parameters by  $\theta$  for short.

8062 The basic structure of an acoustic SCR study is not really much different from  
 8063 ordinary SCR studies. Just as ordinary SCR models require that individuals be  
 8064 encountered at  $> 1$  trap, these acoustic models require that individuals be heard  
 8065 at  $> 1$  recorder. Therefore, the acoustic signals (calls or vocalizations) must be  
 8066 reconcilable and, in fact, reconciled successfully by the investigator. In practice,  
 8067 this would require associating signals that occur at the same instant with the same  
 8068 individual (or making a decision one way or the other). Further, if individuals are  
 8069 actively moving during the sample period (that recorders are functioning) then in-  
 8070 dividuals might be double-counted, thereby biasing estimates of density. In general,  
 8071 the models produce an estimate of density of *sources*, and how that is interpreted  
 8072 depends on whether individuals are stationary or mobile, and other things. In par-  
 8073 ticular, if multiple survey occasions are used (e.g., on different days), then modeling  
 8074 movement of individuals would be essential in order to interpret estimates of den-  
 8075 sity meaningfully. Models that allow some movement should be possible (see Sec.  
 8076 9.4.3 below, and Chaps. 15 and 16).

#### 8077 8.4.1 The signal strength model

8078 We assert that an individual is detected if  $S$  exceeds a threshold,  $c$ . The reason  
 8079 for introducing this threshold  $c$  is that sound recorders will always record some  
 8080 background sound, and so effective use of the acoustic SCR models requires spec-  
 8081 ification of the threshold of measured signal below which the record is censored  
 8082 (non-detection occurs) because the recorded sound is assumed to be background  
 8083 noise. So we assert that an individual is detected if  $S > c$  which occurs with prob-  
 8084 ability  $\Pr(S > c)$ , the encounter probability. To expand on and formalize this, let  
 8085  $S_{ij}$  be the observed value of  $S$  for animal  $i$  at detector  $j$ . The encounter probability  
 8086 is  $\Pr(S_{ij} > c)$  which is  $\Pr(S_{ij} > c) = 1 - \Pr(S_{ij} < c)$ , so that, if we standardize the

8087 variate we have

$$1 - \Pr\left(\frac{(S_{ij} - \mathbb{E}(S))}{\sigma_s} < \frac{(c - \mathbb{E}(S))}{\sigma_s}\right)$$

8088 This probability calculation requires evaluation of the CDF of a standard normal  
 8089 variate say,  $\eta = (S_{ij} - \mathbb{E}(S))/\sigma_s$ , being less than  $\gamma(\boldsymbol{\theta}) = (c - \mathbb{E}(S))/\sigma_s$ , which is  
 8090 a function of all the parameters  $\alpha_0$ ,  $\alpha_1$ ,  $\sigma_s^2$  and also the individual location  $\mathbf{u}$  and  
 8091 trap location  $\mathbf{x}$ . We'll identify it by  $\gamma(\boldsymbol{\theta}, \mathbf{x}, \mathbf{u})$  when we need to be explicit about  
 8092 those things. We can compute  $\Pr(S_{ij} > c) = 1 - \Pr(\eta < \gamma(\boldsymbol{\theta}, \mathbf{x}, \mathbf{u}))$  easily using any  
 8093 software package including **R** which has a standard function, `pnorm`, for computing  
 8094 the normal cdf. To be more precise, we'll use the  $\Phi()$  to represent the normal  
 8095 cdf. Therefore, an individual is encountered whenever  $S_{ij} > c$  which happens with  
 8096 probability  $\Pr(S_{ij} > c) = 1 - \Phi(\gamma(\boldsymbol{\theta}, \mathbf{x}, \mathbf{u}))$ .

8097 Naturally this quantity should depend on *where* an individual is located at the  
 8098 time of recording – what we call it's instantaneous location, say  $\mathbf{u}$ , to distinguish it  
 8099 from it's home-range center  $\mathbf{s}$  (but we outline a model below that contains both  $\mathbf{u}$   
 8100 and  $\mathbf{s}$ ), and also the trap  $\mathbf{x}$ , so we index the quantity  $\gamma$  by those two quantities, in  
 8101 addition to the parameters  $\alpha_0$ ,  $\alpha_1$  and  $\sigma_s$ . The probability of detection is therefore

$$p_{ij} = p(\alpha_0, \alpha_1, \sigma | \mathbf{x}_j, \mathbf{u}_i) = 1 - \Phi(\gamma(\cdot))$$

8102 where  $\mathbf{u}_i$  is the instantaneous location of individual  $i$  and  $\mathbf{x}_j$  is the location of trap  
 8103  $j$ . We'll suppose here that the random variables  $\mathbf{u}_i$  have state-space  $\mathcal{U}$ <sup>3</sup>.

8104 How do we interpret this probability? Well, two things have to happen for an  
 8105 individual to be encountered by a trap: (1) it has to vocalize; (2) the microphone  
 8106 has to record a signal  $> c$ . These two things together are a product of biological and  
 8107 environmental factors which could include time of day, wind direction and speed,  
 8108 or maybe rain, humidity and other things. The bottom line is a lot of factors  
 8109 are balled up in whether or not the microphone records a sound greater than the  
 8110 threshold.

8111 The observations from an acoustic survey are the signal strength measurements,  
 8112 and the likelihood of the observed signal strength from individual  $i$  at detection  
 8113 device  $j$  can be specified by noting that the likelihood is the normal pdf for the  
 8114 observed signal *if* the signal strength is  $> c$  and, otherwise, the contribution to the  
 8115 likelihood is  $\Phi(\gamma(\cdot))$  (see Eq. 8 of Efford et al. (2009b)).

$$\Pr(S_{ij} | \mathbf{u}_i) = \Phi(\gamma(\cdot))^{1-I(S_{ij}>c)} \text{Normal}(S_{ij}; \alpha_0, \alpha_1, \sigma_s, \mathbf{x}_j, \mathbf{u}_i)^{I(S_{ij}>c)}$$

8116 We can use this as the basis for constructing the binomial-form of the likelihood  
 8117 as we did in Chapt. 6, which involves the number of individuals not encountered,  
 8118  $n_0$ . The probability that an individual is *not* captured is equal to the probability

<sup>3</sup>We use  $\mathcal{U}$  here to avoid confusion with definition of signal strength,  $S$ . However,  $\mathcal{U}$  is the same state-space as  $\mathcal{S}$  in the rest of the book

8119 that its signal strength doesn't exceed  $c$  at any microphone. The probability of not  
 8120 being captured at a microphone  $\mathbf{x}_j$  is:

$$1 - p_{\mathbf{u},j} = \Phi(\gamma(\cdot))$$

8121 and therefore the probability of not being captured at any microphone is:

$$\Pr(\text{all } S_{\mathbf{u},j} < c | \mathbf{u}) = \prod_{j=1}^J (1 - p_{\mathbf{u},j}) = \prod_{j=1}^J \Phi(\gamma(\cdot, \mathbf{x}_j, \mathbf{u}))$$

8122 and therefore the marginal probability of not being captured is

$$\pi_0 = [\text{all } S_{\mathbf{u},j} < c | \boldsymbol{\alpha}] = \int_{\mathcal{U}} \left\{ \prod_{j=1}^J \Phi(\gamma(\boldsymbol{\theta}, \mathbf{x}_j, \mathbf{u})) \right\} d\mathbf{u}$$

8123 which can be used to construct the binomial form of the likelihood as we did in  
 8124 Chapt. 6 (see Eq. 6.2.1).

#### 8125 8.4.2 Implementation in **secr**

8126 Fitting acoustic encounter models in **secr** is no more difficult than other SCR  
 8127 models. There is a handy manual (**secr-sound.pdf**) with examples (Efford and  
 8128 Dawson, 2010) which comes with the **secr** package. The basic process is that  
 8129 **make.capthist** will make a **capthist** object from a 3-dimensional encounter array  
 8130 – which is a binary array indicating whether each individual was detected or not  
 8131 at each recorder/microphone. In the case of signal strength data, **secr** handles the  
 8132 case where # occasions = 1, i.e., the recorders obtained data for a single sample  
 8133 occasion, but this is not a general requirement of the model for signal strength  
 8134 data (see next section). The “signal” attribute of the **capthist** object contains the  
 8135 signal strength in decibels. The best way to include the signal attribute is to use  
 8136 **make.capthist** in the usual way, providing it with the encounter data and trap  
 8137 data and, in addition, the variable “cutval” (which is  $c$  in our notation above) and  
 8138 then provide the signal strength data as an extra column of the **capthist** object.  
 8139 See **?make.capthist** for details.

#### 8140 8.4.3 Implementation in **BUGS**

8141 We don't know of any Bayesian applications of acoustic SCR models, although  
 8142 we imagine that implementation of such models in the **BUGS** engines should be  
 8143 achievable. It seems easy enough to write down a general hierarchical model that  
 8144 would accommodate sampling on repeated occasions. Let  $\mathbf{s}_i$  be the home range  
 8145 center, and let  $\mathbf{u}_{ik}$  the instantaneous location of individual  $i$  during sample occasion  
 8146  $k$  (see Chapt. 15 for similar models). The model for  $\mathbf{u}_{ik}$  can be specified conditional

on  $\mathbf{s}_i$ . For example, we could assume that  $\mathbf{u}_{ik}$  are bivariate normal draws with mean  $\mathbf{s}_i$  and some variance  $\sigma_u^2$ . Then, conditional on  $\mathbf{u}_{ik}$  an individual produces a signal according to the signal attenuation model (Eq. 9.4.1), or perhaps some other model. Then we generate the binary encounter data by truncating the observed signal at  $c$ . This general model then is an example of an SCR model in which parameters of a movement model are identifiable (see Sec. 2.6) because there is direct information about movement outcomes from the sampling method, unlike other types of encounter methods (e.g., camera traps) for which animal locations are restricted to a set of fixed, pre-determined points where traps are located. Other types of SCR methods allow for movement information too, including some of the search-encounter models (Chapt. 15).

Instead of developing a Bayesian version of this model here, we leave it to the reader to explore simulating data and devising a Bayesian implementation of the acoustic model in one of the **BUGS** engines. Note that for a single occasion, you can simulate the data using the two stage model (having both  $\mathbf{s}$  and  $\mathbf{u}$ ) or you can simulate  $\mathbf{u}$  uniformly without dealing with  $\mathbf{s}$  in the model. The kernel of the **BUGS** model specification should resemble the following snippet:

```
8164 model {
8165   # Ignoring loops and data augmentation
8166   u[i,1] ~ dunif(xlim[1], xlim[2])
8167   u[i,2] ~ dunif(ylim[1], ylim[2])
8168   mu[i,j] <- alpha0 + alpha1*d[i,j]
8169   #####
8170   ### JAGS has this T() truncation feature
8171   S[i,j] ~ dnorm(mu[i,j], 1/sigma^2)T(c,Inf)
8172   #####
8173   gamma[i,j] <- (c - mu[i,j])/sigma
8174   p[i,j] <- 1 - pnorm(gamma[i,j], 0, 1) # JAGS has pnorm() function
8175   y[i,j] ~ dbern(p[i,j])
8176 }
```

#### 8.4.4 Other types of acoustic data

Efford and Dawson (2010) noted that various other types of acoustic data might arise for which SCR-like models would be useful<sup>4</sup>. For example, we could measure the *time of arrival* of a vocal queue of some sort at multiple recorders to estimate the number and origin of  $N$  queues. 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,

<sup>4</sup>Some of the following is also related to material presented by D.L. Borchers at the ISEC 2012 conference in Norway.

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.

## 8.5 SUMMARY AND OUTLOOK

In this chapter we extended SCR models to accommodate alternative models for the observation process, 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. 15) 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 8.4.** 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. 9.3	<code>single-catch</code>
Acoustic	Sec. 9.4	<code>signal</code>
Search-encounter	Chapt. 15	<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. 16; (2) There are models for which only specific summary statistics are observable (Chandler and Royle, In press; Sollmann et al., 2012) which we cover in Chaps. 18 - 19; (3) We can have multiple observation methods working together as in Gopalaswamy et al. (2012b).

There remains much research to be done to formalize models for certain obser-

vation systems. For example, while we think one will usually be able to analyze  
single-catch systems using the multi-catch model, or even the Bernoulli model if  
encounter probability is sufficiently low, a formalization of the single-catch model  
would be a useful development and, we believe, it should be achievable using one  
or another of the **BUGS** engines. In addition, classical “trapping webs” (Anderson  
et al., 1983; Wilson and Anderson, 1985a; Jett and Nichols, 1987; Parmenter  
and MacMahon, 1989; Link and Barker, 1994) have been around for quite some  
time and it seems like they are amenable to formulation as a type of SCR model  
although we have not pursued that development simply because trapping webs are  
rarely used in practice.



8225  
8226

---

8227

# 9

## SAMPLING DESIGN

### 9.1 GENERAL CONSIDERATIONS

- 8228 9.1.1 Model-based not design-based
- 8229 9.1.2 Sampling space or sampling individuals?
- 8230 9.1.3 Scope of inference vs. state-space

### 9.2 STUDY DESIGN FOR (SPATIAL) CAPTURE-RECAPTURE

### 9.3 TRAP SPACING AND ARRAY SIZE RELATIVE TO ANIMAL MOVEMENT

- 8231 9.3.1 Example: Black bears from Pictured Rocks National Lakeshore:

8232 **9.3.2 Final musings: SCR models, trap spacing and array size**

**9.4 SPACING OF TRAPS WITH TELEMETERED INDIVIDUALS**

**9.5 SAMPLING OVER LARGE SCALES**

**9.6 MODEL-BASED SPATIAL DESIGN**

8233 **9.6.1 Formalization of the Design Problem for SCR Studies**

8234 **9.6.2 An Optimal Design Criterion for SCR**

(9.6.1)

(9.6.2)

(9.6.3)

8235 **9.6.3 Optimization of the criterion**

8236 **9.6.4 Illustration**

**9.7 COVARIATE MODELS**

**9.8 SUMMARY AND OUTLOOK**

8237

## Part III

8238

---

8239

# Advanced SCR Models



8240  
8241

8242  
8243

# 10

---

## MODELING SPATIAL VARIATION IN DENSITY

**10.1 HOMOGENEOUS POINT PROCESS REVISITED**

**10.2 INHOMOGENEOUS POINT PROCESSES**

(10.2.1)

(10.2.2)

(10.2.3)

(10.2.4)

(10.2.5)

(10.2.6)

**10.3 OBSERVED POINT PROCESSES**

**10.4 FITTING INHOMOGENEOUS POINT PROCESS SCR MODELS**

8244   **10.4.1 Continuous space**

8245   **10.4.2 Discrete space**

**10.5 ECOLOGICAL DISTANCE AND DENSITY COVARIATES**

**10.6 THE JAGUAR DATA**

**10.7 SUMMARY AND OUTLOOK**





8246  
8247

# 11

---

## MODELING LANDSCAPE CONNECTIVITY

### 11.1 SHORTCOMINGS OF EUCLIDEAN DISTANCE MODELS

### 11.2 LEAST-COST PATH DISTANCE

(11.2.1)

(11.2.2)

(11.2.3)

8249 11.2.1 Example of Computing Cost-weighted distance

### 11.3 SIMULATING SCR DATA USING ECOLOGICAL DISTANCE

### 11.4 LIKELIHOOD ANALYSIS OF ECOLOGICAL DISTANCE MODELS

(11.4.1)

8250    **11.4.1 Example of SCR with Least-Cost Path**

**11.5 BAYESIAN ANALYSIS**

**11.6 SIMULATION EVALUATION OF THE MLE**

8251    **11.6.1 Simulation Results**

**11.7 DISTANCE IN AN IRREGULAR PATCH**

8252    **11.7.1 Basic Geographic Analysis in R**

**11.8 SUMMARY AND OUTLOOK**





8253  
8254

# 12

---

8255  
8256  
8257

## **INTEGRATING RESOURCE SELECTION WITH SPATIAL CAPTURE-RECAPTURE MODELS**

8258  
8259  
8260

### **12.1 A SIMPLE MODEL OF SPACE USAGE**

**12.1.1 Poisson use model**

**12.1.2 Thinning**

**12.1.3 Capture-recapture Data**

### **12.2 THE JOINT RSF/SCR LIKELIHOOD**

### **12.3 APPLICATION: NEW YORK BLACK BEAR STUDY**

### **12.4 SIMULATION STUDY**

### **12.5 SUMMARY AND OUTLOOK**



8261  
8262

---

8263  
8264

# 13

## STRATIFIED POPULATIONS: MULTI-SESSION AND MULTI-SITE DATA

### 13.1 DATA STRUCTURE

### 13.2 MULTINOMIAL ABUNDANCE MODELS

(13.2.1)

(13.2.2)

(13.2.3)

(13.2.4)

(13.2.5)

8265 **13.2.1 Observation Models**

8266 **13.2.2 Simulating group structured capture-recapture data**

8267 **13.2.3 Fitting in BUGS**

8268 **13.2.4 Approach B modeling  $\psi$**

### **13.3 SPATIAL CAPTURE-RECAPTURE**

(13.3.1)

### **13.4 APPLICATION**

8269 **13.4.1 Results**



**13.5 TOPICS IN MULTI-SESSION MODELS**

- 8270   **13.5.1 Temporal models**
- 8271   **13.5.2 Dependence – is it a problem?**
- 13.6 MULTI-SESSION MODELS IN SECR**
- 8272   **13.6.1 Ovenbird data in WinBUGS?**
- 8273   **13.6.2 Converse data in secr?**
- 13.7 SUMMARY AND OUTLOOK**

8274  
8275

8276  
8277

# 14

---

## MODELS FOR SEARCH-ENCOUNTER DATA

### 14.1 SEARCH-ENCOUNTER SAMPLING DESIGNS

### 14.2 A MODEL FOR SEARCH-ENCOUNTER DATA

(14.2.1)

(14.2.2)

(14.2.3)

8278 14.2.1 Ecological process model

8279 14.2.2 Other stuff

### 14.3 EXAMPLES

8280 14.3.1 Hard plot boundaries

8281 14.3.2 Analysis of other protocols

### 14.4 DESIGN 3: AD HOC IMPLEMENTATION OF DESIGN 1.

### 14.5 CAPRICAILLIE CRAP

8282 14.5.1 model

### 14.6 DESIGN 4 – NO LOCATION INFO

### 14.7 SUMMARY AND OUTLOOK



8283  
8284

8285

# 15

---

## OPEN POPULATION MODELS

### 15.1 INTRODUCTION

- 8286 15.1.1 Overview of Population Dynamics
- 8287 15.1.2 Animal movement related to population demography
- 8288 15.1.3 Basic assumptions of JS and CJS models

### 15.2 TRADITIONAL JOLLY-SEBER MODELS

- 8289 15.2.1 Data Augmentation for the Jolly-Seber Model
- 
- 

- 8290 15.2.2 Mist-netting example
- 8291 15.2.3 Shortcomings of the traditional JS models

### 15.3 SPATIAL JOLLY-SEBER MODELS

(15.3.1)

8292 **15.3.1 Mist-netting example**

## **15.4 TRADITIONAL CJS MODELS**

8293 **15.4.1 Migratory fish example**

## **15.5 MULTI-STATE CJS MODELS**

8294 **15.5.1 Migratory fish example**

## **15.6 SPATIAL CJS MODELS**

8295 **15.6.1 Migratory fish example**

## **15.7 MOVING ACTIVITY CENTERS**

8296 **15.7.1 Migratory Fish Example Notes**

## **15.8 SUMMARY AND OUTLOOK**





8297

## Part IV

8298

---

8299

# Super-Advanced SCR Models



8300  
8301

---

8302

8303

# 16

---

## DEVELOPING MARKOV CHAIN MONTE CARLO SAMPLERS

8304 **16.0.1 Why build your own MCMC algorithm?**

### 16.1 MCMC AND POSTERIOR DISTRIBUTIONS

(16.1.1)

(16.1.2)

### 16.2 TYPES OF MCMC SAMPLING

8305 **16.2.1 Gibbs sampling**

(16.2.1)

---

---

- 8306 **16.2.2 Metropolis-Hastings sampling**  
8307 **16.2.3 Metropolis-within-Gibbs**  
8308 **16.2.4 Rejection sampling and slice sampling**

### **16.3 MCMC FOR CLOSED CAPTURE-RECAPTURE MODEL MH**

### **16.4 MCMC ALGORITHM FOR MODEL SCR0**

- 8309 **16.4.1 SCR model with binomial encounter process**  
8310 **16.4.2 Looking at model output**  
8311 **Markov chain time series plots**  
8312 **16.4.3 Posterior density plots**  
8313 **16.4.4 Serial autocorrelation and effective sample size**  
8314 **16.4.5 Summary results**  
8315 **16.4.6 Other useful commands**  
**16.5 MANIPULATING THE STATE-SPACE**  
**16.6 INCREASING COMPUTATIONAL SPEED**  
8316 **16.6.1 Parallel computing**  
8317 **16.6.2 Using C++**  
**16.7 SUMMARY AND OUTLOOK**





8318  
8319

# 17

---

8320  
8321

## SPATIAL CAPTURE-RECAPTURE FOR UNMARKED POPULATIONS

**17.1 EXISTING MODELS FOR INFERENCE ABOUT DENSITY IN  
UNMARKED POPULATIONS**

**17.2 SPATIAL CORRELATION AS INFORMATION**

**17.3 DATA**

**17.4 MODEL**

(17.4.1)

(17.4.2)

**17.5 NORTHERN PARULA EXAMPLE**

**17.6 IMPROVING PRECISION WITH PRIOR INFORMATION**

**17.7 DESIGN ISSUES**

8322 **17.7.1 How Much Correlation Is Enough?**

8323 **17.7.2 Linear Designs**

8324 **17.7.3 Quadrat counts**

**17.8 ALTERNATIVE OBSERVATION MODELS**

8325 **17.8.1 Spatial point process models**

**17.9 CONCLUSION**



8326

8327

8328

8329

8330

# 18

---

## SPATIAL MARK-RESIGHT MODELS FOR PARTIALLY IDENTIFIABLE POPULATIONS

### 18.1 BACKGROUND

8331 **18.1.1 Types of partial ID data**

8332 **18.1.2 A short history of mark-resight models**

(18.1.1)

8333

### 18.2 KNOWN NUMBER OF MARKED INDIVIDUALS

8334 **18.2.1 MCMC for a spatial mark-resight model**

8335 **18.2.2 Binomial encounter model**

**18.3 UNKNOWN NUMBER OF MARKED INDIVIDUALS****18.4 IMPERFECT IDENTIFICATION OF MARKED INDIVIDUALS****18.5 HOW MUCH INFORMATION DO MARKED AND UNMARKED INDIVIDUALS CONTRIBUTE?****18.6 INCORPORATING TELEMETRY DATA****18.7 SUMMARY AND OUTLOOK**





# 19

8336

8337

8338

8339

## 2012: A SPATIAL CAPTURE-RECAPTURE ODYSSEY

8340 Capture recapture methods have been a cornerstone of ecological modeling and  
8341 analysis for decades. Yet there are essentially no real capture-recapture data sets  
8342 that come *without* auxiliary spatial information about location of capture (but  
8343 sometimes such information is thrown into the trashcan).

8344 The big point is that we provide a framework for spatial analysis of animal popu-  
8345 lations from individual encounter data: MOVEMENT, SPACE USAGE, SPATIAL  
8346 VARIATION IN DENSITY – much to be done: how do individuals interact? how  
8347 is space usage changing over time, etc...

8348 Topics to discuss here:

8349 (1) Strauss process model (2) Need for general purpose software.... all of the  
8350 spatial stuff + open populations in one big model. (3) Efficient computation is still  
8351 an issue. (4) Fit and model selection will continue to be important practical issues.

### 19.1 10 THESIS OR DISSERTATION TOPICS

8352 Future research directions:  
8353     Modeling dynamics of the point process. Transient individuals. Dispersal.  
8354 Things like that.  
8355     Calibration of GoF under meaningful alternatives  
8356     Calibration of AIC/DIC and efficacy study  
8357     Models for non-uniform point processes that exhibit clustering or repulsion  
8358     no-marking model + RSF  
8359     occupancy and counts data + SCR data (AOAS and Sollmann et al.)  
8360     Spatial genetics – can use SCR to study gene flow, related things....  
8361     SCR on dendritic networks (streams and trails).

### 19.2 THREE DIMESIONAL SPACE

8362 Throughout this book we have treated space as two-dimensional, meaning that  
8363 activity centers are assumed to occur on the real plane. This approximation of  
8364 reality is reasonable for many terrestrial species, but aquatic organisms, especially  
8365 marine animals move about in three-dimensional space. Treating space as three-  
8366 dimensional could also conceivably be useful in studies of flying organisms or species  
8367 that use multiple strata of tall forests; however, we suspect that two dimensional  
8368 models of space should suffice in such contexts. Regardless, a three-dimensional  
8369 view of space requires that activity centers  $s_i$  are indexed by  $x, y, z$  coordinates.  
8370 In theory, this presents no problem whatsoever. In practice, estimation based on  
8371 integrated likelihood methods must involve a three-dimensional integration. This  
8372 will clearly be more computationally demanding, but it should be possible using  
8373 packages such as R2Cuba.

### 19.3 GREGARIOUS SPECIES

8374 Many social species move about in large groups rather than as single individuals.  
8375 Even species regarded as solitary often join family groups for some portion of their  
8376 life cycle. The consequences of gregariousness?? are x-fold....  
8377 To account for this, we change our definition of  $s_i$  from the location of an  
8378 individual's activity center, to the location of a group's activity center. We then  
8379 expand our model to include a submodel for group size, and we can estimate both  
8380 the density of group activity centers and total population size.

8381

## Part V

8382

---

8383

## Appendices



8384                   **APPENDIX I - USEFUL SOFTWARE AND**  
8385                   **R PACKAGES**

---

8386     Throughout this book we have used a suite of software and R packages, all of which  
8387     are freely available online. To make life a little easier for you, here we provide you  
8388     with a list of all software and R packages, download links and some (hopefully)  
8389     helpful tips regarding their installation.

#### 19.4 WINBUGS

8390     Although **WinBUGS** (Gilks et al., 1994) is becoming increasingly obsolete with  
8391     the faster and more flexible **OpenBUGS** and **JAGS**, there are still situations  
8392     in which the program comes in handy. The .exe file can be downloaded from  
8393     <http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/contents.shtml>. On 32 bit machines  
8394     you can just go ahead and double-click on the .exe file and follow the installation  
8395     instructions on the screen. On 64 bit machines, according to the BUGS project you  
8396     should download a zip file (from the same page) and unzip it into a folder of your  
8397     choice. There are a couple of additional steps to make BUGS run. First, you need  
8398     to obtain a key (which is free and valid for life) here: '[http://www.mrc-bsu.cam.ac.uk/bugs/WinBUGS14\\_immortality\\_key.txt](http://www.mrc-bsu.cam.ac.uk/bugs/WinBUGS14_immortality_key.txt)'. The key comes  
8399     with instructions on how to activate it. Second, you need to update the basic  
8400     **WinBUGS** version to the most current one (which is from August 2007)  
8401     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)'. **Win-**  
8402     **BUGS** is ready to use after quitting and re-opening it. Remember that **Win-**  
8403     **BUGS** only runs on Windows machines. Also, there appears to be a problem  
8404     installing the program in Vista, although we have no personal experience with this.

##### 19.4.1 WinBUGS through R

8405     While you can run **WinBUGS** as a standalone application, we recommend you ac-  
8406     cess it from within **R** using the package **R2WinBUGS** (Sturtz et al., 2005), so you can  
8407     conveniently process your output, make graphs etc. **R2WinBUGS** also allows you to  
8408     run models in **OpenBUGS** (see below). You can install the package from within

8413 **R** directly from a cran mirror. In addition to the usual package help document  
8414 (<http://cran.r-project.org/web/packages/R2WinBUGS/R2WinBUGS.pdf>) you can  
8415 also download a short manual with some examples (''[http://voteview.com/bayes\\_beach/R2WinBUGS.pdf](http://voteview.com/bayes_beach/R2WinBUGS.pdf)'').

## 19.5 OPENBUGS

8417 **OpenBUGS** is the up-to-date version of **WinBUGS** and can be downloaded  
8418 here: ''<http://www.openbugs.info/w/Downloads>'' (Windows, Mac and Linux  
8419 versions are available). The name '**OpenBUGS**' refers to the software being open  
8420 source, so users do not need to download a license key, like they have to for **Win-**  
8421 **BUGS** (although the license key for **WinBUGS** is free and valid for life). For  
8422 Windows, install by double-clicking on the .exe file and following the instructions  
8423 on the installer screen. Compared to **WinBUGS**, **OpenBUGS** has more built-  
8424 in functions. The method of how to determine the right updater for each model  
8425 parameter has changed and the user can manually control the MCMC algorithm  
8426 used to update model parameters. Several other changes have been implemented  
8427 in **OpenBUGS** and a detailed list of differences between the two **BUGS** versions,  
8428 can be found at <http://www.openbugs.info/w/OpenVsWin>. We have encountered  
8429 convergence problems with simple scr models in this program. There is an extensive  
8430 help archive for both **WinBUGS** and **OpenBUGS** and you can subscribe to a  
8431 mailing list, where people pose and answer questions of how to use these programs  
8432 at <http://www.mrc-bsu.cam.ac.uk/bugs/overview/list.shtml>

### 19.5.1 OpenBUGS through R

8434 Like **WinBUGS**, **OpenBUGS** can be used as a standalone application or through  
8435 **R**. There are several packages that allow **R** to interface with **OpenBUGS**, all of  
8436 which can be installed directly from a cran mirror:

8437 **R2WinBUGS**: One of the options in the `bugs()` call is `program`, which lets you  
8438 specify either **WinBUGS** or **OpenBUGS**. This is a convenient option because  
8439 after having worked through some of this book you will likely be familiar with the  
8440 format of `bugs()` output and other functions of the **R2WinBUGS** package.

8441 **R2OpenBUGS**: **R2OpenBUGS** (Sturtz et al., 2005) is very similar to, and actually  
8442 based on, **R2WinBUGS** and it is unclear to us what can be gained by using the  
8443 former over the latter. Arguments of the `bugs()` call differ slightly between the  
8444 two packages and given that **R2WinBUGS** allows for the use of both **OpenBUGS**  
8445 and **WinBUGS** it is probably easiest to stick with it.

8446 **BRugs**: **BRugs** (Thomas et al., 2006) can be installed from within **R** directly from  
8447 a cran mirror. In addition to the help document at ''<http://www.biostat.umn>.

8448    `edu/~brad/software/BRugs/BRugs_9_21_07.pdf`’’, there is a **WinBUGS** style  
8449    manual you can access at ’’<http://www.rni.helsinki.fi/openbugs/OpenBUGS/Docu/BRugs%20Manual.html>’’. **BRugs** has the convenient feature that all pieces of  
8450    a **BUGS** analysis can be run from within **R**, including checking the model syntax,  
8451    something that requires opening the **BUGS** GUI with other packages.  
8452

## 19.6 JAGS

8453    **JAGS** (Just Another Gibbs Sampler) (Plummer, 2003) runs scr models consider-  
8454    ably faster than **WinBUGS**, does not have the convergence problem with simple  
8455    scr models we have encountered in **OpenBUGS** but similar to the latter program,  
8456    is flexible and constantly updated. Writing a **JAGS** model is virtually identical to  
8457    writing a **WinBUGS** model. However, some functions may have slightly different  
8458    names and you can look up available functions and their use in the **JAGS** man-  
8459    ual. One potential downside is that **JAGS** can be very particular when it comes  
8460    to initial values. These may have to be set as close to truth as possible for the  
8461    model to start. Although **JAGS** lets you run several parallel Markov chains, this  
8462    characteristic interferes with the idea of using overdispersed initial values for the  
8463    different chains. Also, we have found that when running models, sometimes **JAGS**  
8464    crashes for unclear reasons, taking **R** down with it. Oftentimes, in order to make  
8465    it run again you’ll have to go through downloading and installing it again (remove  
8466    the non-functioning version first).

8467    **JAGS** has a variety of functions that are not available in **WinBUGS**. For  
8468    example, **JAGS** allows you to supply observed data for some deterministic func-  
8469    tions of unobserved variables. In **BUGS** we cannot supply data to logical nodes.  
8470    Another useful feature is that the adaptive phase of the model (the burn-in) is run  
8471    separately from the sampling from the stationary Markov chains. This allows you  
8472    to easily add more iterations to the adaptive phase if necessary without the need to  
8473    start from 0. There are other, more subtle differences and there is an entire manual  
8474    section on differences between **JAGS** and **OpenBUGS**.

8475    **JAGS** is available for download at ’’<http://sourceforge.net/projects/mcmc-jags/files/>’’, together with the R package **rjags** (Plummer, 2011), which  
8476    allows running **JAGS** through **R**, user and installation manuals and examples. At  
8477    this site **JAGS** is available for Windows and Mac; Linux binaries are distributed  
8478    separately and you can find links to various sources here: ’’<http://mcmc-jags.sourceforge.net/>’’. **JAGS** comes with a 32 bit and a 64 bit version and can be  
8479    installed by double-clicking on the .exe file and following the instructions on the in-  
8480    staller screen. For questions and problems concerning **JAGS** there is a forum online  
8481    at <http://sourceforge.net/projects/mcmc-jags/forums/forum/610037>.  
8482  
8483

8484 **19.6.1 JAGS through R**

8485 Unlike the two **BUGS** programs, **JAGS** does not have a GUI interface but a com-  
 8486 mand line interface that can be used to run the program as a standalone application.  
 8487 **JAGS** will solely perform the MCMC simulation; analyzing and summarizing the  
 8488 output has to be done outside of **JAGS**. To run **JAGS** through **R** you have two  
 8489 options.

8490 **rjags**: As mentioned above, **rjags** (Plummer, 2011) can be found together with  
 8491 **JAGS** and was developed/is being maintained by the inventor of **JAGS**, which  
 8492 means it is guaranteed to stay up to date when/as **JAGS** changes. The package  
 8493 can be installed from a cran mirror and the help document can be accessed at  
 8494 `' 'http://cran.r-project.org/web/packages/rjags/rjags.pdf'`

8495 **R2jags**: Alternatively, the package **R2jags** (Su and Yajima, 2011) provides a means  
 8496 of accessing **JAGS** through **R**. We prefer **rjags** for the reason named above, as  
 8497 well as because it stores data in a more memory-efficient way and has better `plot()`  
 8498 and `summary()` methods.

**19.7 R**

8499 At the time of the preparation of this list, **R** for Windows is at version 2.15.0, which  
 8500 can be downloaded at <http://cran.r-project.org/bin/windows/base/> This site also  
 8501 contains helpful tips on how to install **R** in Windows Vista, how to update **R**  
 8502 packages etc. Installation of **R** in Windows is straightforward: download the .exe  
 8503 file, double-click on it and follow the instructions of the Windows installer. The **R**  
 8504 later versions of **R** come with versions for both 64 bit and 32 bit machines. The **R**  
 8505 site (`' 'http://mirrors.softliste.de/cran/' '`) has an extensive FAQ section  
 8506 Hornik (2011), which includes instructions on how to install R on Unix and Mac  
 8507 computers.

8508 **19.7.1 R packages**

8509 This section provides an alphabetical list of useful **R** packages. There is a large  
 8510 number of **R** packages and by no means is this list intended to be complete in terms  
 8511 of what is useful. Rather, we list packages that we are familiar with and that we  
 8512 employ at one point or the other in this book. Unless explicitly stated otherwise,  
 8513 all packages can be installed directly from within **R** trough a cran mirror.

8514 **adapt**: **adapt** (Genz et al., 2007) is a package for multidimensional numerical inte-  
 8515 gration. The package has been removed from the CRAN repository but can be ob-  
 8516 tained from `' 'http://cran.r-project.org/src/contrib/Archive/adapt/' '`.

8517 **coda**: **coda** (Plummer et al., 2006) lets you summarize and perform diagnostics on  
 8518 mcmc output. For a list and description of functions, see the manual at `' 'http:`  
 8519 [' '](http://cran.r-project.org/web/packages/coda/coda.pdf).

8520 **gdistance**: **gdistance** (van Etten, 2011) is a package for calculating distances  
8521 and routes on geographical grids and can be used to calculate least cost path  
8522 surfaces. Manual at '<http://cran.r-project.org/web/packages/gdistance/gdistance.pdf>' .

8524 **igraph**: **igraph** (Csardi and Nepusz, 2006) provides routines for graphs and net-  
8525 work analysis. Manual at '<http://cran.r-project.org/web/packages/igraph/igraph.pdf>' .

8527 **inline**: **inline** (Sklyar et al., 2010) allows the user to define R functions with in-  
8528 lined **C**, **C++** or **Fortran** code. Manual at <http://cran.r-project.org/web/packages/inline/inline.pdf>.

8530 **maps**: **maptools** (?) is a library of maps. Manual at '<http://cran.r-project.org/web/packages/maps/index.html>' .

8532 **maptools**: **maptools** (Lewin-Koh et al., 2011) provides a set of tools for reading  
8533 and manipulating spatial data, especially ESRI shapefiles. Manual at '<http://cran.r-project.org/web/packages/maptools/maptools.pdf>' .

8535 **R2cuba**: **R2cuba** (Hahn et al., 2010) is another package for multidimensional integra-  
8536 tion. Manual at '<http://cran.r-project.org/web/packages/R2Cuba/R2Cuba.pdf>' .

8538 **raster**: **raster** (van Etten, 2012) provides functions for geographic analysis and  
8539 modeling with raster data. Manual at '<http://cran.r-project.org/web/packages/raster/raster.pdf>' .

8541 **Rcpp**: **Rcpp** (Eddelbuettel and François, 2011) provides R functions as well as a  
8542 **C++** library which facilitate the integration of **R** and **C++**. Manual at <http://cran.r-project.org/web/packages/Rcpp/Rcpp.pdf>.

8544 **RcppArmadillo**: **RcppArmadillo** (François et al., 2011) is a templated **C++** linear  
8545 algebra library, integrating the **Armadillo** library and **R**. Manual at <http://cran.r-project.org/web/packages/RcppArmadillo/RcppArmadillo.pdf>.

8547 **reshape**: **reshape** (Wickham and Hadley, 2007) allows you to easily manipulate,  
8548 summarize and reshape data. Manual at '<http://cran.r-project.org/web/packages/reshape/reshape.pdf>' .

8550 **rgeos**: **rgeos** (Bivand and Rundel, 2011) provides many useful functions for spatial  
8551 operations such as intersecting or buffering spatial features. Manual at '<http://cran.r-project.org/web/packages/rgeos/rgeos.pdf>' .

8553 **SCRbayes**: (Russell et al., 2012)

8554 **secr: secr** (Efford et al., 2009a)

8555 **shapefiles: shapefiles** (Stabler, 2006) allows you to read and write ESRI shape-  
8556 files (i.e. shapefiles you would use in **ArcGIS**). Manual at '<http://cran.r-project.org/web/packages/shapefiles/shapefiles.pdf>'.

8558 **snow, snowfall:** **snow** (Tierney et al., 2011) and **snowfall** (Knaus, 2010) pro-  
8559 vide functionality for parallel computing. The latter is a more user-friendly wrap-  
8560 per around the former. Manuals at <http://cran.r-project.org/web/packages/snowfall/snowfall.pdf> and <http://cran.r-project.org/web/packages/snow/snow.pdf>.

8563 **sp: sp** (Pebesma and Bivand, 2011) is a package for plotting, selecting, subsetting  
8564 etc. spatial data. **sp** and **spatstat** (see below) are complementary in many ways  
8565 and data formats can be easily converted between the two packages. Manual at  
8566 '<http://cran.r-project.org/web/packages/sp/sp.pdf>'.

8567 **SPACECAP:** (Gopalaswamy et al., 2012a)

8568 **spatstat: spatstat** (Baddeley and Turner, 2005) is an extensive package for an-  
8569 alyzing spatial data. We use it, for example, to generate random points within a  
8570 state space that cannot be described as a rectangle but consists of a (or several)  
8571 arbitrary polygon(s). Manual at '<http://cran.r-project.org/web/packages/spatstat/spatstat.pdf>'.

8573 **unmarked:**

8574

8575 #####

8576 **References**

8577

## BIBLIOGRAPHY

8578

8579

---

- 8580 Aitkin, M. (1991), "Posterior bayes factors," *Journal of the Royal Statistical Society. Series B (Methodological)*, 53, 111–142.
- 8581
- 8582 Alho, J. M. (1990), "Logistic regression in capture-recapture models," *Biometrics*, 46, 623–635.
- 8583
- 8584 Alpízar-Jara, R. and Pollock, K. H. (1996), "A combination line transect and capture-recapture sampling model for multiple observers in aerial surveys," *Environmental and Ecological Statistics*, 3, 311–327.
- 8585
- 8586
- 8587 Amstrup, S. C., McDonald, T. L., and Manly, B. F. J. (2005), *Handbook of capture-recapture analysis*, Princeton Univ Pr.
- 8588
- 8589 Anderson, D. R., Burnham, K. P., White, G. C., and Otis, D. L. (1983), "Density Estimation of Small-Mammal Populations Using a Trapping Web and Distance Sampling Methods," *Ecology*, 64, 674–680.
- 8590
- 8591
- 8592 Arnason, A., Schwarz, C., and Gerrard, J. (1991), "Estimating closed population size and number of marked animals from sighting data," *Journal of Wildlife Management*, 55, 716–730.
- 8593
- 8594
- 8595 Baddeley, A. and Turner, R. (2005), "Spatstat: an R package for analyzing spatial point patterns," *Journal of Statistical Software*, 12, 1–42, ISSN 1548-7660.
- 8596
- 8597 Bales, S., Hellgren, E., Leslie Jr, D., and Hemphill Jr, J. (2005), "Dynamics of a recolonizing population of black bears in the Ouachita Mountains of Oklahoma," *Wildlife Society Bulletin*, 33, 1342–1351.
- 8598
- 8599
- 8600 Balme, G. A., Slotow, R., and Hunter, L. T. B. (2010), "Edge effects and the impact of non-protected areas in carnivore conservation: leopards in the Phinda–Mkuze Complex, South Africa," *Animal Conservation*, 13, 315–323.
- 8601
- 8602
- 8603 Bartmann, R. M., White, G. C., Carpenter, L. H., and Garrott, R. A. (1987), "Aerial Mark-Recapture Estimates of Confined Mule Deer in Pinyon-Juniper Woodland," *The Journal of Wildlife Management*, 51, 41–46.
- 8604
- 8605
- 8606 Berger, J., Liseo, B., and Wolpert, R. (1999), "Integrated likelihood methods for eliminating nuisance parameters," *Statistical Science*, 14, 1–28.
- 8607
- 8608 Best, N. G., Ickstadt, K., and Wolpert, R. L. (2000), "Spatial Poisson Regression for Health and Exposure Data Measured at Disparate Resolutions," *Journal of the American Statistical Association*, 95, 1076.
- 8609
- 8610
- 8611 Bibby, C. J. and Buckland, S. T. (1987), "Bias of bird census results due to detectability varying with habitat," *Acta Ecologica*, 8, 103–112.
- 8612

- 8613 Bibby, C. J., Burgess, N. D., and Hill, D. A. (1992), "Bird census techniques:  
8614 Academic Press," *London, UK*, 257.
- 8615 Bivand, R. and Rundel, C. (2011), *rgeos: Interface to Geometry Engine - Open*  
8616 *Source (GEOS)*, r package version 0.1-8.
- 8617 Blair, W. (1940), "A study of prairie deer-mouse populations in southern Michigan,"  
8618 *American Midland Naturalist*, 24, 273–305.
- 8619 Bolker, B. (2008), *Ecological models and data in R*, Princeton Univ Pr.
- 8620 Borchers, D. L. (2012), "A non-technical overview of spatially explicit capture–  
8621 recapture models," *Journal of Ornithology*, 152, 1–10.
- 8622 Borchers, D. L., Buckland, S. T., and Zucchini, W. (2002), *Estimating animal*  
8623 *abundance: closed populations*, vol. 13, Springer Verlag.
- 8624 Borchers, D. L. and Efford, M. G. (2008), "Spatially explicit maximum likelihood  
8625 methods for capture–recapture studies," *Biometrics*, 64, 377–385.
- 8626 Borchers, D. L., Zucchini, W., and Fewster, R. M. (1998), "Mark-recapture models  
8627 for line transect surveys," *Biometrics*, 54, 1207–1220.
- 8628 Boulanger, J. and McLellan, B. (2001), "Closure violation in DNA-based mark–  
8629 recapture estimation of grizzly bear populations," *Canadian Journal of Zoology*,  
8630 79, 642–651.
- 8631 Bowden, D. C. (1993), "A simple technique for estimating population size. Technical  
8632 Report 93/12." Tech. rep., Department of Statistics, Colorado State University,  
8633 Fort Collins, Colorado, USA.
- 8634 Brooks, S. P., Catchpole, E. A., and Morgan, B. J. T. (2000), "Bayesian Animal  
8635 Survival Estimation," *Statistical Science*, 15, 357–376.
- 8636 Buckland, S., Anderson, D., Burnham, K., Laake, J., Borcher, D., and L. T. (2001),  
8637 *Introduction to distance sampling: estimating abundance of biological populations*,  
8638 Oxford, UK: Oxford University Press.
- 8639 Buckland, S. T. (2004), *Advanced distance sampling*, Oxford University Press, USA.
- 8640 Burnham, K., Anderson, D., and Laake, J. (1980), "Estimation of density from line  
8641 transect sampling of biological populations," *Wildlife monographs*, 3–202.
- 8642 Burnham, K., Anderson, D., White, G., Brownie, C., and Pollock, K. (1987),  
8643 "Design and analysis methods for fish survival experiments based on release–  
8644 recapture," *American Fisheries Society Monograph 5. American Fisheries Society*,  
8645 *Bethesda Maryland. 1987. 737.*
- 8646 Burnham, K. P. and Anderson, D. R. (2002), *Model selection and multimodel in-*  
8647 *ference: a practical information-theoretic approach*, Springer Verlag.
- 8648 Burnham, K. P. and Overton, W. S. (1978), "Estimation of the size of a closed  
8649 population when capture probabilities vary among animals," *Biometrika*, 65,  
8650 625.
- 8651 Burt, W. (1943), "Territoriality and home range concepts as applied to mammals,"  
8652 *Journal of mammalogy*, 24, 346–352.
- 8653 Casella, G. and Berger, R. L. (2002), *Statistical inference*, Duxbury Press.
- 8654 Chandler, R. B. and Royle, J. A. (In press), "Spatially-explicit models for inference  
8655 about density in unmarked or partially marked populations," *Annals of Applied*

- 8656        *Statistics*.
- 8657        Chandler, R. B., Royle, J. A., and King, D. (2011), "Inference about density and  
8658        temporary emigration in unmarked populations," *Ecology*, 92, 1429–1435.
- 8659        Clobert, J., Danchin, E., Dhondt, A., and Nichols, J. (2001), *Dispersal*, Oxford.
- 8660        Converse, S., White, G., Farris, K., and Zack, S. (2006), "Small mammals and forest  
8661        fuel reduction: national-scale responses to fire and fire surrogates," *Ecological  
8662        Applications*, 16, 1717–1729.
- 8663        Converse, S. J. and Royle, J. A. (2012), "Dealing with incomplete and variable  
8664        detectability in multi-year, multi-site monitoring of ecological populations," in  
8665        *Design and Analysis of Long-term Ecological Monitoring Studies*, eds. Gitzen,  
8666        R. R., Millspaugh, J. J., Cooper, A. B., and Licht, D. S., Cambridge University  
8667        Press, pp. 426–442.
- 8668        Cooch, E. and White, G. (2006), "Program MARK: a gentle introduction," *available  
8669        online with the MARK programme*, 7.
- 8670        Coull, B. A. and Agresti, A. (1999), "The Use of Mixed Logit Models to Reflect  
8671        Heterogeneity in Capture-Recapture Studies," *Biometrics*, 55, 294–301.
- 8672        Cressie, N. (1991), *Statistics for spatial data*, Wiley Series in Probability and Mathematical Statistics.
- 8673        Csardi, G. and Nepusz, T. (2006), "The igraph software package for complex network research," *InterJournal, Complex Systems*, 1695.
- 8674        Dawson, D. K. and Efford, M. G. (2009), "Bird population density estimated from  
8675        acoustic signals," *Journal of Applied Ecology*, 46, 1201–1209.
- 8676        DeGraaf, R. M. and Yamasaki, M. (2001), *New England wildlife: habitat, natural  
8677        history, and distribution*, University Press of New England.
- 8678        Dice, L. R. (1938), "Some census methods for mammals," *Journal of Wildlife Management*, 2, 119–130.
- 8679        Dixon, P. (2002), "Bootstrap resampling," *Encyclopedia of environmetrics*.
- 8680        Dorazio, R. M. (2007), "On the choice of statistical models for estimating occurrence  
8681        and extinction from animal surveys," *Ecology*, 88, 2773–2782.
- 8682        Dorazio, R. M. and Royle, J. A. (2003), "Mixture models for estimating the size  
8683        of a closed population when capture rates vary among individuals," *Biometrics*,  
8684        59, 351–364.
- 8685        Durban, J. and Elston, D. (2005), "Mark-recapture with occasion and individual  
8686        effects: abundance estimation through Bayesian model selection in a fixed dimen-  
8687        sional parameter space," *Journal of agricultural, biological, and environmental  
8688        statistics*, 10, 291–305.
- 8689        Eddelbuettel, D. and François, R. (2011), "Rcpp: Seamless R and C++ Integra-  
8690        tion," *Journal of Statistical Software*, 40, 1–18.
- 8691        Efford, M. (2011), *secr: Spatially explicit capture-recapture models*, R package ver-  
8692        sion 2.3.1.
- 8693        Efford, M. G. (2004), "Density estimation in live-trapping studies," *Oikos*, 106,  
8694        598–610.
- 8695        Efford, M. G., Borchers, D. L., and Byrom, A. E. (2009a), "Density estimation by

- 8699 spatially explicit capture–recapture: likelihood-based methods,” *Modeling demo-*  
8700 *graphic processes in marked populations*, 255–269.
- 8701 Efford, M. G. and Dawson, D. K. (2009), “Effect of distance-related heterogeneity  
8702 on population size estimates from point counts,” *The Auk*, 126, 100–111.  
8703 — (2010), “SECR for acoustic data,” .
- 8704 Efford, M. G., Dawson, D. K., and Borchers, D. L. (2009b), “Population density  
8705 estimated from locations of individuals on a passive detector array,” *Ecology*, 90,  
8706 2676–2682.
- 8707 Efford, M. G., Dawson, D. K., and Robbins, C. S. (2004), “DENSITY: software for  
8708 analysing capture-recapture data from passive detector arrays,” *Animal Biodi-*  
8709 *versity and Conservation*, 27, 217–228.
- 8710 Efford, M. G. and Fewster, R. M. (2012), “Estimating population size by spatially  
8711 explicit capture–recapture,” *Oikos*.
- 8712 Efford, M. G., Warburton, B., Coleman, M. C., and Barker, R. J. (2005), “A field  
8713 test of two methods for density estimation,” *Wildlife Society Bulletin*, 33, 731–  
8714 738.
- 8715 F Dormann, C., M McPherson, J., B Araújo, M., Bivand, R., Bolliger, J., Carl, G.,  
8716 G Davies, R., Hirzel, A., Jetz, W., Daniel Kissling, W., et al. (2007), “Methods to  
8717 account for spatial autocorrelation in the analysis of species distributional data:  
8718 a review,” *Ecography*, 30, 609–628.
- 8719 Farnsworth, G. L., Pollock, K. H., Nichols, J. D., Simons, T. R., E., H. J., and  
8720 R., S. J. (2002), “A removal model for estimating detection probabilities from  
8721 point-count surveys,” *The Auk*, 119, 414–425.
- 8722 Fienberg, S. E., Johnson, M. S., and Junker, B. W. (1999), “Classical multilevel and  
8723 Bayesian approaches to population size estimation using multiple lists,” *Journal*  
8724 *of the Royal Statistical Society of London A*, 163, 383–405.
- 8725 Fiske, I. J. and Chandler, R. B. (2011), “unmarked: An R package for fitting  
8726 hierarchical models of wildlife occurrence and abundance,” *Journal Of Statistical*  
8727 *Software*, 43, 1–23.
- 8728 Foster, R. J. and Harmsen, B. J. (2012), “A critique of density estimation from  
8729 camera-trap data,” *The Journal of Wildlife Management*, 76, 224–236.
- 8730 François, R., Eddelbuettel, D., and Bates, D. (2011), *RcppArmadillo: Rcpp inte-*  
8731 *gration for Armadillo templated linear algebra library*, r package version 0.2.25.
- 8732 García-Alaníz, N., Naranjo, E. J., and Mallory, F. F. (2010), “Hair-snares: A  
8733 non-invasive method for monitoring felid populations in the Selva Lacandona,  
8734 Mexico,” *Tropical Conservation Science*, 3, 403–411.
- 8735 Gardner, B., Reppucci, J., Lucherini, M., and Royle, J. (2010a), “Spatially explicit  
8736 inference for open populations: estimating demographic parameters from camera-  
8737 trap studies,” *Ecology*, 91, 3376–3383.
- 8738 Gardner, B., Royle, J. A., and Wegan, M. T. (2009), “Hierarchical models for  
8739 estimating density from DNA mark-recapture studies,” *Ecology*, 90, 1106–1115.
- 8740 Gardner, B., Royle, J. A., Wegan, M. T., Rainbolt, R. E., and Curtis, P. D. (2010b),  
8741 “Estimating black bear density using DNA data from hair snares,” *The Journal*

- 8742       of *Wildlife Management*, 74, 318–325.
- 8743       Garshelis, D. L. and Hristienko, H. (2006), “State and provincial estimates of Amer-  
8744       ican black bear numbers versus assessments of population trend,” *Ursus*, 17, 1–7.
- 8745       Gelman, A. (2006), “Prior distributions for variance parameters in hierarchical  
8746       models,” *Bayesian analysis*, 1, 515–533.
- 8747       Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004), *Bayesian data  
8748       analysis, second edition.*, Bocan Raton, Florida, USA: CRC/Chapman & Hall.
- 8749       Gelman, A., Meng, X. L., and Stern, H. (1996), “Posterior predictive assessment  
8750       of model fitness via realized discrepancies,” *Statistica Sinica*, 6, 733–759.
- 8751       Gelman, A. and Rubin, D. B. (1992), “Inference from iterative simulation using  
8752       multiple sequences,” *Statistical Science*, 7, 457–511.
- 8753       Geman, S. and Geman, D. (1984), “Stochastic relaxation, Gibbs distributions, and  
8754       the Bayesian restoration of images,” *IEEE Transactions on Pattern Analysis and  
8755       Machine Intelligence*, PAMI-6, 721–741.
- 8756       Genz, A. S., Meyer, M. R., Lumley, T., and Maechler, M. (2007), “The adapt  
8757       Package. R package version 1.0-4.” .
- 8758       Gilks, W. R., Thomas, A., and Spiegelhalter, D. J. (1994), “A Language and Pro-  
8759       gram for Complex Bayesian Modelling,” *Journal of the Royal Statistical Society.  
8760       Series D (The Statistician)*, 43, 169–177, ArticleType: primary\_article / Issue  
8761       Title: Special Issue: Conference on Practical Bayesian Statistics, 1992 (3) / Full  
8762       publication date: 1994 / Copyright 1994 Royal Statistical Society.
- 8763       Gopalaswamy, A. M. (2012), “Capture-recapture models, spatially explicit,” in  
8764       *Encyclopedia of Environmentrics Second Edition*, eds. El-Shaarawi, A. H. and  
8765       Piegorsch, W., John Wiley and Sons Ltd, Chichester, UK.
- 8766       Gopalaswamy, A. M., Royle, A. J., Hines, J., Singh, P., Jathanna, D., Kumar,  
8767       N. S., and Karanth, K. U. (2012a), “Program SPACECAP: software for estimat-  
8768       ing animal density using spatially explicit capturerecapture models,” *Methods in  
8769       Ecology and Evolution*, online early, r package version 1.0.4.
- 8770       Gopalaswamy, A. M., Royle, J. A., Delampady, M., Nichols, J. D., Karanth, K. U.,  
8771       and Macdonald, D. W. (2012b), “Density estimation in tiger populations: com-  
8772       bining information for strong inference,” *Ecology*, 93, 1741–1751.
- 8773       Greig-Smith, P. (1964), *Quantitative plant ecology*, Butterworths (Washington).
- 8774       Hahn, T., Bouvier, A., and Kiêu, K. (2010), *R2Cuba: Multidimensional Numerical  
8775       Integration*, r package version 1.0-6.
- 8776       Hall, R. J., Henry, P. F. P., and Bunck, C. M. (1999), “Fifty-year trends in a box  
8777       turtle population in Maryland,” *Biological Conservation*, 88, 165–172.
- 8778       Hanski, I. A. (1999), *Metapopulation Ecology*, Oxford Univiversity Press.
- 8779       Hawkins, C. E. and Racey, P. A. (2005), “Low population density of a tropical forest  
8780       carnivore, Cryptoprocta ferox: implications for protected area management,”  
8781       *Oryx*, 39, 35–43.
- 8782       Hayes, R. J. and Buckland, S. T. (1983), “Radial distance models for the line  
8783       transect method.” *Biometrics*, 39, 29–42.
- 8784       Hayne, D. (1950), “Apparent home range of Microtus in relation to distance between

- 8785 traps," *Journal of mammalogy*, 31, 26–39.
- 8786 Hayne, D. W. (1949), "An Examination of the Strip Census Method for Estimating  
8787 Animal Populations," *The Journal of Wildlife Management*, 13, 145–157.
- 8788 Hedley, S. L., Buckland, S. T., and Borchers, D. L. (1999), "Spatial modelling from  
8789 line transect data," *Journal of Cetacean Research and Management*, 1, 255–264.
- 8790 Hestbeck, J. B. and Malecki, R. A. (1989), "Mark-resight estimate of Canada Goose  
8791 midwinter numbers," *Journal of Wildlife Management*, 53, 749–752.
- 8792 Higdon, D. (1998), "A process-convolution approach to modelling temperatures in  
8793 the North Atlantic Ocean," *Environmental and Ecological Statistics*, 5, 173–190.
- 8794 Hobbs, N. T. (2011), "An Ecological Modeler's Primer on JAGS," .
- 8795 Holdenried, R. (1940), "A population study of the long-eared chipmunk (*Eutamias*  
8796 *quadrimaculatus*) in the central Sierra Nevada," *Journal of Mammalogy*, 21, 405–  
8797 411.
- 8798 Hornik, K. (2011), "The R FAQ," ISBN 3-900051-08-9.
- 8799 Huggins, R. M. (1989), "On the statistical analysis of capture experiments,"  
8800 *Biometrika*, 76, 133.
- 8801 Hurlbert, S. H. (1984), "Pseudoreplication and the design of ecological field exper-  
8802 iments," *Ecological monographs*, 54, 187–211.
- 8803 Illian, J., Penttinen, A., Stoyan, H., and Stoyan, D. (2008), *Statistical analysis and*  
8804 *modelling of spatial point patterns*, Wiley.
- 8805 Ivan, J. (2012), "Density, demography, and seasonal movements of snowshoe hares  
8806 in central Colorado," Ph.D. thesis, Colorado State University.
- 8807 Ivan, J., White, G., and Shenk, T. (2013a), "Using auxiliary telemetry information  
8808 to estimate animal density from capture-recapture data," *Ecology*.
- 8809 — (2013b), "Using simulation to compare methods for estimating density from  
8810 capture-recapture data," *Ecology*.
- 8811 Jackson, R., Roe, J., Wangchuk, R., and Hunter, D. (2006), "Estimating Snow  
8812 Leopard Population Abundance Using Photography and Capture-Recapture  
8813 Techniques," *Wildlife Society Bulletin*, 34, 772–781.
- 8814 Jett, D. A. and Nichols, J. D. (1987), "A Field Comparison of Nested Grid and  
8815 Trapping Web Density Estimators," *Journal of Mammalogy*, 68, 888–892.
- 8816 Johnson (2010), "A Model-Based Approach for Making Ecological Inference from  
8817 Distance Sampling Data," *Biometrics*, 66, 310318.
- 8818 Johnson, D. H. (1999), "The insignificance of statistical significance testing," *The*  
8819 *journal of wildlife management*, 63, 763–772.
- 8820 Kadane, J. B. and Lazar, N. A. (2004), "Methods and criteria for model selection,"  
8821 *Journal of the American Statistical Association*, 99, 279–290.
- 8822 Karanth, K. U. (1995), "Estimating tiger *Panthera tigris* populations from camera-  
8823 trap data using capture–recapture models," *Biological Conservation*, 71, 333–338.
- 8824 Karanth, K. U. and Nichols, J. D. (1998), "Estimation of Tiger Densities in India  
8825 Using Photographic Captures and Recaptures," *Ecology*, 79, 2852–2862.
- 8826 — (2000), "Ecological status and conservation of tigers in India," *WCS, US Fish*  
8827 *and Wildlife Service. Centre for Wildlife Studies. Bangalore, India*, 123.

- 8828 — (2002), *Monitoring tigers and their prey: a manual for researchers, managers*  
8829 *and conservationists in {T}ropical {A}sia*, Bangalore, India: Centre for Wildlife  
8830 Studies.
- 8831 Kass, R. and Wasserman, L. (1996), “The selection of prior distributions by formal  
8832 rules,” *Journal of the American Statistical Association*, 91, 1343–1370.
- 8833 Kays, R. W., Slauson, K. M., Long, R. A., MacKay, P., Zielinski, W. J., and Ray,  
8834 J. C. (2008), “Remote cameras.” *Noninvasive survey methods for carnivores*,  
8835 110–140.
- 8836 Kelly, M., Noss, A., Di Bitetti, M., Maffei, L., Arispe, R., Paviolo, A., De Angelo,  
8837 C., and Di Blanco, Y. (2008), “Estimating puma densities from camera trapping  
8838 across three study sites: Bolivia, Argentina, and Belize,” *Journal of Mammalogy*,  
8839 89, 408–418.
- 8840 Kendall, K. C., Stetz, J. B., Boulanger, J., Macleod, A. C., Paetkau, D., and  
8841 White, G. C. (2009), “Demography and genetic structure of a recovering grizzly  
8842 bear population,” *The Journal of Wildlife Management*, 73, 3–16.
- 8843 Kendall, W. L., Nichols, J. D., and Hines, J. E. (1997), “Estimating Temporary Em-  
8844 igration Using Capture-Recapture Data with Pollock’s Robust Design,” *Ecology*,  
8845 78, 563–578.
- 8846 Kéry, M. (2010), *Introduction to WinBUGS for Ecologists: Bayesian Approach to*  
8847 *Regression, ANOVA, Mixed Models and Related Analyses*, Academic Press.
- 8848 Kéry, M. (2011), “Towards the modelling of true species distributions,” *Journal of*  
8849 *Biogeography*, 38, 617–618.
- 8850 Kéry, M., Gardner, B., and Monnerat, C. (2010), “Predicting species distributions  
8851 from checklist data using site-occupancy models,” *Journal of Biogeography*, 37,  
8852 1851–1862.
- 8853 Kéry, M., Gardner, B., Stoeckle, T., Weber, D., and Royle, J. A. (2011), “Use of  
8854 Spatial Capture-Recapture Modeling and DNA Data to Estimate Densities of  
8855 Elusive Animals,” *Conservation Biology*, 25, 356–364.
- 8856 Kéry, M., Royle, J. A., and Schmid, H. (2005), “Modeling avian abundance from  
8857 replicated counts using binomial mixture models,” *Ecological Applications*, 15,  
8858 1450–1461.
- 8859 Kéry, M. and Schaub, M. (2012), *Bayesian Population Analysis Using WinBugs*,  
8860 Academic Press.
- 8861 King, R. and Brooks, S. (2001), “On the Bayesian analysis of population size,”  
8862 *Biometrika*, 88, 317–336.
- 8863 King, R., Brooks, S., and Coulson, T. (2008), “Analyzing Complex Capture-  
8864 Recapture Data in the Presence of Individual and Temporal Covariates and  
8865 Model Uncertainty,” *Biometrics*, 64, 1187–1195.
- 8866 Knaus, J. (2010), *snowfall: Easier cluster computing (based on snow)*., r package  
8867 version 1.84.
- 8868 Kohn, M., York, E., Kamradt, D., Haught, G., Sauvajot, R., and Wayne, R. (1999),  
8869 “Estimating population size by genotyping faeces,” *Proceedings of the Royal So-*  
8870 *ciety of London. Series B: Biological Sciences*, 266, 657–663.

- 8871 Krebs, C. J. (1999), *Ecological methodology*, Menlo Park, CA: Ben-  
8872 jamin/Cummings.
- 8873 Kucera and Barrett (2011), “missing,” *missing*, missing.
- 8874 Kuo, L. and Mallick, B. (1998), “Variable selection for regression models,” *Sankhy: The Indian Journal of Statistics, Series B*, 60, 65–81.
- 8875 Laird, N. M. and Ware, J. H. (1982), “Random-effects models for longitudinal data,” *Biometrics*, 38, 963–974.
- 8876 Langtimm, C. A., Dorazio, R. M., Stith, B. M., and Doyle, T. J. (2011), “New aerial  
8877 survey and hierarchical model to estimate manatee abundance,” *The Journal of Wildlife Management*, 75, 399–412.
- 8878 Le Cam, L. (1990), “Maximum likelihood: an introduction,” *International Statistical Review/Revue Internationale de Statistique*, 58, 153–171.
- 8879 Lebreton, J. D., Burnham, K. P., Clobert, J., and Anderson, D. R. (1992), “Modeling Survival and Testing Biological Hypotheses Using Marked Animals: A Unified  
8880 Approach with Case Studies,” *Ecological Monographs*, 62, 67–118.
- 8881 Lele, S. R. and Keim, J. L. (2006), “Weighted distributions and estimation of  
8882 resource selection probability functions,” *Ecology*, 87, 3021–3028.
- 8883 Lele, S. R., Moreno, M., and Bayne, E. (2012), “Dealing with detection error in  
8884 site occupancy surveys: What can we do with a single survey?” *Journal of Plant Ecology*.
- 8885 Lewin-Koh, N. J., Bivand, R., contributions by Edzer J. Pebesma, Archer, E.,  
8886 Baddeley, A., Bibiko, H.-J., Dray, S., Forrest, D., Friendly, M., Giraudoux, P.,  
8887 Golicher, D., Rubio, V. G., Hausmann, P., Hufthammer, K. O., Jagger, T.,  
8888 Luque, S. P., MacQueen, D., Niccolai, A., Short, T., Stabler, B., and Turner,  
8889 R. (2011), *maptools: Tools for reading and handling spatial objects*, r package  
8890 version 0.8-10.
- 8891 Link, W. (2003), “Nonidentifiability of Population Size from Capture-Recapture  
8892 Data with Heterogeneous Detection Probabilities,” *Biometrics*, 59, 1123–1130.
- 8893 Link, W. A. and Barker, R. J. (1994), “Density Estimation Using the Trapping  
8894 Web Design: A Geometric Analysis,” *Biometrics*, 50, 733–745.
- 8895 — (2006), “Model weights and the foundations of multimodel inference,” *Ecology*,  
8896 87, 2626–2635.
- 8897 — (2010), *Bayesian Inference: With Ecological Applications*, London, UK: Academic Press.
- 8898 Link, W. A. and Eaton, M. J. (2011), “On thinning of chains in MCMC,” *Methods  
8899 in Ecology and Evolution*, 3, 112–115.
- 8900 Link, W. A., Yoshizaki, J., Bailey, L. L., and Pollock, K. H. (2010), “Uncovering  
8901 a latent multinomial: analysis of markrecapture data with misidentification,”  
8902 *Biometrics*, 66, 178–185.
- 8903 Liu and Wu (1999), “Parameter expansion for data augmentation,” *J Am Stat  
8904 Assoc*, 94, 1264–1274.
- 8905 Lukacs, P. M. and Burnham, K. P. (2005), “Estimating population size from DNA-  
8906 based closed capture-recapture data incorporating genotyping error,” *Journal of*

- 8914      *Wildlife Management*, 69, 396–403.
- 8915      Lunn, D., Spiegelhalter, D., Thomas, A., and Best, N. (2009), “The BUGS project:  
8916      Evolution, critique, and future directions,” *Statistics in Medicine*, 28, 3049–3067.
- 8917      Mace, R., Minta, S., Manley, T., and Aune, K. (1994), “Estimating grizzly bear  
8918      population size using camera sightings,” *Wildlife Society Bulletin*, 22, 74–83.
- 8919      MacEachern, S. N. and Berliner, L. M. (1994), “Subsampling the Gibbs sampler,”  
8920      *American Statistician*, 48, 188–190.
- 8921      MacKay, P., Smith, D. A., Long, R. A., Parker, M., Long, R. A., MacKay, P.,  
8922      Zielinski, W. J., and Ray, J. C. (2008), “Scat detection dogs,” *Noninvasive survey  
8923      methods for carnivores*, 183–222.
- 8924      MacKenzie, D. I., Nichols, J. D., Lachman, G. B., Droege, S., Royle, J. A., and  
8925      Langtimm, C. A. (2002), “Estimating site occupancy rates when detection prob-  
8926      abilities are less than one,” *Ecology*, 83, 2248–2255.
- 8927      MacKenzie, D. I., Nichols, J. D., Royle, J. A., Pollock, K. H., Bailey, L. L., and  
8928      Hines, J. E. (2006), *Occupancy estimation and modeling: inferring patterns and  
8929      dynamics of species occurrence*, Academic Press.
- 8930      Magoun, A. J., Long, C. D., Schwartz, M. K., Pilgrim, K. L., Lowell, R. E., and  
8931      Valkenburg, P. (2011), “Integrating motion-detection cameras and hair snags for  
8932      wolverine identification,” *The Journal of Wildlife Management*, 75, 731–739.
- 8933      Manly, B., McDonald, L., Thomas, D., McDonald, T., and Erickson, W. (2002),  
8934      *Resource selection by animals: statistical design and analysis for field studies*,  
8935      Springer, 2nd ed.
- 8936      Marques, T., Thomas, L., Ward, J., DiMarzio, N., and Tyack, P. (2009), “Es-  
8937      timating cetacean population density using fixed passive acoustic sensors: An  
8938      example with Blainvilles beaked whales,” *The Journal of the Acoustical Society  
8939      of America*, 125, 1982.
- 8940      McCarthy, M. A. (2007), *Bayesian Methods for Ecology*, Cambridge: Cambridge  
8941      University Press.
- 8942      McClintock, B. and Hoeting, J. (2010), “Bayesian analysis of abundance for bino-  
8943      mial sighting data with unknown number of marked individuals,” *Environmental  
8944      and Ecological Statistics*, 17, 317–332.
- 8945      McClintock, B. and White, G. (2012), “From NOREMARK to MARK: software for  
8946      estimating demographic parameters using mark-resight methodology,” *Journal of  
8947      Ornithology*, 152, 641–650.
- 8948      McClintock, B., White, G., Antolin, M., and Tripp, D. (2009a), “Estimating abun-  
8949      dance using mark-resight when sampling is with replacement or the number of  
8950      marked individuals is unknown,” *Biometrics*, 65, 237–246.
- 8951      McClintock, B., White, G., and Burnham, K. (2006), “A robust design mark-resight  
8952      abundance estimator allowing heterogeneity in resighting probabilities,” *Journal  
8953      of agricultural, biological, and environmental statistics*, 11, 231–248.
- 8954      McClintock, B. T., White, G. C., Burnham, K. P., and Pryde, M. A. (2009b),  
8955      “A generalized mixed effects model of abundance for mark-resight data when  
8956      sampling is without replacement,” in *Modeling demographic processes in marked*

- 8957      *populations*, ed. Thomson, D., New York: Springer, pp. 271–289.
- 8958      McCullagh, P. and Nelder, J. A. (1989), *Generalized linear models*, Chapman &  
8959      Hall/CRC.
- 8960      Millar, R. B. (2009), “Comparison of hierarchical Bayesian models for overdispersed  
8961      count data using DIC and Bayes’ Factors,” *Biometrics*, 65, 962–969.
- 8962      Mills, L. S., Citta, J. J., Lair, K. P., Schwartz, M. K., and Tallmon, D. A. (2000),  
8963      “Estimating animal abundance using noninvasive DNA sampling: promise and  
8964      pitfalls,” *Ecological Applications*, 10, 283–294.
- 8965      Minta, S. and Mangel, M. (1989), “A Simple Population Estimate Based on Simula-  
8966      tion for Capture-Recapture and Capture-Resight Data,” *Ecology*, 70, 1738–1751.
- 8967      Møller, J. and Waagepetersen, R. P. (2004), *Statistical inference and simulation  
8968      for spatial point processes*, Chapman & Hall/CRC.
- 8969      Mohr, C. (1947), “Table of equivalent populations of North American small mam-  
8970      mals,” *American midland naturalist*, 37, 223–249.
- 8971      Moldenhauer, R. R. and Regelski, D. J. (1996), “Northern Parula (*Parula ameri-  
8972      cana*),” in *The Birds of North America Online*, ed. Poole, A., Ithaca, NY: Cornell  
8973      Lab of Ornithology.
- 8974      Molinari-Jobin, A., Kéry, M., Marboutin, E., Marucco, F., Zimmermann, F., Molini-  
8975      nari, P., Frick, H., Wölfl, S., Bled, F., Breitenmoser-Würsten, C., Fuxjäger, C.,  
8976      Huber, T., I., K., Kos, I., Manfred Wölfl, M., and Breitenmoser, U. (2013),  
8977      “Mapping range dynamics from opportunistic data: spatio-temporal distribu-  
8978      tion modeling of lynx *Lynx lynx* L. in the Alps,” *Biological Conservation*, xx,  
8979      xxxx–xxxx.
- 8980      Neal, A., White, G., Gill, R., Reed, D., and Olterman, J. (1993), “Evaluation  
8981      of mark-resight model assumptions for estimating mountain sheep numbers,”  
8982      *Journal of Wildlife Management*, 57, 436–450.
- 8983      Neal, A. K. (1990), “Evaluation of mark-resight population estimates using simula-  
8984      tions and field data from mountain sheep.” M.S. thesis, Colorado State University,  
8985      Fort Collins, Colorado, USA.
- 8986      Nelder, J. A. and Wedderburn, R. W. M. (1972), “Generalized linear models,”  
8987      *Journal of the Royal Statistical Society. Series A (General)*, 135, 370–384.
- 8988      Nichols, J. D., Hines, J. E., Sauer, J. R., Fallon, F. W., Fallon, J. E., and Heglund,  
8989      P. J. (2000), “A double-observer approach for estimating detection probability  
8990      and abundance from point counts,” *The Auk*, 117, 393–408.
- 8991      Nichols, J. D. and Karanth, K. U. (2002), “Statistical concepts: assessing spatial  
8992      distributions,” in *Monitoring tigers and their prey: a manual for researchers,  
8993      managers and conservationists in Tropical Asia*, eds. Karanth, K. U. and Nichols,  
8994      J. D., Bangalore, India: Centre for Wildlife Studies, pp. 29–38.
- 8995      Niemi, A. and Fernández, C. (2010), “Bayesian Spatial Point Process Modeling  
8996      of Line Transect Data,” *Journal of Agricultural, Biological, and Environmental  
8997      Statistics*, 15, 327–345.
- 8998      Norris, J. L. and Pollock, K. H. (1996), “Nonparametric MLE under two closed  
8999      capture-recapture models with heterogeneity,” *Biometrics*, 52, 639–649.

- 9000 O'Brien, T. (2011), "Abundance, density and relative abundance: A conceptual  
9001 framework," in *Camera traps in animal ecology: methods and analyses*, eds.  
9002 O'Connell, A. F. J., Nichols, J. D., and Karanth, U., Tokyo, Japan: Springer  
9003 Verlag, pp. 71–96.
- 9004 O'Connell, A. F., Nichols, J. D., and Karanth, U. K. (2010), *Camera traps in*  
9005 *animal ecology: Methods and analyses*, Springer.
- 9006 O'Hara, R. and Sillanpää, M. (2009), "A review of Bayesian variable selection  
9007 methods: what, how and which," *Bayesian Analysis*, 4, 85–118.
- 9008 Otis, D. L., Burnham, K. P., White, G. C., and Anderson, D. R. (1978), "Statistical  
9009 inference from capture data on closed animal populations," *Wildlife monographs*,  
9010 3–135.
- 9011 Ovaskainen, O. (2004), "Habitat-specific movement parameters estimated using  
9012 mark-recapture data and a diffusion model," *Ecology*, 85, 242–257.
- 9013 Ovaskainen, O., Rekola, H., Meyke, E., and Arjas, E. (2008), "Bayesian methods for  
9014 analyzing movements in heterogeneous landscapes from mark-recapture data,"  
9015 *Ecology*, 89, 542–554.
- 9016 Parmenter, R. R. and MacMahon, J. A. (1989), "Animal Density Estimation Using  
9017 a Trapping Web Design: Field Validation Experiments," *Ecology*, 70, 169–179.
- 9018 Parmenter, R. R., Yates, T. L., Anderson, D. R., Burnham, K. P., Dunnum, J. L.,  
9019 Franklin, A. B., Friggens, M. T., Lubow, B. C., Miller, M., Olson, G. S., and  
9020 Others (2003), "Small-mammal density estimation: A field comparison of grid-  
9021 based vs. web-based density estimators," *Ecological Monographs*, 73, 1–26.
- 9022 Pebesma, E. and Bivand, R. (2011), *Package 'sp'*, r package version 0.9-91.
- 9023 Pledger, S. (2004), "Unified maximum likelihood estimates for closed capture-  
9024 recapture models using mixtures," *Biometrics*, 56, 434–442.
- 9025 Plummer, M. (2003), "JAGS: A program for analysis of Bayesian graphical models  
9026 using Gibbs sampling," in *Proceedings of the 3rd International Workshop on*  
9027 *Distributed Statistical Computing (DSC 2003)*. March, pp. 20–22.
- 9028 — (2009), "rjags: Bayesian graphical models using mcmc. R package version 1.0  
9029 3-12," .
- 9030 — (2011), "rjags: Bayesian graphical models using mcmc. R package version 3-5,"  
9031 .
- 9032 Plummer, M., Best, N., Cowles, K., and Vines, K. (2006), "CODA: Convergence  
9033 Diagnosis and Output Analysis for MCMC," *R News*, 6, 7–11.
- 9034 Pollock, K. H. (1982), "A Capture-Recapture Design Robust to Unequal Probability  
9035 of Capture." *Journal of Wildlife Management*, 46, 752–757.
- 9036 Pollock, K. H., Nichols, J. D., Brownie, C., and Hines, J. E. (1990), "Statistical  
9037 inference for capture-recapture experiments," *Wildlife monographs*, 3–97.
- 9038 Reich, B. J., Gardner, B., and Wilting, A. (2012), "A spatial capture-recapture  
9039 model for territorial species," *Biometrics in review*, xx, xx–xxx.
- 9040 Robert, C. P. and Casella, G. (2010), *Introducing Monte Carlo Methods with R*,  
9041 New York, USA: Springer.
- 9042 Royle, J., Kéry, M., and Guélat, J. (2011a), "Spatial capture-recapture models for

- 9043     search-encounter data," *Methods in Ecology and Evolution*, 2, 602–611.
- 9044     Royle, J. A. (2004a), "Generalized estimators of avian abundance from count survey  
9045        data," *Animal Biodiversity and Conservation*, 27, 375–386.
- 9046     — (2004b), "N-Mixture Models for Estimating Population Size from Spatially  
9047        Replicated Counts," *Biometrics*, 60, 108–115.
- 9048     — (2006), "Site occupancy models with heterogeneous detection probabilities,"  
9049        *Biometrics*, 62, 97–102.
- 9050     — (2008), "Modeling individual effects in the Cormack–Jolly–Seber model: a state–  
9051        space formulation," *Biometrics*, 64, 364–370.
- 9052     — (2009), "Analysis of capture-recapture models with individual covariates using  
9053        data augmentation," *Biometrics*, 65, 267–274.
- 9054     Royle, J. A. and Chandler, R. B. (2012), "Integrating Resource Selection Informa-  
9055        tion with Spatial Capture-Recapture," *arXiv preprint arXiv:1207.3288*.
- 9056     Royle, J. A., Chandler, R. B., Gazenski, K. D., and Graves, T. A. (2012a), "Spa-  
9057        tial capture-recapture for jointly estimating population density and landscape  
9058        connectivity," *Ecology*, to appear.
- 9059     Royle, J. A., Chandler, R. B., Sun, C., and Fuller, A. (2012b), "Integrating Resource  
9060        Selection Information with Spatial Capture-Recapture," *needs updated MEE*.
- 9061     Royle, J. A. and Converse, S. J. (in review), "Hierarchical spatial capture-recapture  
9062        models: Modeling population density from replicated capture-recapture experi-  
9063        ments," *Ecology*.
- 9064     Royle, J. A., Converse, S. J., and Link, W. A. (2012c), "Data Augmentation for  
9065        Hierarchical Capture-recapture Models," *arXiv preprint arXiv:1211.5706*.
- 9066     Royle, J. A., Dawson, D. K., and Bates, S. (2004), "Modeling abundance effects in  
9067        distance sampling," *Ecology*, 85, 1591–1597.
- 9068     Royle, J. A. and Dorazio, R. M. (2006), "Hierarchical models of animal abundance  
9069        and occurrence," *Journal of Agricultural, Biological, and Environmental Statis-  
9070        tics*, 11, 249–263.
- 9071     — (2008), *Hierarchical modeling and inference in ecology: the analysis of data from  
9072        populations, metapopulations and communities*, Academic Press.
- 9073     — (2012), "Parameter-expanded data augmentation for Bayesian analysis of  
9074        capture-recapture models," *Journal of Ornithology*, 152, S521–S537.
- 9075     Royle, J. A., Dorazio, R. M., and Link, W. A. (2007), "Analysis of multinomial  
9076        models with unknown index using data augmentation," *Journal of Computational  
9077        and Graphical Statistics*, 16, 67–85.
- 9078     Royle, J. A. and Dubovský, J. A. (2001), "Modeling spatial variation in waterfowl  
9079        band-recovery data," *The Journal of wildlife management*, 65, 726–737.
- 9080     Royle, J. A. and Gardner, B. (2011), "Hierarchical models for estimating density  
9081        from trapping arrays," in *Camera traps in animal ecology: methods and analyses*,  
9082        eds. O'Connel, A. F. J., Nichols, J. D., and Karanth, U., Tokyo, Japan: Springer  
9083        Verlag, pp. 163–190.
- 9084     Royle, J. A., Karanth, K. U., Gopalaswamy, A. M., and Kumar, N. S. (2009a),  
9085        "Bayesian inference in camera trapping studies for a class of spatial capture–

- 9086      recapture models," *Ecology*, 90, 3233–3244.
- 9087      Royle, J. A. and Kéry, M. (2007), "A Bayesian state-space formulation of dynamic  
9088      occupancy models." *Ecology*, 88, 1813–1823.
- 9089      Royle, J. A. and Link, W. A. (2006), "Generalized site occupancy models allowing  
9090      for false positive and false negative errors," *Ecology*, 87, 835–841.
- 9091      Royle, J. A., Magoun, A. J., Gardner, B., Valkenburg, P., and Lowell, R. E. (2011b),  
9092      "Density estimation in a wolverine population using spatial capture–recapture  
9093      models," *The Journal of Wildlife Management*, 75, 604–611.
- 9094      Royle, J. A. and Nichols, J. D. (2003), "Estimating abundance from repeated  
9095      presence-absence data or point counts," *Ecology*, 84, 777–790.
- 9096      Royle, J. A., Nichols, J. D., Karanth, K. U., and Gopalaswamy, A. M. (2009b),  
9097      "A hierarchical model for estimating density in camera-trap studies," *Journal of  
9098      Applied Ecology*, 46, 118–127.
- 9099      Royle, J. A. and Young, K. V. (2008), "A Hierarchical Model For Spatial Capture–  
9100      Recapture Data," *Ecology*, 89, 2281–2289.
- 9101      Russell, R. E., Royle, J. A., Desimone, R., Schwartz, M. K., Edwards, V. L.,  
9102      Pilgrim, K. P., and McKelvey, K. S. (2012), "Estimating abundance of mountain  
9103      lions from unstructured spatial samples," *Journal of Wildlife Management*.
- 9104      Rutledge, M. (2012), "XXX to be determined XXX," Ph.D. thesis, North Carolina  
9105      State University.
- 9106      Salom-Pérez, R., Carrillo, E., Sáenz, J., and Mora, J. (2007), "Critical condition of  
9107      the jaguar *Panthera onca* population in Corcovado National Park, Costa Rica,"  
9108      *Oryx*, 41, 51.
- 9109      Sanathanan, L. (1972), "Estimating the size of a multinomial population," *The  
9110      Annals of Mathematical Statistics*, 43, 142–152.
- 9111      Schofield, M. and Barker, R. (2008), "A unified capture-recapture framework,"  
9112      *Journal of agricultural, biological, and environmental statistics*, 13, 458–477.
- 9113      Schwartz, M. K. and Monfort, S. L. (2008), *Genetic and endocrine tools for carni-  
9114      vore surveys*, Island Press Washington, DC, USA.
- 9115      Seber, G. A. F. (1982), *The estimation of animal abundance and related parameters*,  
9116      Macmillan Publishing Co.
- 9117      Sepúlveda, M. A., Bartheld, J. L., Monsalve, R., Gómez, V., and Medina-Vogel,  
9118      G. (2007), "Habitat use and spatial behaviour of the endangered Southern river  
9119      otter (*Lontra provocax*) in riparian habitats of Chile: conservation implications,"  
9120      *Biological Conservation*, 140, 329–338.
- 9121      Sillett, S., Chandler, R. B., Royle, J. A., Kéry, M., and Morrison, S. (2012), "Hierar-  
9122      chical distance sampling models to estimate population size and habitat-specific  
9123      abundance of an island endemic," *Ecological Applications*.
- 9124      Simons, T. R., Pollock, K. H., Wetters, J. M., Alldredge, M. W., Pacifici, K., and  
9125      Brewster, J. (2009), "Sources of Measurement Error, Misclassification Error, and  
9126      Bias in Auditory Avian Point Count Data," in *Modeling Demographic Processes  
9127      In Marked Populations*, eds. Thomson, D. L., Cooch, E. G., and Conroy, M. J.,  
9128      Boston, MA: Springer US, vol. 3, pp. 237–254.

- 9129 Skalski, J. R., Millspaugh, J. J., and Spencer, R. D. (2005), "Population estimation and biases in paintball, mark-resight surveys of elk," *Journal of Wildlife Management*, 69, 1043–1052.
- 9130
- 9131
- 9132 Sklyar, O., Murdoch, D., Smith, M., Eddelbuettel, D., and François, R. (2010), *inline: Inline C, C++, Fortran function calls from R*, r package version 0.3.8.
- 9133
- 9134 Smith, M. H., Blessing, R., Chelton, J. G., Gentry, J. B., Golley, F. B., and McGinnis, J. T. (1971), "Determining density for small mammal populations using a grid and assessment lines," *Acta Theriologica*, 16, 105–125.
- 9135
- 9136
- 9137 Soisalo, M. K. and Cavalcanti, S. M. C. (2006), "Close-up Space in "Radio-telemetry"," *Biological Conservation*, 129, 487–496.
- 9138
- 9139 Sollmann, R., Furtado, M. M., Gardner, B., Hofer, H., Jacomo, A. T. A., Trzes, N. M., and Silveira, L. (2011), "Improving density estimates for elusive carnivores: Accounting for sex-specific detection and movements using spatial capture-recapture models for jaguars in central Brazil," *Biological Conservation*, 144, 1017–1024.
- 9140
- 9141
- 9142
- 9143
- 9144 Sollmann, R., Gardner, B., Chandler, R. B., Shindle, D., Onorato, D. P., Royle, J. A., and O'Connell, A. F. (in revision), "Using multiple data sources provides density estimates for endangered Florida panther," *Journal of Applied Ecology*.
- 9145
- 9146
- 9147 Sollmann, R., Gardner, B., Parsons, A., Stocking, J., McClintock, B., Simons, T., Pollock, K., and O'Connell, A. (2012), "A spatial mark-resight model augmented with telemetry data," *Ecology*.
- 9148
- 9149
- 9150 Sólymos, P., Lele, S., and Bayne, E. (2012), "Conditional likelihood approach for analyzing single visit abundance survey data in the presence of zero inflation and detection error," *Environmetrics*, 23, 197–205.
- 9151
- 9152
- 9153 Spiegelhalter, D., Best, N., Carlin, B., and Van Der Linde, A. (2002), "Bayesian measures of model complexity and fit," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64, 583–639.
- 9154
- 9155
- 9156 Stabler, B. (2006), *shapefiles: Read and Write ESRI Shapefiles*, r package version 0.6.
- 9157
- 9158 Stanley, T. and Burnham, K. (1999), "A closure test for time-specific capture-recapture data," *Environmental and Ecological Statistics*, 6, 197–209.
- 9159
- 9160 Stoyan, D. and Penttinen, A. (2000), "Recent Applications of Point Process Methods in Forestry Statistics," *Statistical Science*, 15, 61–78.
- 9161
- 9162 Sturtz, S., Ligges, U., and Gelman, A. (2005), "R2WinBUGS: A Package for Running WinBUGS from R," *Journal of Statistical Software*, 12, 1–16.
- 9163
- 9164 Su, Y.-S. and Yajima, M. (2011), *R2jags: A Package for Running jags from R*, r package version 0.02-17.
- 9165
- 9166 Taberlet, P. and Bouvet, J. (1992), "Bear conservation genetics," *Nature*, 358, 197–197.
- 9167
- 9168 Tanner, M. A. and Wong, W. H. (1987), "The calculation of posterior distributions by data augmentation," *J Am Stat Assoc*, 82, 528–540.
- 9169
- 9170 Thomas, A., O'Hara, B., Ligges, U., and Sturtz, S. (2006), "Making BUGS Open," *R News*, 6, 12–17.
- 9171

- 9172 Thomas, M. (1949), "A generalization of Poisson's binomial limit for use in ecology,"  
9173     *Biometrika*, 36, 18–25.
- 9174 Thompson, C., Royle, J. A., and Garner, J. (2012), "A framework for inference  
9175     about carnivore density from unstructured spatial sampling of scat using detector  
9176     dogs," *The Journal of Wildlife Management*, 76, 863–871.
- 9177 Tierney, L., Rossini, A. J., Li, N., and Sevcikova, H. (2011), *snow: Simple Network  
9178     of Workstations*, r package version 0.3-7.
- 9179 Tilman, D. and Kareiva, P. (1997), *Spatial ecology: the role of space in population  
9180     dynamics and interspecific interactions*, vol. 30, Princeton University Press.
- 9181 Tobler, M. W., Hibert, F., Debeir, L., and Hansen, C. (2012), "Density and sus-  
9182     tainable harvest estimates for the lowland tapir in the Amazon of French Guiana  
9183     using a spatial capture-recapture model," *none yet*, none, none.
- 9184 Trolle, M. and Kéry, M. (2003), "Estimation of ocelot density in the Pantanal using  
9185     capture-recapture analysis of camera-trapping data," *Journal of Mammalogy*, 84,  
9186     607–614.
- 9187 — (2005), "Camera-trap study of ocelot and other secretive mammals in the north-  
9188     ern Pantanal," *Mammalia*, 69, 409–416.
- 9189 Tyre, A. J., Tenhumberg, B., Field, S. A., Niejalke, D., Parris, K., and Possing-  
9190     ham, H. P. (2003), "Improving precision and reducing bias in biological surveys:  
9191     estimating false-negative error rates," *Ecological Applications*, 13, 1790–1801.
- 9192 Valiere, N. and Taberlet, P. (2000), "Urine collected in the field as a source of DNA  
9193     for species and individual identification," *Molecular Ecology*, 9, 2150–2152.
- 9194 van Etten, J. (2011), *Package gdistance*, r package version 1.1-2.
- 9195 van Etten, R. J. H. . J. (2012), *raster: Geographic analysis and modeling with raster  
9196     data*, r package version 1.9-67.
- 9197 Venables, W. and Ripley, B. (2002), *Modern applied statistics with S*, Springer  
9198     verlag.
- 9199 Venables, W., Smith, D., and Team, R. D. C. (2012), "An introduction to R," .
- 9200 Ver Hoef, J. and Boveng, P. (2007), "Quasi-poisson vs. negative binomial regression:  
9201     How should we model overdispersed count data?" *Ecology*, 88, 2766–2772.
- 9202 Ver Hoef, J. M. (2012), "Who Invented the Delta Method?" *The American Statis-  
9203     tistician*, 66, 124–127.
- 9204 Wallace, R. B., Gomez, H., Ayala, G., and Espinoza, F. (2003), "Camera trapping  
9205     for jaguar (*Panthera onca*) in the Tuichi Valley, Bolivia," *Journal of Neotropical  
9206     Mammalogy*, 10, 133–139.
- 9207 Wegan, M., Curtis, P., Rainbolt, R., and Gardner, B. (2012), "Temporal sampling  
9208     frame selection in DNA-based capture mark-recapture investigations." *Ursus*, 23,  
9209     42 – 51.
- 9210 Wegan, M. T. (2008), "Aversive conditioning, population estimation, and habitat  
9211     preference of black bears (*Ursus americanus*) on Fort Drum Military Installation  
9212     in northern New York," Ph.D. thesis, Cornell University, Jan.
- 9213 White, G. (1996), "NOREMARK: population estimation from mark-resighting sur-  
9214     veys," *Wildlife Society Bulletin*, 24, 50–52.

- 9215 White, G. and Bennetts, R. (1996), "Analysis of frequency count data using the  
9216 negative binomial distribution," *Ecology*, 77, 2549–2557.
- 9217 White, G. and Shenk, M. (2001), "Poplation estimation with radio-marked inividu-  
9218 als." in *Radio tracking adn animal populations*, eds. Millspaugh, J. and Marzluff,  
9219 J., San Diego, USA: Academic Press, pp. 329–350.
- 9220 White, G. C., Anderson, D. R., Burnham, K. P., and Otis, D. (1982), *Capture-  
9221 recapture and removal methods for sampling closed populations*, Los Alamos: Los  
9222 Alamos National Laboratory.
- 9223 White, G. C. and Garrot, R. (1990), *Analysis of wildlife radiolocation data*, New  
9224 York, USA: Academic Press.
- 9225 White, G. C. and Shenk, T. M. (2000), *Population estimation with radio-marked  
9226 animals*, San Diego, California: Academic Press.
- 9227 Whitman, J., Ballard, W., and Gardner, C. (1986), "Home range and habitat use  
9228 by wolverines in southcentral Alaska," *The Journal of wildlife management*, 460–  
9229 463.
- 9230 Wickham and Hadley (2007), "Reshaping data with the reshape package," *Journal  
9231 of Statistical Software*, 21.
- 9232 Williams, B. K., Nichols, J. D., and Conroy, M. J. (2002), *Analysis and management  
9233 of animal populations: modeling, estimation, and decision making*, Academic Pr.
- 9234 Wilson, K. R. and Anderson, D. R. (1985a), "Evaluation of a Density Estimator  
9235 Based on a Trapping Web and Distance Sampling Theory," *Ecology*, 66, 1185–  
9236 1194.
- 9237 — (1985b), "Evaluation of Two Density Estimators of Small Mammal Population  
9238 Size," *Journal of Mammalogy*, 66, 13–21.
- 9239 Wolpert, R. L. and Ickstadt, K. (1998), "Poisson/gamma random field models for  
9240 spatial statistics," *Biometrika*, 85, 251 –267.
- 9241 Woods, J. G., Paetkau, D., Lewis, D., McLellan, B. N., Proctor, M., and Strobeck,  
9242 C. (1999), "Genetic tagging of free-ranging black and brown bears," *Wildlife  
9243 Society Bulletin*, 27, 616–627.
- 9244 Wright, J., Barker, R., Schofield, M., Frantz, A., Byrom, A., and Gleeson, D.  
9245 (2009), "Incorporating Genotype Uncertainty into Mark–Recapture-Type Models  
9246 For Estimating Abundance Using DNA Samples," *Biometrics*, 65, 833–840.
- 9247 Yang, H. C. and Chao, A. (2005), "Modeling Animals' Behavioral Response by  
9248 Markov Chain Models for Capture–Recapture Experiments," *Biometrics*, 61,  
9249 1010–1017.
- 9250 Yoshizaki, J., Pollock, K. H., Brownie, C., and Webster, R. A. (2009), "Modeling  
9251 misidentification errors in capture-recapture studies using photographic identifi-  
9252 cation of evolving marks," *Ecology*, 90, 3–9.
- 9253 Zuur, A. F., Ieno, E. N., Walker, N. J., Saveliev, A. A., and Smith, G. M. (2009),  
9254 *Mixed effects models and extensions in ecology with R*, Springer Verlag.
- 9255 Zylstra, E., Steidl, R., and Swann, D. (2010), "Evaluating survey methods for mon-  
9256 itoring a rare vertebrate, the Sonoran desert tortoise," *The Journal of Wildlife  
Management*, 74, 1311–1318.