

¹

² **Spatial Capture-Recapture**

³ **J. Andrew Royle**
⁴ **Richard B. Chandler**
⁵ **Rahel Sollmann**
⁶ **Beth Gardner**

⁷ USGS Patuxent Wildlife Research Center
⁸ North Carolina State University

⁹ ACADEMIC PRESS

CONTENTS

10

11

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

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

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

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

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

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

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

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

319	16.5 Summary and Outlook	425
320	IV Super-Advanced SCR Models	427
321	17 Developing Markov Chain Monte Carlo Samplers	429
322	17.1 Why Build Your Own MCMC Algorithm?	429
323	17.2 MCMC and Posterior Distributions	430
324	17.3 Types of MCMC Sampling	432
325	17.3.1 Gibbs sampling	432
326	17.3.2 Metropolis-Hastings sampling	436
327	17.3.3 Metropolis-within-Gibbs	440
328	17.3.4 Rejection sampling and slice sampling	444
329	17.4 MCMC for Closed Capture-Recapture Model M_h	445
330	17.5 MCMC Algorithm for Model SCR0	448
331	17.5.1 SCR model with binomial encounter process	450
332	17.6 Looking at Model Output	451
333	17.6.1 Markov chain time series plots	452
334	17.6.2 Posterior density plots	452
335	17.6.3 Serial autocorrelation and effective sample size	454
336	17.6.4 Summary results	455
337	17.6.5 Other useful commands	456
338	17.7 Manipulating the State-Space	457
339	17.8 Increasing Computational Speed	460
340	17.8.1 Parallel computing	461
341	17.8.2 Using C++	463
342	17.9 Summary and Outlook	465
343	18 Unmarked Populations	469
344	18.1 Existing Models for Inference About Density in Unmarked Populations	470
345	18.2 Spatial Correlation in Count Data	472
346	18.2.1 Spatial correlation as information	472
347	18.2.2 Types of spatial correlation	473
348	18.3 Spatial Count Model	474
349	18.3.1 Data	474
350	18.3.2 Model	474
351	18.3.3 On N being unknown	476
352	18.3.4 Inference	476
353	18.4 How Much Correlation Is Enough?	477
354	18.5 Applications	477
355	18.5.1 Simulation example	477
356	18.5.2 Northern parula in Maryland	482
357	18.6 Extensions of the Spatial Count Model	483

358	18.6.1 Improving Precision	483
359	18.6.2 Dusky Salamanders in Maryland	485
360	18.7 Summary and Outlook	487
361	19 Spatial Mark-Resight Models	491
362	19.1 Background	493
363	19.1.1 Resighting techniques	493
364	19.1.2 Types of mark-resighting data	493
365	19.1.3 A short history of mark-resight models	494
366	19.1.4 The random sample assumption	496
367	19.2 Known number of marked individuals	498
368	19.2.1 Implementing spatial mark-resight models	499
369	19.3 Unknown number of marked individuals	502
370	19.3.1 Canada geese in North Carolina	503
371	19.4 Imperfect identification of marked individuals	505
372	19.5 How Much Information Do Marked and Unmarked Individuals Contribute?	508
373	19.6 Incorporating telemetry data	510
374	19.6.1 Raccoons on the Outer Banks of North Carolina	512
375	19.7 Point process models for marked individuals	514
376	19.7.1 Homogeneous point process in a subset of \mathcal{S}	514
377	19.7.2 Inhomogenous point processes	515
378	19.8 Summary and Outlook	518
380	20 2012: A Spatial Capture-Recapture Odyssey	521
381	20.1 Emerging Topics	524
382	20.1.1 Modeling territoriality	524
383	20.1.2 Combining data from different surveys	525
384	20.1.3 Misidentification	526
385	20.1.4 Gregarious species	527
386	20.1.5 Single Catch Traps	527
387	20.1.6 Model Fit and Selection	528
388	20.1.7 Explicit movement models	528
389	20.2 Final Remarks	529
390	V Appendices	531
391	20.3 WinBUGS	533
392	20.3.1 WinBUGS through R	533
393	20.4 OpenBUGS	534
394	20.4.1 OpenBUGS through R	534
395	20.5 JAGS	534
396	20.5.1 JAGS through R	535

397	20.6 R	536
398	20.6.1 R packages	536
399	Bibliography	539

400 Foreword
401

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



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

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

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

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

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

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

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

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

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

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

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

501 Capture-Recapture will be an extremely important book.

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

515 The second major section, “Basic SCR Models”, begins with a complete development of SCRs as hierarchical models with observation and spatial point process components. Included is a clear discussion of space use by animals, important because any model of the detection process implies a model for space use. A chapter is devoted to likelihood analysis of SCR models including both model development and an introduction to software available for fitting models. Another chapter is devoted to various approaches to modeling variation in encounter probability. A variety of basic models is introduced, as well as approaches to modeling covariates associated with traps, time, individual capture history, and individual animals (e.g., sex, body mass, random effects models as well). The chapter on model selection and assessment does not provide an omnibus, one-size-fits-all statistic. Rather, it describes useful approaches including AIC for likelihood analyses and both DIC and the Kuo and Mallick (1998) indicator variable approach for Bayesian analyses. For assessing model adequacy, they use the Bayesian p-value approach (Gelman et al., 529 1996) applied to different components of model fit. Another chapter is devoted to the encounter process which requires attention to the nature of the detection device (e.g., can an animal be caught only once or multiple times during an occasion, do traps permit catches of multiple or only single individuals, can an individual be detected multiple times by the same device) and the kinds of data produced by these devices. The final chapter in this section deals with the important topic of study design. A fundamental design trade-off involves the competing needs to capture a good number of animals (sample size) and to attain a reasonably high average capture probability, and the authors emphasize the need for designs that represent a good compromise rather than those that emphasize one component to the exclusion of the other. General recommendations about trap spacing and clustering, and use of ancillary data (telemetry) are discussed as well. The material in this section is extremely important in conveying the basic principles underlying SCR modeling and, as such, will be the section of primary interest to many readers.

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

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

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

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

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

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

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

597 James D. Nichols
598 Patuxent Wildlife Research Center
599 March 14, 2013

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

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

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

THEMES OF THIS BOOK

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

655 The conceptual and methodological themes of this book can be summarized as
 656 follows:

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

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

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

COMPUTING

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

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

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

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

730 **The R package scrbook**

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

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

ORGANIZATION OF THIS BOOK

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

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

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

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

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

776 The 3rd section of the book covers more advanced SCR models. We have a
777 number of chapters on spatial modeling aspects related to SCR, including mod-
778eling spatial variation in density (Chapt. 11, modeling landscape connectivity or
779 “ecological distance” using SCR models (Chapt. 12), and modeling space usage
780 or resource selection (Chapt 13), which includes material on integrating telemetry
781 data into SCR models. After this there are a series of 3 chapters that involve
782 some elements of modeling spatially or temporally stratified populations. We cover
783 Bayesian multi-session models in Chapt. 14, what we call “search-encounter” mod-
784els in Chapt. 15 and, finally, fully open models involving movement or population
785 dynamics in Chapt. 16. The reason we view the search-encounter models chap-
786 ter, Chapt. 15, as a prelude to fully open models is that these models apply to
787 situations where we observe the animal locations “unbiased by fixed sampling lo-
788cations” – so we get to observe clean measurements of movement outcomes. When
789 this is possible, we can resolve parameters of explicit movement models free of those
790 that involve encounter probability. For example, one such models has two “scale”
791 parameters: σ that determines the rate of decay in encounter probability from a
792 sampling point or line, and τ which is the standard deviation of movements about
793 an individuals activity center.

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

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

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

812 Acknowledgements
813

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

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

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

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

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

853 with encouragement and feedback throughout this project and we thank them for
854 their continued support.

855

Part I

856

857

Background and Concepts

858
859

860

1

INTRODUCTION

861 Space plays a vital role in virtually all ecological processes (Tilman and Kareiva,
862 1997; Hanski, 1999; Clobert et al., 2001). The spatial arrangement of habitat can
863 influence movement patterns during dispersal, habitat selection, and survival. The
864 distance between an organism and its competitors and prey can influence activity
865 patterns and foraging behavior. Further, understanding distribution and spatial
866 variation in abundance is necessary in the conservation and management of popu-
867 lations. The inherent spatial aspect of *sampling* populations also plays an important
868 role in ecology as it strongly affects, and biases, how we observe population struc-
869 ture (Seber, 1982; Buckland et al., 2001; Borchers et al., 2002; Williams et al.,
870 2002). However, despite the central role of space and spatial processes to both
871 understanding population dynamics and how we observe or sample populations, a
872 coherent framework that integrates these two aspects of ecological systems has not
873 been fully realized either conceptually or methodologically.

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

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

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

1.1 THE STUDY OF POPULATIONS BY CAPTURE-RECAPTURE

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

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

912 Capture-recapture has been important in studies of animal populations for many
913 decades, and its importance is growing dramatically in response to technological
914 advances that improve our ability and efficiency to obtain encounter history data.
915 Historically, such information was obtainable using methods requiring physical cap-
916 ture of individuals. However, new methods do not require physical capture or
917 handling of individuals. A large number of passive detection devices produce indi-
918 vidual encounter history data including camera traps (Karanth and Nichols, 1998;
919 O'Connell et al., 2010), acoustic recording devices (Dawson and Efford, 2009), and
920 methods that obtain DNA samples such as hair snares for bears, scent posts for
921 many carnivores, and related methods which allow DNA to be extracted from scat,
922 urine or animal tissue in order to identify individuals. This book is concerned with
923 how such data can be used to carry out inference about animal abundance or den-
924 sity, and other parameters such as survival, recruitment, resource selection, and
925 movement using new classes of capture-recapture models which utilize auxiliary
926 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 DATA⁵

927 spatial capture-recapture (SCR) models¹.

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

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

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

953 1.2.1 Camera trapping

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

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

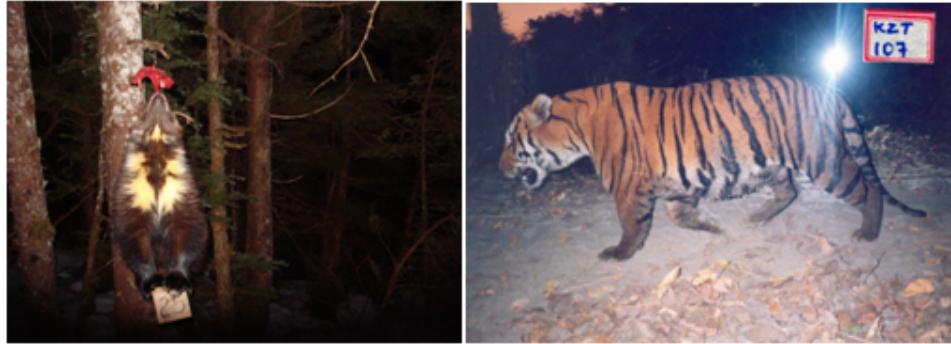


Figure 1.1. Left: Wolverine being encounter by a camera trap (Photo credit: Audrey Magoun). Right: Tiger encountered by camera trap (Photo credit: Ullas Karanth).

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

1.2.2 DNA sampling

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



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

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

1.2.3 Acoustic sampling

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

1.2.4 Search-encounter methods

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



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

1007 that involve reptiles and amphibians, e.g., we might walk transects picking up box
 1008 turtles (Hall et al., 1999), or desert tortoises (Zylstra et al., 2010), or search space
 1009 for lizards (Royle and Young, 2008).

1010 These methods don't seem like normal capture-recapture in the sense that the
 1011 encounter of individuals is not associated with specific trap location, but SCR
 1012 models are equally relevant for analysis of such data as we discuss in Chapt. 15.

1.3 CAPTURE-RECAPTURE FOR MODELING ENCOUNTER PROBABILITY

1013 We briefly introduced techniques used for the study of animal populations. These
 1014 methods produce individual encounter history data, a record of where and when
 1015 each individual was captured. We refer to this as a *spatial encounter history*. Histori-
 1016 cally, auxiliary spatial information has been ignored, and encounter history data
 1017 have been *summarized* to simple “encounter or not” for the purpose of applying
 1018 ordinary CR models. The basic problem with these ordinary (or “non-spatial”)
 1019 capture-recapture models is they don’t have any sense of space in them, the spatial
 1020 information is summarized out of the data set, so we aren’t able to use such mod-
 1021 els for studying things such as movement, or resource selection, etc.*dots*. Instead,
 1022 ordinary capture-recapture models usually resort to models of “encounter prob-

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

1.3.1 Example: Fort Drum bear study

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

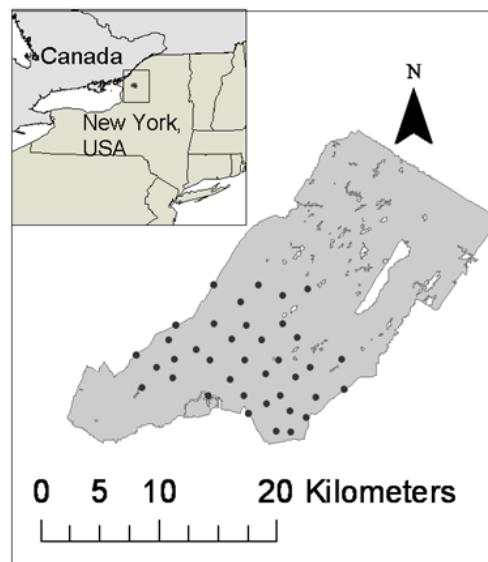


Figure 1.4. Locations of hair snares on Fort Drum, New York, operated during the summer of 2006 to sample black bears.

1040 Although each bear was captured, or not, in each of the 38 hair snares, we start
1041 by treating this data set as a standard capture-recapture data set and summarize
1042 to an encounter history matrix with 47 rows and 8 columns with entries y_{ik} , where
1043 $y_{ik} = 1$ if individual i was captured, at any trap, in sample occasion k and $y_{ik} = 0$
1044 otherwise. There is a standard closed population model, colloquially referred to
1045 as “model M_0 ” (see Chapt. 4), which assumes that encounter probability p is
1046 constant for all individuals and sample periods. We fitted model M_0 to the Fort
1047 Drum data using traditional likelihood methods, yielding the maximum likelihood
1048 estimate (MLE) of $\hat{N} = 49.19$ with an asymptotic standard error (SE) of 1.9.

1049 The key issue in using such a closed population model regards how we should
1050 interpret this estimate of $N = 49.19$ bears. Does it represent the entire population
1051 of Fort Drum? Certainly not – the trapping array covers less than half of Fort
1052 Drum as we see in Fig. 1.4. So to get at the total bear population size of Fort
1053 Drum, we would have to convert our \hat{N} to an estimate of density and extrapolate.
1054 To get at density, then, should we assert that N applies to the southern half of
1055 Fort Drum below some arbitrary line? Surely bears move on and off of Fort Drum
1056 without regard to hypothetical boundaries. Without additional information there
1057 is simply no way of converting this estimate of N to density, and hence it is really
1058 not meaningful biologically. To resolve this problem, we will adopt the customary
1059 approach of converting N to D by buffering the convex hull around the trap array.
1060 The convex hull has area 157.135 km^2 . We follow Bales et al. (2005) in buffering
1061 the convex hull of the trap array by the radius of the mean female home range size.

1062 The mean female home range radius was estimated (Wegan, 2008) for this study
1063 region to be 2.19 km, and the area of the convex hull buffered by 2.19 km is
1064 277.01 km^2 . (**R** commands to compute the convex hull, buffer it, and compute the
1065 area are given in the **R** package **scrbook** which accompanies the book). Hence,
1066 the estimated density here is approximately 0.178 bears/ km^2 using the estimated
1067 population size obtained by model M_0 . We could assert that the problem has been
1068 solved, go home, and have a beer. But then, on the other hand, maybe we should
1069 question the use of the estimated home range radius – after all, this is only the
1070 female home range radius and the home ranges change for many reasons. Instead,
1071 we may decide to rely on a buffer width based on one-half mean maximum distance
1072 moved (MMDM) estimated from the actual hair snare data as is more customary
1073 (Dice, 1938). In that case the buffer width is 1.19 km, and the resulting estimated
1074 density is increased to 0.225 bears/ km^2 about 27 % larger. But wait – some studies
1075 actually found the full MMDM (Parmenter et al., 2003) to be a more appropriate
1076 measure of movement (e.g. Soisalo and Cavalcanti (2006)). So maybe we should use
1077 the full MMDM which is 2.37 km, pretty close to the telemetry-based estimate and
1078 therefore providing a similar estimate of density (0.171 bears/ km^2). So in trying to
1079 decide how to buffer our trap array we have already generated 3 density estimates.
1080 The crux of the matter is obvious: Although it is intuitive that N should scale with
1081 area – the number of bears should go up as area increases and go down as area
1082 decreases – in this ad hoc approach of accounting for animal movement N remains

1083 the same, no matter what area we assert was sampled. The number of bears and the
 1084 area they live in are not formally tied together within the model, because estimating
 1085 N and estimating the area N relates to are two completely independent analytical
 1086 steps which are unrelated to one another by a formal model.

1087 Unfortunately, our problems don't end here. In thinking about the use of model
 1088 M_0 , we might naturally question some of the basic assumptions that go into that
 1089 model. The obvious one to question is that which declares that p is constant.
 1090 One obvious source of variation in p is variation *among individuals*. We expect
 1091 that individuals may have more or less exposure to trapping due to their location
 1092 relative to traps, and so we try to model this "heterogeneous" encounter probability
 1093 phenomenon. To illustrate this phenomenon, here are the number of traps that each
 1094 individual was encountered in:

```
1095 # traps: 1 2 3 4 5 6 9
1096 # bears: 23 13 6 2 1 1 1
```

1097 meaning, for example, 23 bears were captured in only 1 trap, and 1 bear was
 1098 captured in 9 distinct traps. The variation in trap-encounter frequencies suggests
 1099 quite a range in traps exposed to bears in the sampled population. Historically,
 1100 researches try to reduce spatial heterogeneity in capture probability by placing > 1
 1101 trap per home range (Otis et al., 1978; Williams et al., 2002). This seems like a
 1102 sensible idea but it is difficult to do in practice since you don't know where all
 1103 the home ranges are and so we try to impose a density of traps that averages
 1104 something > 1 per home range. An alternative solution is to fit models that allow
 1105 for individual heterogeneity in p (Karanth, 1995). Such models have the colloquial
 1106 name of "model M_h " (Otis et al., 1978). We fitted this model (see Chapt. 4
 1107 for details) to the Fort Drum data using each of the 3 buffer widths previously
 1108 described (telemetry, 1/2 MMDM and MMDM), producing the estimates reported
 1109 in Table 1.1. While we can tell by the models' AIC that M_h is clearly favored by
 1110 more than 30 units, we might still not be entirely happy with our results. Clearly
 1111 there is information in our data that could tell us something about the exposure
 1112 of individual bears to the trap array – where they were captured, and how many
 1113 times – but since space has no representation in our model, we can't make use
 1114 of this information. Model M_h thus merely accounts for what we observe in our
 1115 data (some bears were more frequently captured than others) rather than explicitly
 1116 accounting for the processes that generated the data.

1117 So what are we left with? Our density estimates span a range from 0.17 to
 1118 0.43 bears/km² depending on which estimator of N we use and what buffer strip
 1119 we apply. Should we feel strongly about one or the other? Which buffer should
 1120 we prefer? AIC favors model M_h , but did it adequately account for the differ-
 1121 ences in exposure of individuals to the trap array? Are we happy with a purely
 1122 phenomenological model for heterogeneity? It assumes that all individuals are in-
 1123 dependent and identically distributed (*iid*) draws from some distribution, but does
 1124 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 (Holdenried, 1940; Hayne, 1949), nested grids (Otis et al., 1978), and assessment lines (Smith et al., 1971)."

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

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

1.4.2 Temporary emigration

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

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

1236 closed population models and fully spatial capture-recapture models, which we
1237 address directly in Chapt. 4.

1238 While these procedures are all heuristically appealing, they are also essentially
1239 ad hoc in the sense that the underlying model remains unspecified or at least im-
1240 precisely characterized and so there is little or no basis for modifying, extending
1241 or generalizing the methods. These methods are distinctly *not* model-based pro-
1242 cedures. Despite this, there seems to be an enormous amount of literature developing,
1243 evaluating and “validating” these literally dozens of heuristic ideas that solve spe-
1244 cific problems, as well as various related tweaks and tunings of them and really it
1245 hasn’t led to any substantive breakthroughs that are sufficiently general or theo-
1246 retically rigorous.

1.5 EXTENSION OF CLOSED POPULATION MODELS

1247 The deficiency with classical closed population models is that they have no spatial
1248 context. N is just an integer parameter that applies equally well to estimating the
1249 number of unique words in a book, the size of some population that exists in a
1250 computer, or a bucket full of goldfish. The question of *where* the N items belong
1251 is central both to interpretation of data and estimates from all capture-recapture
1252 studies and, in fact, to the construction of spatial capture-recapture models con-
1253 sidered in this book. Surely it must matter whether the N items exist as words in
1254 a book, or goldfish in a bowl, or tigers in a patch of forest! That classical closed
1255 population models have no spatial context leads to a number of conceptual and
1256 methodological problems or limitations as we have encountered previously. More
1257 important, ecologists seldom care only about N – space is often central to objec-
1258 tives of many population studies – movement, space usage, resource selection, how
1259 individuals are distributed in space and in response to explicit factors related to
1260 landuse or habitat. Because space is central to so many real problems, this is proba-
1261 bly the number 1 reason that many ecologists don’t bother with capture-recapture.
1262 They haven’t seen capture-recapture methods as being able to solve their problems.
1263 Thus, the essential problem is that classical closed population models are too sim-
1264 ple – they ignore the spatial attribution of traps and encounter events, movement
1265 and variability in exposure of individuals to trap proximity. These problems can be
1266 addressed formally by the development of more general capture-recapture models.

1267 1.5.1 Towards spatial explicitness: Efford’s formulation

1268 The solution to the various issues that arise in the application of ordinary capture-
1269 recapture models is to extend the closed population model so that N becomes
1270 spatially explicit. Efford (2004) was the first to formalize an explicit model for
1271 spatial capture-recapture problems in the context of trapping arrays. He adopted
1272 a Poisson point process model to describe the distribution of individuals and essen-
1273 tially a distance sampling formulation of the observation model which describes the

1274 probability of detection as a function of individual location, regarded as a latent
 1275 variable governed by the point process model. While earlier (and contemporary)
 1276 methods of estimating density from trap arrays have been ad hoc in the sense of
 1277 lacking a formal description of the spatial model, Efford achieved a formalization
 1278 of the model, describing explicit mechanisms governing the spatial distribution of
 1279 individuals and how they are encountered by traps, but adopted a more or less
 1280 ad hoc framework for inference under that spatial model using a simulation based
 1281 method known as inverse prediction (Gopalaswamy, 2012).

1282 Recently, there has been a flurry of effort devoted to formalizing inference un-
 1283 der this model-based framework for the analysis of spatial capture-recapture data
 1284 (Borchers and Efford, 2008; Royle and Gardner, 2011; Borchers, 2012; Gopalaswamy,
 1285 2012). There are two distinct lines of work which adopt the model-based formula-
 1286 tion in terms of the underlying point process but differ primarily by the manner in
 1287 which inference is achieved. One approach (Borchers and Efford, 2008) uses classi-
 1288 cal inference based on likelihood (see Chapt. 6), and the other (Royle and Young,
 1289 2008) adopts a Bayesian framework for inference (Chapts. 5 and 17).

1290 **1.5.2 Abundance as the aggregation of a point process**

1291 Spatial point process models represent a major methodological theme in spatial
 1292 statistics (Cressie, 1991) and they are widely applied as models for many ecological
 1293 phenomena (Stoyan and Penttinen, 2000; Illian et al., 2008). Point process models
 1294 apply to situations in which the random variable in question represents the locations
 1295 of events or objects: trees in a forest, weeds in a field, bird nests, etc. . . As such,
 1296 it seems natural to describe the organization of individuals in space using point
 1297 process models. SCR models represent the extension of ordinary capture-recapture
 1298 by augmenting the model with a point process to describe individual locations.

1299 Specifically, let $s_i; i = 1, 2, \dots, N$ be the locations of all individuals in the popu-
 1300 lation. One of the key features of SCR models is that the point locations are latent,
 1301 or unobserved, and we only obtain imperfect information about the point locations
 1302 by observing individuals at trap or observation locations. Thus, the realized loca-
 1303 tions of individuals represent a type of “thinned” point process, where the thinning
 1304 mechanism is not random but, rather, biased by the observation mechanism. It is
 1305 also natural to think about the observed point process as some kind of a compound
 1306 or aggregate point process with a set of “parent” nodes being the locations of in-
 1307 dividual home ranges or their centroids, and the observed locations as “offspring”
 1308 - i.e., a Poisson cluster process (PCP). In that context, density estimation in SCR
 1309 models is analogous to estimating the number of parents of a Poisson cluster process
 1310 (Chandler and Royle, In press).

1311 Most of the recent developments in modeling and inference from spatial en-
 1312 counter history data, including most methods discussed in this book, are predicated
 1313 on the view that individuals are organized in space according to a relatively simple
 1314 point process model. More specifically, we assume that the collection of individ-

1315 ual activity centers are independent and identically distributed random variables
 1316 distributed uniformly over some region. This is consistent with the assumption
 1317 that the activity centers represent the realization of a Poisson point process or, if
 1318 the total number of activity centers fixed, then this is usually referred to as a
 1319 binomial point process.

1320 1.5.3 The activity center concept

1321 In the context of SCR models, and because most animals we study by capture-
 1322 recapture are not sessile, there is not a unique and precise mathematical definition
 1323 of the point locations \mathbf{s} . Rather, we imagine these to be the centroid of individ-
 1324 uals home ranges, or the centroid of an individual's activities during the time of
 1325 sampling, or even it's average location measured with error (e.g., from a long series
 1326 of telemetry measurements). In general, this point is unknown for any individual
 1327 but if we could track an individual over time and take many observations then we
 1328 could perhaps get a good idea of where that point is. We'll think of the collection
 1329 of these points as defining the spatial distribution of individuals in the population.

1330 We use the terms home range or activity center interchangeably. The term
 1331 "home range center" suggests that models are only relevant to animals that exhibit
 1332 behavior of establishing home ranges or territories, or central place foragers, and
 1333 since not all species do that, perhaps the construction of SCR models based on this
 1334 idea is flawed. However, the notion of a home range center is just a conceptual
 1335 device and we don't view this concept as being strictly consistent with classical
 1336 notions of animal territories. Rather our view is that a home range or territory
 1337 is inherently dynamic, temporally, and thus it is a transient quantity - where the
 1338 animal lived during the period of study, a concept that is completely analogous to
 1339 the more conventional notion of utilization distributions. Therefore, whether or not
 1340 individuals of a species establish home ranges is irrelevant because, once a precise
 1341 time period is defined, this defines a distinct region of space that an individual must
 1342 have occupied.

1343 1.5.4 The state-space

1344 Once we introduce the collection of activity centers, $\mathbf{s}_i; i = 1, 2, \dots, N$, then the
 1345 question "what are the possible values of \mathbf{s} ?" needs to be addressed because the
 1346 individual \mathbf{s}_i are *unknown*. As a technical matter, we will regard them as random
 1347 effects and in order to apply standard methods of statistical inference we need to
 1348 provide a distribution for these random effects. In the context of the point process
 1349 model, the possible values of the point locations referred to as the "state-space" of
 1350 the point process and this is some region or set of points which we will denote by
 1351 \mathcal{S} . This is analogous to what is sometimes called the *observation window* for \mathbf{s} in
 1352 the point process literature. The region \mathcal{S} serves as a prior distribution for \mathbf{s}_i (or,
 1353 equivalently, the random effects distribution). In animal studies, as a description

1354 of where individuals that could be captured are located, it includes our study area,
 1355 and should accommodate all individuals that could have been captured in the study
 1356 area. In the practical application of SCR models, in most cases estimates of density
 1357 will be relatively insensitive to choice of state-space which we discuss further in
 1358 Chapt. 5 and elsewhere.

1359 **1.5.5 Abundance and density**

1360 When the underlying point process is well-defined, including a precise definition
 1361 of the state-space, this in turn induces a precise definition of the parameter N ,
 1362 “population size”, as the number of individual activity centers located within the
 1363 prescribed state-space, and its direct linkage to density, D . That is, if $A(\mathcal{S})$ is the
 1364 area of the state-space then

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

1365 A deficiency with some classical methods of “adjustment” is they attempted to
 1366 prescribe something like a state-space - a “sampled area” - except absent any pre-
 1367 cise linkage of individuals with the state-space. SCR models formalize the linkage
 1368 between individuals and space and, in doing so, provide an explicit definition of N
 1369 associated with a well-defined spatial region, and hence density. That is, the pro-
 1370 vide a model in which N scales, as part of the model, with the size of the prescribed
 1371 state-space. In a sense, the whole idea of SCR models is that by defining a point
 1372 process and its state-space \mathcal{S} , this gives context and meaning to N which can be
 1373 estimated directly for that specific state-space. Thus, it is fixing \mathcal{S} that resolves
 1374 the problem of “unknown area” that we have previously discussed.

1.6 CHARACTERIZATION OF SCR MODELS

1375 Formulation of capture-recapture models conditional on the latent point process is
 1376 the critical and unifying element of *all* SCR models. However, SCR models differ
 1377 in how the underlying process model is formulated, and its complexity. Most of the
 1378 development and application of SCR models has focused on their use to estimate
 1379 density and touting the fact that they resolve certain specific technical problems
 1380 related to the use of ordinary capture-recapture models. This is achieved with a sim-
 1381 ple process model being a basic point process of independently distributed points.
 1382 At the same time, there are models of CR data that focus exclusively on *movement*
 1383 modeling, or models with explicit dynamics (Ovaskainen, 2004; Ovaskainen et al.,
 1384 2008). Conceptually, these are akin to spatial versions of so-called Cormack-Jolly-
 1385 Seber (CJS) models in the traditional capture-recapture literature, except they
 1386 involve explicit mathematical models of movement based on diffusion or Brownian
 1387 motion. Finally, there are now a very small number of papers that focus on *both*
 1388 movement and density simultaneously (Royle and Young, 2008; Royle et al., 2011a;

1389 Royle and Chandler, 2012) or population dynamics and density (Gardner et al.,
1390 2010b).

1391 A key thing is that these models, whether focused just on density, or just on
1392 movement, or both, are similar models in terms of the underlying concepts, the
1393 latent structure, and the observation model. They differ primarily in terms of the
1394 ecological focus. Understanding movement is an important topic in ecology, but
1395 models that strictly focus on movement will be limited by two practical consider-
1396 ations: (1) most capture-recapture data e.g., by camera trapping or whatever,
1397 produces only a few observations of each individual (between 1-5 would be typi-
1398 cal). So there is not too much information about complex movement models. (2)
1399 Typically people have an interest in density of individuals and therefore we need
1400 models that can be extrapolated from the sample to the unobserved part of the
1401 population. That said, there are clearly some cases where more elaborate move-
1402 ment models should come into play. If one has some telemetry data in addition to
1403 SCR then there is additional information on fine-scale movements that should be
1404 useful.

1.7 SUMMARY AND OUTLOOK

1405 Spatial capture-recapture models are an extension of traditional capture-recapture
1406 models to accommodate the spatial organization of both individuals in a population
1407 and the observation mechanism (e.g., locations of traps). They resolve problems
1408 which have been recognized historically and for which various ad hoc solutions
1409 have been suggested: heterogeneity in encounter probability due to the spatial
1410 organization of individuals relative to traps, the need to model trap-level effects
1411 on encounter, and that a well-defined sample area does not exist in most studies,
1412 and thus estimates of N using ordinary capture-recapture models cannot be related
1413 directly to density.

1414 As we have shown already, SCR models are not simply an extension of a tech-
1415 nique to resolve certain technical problems. Rather, they provide a coherent, flex-
1416 ible framework for making ecological processes explicit in models of individual
1417 encounter history data, and for studying animal populations processes such as individ-
1418 ual movement, resource selection, space usage, population dynamics, and density.
1419 Historically, researchers studied these questions independently, using ostensibly un-
1420 related study designs and statistical procedures. For example, resource selection
1421 function (RSF) models for resource selection, state-space models for movement,
1422 density using closed capture-recapture methods, and population dynamics with
1423 various “open” capture-recapture models. SCR can bring all of these problems
1424 together into a single unified framework for modeling and inference. Most impor-
1425 tantly, spatial capture-recapture models promise the ability to integrate explicit
1426 ecological theories directly into the models so that we can directly test hypoth-
1427 eses about either space usage (e.g., Chapt. 13), landscape connectivity (Chapt.
1428 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.

1434
1435

1436

2

STATISTICAL MODELS AND SCR

1437 In the previous chapter we described the basics of capture-recapture methods and
1438 the advantages that spatial models have over traditional non-spatial models. We
1439 avoided statistical terminology like the plague so that we could focus on a few key
1440 concepts. Although it is critical to understand the non-technical motivation for this
1441 broad class of models, it is impossible to fully appreciate them, and apply them to
1442 real data, without a solid grasp of the fundamentals of statistical inference.

1443 In this chapter, we present a brief overview of the basic statistical principals that
1444 are referenced throughout the remainder of this book. Emphasis is placed on the
1445 definition of a random variable, the common probability distributions used to model
1446 random variables, and how hierarchical models can be used to describe conditionally
1447 related random variables. For some readers, this material will be familiar, perhaps
1448 even elementary, and thus you may want to skip to the next chapter. However, our
1449 experience is that many basic statistics courses taken by ecologists do not emphasize
1450 the important subjects covered in this chapter. Instead, there seems to be much
1451 attention paid to minor details such as computing the number of degrees of freedom
1452 in various F -tests, which, although useful in some contexts, do not provide the basis
1453 for drawing conclusions from data and evaluating scientific hypotheses.

1454 The material in the beginning of this chapter is explained in numerous other
1455 texts. Technical treatments that emphasize ecological problems are given by Williams
1456 et al. (2002), Royle and Dorazio (2008) and Link and Barker (2010), to name just
1457 a few. A very accessible introduction to some of the topics covered in this chapter
1458 is presented in Chapt. 3 of MacKenzie et al. (2006). With all these resources, one
1459 might wonder why we bother rehashing these concepts here. Our motivation is
1460 two-fold: first, we wish to develop this material using examples relevant to spatial
1461 capture-recapture, and second, we find that most introductory texts are not accom-
1462 panied by code that can be helpful to the novice. We therefore attempt to present

1463 simple **R** code throughout this chapter so that those who struggle with equations
1464 and mathematical notation can learn by doing. As mentioned in the Preface, we
1465 rely on **R** because it provides tremendous flexibility for analyzing data and because
1466 it is free. We do not, however, try to explain how to use **R** because there are so
1467 many good references already, including Venables and Ripley (2002); Bolker (2008);
1468 Venables et al. (2012).

1469 After covering some basic concepts of hierarchical modeling, we end the chapter
1470 by describing spatial capture-recapture models using hierarchical modeling nota-
1471 tion. This makes the concepts outlined in the previous chapter more precise, and
1472 it highlights the fact that SCR models include explicit models for the ecological
1473 processes of interest (e.g. spatial variation in density) and the observation process,
1474 which describes how individuals are encountered.

2.1 RANDOM VARIABLES AND PROBABILITY DISTRIBUTIONS

2.1.1 Stochasticity in ecology

1476 Few ecological processes can be described using purely deterministic models, and
1477 thus we need a formal method for drawing conclusions from data while acknowl-
1478 edging the stochastic nature of ecological systems. This is the role of statistical
1479 inference, which is founded on the laws of probability. For our purposes, it suffices
1480 to be familiar with a small number of concepts from probability theory—the most
1481 important of which is the concept of a random variable, say X . A random variable
1482 is a variable whose realized value is the outcome of some stochastic process. To
1483 be more precise, a random variable is characterized by a function that describes
1484 the probability of observing the value x . This probability function can be written
1485 $\Pr(X = x|\theta)$ where θ is a parameter, or set of parameters of the function. If x is
1486 discrete, e.g. binary or integer, then we call the probability function a probability
1487 mass function (pmf). If x is continuous, the function is called a probability density
1488 function (pdf).

1489 To clarify the concept of a random variable, let X be the number of American
1490 shad (*Alosa sapidissima*) caught after $K = 20$ casts at the shad hole on Deerfield
1491 River in Massachusetts. Suppose that we had a good day and caught $x = 7$ fish.
1492 If there were no random variation at play, we would say that the probability of
1493 catching a fish, which we will call p , is $p = 7/20 = 0.35$, and we would always
1494 expect to catch 7 shad after 20 casts. In other words, our deterministic model is
1495 $x = 0.35 \times K$. In reality, however, we can be pretty sure that this deterministic
1496 model would not be very good. Even if we knew for certain that $p \equiv 0.35$, we would
1497 expect some variation in the number of fish caught on repeated fishing outings.
1498 To describe this variation, we need a model that acknowledges uncertainty (i.e.,
1499 stochasticity), and specifically we need a model that describes the probability of
1500 catching x fish given K and p , $\Pr(X = x|K, p)$. Since x is discrete, not continuous,
1501 we need a pmf. Before contemplating which pmf is most appropriate in this case,

1502 we need to first mention a few issues related to notation.

1503 Statisticians make things easier for themselves, and more complicated for ev-
 1504 eryone else, by using different notation for probability distributions. Sometimes
 1505 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
 1506 $\pi(X|K, p)$ or $\mathbb{P}(X|K, p)$ or $[X|K, p]$ or even just $[X]!$ Just remember that these
 1507 expressions all have the same meaning—they are all probability distributions that
 1508 tell us the probability of observing any possible realization of the random variable
 1509 X . In this book, we will almost always use bracket notation (the last two examples
 1510 above) to represent arbitrary probability distributions. Hence, from here on out,
 1511 when you see $[X|K, p]$, just remember that this is equivalent to the more traditional
 1512 expression $\Pr(X = x|K, p)$. In addition, from here on, to achieve a more concise
 1513 presentation, we will no longer use uppercase letters to denote random variables
 1514 and lowercase letters for realized values. Rather, we will define a random vari-
 1515 able by some symbol (x , N , etc...) and let the context determine whether we are
 1516 talking about the random variable itself, or realized values of it. In some limited
 1517 cases, we will want upper- and lower-case letters to represent different variables.
 1518 For example, we will often let N denote population size and n denote the number
 1519 of individuals actually detected.

1520 When we wish to be specific about a probability distribution, we will do so in
 1521 one of two ways, one mathematically precise and one symbolic. Before explaining
 1522 these two options, let's choose a specific distribution as a model for the data in our
 1523 example. In this case, the natural choice for $[x|K, p]$ is the binomial distribution,
 1524 the mathematically precise representation of which is

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

1525 The right-hand side of this equation is the binomial pmf (described in more detail
 1526 in Sec. 2.2), and plugging in values for the parameters K , and p will return the
 1527 probability of observing any realized value of the random variable x . This is precise,
 1528 but it is also cumbersome to write repetitively, and it may make the eyes glaze over
 1529 when seen too often. Thus, we will often simplify Eq. 2.1.1 using the symbolic
 1530 notation:

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

1531 The “ \sim ” symbol is meant to represent a stochastic relationship, and can be read
 1532 “is distributed as.” Another reason for using this notation is that it resembles the
 1533 syntax of the **BUGS** language, which we will frequently use to conduct Bayesian
 1534 inference.

1535 Note that once we choose a probability distribution, we have chosen a model. In
 1536 our example, we have specified our model as $x \sim \text{Binomial}(K, p)$, and because we
 1537 are assuming that the parameters are known, we can make probability statements
 1538 about future outcomes. Continuing with our fish example, we might want to know
 1539 the probability of catching $x = 7$ again after $K = 20$ casts on a future fishing

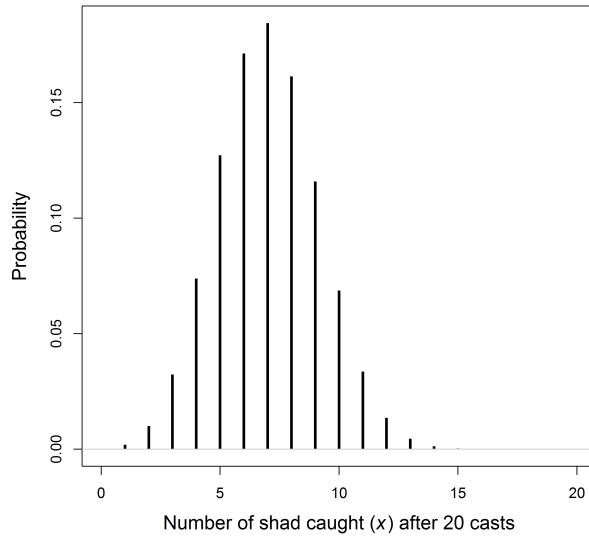


Figure 2.1. The binomial probability mass function with $N = 20$ and $p = 0.35$.

1540 outing, assuming that we know $p = 0.35$. Evaluating the binomial pmf returns a
 1541 probability of approximately 0.18, as show using this bit of **R** code:

```
1542 > dbinom(7, 20, 0.35)
1543 [1] 0.1844012
```

1544 By definition, the pmf allows us to evaluate the probability of observing any x given
 1545 $K = 20$ and $p = 0.35$, thus the distribution of the random variable can be visualized
 1546 by evaluating it for all values of x that have non-negligible probabilities, as can be
 1547 easily done in **R**:

```
1548 plot(0:20, dbinom(0:20, 20, 0.35), type="h", ylab="Probability",
1549 xlab="Number of shad caught (X)")
```

1550 the result of which is shown in Fig. 2.1 with some extra details.

1551 The purpose of this little example is to show that once we specify a model for the
 1552 random variable(s) being studied, we can begin drawing conclusions, i.e. making
 1553 inferences, about the processes of interest, even in the face of uncertainty. Prob-
 1554 ability distributions are essential to this process, and thus we need to understand
 1555 them in more depth.

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

Distribution	Notation	pmf or pmf	Support	Mean $\mathbb{E}(x)$	Variance $\text{Var}(x)$
Discrete random variables					
Poisson	$x \sim \text{Pois}(\lambda)$	$\exp(-\lambda)\lambda^x/x!$	$x \in \{0, 1, \dots\}$	λ	λ
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	$x \sim \text{N}(\mu, \sigma^2)$	$\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$	$x \in [-\infty, \infty]$	μ	σ^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	$x \sim \text{Gamma}(a, b)$	$\frac{1}{\Gamma(a)} x^{a-1} \exp(-bx)$	$x \in [0, \infty]$	a/b	a/b^2
Multivariate Normal	$\mathbf{x} \sim \text{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	$(2\pi)^{-k/2} \boldsymbol{\Sigma} ^{-1/2} \exp(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}))$	$x_k \in [-\infty, \infty]$	$\boldsymbol{\mu}$	$\boldsymbol{\Sigma}$

1556 **2.1.2 Properties of probability distributions**

1557 A pdf or a pmf is a function like any other function in the sense that it has one
 1558 or more arguments whose values determine the result of the function. However,
 1559 probability functions have a few properties that distinguish them from other func-
 1560 tions. The first is that the function must be non-negative for all possible values of
 1561 the random variable, i.e. $[x] \geq 0$. The second requirement is that the integral of
 1562 a pdf must be unity, $\int_{-\infty}^{\infty} [x] dx = 1$, and similarly for a pmf, the summation over
 1563 all possible values is unity, $\sum_x [x] = 1$. The following **R** code demonstrates this for
 1564 the normal and binomial distributions:

```
1565 > integrate(dnorm, -Inf, Inf, mean=0, sd=1)$value
1566 [1] 1
1567 > sum(dbinom(0:5, size=5, p=0.1))
1568 [1] 1
```

1569 This requirement is important to remember when one develops a non-standard
 1570 probability distribution. For example, in Chapt. 11 and 13, we work with resource
 1571 selection functions whose probability density function is not one that is pre-defined
 1572 in software packages such as **R** or **BUGS**.

1573 Another feature of probability distributions is that they can be used to compute
 1574 important summaries of random variables. The two most important summaries
 1575 are the expected value, $\mathbb{E}(x)$, and the variance $\text{Var}(x)$. The expected value, or
 1576 mean, can be thought of as the average of a very large sample from the specified
 1577 distribution. For example, one way of approximating the expected values of a
 1578 binomial distribution with $K = 20$ trials and $p = 0.35$ can be implemented in
 1579 **R** using:

```
1580 > mean(rbinom(10000, 20, 0.3))
1581 [1] 6.9865
```

1582 For most probability distributions used in this book, the expected values are known
 1583 exactly, as shown in Table 2.1, and thus we don't need to resort to such Monte Carlo
 1584 approximations. For instance, the expected value of the binomial distribution is
 1585 exactly $\mathbb{E}(x) = Kp = 20 \times 0.35 = 7$. In this case, it happens to take an integer
 1586 value, but this is not a necessary condition, even for discrete random variables.

1587 A more formal definition of an expected value is the average of all possible
 1588 values of the random variable, weighted by their probabilities. For continuous
 1589 random variables, this weighted average is found by integration:

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

1590 For example, if $[x]$ is normally distributed with mean 3 and unit variance, we could
 1591 find the expected value using the following code.

```
1592 > integrate(function(x) x*dnorm(x, 3, 1), -Inf, Inf)
1593 3 with absolute error < 0.00033
```

1594 Of course, the mean *is* the expected value of the normal distribution, so we didn't
 1595 need to compute the integral but, the point is, that Eq. 2.1.3 is generic. For
 1596 discrete random variables, the expected value is found by summation rather than
 1597 integration:

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

1598 where the summation is over all possible values of x . Earlier we approximated the
 1599 expected value of the binomial distribution with $K = 20$ trials and $p = 0.35$ by
 1600 taking a Monte Carlo average. Eq. 2.1.4 let's us find the exact answer, using this
 1601 bit of R code:

```
1602 > sum(dbinom(0:100, 20, 0.35)*0:100)
1603 [1] 7
```

1604 This is great. But of what use is it? One very important concept to understand is
 1605 that when we fit models, we are often modeling changes in the expected value of
 1606 some random variable. For example, in Poisson regression, we model the expected
 1607 value of the random variable, which may be a function of environmental variables.

1608 The ability to model the expected value of a random variable gets us very far,
 1609 but we also need a model for the variance of the random variable. The variance
 1610 describes the amount of variation around the expected value. Specifically, $\text{Var}(x) =$
 1611 $\mathbb{E}((x - \mathbb{E}(x))^2)$. Clearly, if the variance is zero, the variable is not random as
 1612 there is no uncertainty in its outcome. For some distributions, notably the normal
 1613 distribution, the variance is a parameter to be estimated. Thus, in ordinary linear
 1614 regression, we estimate both the expected value $\mu = \mathbb{E}(x)$, which may be a function
 1615 of covariates, and the variance σ^2 , or similarly the residual standard error σ . For
 1616 other distributions, the variance is not an explicit parameter to be estimated, and
 1617 instead, the mean to variance ratio is fixed. In the case of the Poisson distribution,
 1618 the mean is equal to the variance, $\mathbb{E}(x) = \text{Var}(x) = \lambda$. A similar situation is true
 1619 for the binomial distribution—the variance is determined by the two parameters K
 1620 and p , $\text{Var}(x) = Kp(1-p)$. In our earlier example with $K = 20$ and $p = 0.35$, the
 1621 variance is 4.55. Toying around with these ideas using random number generators
 1622 may be helpful. Here is some code to illustrate some of these basic concepts:

```
1623 > 20*0.35*(1-0.35)                      # Exact variance, Var(x)
1624 [1] 4.55
1625 > x <- rbinom(100000, 20, 0.35)
1626 > mean((x-mean(x))^2)                   # Monte Carlo approximation
1627 [1] 4.545525
```

2.2 COMMON PROBABILITY DISTRIBUTIONS

1628 We got a little ahead of ourselves in the previous sections by using the binomial
 1629 and Poisson distributions without describing them in detail. A solid understanding
 1630 of the binomial, Poisson, multinomial, uniform, and normal (or Gaussian) distri-
 1631 butions is absolutely essential throughout the remainder of the book. We will
 1632 occasionally make use of other distributions such as the beta, log-normal, gamma,
 1633 Dirichlet, etc... that can be helpful when modeling capture-recapture data, but
 1634 these distributions can be readily understood once you are comfortable with the
 1635 more commonly used distributions described in this section.

1636 **2.2.1 The binomial distribution**

1637 The binomial distribution plays a critical role in ecology. It is used for purposes
 1638 as diverse as modeling count data, survival probability, occurrence probability, and
 1639 capture probability, just to name a few. To describe the properties of the binomial
 1640 distribution, and related distributions, we will introduce a new example. Suppose
 1641 we are conducting a bird survey at a site in which $N = 10$ chestnut-sided warblers
 1642 (*Setophaga pensylvanica*) occur, and each of these individuals has a detection prob-
 1643 ability of $p = 0.5$. The binomial distribution is the natural choice for describing
 1644 the number of individuals that we would expect to detect (n) in this situation, and
 1645 using our notation, we can write the model as: $n \sim \text{Binomial}(10, 0.5)$. When $p < 1$,
 1646 we can expect that we will observe a different number of warblers on each of K
 1647 replicate survey occasions. To see this, we simulate data under this simple model
 1648 with $K = 3$.

```
1649 > n <- rbinom(3, size=10, prob=0.5) # Generate 3 binomial outcomes
1650 > n                                     # Display the 3 values
1651 [1] 6 4 8
```

1652 The vector of counts will typically differ each time you issue this command; however,
 1653 we know the probability of observing any value of n_k because it is defined by the
 1654 binomial pmf. As we demonstrated earlier, in R this probability can be found using
 1655 the `dbinom` function. For example, the probability of observing $n_k = 5$ is given by:

```
1656 > dbinom(5, 10, 0.5)
```

1657 This simply evaluates the function shown in Table 2.1. We could do the same more
 1658 transparently, but less efficiently, using any of the following:

```
1659 > n <- 5; N <- 10; p <- 0.5
1660 > factorial(N)/(factorial(n)*factorial(N-n))*p^n*(1-p)^(N-n)
1661 > exp(lgamma(N+1) - (lgamma(n+1) + lgamma(N-n+1)))*p^n*(1-p)^(N-n)
1662 > choose(N, n)*p^n*(1-p)^(N-n)
```

1663 Note that the last three lines of code differ only in how they compute the binomial
 1664 coefficient $\binom{N}{n}$, which is the number of different ways we could observe $n = 5$ of
 1665 the $N = 10$ chestnut-sided warblers at the site. The binomial coefficient, which is
 1666 read “N choose n” is defined as

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

1667 Now that we know how to simulate binomial data and compute the probabilities
 1668 of observing any particular outcome n , conditional on the parameters N and
 1669 p , we can contemplate the relevance of the binomial distribution in spatial capture-
 1670 recapture models. One important application of the binomial distribution is as a
 1671 model encounter frequencies. Indeed, one of the most important encounter models
 1672 in SCR will be referred to as the “binomial encounter model”, in which the number
 1673 of times individual i is captured at “trap” j after K survey occasions is modeled as
 1674 $y_{ij} \sim \text{Binomial}(K, p_{ij})$. Here, p_{ij} is the encounter probability determined, in part,
 1675 by the distance between an animal’s activity center and the trap location. This
 1676 binomial encounter model is described in detail in Sec. 7.1. Another important application
 1677 of the binomial distribution is as a prior for the population size parameter
 1678 in Bayesian analyses, as is discussed in Chapt. 4.

1679 2.2.2 The Bernoulli distribution

1680 Above, we showed 3 alternatives to `dbinom` for evaluating the binomial pmf. These
 1681 three commands differed only in how they computed the binomial coefficient, which
 1682 we needed because of the numerous ways in which we could observe $n = 5$ given
 1683 $N = 10$. To conceptualize this, let y_i be a binary variable indicating if individual i
 1684 was detected or not. Hence, given that 5 individuals were detected, the vector of
 1685 individual detections could be something like $\mathbf{y} = (0, 0, 1, 1, 1, 1, 0, 0, 0)$, indicating
 1686 that we detected individuals 3-7 but not 1-2 or 8-10. For $N = 10$ and $n = 5$,
 1687 the binomial coefficient tells us that there are 252 possible vectors \mathbf{y} with 5 ones.
 1688 However, when $N \equiv 1$, this term drops from the pmf and the result is the pmf for
 1689 the Bernoulli distribution. That is, the Bernoulli distribution is simply the binomial
 1690 distribution when $N \equiv 1$. Alternatively, we could say that the binomial distribution
 1691 is the outcome of N iid Bernoulli trials. We use the standard abbreviation “iid”
 1692 to mean *independent, identically distributed*.

1693 The utility of the Bernoulli distribution is evident when we imagine that not all
 1694 of the chestnut-sided warblers have the same detection probability. Thus, if some
 1695 individuals can be detected with probability 0.3 and others have a 0.7 detection
 1696 probability, then the model $n \sim \text{Binomial}(N, p)$ is no longer an accurate description
 1697 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

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

1698 This states that individual i is detected with probability p_i , and the observed count
 1699 is the sum of the N Bernoulli outcomes.

1700 An important point is that the individual-specific data y_{ik} can only be observed
 1701 if the individuals are uniquely distinguishable, such as when they are marked by
 1702 biologists with color bands. In such cases, the Bernoulli distribution allows us
 1703 to model variation in detection probability among individuals and thus would be
 1704 preferable to the binomial distribution, which assumes that each of the N indi-
 1705 viduals have the same p . For this reason, the Bernoulli distribution, as simple as
 1706 it is, is of paramount importance in capture-recapture models, including spatial
 1707 capture-recapture models in which there is virtually always substantial and impor-
 1708 tant variation in capture probability among individuals. Indeed, it could be said
 1709 that the Bernoulli model is the canonical model in capture-recapture studies, and
 1710 most of the different flavors of capture-recapture models differ primarily in how p_i
 1711 is specified.

1712 The Bernoulli pmf is given by $p^n(1-p)^{1-n}$ and hence we do not need canned
 1713 functions to facilitate its evaluation. Of course, if you wanted to, you could always
 1714 use `dbinom` with the `size` argument set to 1. For example, `dbinom(1, 1, 0.3)`
 1715 returns the Bernoulli probability of observing $n = 1$ given $p = 0.3$.

1716 2.2.3 The multinomial and categorical distributions

1717 The binomial distribution is used when we are accumulating a binary response—
 1718 that is, one in which there are two possible categories such as success/failure or
 1719 captured/not-captured. The multinomial distribution is a multivariate extension
 1720 of the binomial used when there are $G > 2$ categories. The multinomial distribution
 1721 can be thought of as a model for placing N items in the G categories, which are
 1722 also called bins or cells. Each bin has its own probability π_g and these probabilities
 1723 must sum to one. In ecology, N is often population size or the number of individuals
 1724 detected, but the definition of the G bins varies among applications. For example,
 1725 in distance sampling, when the distance data are aggregated into intervals, the
 1726 bins are the distance intervals, and the cell probabilities are functions of detection
 1727 probability in each interval (Royle et al., 2004).

1728 The multinomial distribution is widely used to model data from traditional,
 1729 non-spatial capture-recapture studies. Earlier we let y_{ik} denote a binary random
 1730 variable indicating if warbler i was detected on survey k . The vector of observations
 1731 for an individual, \mathbf{y}_i , is often referred to as the individual's "encounter history".

1732 The number of possible encounter histories depends on K , the number of survey
 1733 occasions. Specifically, there are 2^K possible encounter histories¹. If we tabulate the
 1734 number of individuals with each encounter history, the frequencies can be modeled
 1735 using the multinomial distribution.

1736 Going back to our chestnut-sided warbler example, suppose the 10 individuals
 1737 are marked and we make $K = 2$ visits to the site such that there are $2^K = 4$ pos-
 1738 sible encounter histories: (11, 10, 01, 00), where, for example, “10” is the encounter
 1739 history for an individual detected on the first visit but not the second. If $p = 1$,
 1740 then the encounter history for each of the 10 individuals must be “11”. That is, we
 1741 would detect each individual on both occasions. In this case, we the data would be:
 1742 $\mathbf{h} = (10, 0, 0, 0)$, which indicates that all 10 warblers had the first encounter history.
 1743 The corresponding cell probabilities would be $\boldsymbol{\pi} = (1, 0, 0, 0)$. What about the sit-
 1744 uation where $p < 1$, e.g. $p = 0.3$? In this case, the probability of observing the
 1745 capture history “11” (detected on both occasions) is $p \times p = 0.3 \times 0.3 = 0.09$. The
 1746 probability of observing “10” is $p \times (1 - p) = 0.21$. Following this logic, the vector
 1747 of cell probabilities is $\boldsymbol{\pi} = (0.09, 0.21, 0.21, 0.49)$. We can simulate data under this
 1748 model as follows:

```
1749 > caphist.probs <- c("11"=0.09, "10"=0.21, "01"=0.21, "00"=0.49)
1750 > drop(rmultinom(1, 10, caphist.probs))
1751 11 10 01 00
1752 0 3 2 5
```

1753 The result of our simulation is that zero individuals were observed with the capture
 1754 history “11” and 5 individuals were observed with the capture history “00”. The
 1755 other 5 individuals were observed one out of the two occasions. This is not such a
 1756 surprising outcome given $p = 0.3$.

1757 As in non-spatial capture-recapture studies, the multinomial distribution turns
 1758 out to be very important in spatial capture-recapture studies. However, N is not
 1759 defined as population size. Rather, we use the multinomial distribution when an
 1760 individual can only be captured in a single trap during an occasion. Thus $N = 1$
 1761 and the cell probabilities are the probabilities of being captured in each trap. A
 1762 thorough discussion of this point can be found in Chapt. 9. Another application
 1763 of the multinomial distribution in SCR models is discussed in Chapt. 11 where we
 1764 discuss how to model the probability that an individual’s activity center is located
 1765 in one of the cells of a raster defining the spatial region of interest.

1766 Just as the Bernoulli distribution is the elemental form of the binomial distri-
 1767 bution (being the case $N = 1$), the categorical distribution is essentially equivalent
 1768 to the multinomial distribution with size parameter $N \equiv 1$. The only difference is
 1769 that, rather than returning a vector with a single element equal to 1, it returns the
 1770 element *location* where the 1 occurs. For example, if $\mathbf{y} = (0, 0, 1, 0)$ is an outcome

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

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

```
1777 > which(rmultinom(1, 1, c(0.1, 0.7, 0.2)) == 1)
1778 [1] 2
```

1779 2.2.4 The Poisson distribution

1780 The Poisson distribution is the canonical model for count data in ecology. More
 1781 generally, the Poisson distribution is a model for random variables taking on non-
 1782 negative, integer values. Although it is a simple model having just one parameter,
 1783 $\lambda = \mathbb{E}(x) = \text{Var}(x)$, its applications are highly diverse, including as a model of
 1784 spatial variation in abundance or as a model for the frequency of behaviors over
 1785 time. Just as logistic regression is the standard generalized linear model (GLM)
 1786 used to model binary data, Poisson regression is the default GLM for modeling
 1787 count data and variation in λ .

1788 The Poisson distribution is related to both the binomial and multinomial distri-
 1789 butions, and the following three bits of trivia are occasionally worth knowing. First,
 1790 it is the limit of the binomial distribution as $N \rightarrow \infty$ and $p \rightarrow 0$, which means that
 1791 for high values of N and low values of p , $\text{Poisson}(N \times p)$ is approximately equal
 1792 to $\text{Binomial}(N, p)$. Second, if $\{n_1 \sim \text{Poisson}(\lambda_1), \dots, n_K \sim \text{Poisson}(\lambda_K)\}$ then the
 1793 vector of counts is multinomial, $\{n_1, \dots, n_K\} \sim \text{Multinomial}(\sum_k n_k, \{\frac{\lambda_1}{\sum_k \lambda_k}, \dots, \frac{\lambda_K}{\sum_k \lambda_k}\})$.
 1794 Third, the sum of two Poisson random variables $x_1 \sim \text{Poisson}(\lambda_1)$ and $x_2 \sim$
 1795 $\text{Poisson}(\lambda_2)$ is also Poisson: $x_1 + x_2 \sim \text{Poisson}(\lambda_1 + \lambda_2)$.

1796 The Poisson distribution has two important uses in spatial capture-recapture
 1797 models: (1) as a prior distribution for the population size parameter N , and (2) as a
 1798 model for the frequency of captures in a trap. In the first context, the Poisson prior
 1799 for N results in a Poisson point process for the location of the N activity centers
 1800 in the region of interest. This topic is discussed in Chapt. 5 and Chapt 11. The
 1801 second use of the Poisson distribution in spatial capture-recapture is to describe
 1802 data from sampling methods in which an individual can be detected multiple times
 1803 at a trap during a single occasion. For example, in camera trapping studies we
 1804 might obtain multiple pictures of the same individual at a trap during a single
 1805 sampling occasion. Thus, λ in this case would be defined as the expected number
 1806 of detections or captures per occasion.

1807 2.2.5 The uniform distribution

1808 The lowly uniform distribution is a continuous distribution whose only two pa-
 1809 rameters are the lower and upper bounds that restrict the possible values of the

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

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

```
1825 D <- 100      # points per unit area
1826 A <- 1        # Area of unit square
1827 N <- rpois(1, D*A)
1828 plot(s <- cbind(runif(N), runif(N)))
```

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

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

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

2.2.6 Other distributions

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

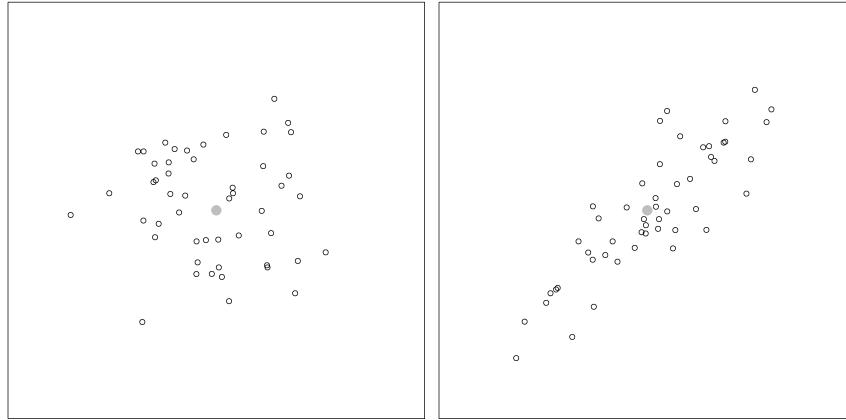


Figure 2.2. Two realized point patterns from the bivariate normal distribution.

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

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

```

1861 library(mvtnorm)
1862 set.seed(3)
1863 mu <- c(0,0)
1864 Sigma <- matrix(c(1, .9, .9, 1), 2, 2)
1865 X1 <- cbind(rnorm(50, mu[1], Sigma[1,1]), # No correlation (rho=0)
1866                 rnorm(50, mu[2], Sigma[2,2]))
1867 X2 <- rmvnorm(50, mu, Sigma)           # rho=0.9

```

Fig. 2.2 shows the simulated points.

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

1871 valued living between zero and ∞ . The beta distribution is the standard prior
 1872 used for probabilities because it can be used to express either a lack of knowledge
 1873 or very precise knowledge about a parameter. For example, a Beta(1,1) distribu-
 1874 tion is equivalent to a Uniform(0,1) distribution. However, unlike the uniform
 1875 distribution, the beta distribution can be used as an informative prior; for exam-
 1876 ple if published estimates of detection probability exist we can choose parameters
 1877 of the beta distribution to reflect that. To gain some familiarity with the beta
 1878 distribution, execute the following R commands:

```
1879 curve(dbeta(x, 1, 1), col="black", ylim=c(0,5))
1880 curve(dbeta(x, 10, 10), col="blue", add=TRUE)
1881 curve(dbeta(x, 10, 20), col="darkgreen", add=TRUE)
```

1882 Other parameters in SCR models are continuous but positive-valued and can be
 1883 modeled using the gamma distribution. As with the beta distribution, the gamma
 1884 distribution is typically favored over the uniform distribution when one is interested
 1885 in using an informative prior. It is also frequently used as a vague prior for the
 1886 inverse of variance parameters, but it is wise to compare this prior to a uniform to
 1887 assess its influence on the posterior.

2.3 STATISTICAL INFERENCE AND PARAMETER ESTIMATION

1888 If the parameters of a statistical model were known with absolute certainty, then it
 1889 would be possible to use pdfs and pmfs to make direct probability statements about
 1890 unknowns such as future outcomes. However, we almost never know the actual
 1891 values of parameters, and instead we have to estimate them from observations
 1892 (i.e., data). Our inferences must then acknowledge the uncertainty associated with
 1893 our imperfect knowledge of the parameters. Doing so is most often accomplished
 1894 using one of two approaches: classical (frequentist) inference or Bayesian inference.
 1895 These two modes of inference regard the uncertainty about parameters in entirely
 1896 different ways. In the next chapter, we will review some of the important concepts
 1897 in Bayesian inference, so here, we will focus on the frequentist perspective.

1898 Suppose we count oak trees at J sites, and the resulting data $\{y_1, \dots, y_J\}$ can
 1899 be assumed to be *iid* outcomes from some distribution, such as the Poisson with
 1900 unknown parameter λ . We want to estimate this parameter. In classical inference,
 1901 the only uncertainty about λ is that attributable to sampling. For instance, we can
 1902 imagine repeatedly sampling the population (sites in this example) and obtaining
 1903 sample-specific estimates of λ . Typically, we entertain the idea that there are an
 1904 infinite number of possible samples and so we could obtain an infinite number of
 1905 estimates: $\{\hat{\lambda}_1, \hat{\lambda}_2, \dots, \hat{\lambda}_\infty\}$. If these estimates are produced using the method
 1906 of maximum likelihood, and as n tends to infinity, the distribution of estimates,
 1907 called the sampling distribution, will be normally distributed with $E(\hat{\lambda}) = \lambda$. The
 1908 standard deviation of the sampling distribution is called the standard error, which
 1909 can also be estimated as part of the maximum likelihood procedure. Of course, we

1910 almost always have just a single sample of data, and hence a single $\hat{\lambda}$ and a single
 1911 estimate of the standard error. However, under the assumption of a normally
 1912 distributed sampling distribution, we can construct a confidence interval that will
 1913 include the true value of λ with coverage probability $1 - \alpha$, where α is a prescribed
 1914 value like 0.05. An important point is that there is no uncertainty associated with
 1915 the actual parameter—it is regarded as a fixed value, and hence probability is only
 1916 used to characterize the estimator via its sampling distribution.

1917 Maximum likelihood is heuristically a method of finding the most “likely” value
 1918 of λ , given the observed data, and of characterizing the variance of the sampling dis-
 1919 tribution. Of course, it also applies to cases where the observations are multivariate,
 1920 or the probability distribution is a function of multiple parameters. Endless num-
 1921 bers of textbooks and online resources are available for those interested in a detailed
 1922 explanation of maximum likelihood. For our purposes, we wish to keep it simple
 1923 and focus on *how* to do it. The first step is to define the likelihood function, which
 1924 is the joint distribution of the data regarded as a function of the parameter(s). If
 1925 the joint distribution of the observations is denoted by $[y_1, y_2, \dots, y_n | \lambda]$, we usually
 1926 denote the likelihood by flipping the arguments: $\mathcal{L}(\lambda | \mathbf{y}) = [\lambda | y_1, y_2, \dots, y_n]$.

1927 If the observations are *iid*, the likelihood simplifies to

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

1928 where $[y_i | \lambda]$ is a probability distribution, like those discussed in the previous sec-
 1929 tions. For example, if y_i is Poisson distributed, then $[y_i | \lambda] = \text{Poisson}(\lambda) = \frac{\lambda^{y_i} e^{-\lambda}}{y_i!}$.
 1930 Although likelihoods are typically shown on the natural scale, we almost always
 1931 maximize the logarithm of the likelihood to avoid computational problems that
 1932 arise when multiplying very small probabilities. Thus, we rewrite Eq. 2.3.1 as

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

1933 Here is some simple **R** code to simulate independent Poisson outcomes and esti-
 1934 mate λ (as though we did not know it) using the method of maximum likelihood.
 1935 Actually, we will minimize the negative log-likelihood because it is equivalent and
 1936 is the default for **R**’s optimizers like `optim` and `nls`.

```
1937 > lambda <- 3 # Actual parameter value
1938 > y1 <- rpois(100, lambda) # Realized values (data)
1939 > negLogLike1 <- function(par) -sum(dpois(y1, par, log=TRUE))
1940 > starting.value <- c('lambda'=1)
1941 > optim(starting.value, negLogLike1)$par # MLE
1942   lambda
1943 3.039844
```

1944 Explicitly maximizing the likelihood, numerically, isn't actually necessary here be-
 1945 cause the MLE of λ is given by the mean of the observations. A more interesting
 1946 example is when there are covariates of λ . For example, suppose λ is a function of
 1947 elevation and vegetation height according to: $\log(\lambda_i) = \beta_0 + \beta_1 ELEV_i + \beta_2 VEGHT_i$.
 1948 This is a standard Poisson regression problem, with likelihood:

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

1949 This likelihood is almost identical to the previous one except that λ is now a
 1950 function, and so we need to estimate the parameters of the function, i.e. the β 's.
 1951 Some code to fit this model to simulated data is shown here:

```
1952 > nsites <- 100
1953 > elevation <- rnorm(100)
1954 > veght <- rnorm(100)
1955 > beta0 <- 1
1956 > beta1 <- -1
1957 > beta2 <- 0
1958 > lambda <- exp(beta0 + beta1*elevation + beta2*vegght)
1959 > y2 <- rpois(nsites, lambda)
1960 > negLogLike2 <- function(pars) {
1961   +   beta0 <- pars[1]
1962   +   beta1 <- pars[2]
1963   +   beta2 <- pars[3]
1964   +   lambda <- exp(beta0 + beta1*elevation + beta2*vegght)
1965   +   -sum(dpois(y2, lambda, log=TRUE))
1966   +
1967 > starting.values <- c('beta0'=0, 'beta1'=0, 'beta2'=0)
1968 > optim(starting.values, negLogLike2)$par
1969   beta0      beta1      beta2
1970 0.98457756 -1.03025173 -0.01218292
```

1971 We see that the maximum likelihood estimates (MLEs) are very close to the true
 1972 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 μ and σ^2 . In this case, β_i would

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

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

1973 This is an example of a mixed effects model or a hierarchical model. How do we
 1974 estimate the parameters of a model that includes random effects? Earlier the like-
 1975 lihood function was written as the product of probabilities determined by a single
 1976 pmf or pdf, $[y|\lambda]$, but now we have an additional random variable, and we are forced
 1977 to think about conditional relationships, because y depends upon β_i and β_i depends
 1978 upon other parameters, specifically μ and σ^2 . This type of conditional dependence
 1979 among parameters is the essence of hierarchical models, and statistical analysis
 1980 of hierarchical models requires that we discuss joint distributions, marginal distri-
 1981 butions and conditional distributions. These concepts will be used extensively in
 1982 Chapt. 6 where we demonstrate how to estimate parameters of hierarchical models
 1983 using maximum likelihood.

2.4 JOINT, MARGINAL, AND CONDITIONAL DISTRIBUTIONS

1984 So far we have restricted our attention to situations in which we wish to make
 1985 inference about a single random variable. However, in ecology, we often are inter-
 1986 ested in multiple random variables and how they are related. Let Y be a random
 1987 variable that may or may not be independent of X (here again we will distinguish
 1988 between random variables and realized values for conceptual clarity). Inference
 1989 about these two random variables can be made using the joint, marginal, or condi-
 1990 tional distributions—or, we may make use of all of them depending on the question
 1991 being asked. In the case of discrete random variables, the joint distribution is the
 1992 probability that X takes on the value x and that Y takes on the value y , which
 1993 is written $[X = x, Y = y]$. To clarify this concept, let's go back to our original
 1994 example where X was the number of fish caught after 20 casts, which we said
 1995 was an *iid* binomial random variable. Now, let's suppose that X depends on the
 1996 random variable Y , which is the number of other fisherman at the hole. Specifi-
 1997 cally, let's say that the probability of catching a fish p is related to Y according
 1998 to $\text{logit}(p) = -0.6 + -2y$. Furthermore, let's make the intuitive assumption that
 1999 the number of fishermen at the hole is a Poisson random variable with mean 0.6,
 2000 i.e. $Y \sim \text{Poisson}(0.6)$. Our model is now fully specified, and so we can answer the
 2001 question: “what is the probability of catching x fish and of there being y fishermen
 2002 at the hole”. This joint distribution is given by the product of the binomial pmf
 2003 (with p determined by y) and the Poisson pmf with $\lambda = 0.6$. The following R code
 2004 creates the joint distribution.

```
2005 > X <- 0:20 # All possible values of X
2006 > Y <- 0:10 # All possible values of Y
2007 > lambda <- 0.6
```

```

2008 > p <- plogis(-0.62 + -2*Y) # p as function of Y
2009 > round(p,2)
2010 [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
2011 > joint <- matrix(NA, length(X), length(Y))
2012 > rownames(joint) <- paste("X=", X, sep="")
2013 > colnames(joint) <- paste("Y=", Y, sep="")
2014 >
2015 > # Joint distribution [X,Y]
2016 > for(i in 1:length(Y)) {
2017 +   joint[,i] <- dbinom(X, 20, p[i]) * dpois(Y[i], lambda)
2018 + }
2019 > round(joint,2)
2020   Y=0  Y=1  Y=2  Y=3  Y=4  Y=5  Y=6  Y=7  Y=8  Y=9  Y=10
2021 X=0  0.00 0.08 0.08 0.02  0  0  0  0  0  0  0
2022 X=1  0.00 0.12 0.02 0.00  0  0  0  0  0  0  0
2023 X=2  0.01 0.08 0.00 0.00  0  0  0  0  0  0  0
2024 X=3  0.02 0.04 0.00 0.00  0  0  0  0  0  0  0
2025 X=4  0.04 0.01 0.00 0.00  0  0  0  0  0  0  0
2026 X=5  0.07 0.00 0.00 0.00  0  0  0  0  0  0  0
2027 X=6  0.09 0.00 0.00 0.00  0  0  0  0  0  0  0
2028 X=7  0.10 0.00 0.00 0.00  0  0  0  0  0  0  0
2029 X=8  0.09 0.00 0.00 0.00  0  0  0  0  0  0  0
2030 X=9  0.06 0.00 0.00 0.00  0  0  0  0  0  0  0
2031 X=10 0.04 0.00 0.00 0.00  0  0  0  0  0  0  0
2032 X=11 0.02 0.00 0.00 0.00  0  0  0  0  0  0  0
2033 X=12 0.01 0.00 0.00 0.00  0  0  0  0  0  0  0
2034 X=13 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
2035 X=14 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
2036 X=15 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
2037 X=16 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
2038 X=17 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
2039 X=18 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
2040 X=19 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
2041 X=20 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0

```

2042 This matrix tells us the probability of all possible combinations of x and y , and
 2043 we see that the most likely value is $(X = 1, Y = 1)$, i.e. we will catch 1 fish and
 2044 there will be 1 other fisherman. This matrix also demonstrates the law of total
 2045 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:

```
2051 > margX <- rowSums(joint)
2052 > round(margX, 2)
2053   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
2054 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
2055 X=15  X=16  X=17  X=18  X=19  X=20
2056 0.00 0.00 0.00 0.00 0.00 0.00
```

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

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

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

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

```
2059 > XgivenY <- joint/matrix(margY, nrow(joint), ncol(joint), byrow=TRUE)
2060 > round(XgivenY, 2)
2061   Y=0  Y=1  Y=2  Y=3  Y=4  Y=5  Y=6  Y=7  Y=8  Y=9  Y=10
2062   X=0  0.00 0.25 0.82 0.97  1  1  1  1  1  1  1
2063   X=1  0.00 0.36 0.16 0.03  0  0  0  0  0  0  0
2064   X=2  0.01 0.25 0.02 0.00  0  0  0  0  0  0  0
2065   X=3  0.03 0.11 0.00 0.00  0  0  0  0  0  0  0
2066   X=4  0.07 0.03 0.00 0.00  0  0  0  0  0  0  0
2067   X=5  0.13 0.01 0.00 0.00  0  0  0  0  0  0  0
2068   X=6  0.17 0.00 0.00 0.00  0  0  0  0  0  0  0
2069   X=7  0.18 0.00 0.00 0.00  0  0  0  0  0  0  0
```

2072	X=8	0.16	0.00	0.00	0.00	0	0	0	0	0	0	0
2073	X=9	0.12	0.00	0.00	0.00	0	0	0	0	0	0	0
2074	X=10	0.07	0.00	0.00	0.00	0	0	0	0	0	0	0
2075	X=11	0.03	0.00	0.00	0.00	0	0	0	0	0	0	0
2076	X=12	0.01	0.00	0.00	0.00	0	0	0	0	0	0	0
2077	X=13	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
2078	X=14	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
2079	X=15	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
2080	X=16	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
2081	X=17	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
2082	X=18	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
2083	X=19	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
2084	X=20	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0

2085 Note that we have 11 probability distributions for X , one for each possible value of
 2086 Y , and each pmf sums to unity as it should. Note also that if you show up at the
 2087 hole and there are > 2 fisherman, your chance of catching a fish is very low. Go
 2088 home. These concepts are explained in more detail in other texts such as Casella
 2089 and Berger (2002), Royle and Dorazio (2008), and Link and Barker (2010), but
 2090 hopefully, the code shown here complements the equations and makes it easier for
 2091 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:

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

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

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

2092 From here on out, when you see such notation, you should immediately grasp
 2093 the fact that Y is a random variable independent of X , but X depends upon
 2094 Y through p . Now you have the tools to make probability statements about the
 2095 random variables in this system. The one caveat faced in reality is that we typically
 2096 do not know the values of the parameters, and instead we have to estimate them.
 2097 Maximum likelihood methods for hierarchical models are covered in Chapt. 6.

2.5 HIERARCHICAL MODELS AND INFERENCE

2098 The term hierarchical modeling (or hierarchical model) has become something of
 2099 a buzzword over the last decade with hundreds of papers published in ecological
 2100 journals using that term. So then, what exactly is a hierarchical model, anyhow?
 2101 Obviously, this term stems from the root “hierarchy” which means:

2102 **Definition:** *hierarchy* (noun) – a series of ordered groupings of people or things
 2103 within a system;

2104 In the case of a hierarchical model (hierarchical being the adjective form of hi-
 2105 erarchy), the “things” are probability distributions, and they are ordered according
 2106 to their conditional probability structure. Thus, a hierarchical model is *an ordered*
 2107 *series of models, ordered by their conditional probability structure.*

2108 A canonical hierarchical model in ecology is this elemental model of species
 2109 occurrence or distribution (MacKenzie et al., 2002; Tyre et al., 2003; Kéry, 2011):

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

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

2111 where y_i = observation of presence/absence at a site i and z_i = occurrence status
 2112 ($z_i = 1$ if a species occurs at site i and $z_i = 0$ if not). Note that if $p = 1$, then we
 2113 would perfectly observe z and the model would no longer be hierarchical—it would
 2114 be a simple logistic regression model. Note also that this hierarchical model has an
 2115 important conceptual distinction between other types of classical multi-level models
 2116 such as repeated measures on subjects, in that z_i is an actual state of nature. In
 2117 that sense, z is a random variable that is the outcome of a “real” process. Royle
 2118 and Dorazio (2008) used the term *explicit* hierarchical model to describe this type of
 2119 model to distinguish from hierarchical models (*implicit* hierarchical models) where
 2120 the latent variables don’t correspond to an actual state of nature—but rather just
 2121 soak up variation that is unmodeled by explicit elements of the model. At best,
 2122 latent variables in such models are surrogates for something of ecological relevance
 2123 (“time effects”, “space effects” etc.).

2124 With these examples, we expand on our definition of a hierarchical model as we
 2125 will use it in this book:

2126 **Definition: Hierarchical Model:** A model with explicit component models that de-
 2127 scribe variation in the data due to (spatial/temporal) variation in *ecological process*,
 2128 and due to *imperfect observation* of the process.

2129 Most models considered in this book describe the encounter of individuals con-
 2130 ditional on the “activity center” of the individual, which is a latent variable (i.e.,
 2131 unobserved random effect). The definition of an activity center will be context-
 2132 dependent as discussed in Chapt. 5, but often it can be thought of as an individual’s
 2133 home range center. The collection of these latent variables represents the outcome
 2134 of an ecological process describing how individuals distribute themselves over the
 2135 landscape. Moreover, how individuals are encountered in traps is, in some cases,
 2136 the result of a model governing movement. As such, these models are examples of
 2137 hierarchical models that contain formal model components representing both eco-
 2138 logical process and also the observation of that process. That is, they are explicit
 2139 hierarchical models (Royle and Dorazio, 2008) as opposed to implicit hierarchical
 2140 models.

2.6 CHARACTERIZATION OF SCR MODELS

2141 For the purposes of this book, an SCR model is any “individual encounter model”
 2142 (not just “capture-recapture”!) where auxiliary spatial information is also obtained.
 2143 To be more precise we could as well use the term “spatial capture and/or recap-
 2144 ture” but that is slightly unwieldy and, besides, it also abbreviates to SCR. The
 2145 class of SCR models includes traditional capture-recapture models with auxiliary
 2146 spatial information and even some models that do not even require “recapture”
 2147 (e.g., distance sampling). There is even a class of models (Chapt. 18) which don’t
 2148 require capture or unique identification of individuals.

2149 Conceptually, SCR models involve a collection of random variables, \mathbf{s} , \mathbf{u} and
 2150 y where \mathbf{s} is the activity center, or home range center, \mathbf{u} is the location of the
 2151 individual at the time of sampling, which we may think of as a realization from some
 2152 movement model, and y is the “response variable”—what the observer records. For
 2153 example, $y = 1$ means “detected” and $y = 0$ means “not detected”, but many other
 2154 types of responses are possible (Chapt 9). A broad class of models for estimating
 2155 density are unified by a hierarchical model involving explicit models for animal
 2156 activity centers \mathbf{s} , movement outcomes \mathbf{u} , and encounter data y . In some cases, we
 2157 don’t observe y but rather summaries of y , say $n(y)$, yet it might be convenient
 2158 in such cases to retain an explicit focus on y in terms of model construction. We
 2159 thus introduce a sequence of models—a hierarchical model—to relate these random
 2160 variables, which can be written as

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

2161 Every model we talk about in this book has a subset of these components although
 2162 we never fit the full model because we have not encountered a situation requiring
 2163 that we do so. However, a detailed description of this model and its various com-
 2164 ponents is the subject of this book, and we will not pretend to condense hundreds
 2165 of pages of material into the next few paragraphs. However, we give a cursory
 2166 overview here to whet the appetite and provide some indication of where we are
 2167 going. Don’t worry if some of this material doesn’t sink in just yet—we will walk
 2168 through it slowly in the subsequent chapters.

2169 Let’s begin with the model $[\mathbf{s}]$ that describes the distribution of the activity
 2170 centers of each animal in the spatial region \mathcal{S} (the state-space as we called it previ-
 2171 ously). As will be explained in Chapt. 5 and Chapt. 11, $[\mathbf{s}]$ defines a spatial point
 2172 process, which may be inhomogeneous if there exists spatial variation in density, or
 2173 it may be homogeneous if density is constant throughout \mathcal{S} . In the later case, we can
 2174 write $[\mathbf{s}] = \text{Uniform}(\mathcal{S})$, which is to say that the N activity centers are uniformly
 2175 distributed in the polygon \mathcal{S} . A point process is also a model for the number of indi-
 2176 viduals in the population N . So we could write $[\mathbf{s}|\mu]$ where μ is an intensity param-
 2177 eter defined as the number of points per unit area. In other words, μ is population
 2178 density, and we often model population size as either $N \sim \text{Poisson}(\mu A(\mathcal{S}))$, where
 2179 $A(\mathcal{S})$ is the area of the state-space; or, $N \sim \text{Binomial}(M, \psi)$ where $\psi = \mu A(\mathcal{S})/M$

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

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

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

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

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

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

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

2214 population size N is Poisson distributed (2) activity centers are uniformly distributed in two-dimensional space, and (3) capture probability is a function of the
 2215 distance between the activity and the trap. Each of these model components can
 2216 be modified as needed to match specific hypotheses, study designs, and data struc-
 2217 tures. For example, spatial variation in abundance or density can be easily modeled
 2218 as a function of habitat covariates (Chapt. 11).

2219 We realize that many the model description in Eq. 2.6.2 may not be self-evident
 2220 to some ecologists. However, it is absolutely essential that one can understand
 2221 such a model description—not just for being able to read this book, but also for
 2222 understanding any statistical model in ecology. One of the best ways of familiarizing
 2223 oneself with this notation is to translate it into **R** code that simulates outcomes
 2224 from the model. The following code is an example.

```
2226 set.seed(36372)
2227 Area <- 1                                # area of state-space (unit square)
2228 x <- cbind(rep(seq(.1,.9,.2), each=5),    # trap locations
2229             rep(seq(.1,.9,.2), times=5))
2230 p0 <- 0.3                                 # baseline capture probability
2231 sigma <- 0.05                             # Gaussian scale parameter
2232 mu <- 50                                  # population density
2233 N <- rpois(1, mu*Area)                   # population size
2234 s <- cbind(runif(N, 0, 1),                # activity centers in unit square
2235             runif(N, 0, 1))
2236 K <- 5
2237 y <- matrix(NA, N, nrow(x))            # capture data
2238 for(i in 1:N) {
2239   d.ij <- sqrt((x[,1] - s[i,1])^2 +      # distance between x and s[i]
2240                 (x[,2] - s[i,2])^2)
2241   p.ij <- p0*exp(-d.ij^2 / (2*sigma^2)) # capture probability
2242   y[i,] <- rbinom(nrow(x), K, p.ij)       # capture history for animal i
2243 }
```

2244 Fig. 2.3 shows the results of this simulation from a basic, yet very useful, SCR
 2245 model.

2246 Having briefly explained each of the model components in Eq. 2.6.1, and having
 2247 shown how a subset of these components results in a basic SCR model, we can
 2248 now discuss other relevant arrangements. Examples include: (1) Classical distance
 2249 sampling (Buckland et al., 2001; Borchers et al., 2002), (2) Spatial capture-recapture
 2250 models with fixed arrays of traps (Efford, 2004; Borchers and Efford, 2008; Royle
 2251 et al., 2009a,b; Gardner et al., 2010a; Royle et al., 2011b), and (3) Search-encounter
 2252 models (Royle and Young, 2008; Royle et al., 2011a). We will now elaborate on
 2253 some of these distinctions.

2254 1. **Distance sampling.** The last 2 stages of the hierarchy are confounded
 2255 (implicitly) and so analysis is based on the model $[y|\mathbf{u}][\mathbf{u}]$. The “process
 2256 model” is that of “uniformity”: $\mathbf{u} \sim \text{Uniform}(\mathcal{S})$.

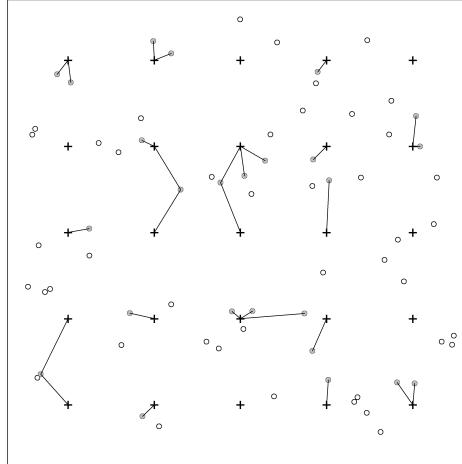


Figure 2.3. Population of $N = 69$ home-range centers (s , circles) and 25 trap locations (x , crosses). Lines connect activity centers to the traps where the individuals were detected. As in many SCR models, movement outcomes (u) are ignored.

2257 2. **Spatial capture-recapture model with a fixed array of traps.** SCR
 2258 models appear to have little in common with distance sampling because ob-
 2259 servations are made only at a pre-defined set of discrete locations—where
 2260 traps are placed. However, the models are closely related in terms of our
 2261 hierarchical representation above. In SCR models based on fixed arrays, we
 2262 cannot estimate both $\Pr(y = 1|u)$ and $\Pr(u|s)$ —the probability that an in-
 2263 dividual “moves to u ” cannot be separated from the probability that it is
 2264 detected given that it moves to u , because of the fact that the observation
 2265 locations are fixed by design. Formally, such SCR models confound $[y|u]$
 2266 with $[u|s]$ so that the observation model arises as:

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

2267 This confounding happens because SCR sampling is spatially biased—restricted
 2268 to a fixed pre-determined set of locations. Conversely, distance sampling
 2269 confounds $[u|s][s]$ because, essentially, there is only a single realization of the
 2270 encounter process. It is probably reasonable to assume that $\Pr(y = 1|u) = 1$
 2271 or at least it is locally constant for most devices (e.g., cameras, etc..), and
 2272 thus the detection model will have the interpretation in terms of movement
 2273 (see Chapt. 13 and 12).

2274 3. **Search-encounter models.** What we call “search-encounter” models (Royle

2275 and Young, 2008; Royle et al., 2011a) are kind of a hybrid model combining
2276 features of SCR models and features of distance sampling. Like distance
2277 sampling they allow for encounters in continuous space which provide di-
2278 rect observations from $[\mathbf{u}|\mathbf{s}]$. Thus, the hierarchical model is fully identified.
2279 These models are described in Chapt. chapt.search-encounter.

2.7 SUMMARY AND OUTLOOK

2280 Spatial capture-recapture models are hierarchical models, and hierarchical models
2281 are models of multiple random variables that are conditionally related. It is there-
2282 fore important that the basic rules of modeling random variables are understood,
2283 and we hope that this chapter has made some of the basic concepts accessible to
2284 ecologists with rudimentary background in statistics. If some of this material still
2285 seems difficult to grasp, we recommend working with the provided **R** code, which
2286 is perhaps the best way of making the equations more tangible.

2287 In some respects, it is possible to understand the jist of SCR without knowing
2288 anything about marginal and conditional relationships. One can always fit models
2289 using canned software and interpret the output without understanding the guts of
2290 the model or the details of the estimation process. For some applied ecologists,
2291 this may be perfectly fine, and this book is meant to be useful for both statistical
2292 novices and ecologists with more advanced quantitative skills. In most chapters, we
2293 begin with a basic conceptual discussion, then we explain the technical details that
2294 require an understanding of the concepts in this chapter, and finally we end with
2295 one or more worked examples. For those not interested in the technical details,
2296 we recommend focusing on the chapter introductions and the examples. However,
2297 taking the time to understand the concepts presented in this chapter can only
2298 increase one's ability to tackle the unique and complex problems that often present
2299 themselves when modeling spatial and temporal aspects of population dynamics.

2300
2301

3

2302

GLMS AND BAYESIAN ANALYSIS

2303 A major theme of this book is that spatial capture-recapture models are, for the
2304 most part, just generalized linear models (GLMs) wherein the covariate, distance
2305 between trap and home range center, is partially or fully unobserved – and therefore
2306 regarded as a random effect. Outside of capture-recapture, such models are usually
2307 referred to as generalized linear mixed models (GLMMs) and, therefore, SCR mod-
2308 els can be thought of as a specialized type of GLMM. Naturally then, we should
2309 consider analysis of these slightly simpler models in order to gain some experience
2310 and, hopefully, develop a better understanding of spatial capture-recapture models.

2311 In this chapter, we consider classes of GL(M)Ms – Poisson and binomial (i.e.,
2312 logistic regression) models – that will prove to be enormously useful in the analysis
2313 of capture-recapture models of all kinds. Many readers are likely familiar with these
2314 models already because they are among the most useful models in ecology and,
2315 as such, have received considerable attention in many introductory and advanced
2316 texts. We focus on them here in order to introduce the readers to the analysis of
2317 such models in **R** and **WinBUGS** or **JAGS**, which we will translate directly to
2318 the analysis of SCR models in subsequent chapters.

2319 Bayesian analysis is convenient for analyzing GL(M)Ms because it allows us to
2320 work directly with the conditional model – i.e., the model that is conditional on the
2321 random effects, using computational methods known as Markov chain Monte Carlo
2322 (MCMC). Learning how to do Bayesian analysis of GLMs and GLMMs using the
2323 **BUGS** language is, in part, the purpose of this chapter. We focus here on the use of
2324 **WinBUGS** because it is the most popular “**BUGS** engine”. However, later in the
2325 book we transition to another popular **BUGS** engine known as **JAGS** (Plummer,
2326 2009) which stands for *Just Another Gibbs Sampler*. For most of our purposes, the
2327 specification of models in either platform is the same, but **JAGS** is under active
2328 development at the present time while **WinBUGS** no longer is, having transitioned

2329 to **OpenBUGS** (Lunn et al., 2009) which is still in active development. While we
 2330 use **BUGS** of one sort or another to do the Bayesian computations, we organize and
 2331 summarize our data and execute **WinBUGS** or **JAGS** from within **R** using the
 2332 packages **R2WinBUGS** (Sturtz et al., 2005), **R2jags** (Su and Yajima, 2011) or **rjags**
 2333 (Plummer, 2009). Kéry (2010), and Kéry and Schaub (2012) provide excellent
 2334 and accessible introductions to the basics of Bayesian analysis and GL(M)Ms using
 2335 **WinBUGS**. We don't want to be too redundant with those books and so we avoid
 2336 a detailed treatment of Bayesian methodology and software usage - instead just
 2337 providing a cursory overview so that we can move on and attack the problems
 2338 we're most interested in related to spatial capture-recapture. In addition, there are
 2339 a number of texts that provide general introductions to Bayesian analysis, MCMC,
 2340 and their applications in ecology including McCarthy (2007), Kéry (2010), Link
 2341 and Barker (2010), and King et al. (2008).

2342 While this chapter is about Bayesian analysis of GL(M)Ms, such models are
 2343 routinely analyzed using likelihood methods too. Later in this book (Chapt. 6), we
 2344 will use likelihood methods to analyze SCR models but, for now, we concentrate on
 2345 providing a basic introduction to Bayesian analysis because that is the approach
 2346 we will use in a majority of cases in later chapters.

3.1 GLMS AND GLMMS

2347 We have asserted already that SCR models work out most of the time to be variations
 2348 of GL(M)Ms. You might therefore ask: What are these GLM and GLMM
 2349 models, anyhow? These models are covered extensively in many very good applied
 2350 statistics books and we refer the reader elsewhere for a detailed introduction. The
 2351 classical references for GLMs are Nelder and Wedderburn (1972) and McCullagh
 2352 and Nelder (1989). In addition, we think Kéry (2010), Kéry and Schaub (2012),
 2353 and Zuur et al. (2009) are all accessible treatments. Here, we'll give the 1 minute
 2354 treatment of GL(M)Ms, not trying to be complete but rather only to preserve a
 2355 coherent organization to the book.

2356 The GLM is an extension of standard linear models allowing the response variable
 2357 to have some distribution from the exponential family of distributions. This
 2358 includes the normal distribution but also others such as the Poisson, binomial,
 2359 gamma, exponential, and many more. In addition, GLMs allow the response variable
 2360 to be related to the predictor variables (i.e., covariates) using a link function,
 2361 which is usually nonlinear. The GLM consists of three components:

- 2362 1. A probability distribution for the dependent (or response) variable y , from the
 2363 exponential family of probability distributions.
- 2364 2. A "linear predictor" $\eta = \beta_0 + x\beta_1$, where x is a predictor variable (i.e., a covariate).
- 2365 3. A link function g that relates the expected value of y , $\mathbb{E}(y)$, to the linear predictor,
 2366 $\mathbb{E}(y) = \mu = g^{-1}(\eta)$. Therefore $g(\mathbb{E}(y)) = \eta = \beta_0 + x\beta_1$.

2368 A key aspect of GLMs is that $g(\mathbb{E}(y))$ is assumed to be a linear function of the
 2369 predictor variable(s), here x , with unknown parameters, here β_0 and β_1 , to be
 2370 estimated. In standard GLMs, the variance of y is a function V of the mean of y :
 2371 $\text{Var}(y) = V(\mu)$ (see below for examples). As an example, a Poisson GLM posits
 2372 that $y \sim \text{Poisson}(\lambda)$ with $\mathbb{E}(y) = \lambda$ and usually the model for the mean is specified
 2373 using the *log link function* by

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

2374 The variance function is $V(y_i) = \lambda_i$. To see how a Poisson GLM works, use the **R**
 2375 code below to simulate some data and then estimate the parameters:

```
2376 > set.seed(13)
2377 > n <- 100          # set sample size
2378 > beta0 <- -2       # set intercept term
2379 > beta1 <- 1.5      # set coefficient
2380 > x <- rnorm(n, 0,1) # generate a predictor variable, x
2381
2382 > linpred <- beta0 + beta1*x # calculate linear predictor of E(y)
2383 > y <- rpois(n, exp(linpred)) # generate observations from model
```

2384 The **R** function `glm()` fits a GLM to the data we just generated and returns estimates of
 2385 β_0 and β_1 , which we see are fairly close to the data generating values above:

```
2386 > glm(y ~ 1 + x, family='poisson')      # the fit model
```

2387 This produces the output:

```
2388 Call: glm(formula = y ~ 1 + x, family = "poisson")
2389
2390 Coefficients:
2391 (Intercept)      x
2392     -2.007      1.446
2393
2394 [... some output deleted ...]
```

2395 In this summary output, the maximum likelihood estimates (MLEs) of the regression
 2396 parameters β_0 and β_1 are labeled “Coefficients.” We see that these are not too different
 2397 from the data-generating values (-2 and 1.5, respectively).

2398 The binomial GLM posits that $y_i \sim \text{Binomial}(K, p)$ where K is the fixed sample size
 2399 parameter and $\mathbb{E}(y_i) = K \times p_i$. Usually the model for the mean is specified using the *logit*
 2400 *link function* according to

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

2401 Where $\text{logit}(p) = \log(p/(1-p))$. The inverse-logit function, consequently, is $\text{logit}^{-1}(p) =$
 2402 $\exp(p)/(1+\exp(p))$.

2403 A GLMM is the extension of GLMs to accommodate “random effects”. Often this
 2404 involves adding a normal random effect to the linear predictor. One simple example is
 2405 using a random intercept, α :

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

2406 where

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

2407 Many other probability distributions and formulations of the linear predictor might be
 2408 considered. GLMMs are enormously useful in ecological modeling applications for mod-
 2409 eling variation due to subjects, observers, spatial or temporal stratification, clustering,
 2410 and dependence that arises from any kind of group structure and, of course, because SCR
 2411 models prove to be a type of GLM with a random effect, but one that does not enter the
 2412 mean linearly.

3.2 BAYESIAN ANALYSIS

2413 Bayesian analysis is less familiar to many ecological researchers because they are often
 2414 educated only in the classical statistical paradigm of frequentist inference. But advances
 2415 in technology and increasing exposure to the benefits of Bayesian analysis are fast mak-
 2416 ing Bayesians out of people or at least making Bayesian analysis an acceptable, general
 2417 alternative to classical, frequentist inference.

2418 Conceptually, the main thing about Bayesian inference is that it uses probability
 2419 directly to characterize uncertainty about things we don't know. "Things", in this case,
 2420 are parameters of models and, just as it is natural to characterize uncertain outcomes of
 2421 stochastic processes using probability, it seems natural also to characterize information
 2422 about unknown parameters using probability. At least this seems natural to us and, we
 2423 think, most ecologists either explicitly adopt that view or tend to fall into that point
 2424 of view naturally. Conversely, frequentists use probability in many different ways, but
 2425 never to characterize uncertainty about parameters¹. Instead, frequentists use probability
 2426 to characterize the behavior of *procedures* such as estimators or confidence intervals (see
 2427 below). It is surprising that people readily adopt a philosophy of statistical inference in
 2428 which the things you don't know (i.e., parameters) should *not* be regarded as random
 2429 variables, so that, as a consequence, one cannot use probability to characterize one's state
 2430 of knowledge about them.

2431 3.2.1 Bayes' rule

2432 As its name suggests, Bayesian analysis makes use of Bayes' rule in order to make direct
 2433 probability statements about model parameters. Given two random variables z and y ,
 2434 Bayes' rule relates the two conditional probability distributions $[z|y]$ and $[y|z]$ by the
 2435 relationship:

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

2436 Bayes' rule itself is a mathematical fact and there is no debate in the statistical community
 2437 as to its validity and relevance to many problems. Generally speaking, these distributions
 2438 are characterized as follows: $[y|z]$ is the conditional probability distribution of y given z ,
 2439 $[z]$ is the marginal distribution of z and $[y]$ is the marginal distribution of y . In the context
 2440 of Bayesian inference we usually associate specific meanings in which $[y|z]$ is thought of
 2441 as "the likelihood", $[z]$ as the "prior" and so on. We leave this for later because here the
 2442 focus is on this expression of Bayes' rule as a basic fact of probability.

¹To hear this will be shocking to some readers perhaps.

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

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

Mathematically, this is

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

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

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

3.2.2 Principles of Bayesian inference

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

2478 all unknown quantities of a model as realizations of random variables – this includes data,
 2479 latent variables, and also parameters. Classical (non-Bayesian) analysts sometimes object
 2480 to regarding parameters as outcomes of random variables. Classically, parameters are
 2481 thought of as “fixed but unknown” (using the terminology of classical statistics). Indeed,
 2482 a common misunderstanding on the distinction between Bayesian and frequentist infer-
 2483 ence goes something like this “in frequentist inference parameters are fixed but unknown
 2484 but in a Bayesian analysis parameters are random.” At best this is a sad caricature of the
 2485 distinction and at worst it is downright wrong. In Bayesian analysis the parameters are
 2486 also unknown and, in fact, there is a single data-generating value of each parameter, and
 2487 so they are also fixed. The difference is that the fixed but unknown values are regarded
 2488 as having been generated from some probability distribution. Specification of that prob-
 2489 ability distribution is necessary to carry out Bayesian analysis, but it is not required in
 2490 classical frequentist inference.

2491 To see the general relevance of Bayes’ rule in the context of statistical inference, let y
 2492 denote observations - i.e., data - and let $[y|\theta]$ be the observation model (often colloquially
 2493 referred to as the “likelihood”). Suppose θ is a parameter of interest having (prior)
 2494 probability distribution $[\theta]$ (also simply referred to as the prior). These are combined to
 2495 obtain the posterior distribution using Bayes’ rule, which is:

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

2496 Asserting the general relevance of Bayes’ rule to all statistical problems, we can conclude
 2497 that the two main features of Bayesian inference are that: (1) parameters, θ , are regarded
 2498 as realizations of a random variable and, as a result, (2) inference is based on the prob-
 2499 ability distribution of the parameters given the data, $[\theta|y]$, which is called the posterior
 2500 distribution. This is the result of using Bayes’ rule to combine the “likelihood” and the
 2501 prior distribution. The key concept is regarding parameters as realizations of a random
 2502 variable because, once you admit this conceptual view, this leads directly to the posterior
 2503 distribution, a very natural quantity upon which to base inference about things we don’t
 2504 know - including parameters of statistical models. In particular, $[\theta|y]$ is a probability
 2505 distribution for θ and therefore we can make direct probability statements to characterize
 2506 uncertainty about θ .

2507 The denominator of our invocation of Bayes’ rule, $[y]$, is the marginal distribution of
 2508 the data y . We note without further remark right now that, in many practical problems,
 2509 this can be an enormous pain to compute. The main reason that the Bayesian paradigm
 2510 has become so popular in the last 20 years or so is because methods have been developed
 2511 for characterizing the posterior distribution that do not require that we possess a math-
 2512 ematical understanding of $[y]$. This means we never have to compute it or know what it
 2513 looks like, or know anything specific about it.

2514 While we can understand the conceptual basis of Bayesian inference merely by under-
 2515 standing Bayes’ rule – that’s really all there is to it – it is not so easy to understand the
 2516 basis of classical frequentist inference. What is mostly coherent in frequentist inference is
 2517 the manner in which procedures are evaluated – the performance of a given procedure is
 2518 evaluated by “averaging over” hypothetical realizations of y , regarding the *estimator* as a
 2519 random variable. For example, if $\hat{\theta}$ is an estimator of θ then the frequentist is interested
 2520 in $E_y(\hat{\theta}|y)$ which is used to characterize bias. If the expected value of $\hat{\theta}$, when averaged
 2521 over realizations of y , is equal to θ , then $\hat{\theta}$ is unbiased.

2522 The view of parameters as being random variables allows Bayesians to use probability
2523 to make direct probability statements about parameters. Frequentist inference procedures
2524 do not permit direct probability statements to be made about parameter values. Instead,
2525 the view of parameters as fixed constants and estimators as random variables leads to
2526 interpretations that are not so straightforward. For example confidence intervals having
2527 the interpretation “95% probability that the interval contains the true value” and p-values
2528 being “the probability of observing an outcome of the test statistic as extreme or more
2529 than the one observed.” These are far from intuitive interpretations to most people.
2530 Moreover, this is conceptually problematic to some because we will never get to observe
2531 the hypothetical realizations that characterize the performance of our procedure.

2532 While we do tend to favor Bayesian inference for the conceptual simplicity (parameters
2533 are random, posterior inference), we mostly advocate for a pragmatic non-partisan
2534 approach to inference because, frankly, some of the frequentist methods are actually very
2535 convenient in certain situations, and will generally yield very similar inferences about
2536 parameters, as we will see in later chapters.

2537 3.2.3 Prior distributions

2538 The prior distribution $[\theta]$ is an important feature of Bayesian inference. As a conceptual
2539 matter, the prior distribution characterizes “prior beliefs” or “prior information” about
2540 a parameter. Indeed, an oft-touted benefit of Bayesian analysis is the ease with which
2541 prior information can be included in an analysis. However, more commonly, the prior
2542 is chosen to express a lack of prior information, even if previous studies have been done
2543 and even if the investigator does in fact know quite a bit about a parameter. This is
2544 because the manner in which prior information is embodied in a prior (and the amount
2545 of information) is usually very subjective and thus the result can wind up being very
2546 contentious; e.g., different investigators might report different results based on subjective
2547 assessments of prior information. Thus it is usually better to “let the data speak” and
2548 use priors that reflect absence of information beyond the data set being analyzed. An
2549 example for an uninformative prior is a Uniform(0, 1) for a probability, or a Uniform($-\infty$,
2550 ∞) (also called a “flat” or “improper” prior) for an unbounded continuous parameter.
2551 Alternatively, people use “diffuse priors”; these contain some information, but (ideally)
2552 not enough to exert meaningful influence on the posterior. An example for a diffuse prior
2553 could be a normal distribution with a large standard deviation.

2554 But still the need occasionally arises to embody prior information or beliefs about a
2555 parameter formally into the estimation scheme. In SCR models we often have a parameter
2556 that is closely linked to “home range size” and thus auxiliary information on the home
2557 range size of a species can be used as prior information, which may improve parameter
2558 estimation (e.g., see Chandler and Royle (In press); also Chapt. 18).

2559 At times the situation arises where a prior can inadvertently impose substantial effect
2560 on the posterior of a parameter, and that is not desirable. For example, we use data
2561 augmentation to deal with the fact that the population size N is an unknown parameter
2562 (Royle et al., 2007) which is equivalent to imposing a Binomial(M, ψ) prior on N for some
2563 integer M (see Sec. 4.2). One has to take care to make sure that M is sufficiently large so
2564 as to not affect the posterior distribution on N (see Fig. 17.6, and also Kéry and Schaub
2565 (2012, Ch. 5)). Another situation that we have to be careful of is that prior distributions

2566 are *not* invariant to transformation of the parameter, and therefore neither are posterior
 2567 distributions (Link and Barker, 2010, Sec. 6.2.1). Thus, a prior that is ostensibly non-
 2568 informative on one scale, may be very informative on another scale. For example, if we
 2569 have a flat prior on $\text{logit}(p)$ for some probability parameter p , this is very different from
 2570 having a Uniform(0,1) prior on p . We show an example where this makes a difference in
 2571 Chapt. 5. Nonetheless, it is always possible to assess the influence of prior choice, and
 2572 it is often the case (with sufficient data and a structurally identifiable model) that the
 2573 influence of priors is negligible.

2574 **3.2.4 Posterior inference**

2575 In Bayesian inference, we are not focusing on estimating a single point or interval but
 2576 rather on characterizing a whole distribution – the posterior distribution – from which
 2577 one can report any summary of interest. A point estimate might be the posterior mean,
 2578 median, mode, etc.. In many applications in this book, we will compute 95% Bayesian
 2579 confidence intervals using the 2.5% and 97.5% quantiles of the posterior distribution. For
 2580 such intervals, it is correct to say $\Pr(L < \theta < U) = 0.95$. That is, “the probability that θ
 2581 lies between L and U is 0.95”.

2582 As an example, suppose we conducted a Bayesian analysis to estimate detection probability
 2583 (p) of some species at a study site, and we obtained a posterior distribution of
 2584 $\text{beta}(20,10)$ for the parameter p . The following R commands demonstrate how we make
 2585 inferences based upon summaries of the posterior distribution:

```
2586 > post.median <- qbeta(0.5, 20, 10)
2587 [1] 0.6704151
2588
2589 > post.95ci <- qbeta(c(0.025, 0.975), 20, 10)
2590 [1] 0.4916766 0.8206164
```

2591 Thus, we can state that there is a 95% probability that θ lies between 0.49 and 0.82. Fig.
 2592 3.1 shows the posterior along with the summary statistics. It is not a subtle thing that
 2593 such statements cannot be made using frequentist methods, although people tend to say
 2594 it anyway and not really understand why it is wrong or even that it is wrong.

2595 **3.2.5 Small sample inference**

2596 The posterior distribution is an exhaustive summary of the state-of-knowledge about an
 2597 unknown quantity. It is *the* posterior distribution - not an estimate of that thing. It is
 2598 also not, usually, an approximation except to within Monte Carlo error (in cases where
 2599 we use simulation to calculate it, see Sec. 3.5.2). One of the great virtues of Bayesian
 2600 analysis which is not widely appreciated is that posterior inference is not “asymptotic”,
 2601 which is to say, valid in a limiting sense as the sample size tends to infinity. Rather,
 2602 posterior inference is valid for *any* sample size and, in particular, *the* sample size on-hand.
 2603 Conversely, almost all frequentist procedures are based on asymptotic approximations to
 2604 the procedure which is being employed.

2605 There seems to be a prevailing view in statistical ecology that classical likelihood-based
 2606 procedures are virtuous because of the availability of simple formulas and procedures for

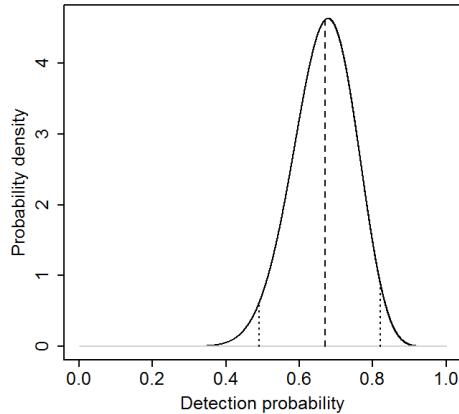


Figure 3.1. Probability density plot of a hypothetical posterior distribution of $\text{beta}(20,10)$; dashed lines indicate mean and upper and lower 95% interval

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

The biologist thus faces a dilemma with such data. On one hand, these data sets, and the resulting inference, are often criticized as being poor and unreliable. Or, even worse², "the data set is so small, this is a poor analysis." On the other hand, such data

²Actual quote from a referee

2629 may be all that is available for species that are extraordinarily important for conservation
2630 and management. The Bayesian framework for inference provides a valid, rigorous, and
2631 flexible framework that is theoretically justifiable in arbitrary sample sizes. This is not to
2632 say that one will obtain precise estimates of density or other parameters, just that your
2633 inference is coherent and justifiable from a conceptual and technical statistical point of
2634 view. That is, for example when we estimate the density D of some animal population,
2635 we report the posterior probability $\Pr(D|data)$ which is easily interpretable and just what
2636 it is advertised to be and we don't need to do a simulation study to evaluate how well
2637 the reported $\Pr(D|data)$ deviates from the "true" $\Pr(D|data)$ because they are the same
2638 quantity.

3.3 CHARACTERIZING POSTERIOR DISTRIBUTIONS BY MCMC SIMULATION

2639 In practice, it is not really feasible to ever compute the marginal probability distribution
2640 [y], the denominator resulting from application of Bayes' rule (Eq. 3.2.1). For decades
2641 (even centuries!) this impeded the adoption of Bayesian methods by practitioners. Or,
2642 the few Bayesian analyses done were based on asymptotic normal approximations to the
2643 posterior distribution. While this was useful from a theoretical and technical standpoint
2644 and, practically, it allowed people to make the probability statements that they naturally
2645 would like to make, it was kind of a bad joke around the Bayesian water-cooler to, on
2646 one hand, criticize classical statistics for being, essentially, completely ad hoc in their
2647 approach to things but then, on the other hand, have to devise various approximations to
2648 what they were trying to characterize. The advent of Markov chain Monte Carlo (MCMC)
2649 methods has made it easier to calculate posterior distributions for just about any problem
2650 to sufficient levels of precision.

2651 Broadly speaking, MCMC is a class of methods for drawing random samples (i.e.,
2652 simulating from or just "sampling") from the target posterior distribution. Thus, even
2653 though we might not recognize the posterior as a named distribution or be able to analyze
2654 its features analytically, e.g., devise mathematical expressions for the mean and variance,
2655 we can use these MCMC methods to obtain a large sample from the posterior and then
2656 use that sample to characterize features of the posterior. What we do with the sample
2657 depends on our intentions – typically we obtain the mean or median for use as a point
2658 estimate, and take a confidence interval based on Monte Carlo estimates of the quantiles.

2659 3.3.1 What goes on under the MCMC hood

2660 We will develop and apply MCMC methods in some detail for spatial capture-recapture
2661 models in Chapt. 17. Here we provide a simple illustration of some basic ideas related to
2662 the practice of MCMC.

2663 A type of MCMC method relevant to most problems is Gibbs sampling (Geman and
2664 Geman, 1984) which we address in more detail in Chapt. 17. Gibbs sampling involves iterative
2665 simulation from the "full conditional" distributions (also called conditional posterior
2666 distributions). The full conditional distribution for an unknown quantity is the conditional
2667 distribution of that quantity given every other random variable in the model - the data
2668 and all other parameters (see Sec. 3.3.2 for rules of how to construct full conditionals).

2669 For example, for a normal regression model ³ with $y \sim \text{Normal}(\beta_0 + \beta_1(x - \bar{x}), \sigma^2)$ where
 2670 lets say σ^2 is known, the full conditionals are, using “bracket notation”,

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

2671 and

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

2672 We might use our knowledge of probability to identify these mathematically. In particular,
 2673 by Bayes' Rule, $[\beta_0|y, \beta_1] = [y|\beta_0, \beta_1][\beta_0|\beta_1]/[y|\beta_1]$ and similarly for $[\beta_1|y, \beta_0]$. For
 2674 example, if we have priors for $[\beta_0] = \text{Normal}(\mu_{\beta_0}, \sigma_{\beta_0}^2)$ and $[\beta_1] = \text{Normal}(\mu_{\beta_1}, \sigma_{\beta_1}^2)$ then
 2675 some algebra reveals that

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

2676 where $\tau = 1/\sigma^2$ and $\tau_{\beta_0} = 1/\sigma_{\beta_0}^2$ (the inverse of the variance is sometimes called *precision*),
 2677 and $w = \tau n / (\tau n + \tau_{\beta_0})$. We see in this case that the posterior mean is a *precision-weighted*
 2678 sum of the sample mean \bar{y} and the prior mean μ_{β_0} , and the posterior *precision* is the
 2679 sum of the precision of the likelihood and that of the prior. These results are typical of
 2680 many classes of problems. In particular, note that as the prior precision tends to 0, i.e.,
 2681 $\tau_{\beta_0} \rightarrow 0$, then the posterior of β_0 tends to $\text{Normal}(\bar{y}, \sigma^2/n)$. We recognize the variance of
 2682 this distribution as that of the variance of the sampling distribution of \bar{y} and its mean is
 2683 in fact the MLE of β_0 for this model. The conditional posterior of β_1 has a very similar
 2684 form:

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

2685 which might look slightly unfamiliar, but note that if $\tau_{\beta_1} = 0$, then the mean of this
 2686 distribution is the familiar $\hat{\beta}_1$, and the variance is, in fact, the sampling variance of $\hat{\beta}_1$.
 2687 The MCMC algorithm for this model has us simulate in succession, repeatedly, from
 2688 those two distributions. See Gelman et al. (2004) for more examples of Gibbs sampling
 2689 for the normal model, and we also provide another example in Chapt. 17. A conceptual
 2690 representation of the MCMC algorithm for this simple model is therefore:

Algorithm: Gibbs Sampling for linear regression

```

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

```

2692 As we just saw for this simple “normal-normal” model, it is sometimes possible to
 2693 specify the full conditional distributions analytically. In general, when certain so-called
 2694 conjugate prior distributions are used, which have an analytic form that, in a statistical

³We center the independent variable here so that things look more familiar in the result

2695 sense, “matches” the likelihood, then the form of the full conditional distributions is also
 2696 similar to that of the observation model. In this normal-normal case, the normal distribu-
 2697 tion for the mean parameters is the conjugate prior for the normal observation model, and
 2698 thus the full-conditional distributions are also normal. This is convenient because, in such
 2699 cases, we can simulate directly from them using standard methods (or **R** functions). But,
 2700 in practice, we don’t really ever need to know such things because most of the time we
 2701 can get by using a simple algorithm, called the Metropolis-Hastings (henceforth “MH”)
 2702 algorithm, to obtain samples from these full conditional distributions without having to
 2703 recognize them as specific, named, distributions. This gives us enormous freedom in devel-
 2704 oping models and analyzing them without having to resolve them mathematically because
 2705 to implement the MH algorithm we need only identify the full conditional distribution up
 2706 to a constant of proportionality, that being the marginal distribution in the denominator
 2707 (e.g., $[y|\beta_1]$ above).

2708 We will talk about the Metropolis-Hastings algorithm shortly, and we will use it ex-
 2709 tensively in the analysis of SCR models (e.g., Chapt. 17).

2710 3.3.2 Rules for constructing full conditional distributions

2711 The basic strategy for constructing full-conditional distributions for devising MCMC al-
 2712 gorithms can be reduced conceptually to a couple of basic steps summarized as follows:

- 2713 **(step 1)** Identify all stochastic components of the model and collect their probability
 distributions;
- 2715 **(step 2)** Express the full conditional in question as proportional to the product of all
 probability distributions identified in step 1;
- 2717 **(step 3)** Remove the ones that don’t have the focal parameter in them.
- 2718 **(step 4)** Do some algebra on the result in order to identify the resulting probability
 distribution function (pdf) or mass function (pmf).

2720 Of the 4 steps, the last of those is the main step that requires quite a bit of statistical
 2721 experience and intuition because various algebraic tricks can be used to reshape the mess
 2722 into something recognizable – i.e., a standard, named distribution. But step 4 is not
 2723 necessary if we decide instead to use the Metropolis-Hastings algorithm as described below.

2724 In the context of our simple linear regression model that we’ve been working with,
 2725 to characterize $[\beta_0|y, \beta_1]$ we first apply step 1 and identify the model components as:
 2726 $[y|\beta_0, \beta_1]$, with prior distributions $[\beta_0]$ and $[\beta_1]$. Step 2 has us write $[\beta_0|y, \beta_1] \propto [y|\beta_0, \beta_1][\beta_0][\beta_1]$.
 2727 Step 3: We note that $[\beta_1]$ is not a function of β_0 and therefore we remove it to obtain
 2728 $[\beta_0|y, \beta_1] \propto [y|\beta_0, \beta_1][\beta_0]$. Similarly, applying step 2 and 3 for β_1 we obtain $[\beta_1|y, \beta_0] \propto$
 2729 $[y|\beta_0, \beta_1][\beta_1]$. We apply step 4 and manipulate these algebraically to arrive at the re-
 2730 sult (which we provided in Eqs. 3.3.1 and 3.3.2) or, alternatively, we can sample them
 2731 indirectly using the Metropolis-Hastings algorithm, which we discuss now.

2732 3.3.3 Metropolis-Hastings algorithm

2733 The Metropolis-Hastings (MH) algorithm is a completely generic method for sampling
 2734 from any distribution, say $[\theta]$. In our applications, $[\theta]$ will typically be the full conditional
 2735 distribution of θ . While we sometimes use Gibbs sampling, we seldom use “pure” Gibbs

2736 sampling because full conditionals do not always take the form of known distributions we
 2737 can sample from directly. In such cases, we use MH to sample from the full conditional
 2738 distributions. When the MH algorithm is used to sample from full conditional distributions
 2739 of a Gibbs sampler the resulting hybrid algorithm is called *Metropolis-within-Gibbs*. In
 2740 Sec. 3.6.3 we will construct such an algorithm for a simple class of models. We discuss
 2741 both the Gibbs and the MH algorithm, as well as their hybrid in more depth in Chapt.
 2742 17.

2743 The MH algorithm generates candidate values for the parameter(s) we want to estimate
 2744 from some proposal or candidate-generating distribution that may be conditional on the
 2745 current value of the parameter, denoted by $h(\theta^*|\theta^{t-1})$. Here, θ^* is the *candidate* or
 2746 proposed value and θ^{t-1} is the value of θ at the previous time step, i.e., at iteration $t - 1$
 2747 of the MCMC algorithm. The proposed value is accepted with probability

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

2748 which is called the MH acceptance probability. This ratio can sometimes be > 1 in which
 2749 case we set it equal to 1. It is useful to note that $h()$ can be any probability distribution.

2750 In the context of using the MH algorithm to do MCMC (in which case the target
 2751 distribution is a full-conditional or posterior distribution), an important fact is, no matter
 2752 the choice of $h()$, we can compute the MH acceptance probability directly because the
 2753 marginal distribution of y cancels from both the numerator and denominator of r . This
 2754 is the magic of the MH algorithm.

3.4 BAYESIAN ANALYSIS USING THE BUGS LANGUAGE

2755 We won't be too concerned with devising our own MCMC algorithms for every analysis,
 2756 although we will do that a few times for fun. More often, we will rely on the freely available
 2757 software package **WinBUGS** or **JAGS** for doing this. We will always execute these
 2758 **BUGS** engines from within **R** using the **R2WinBUGS** (Sturtz et al., 2005) or, for **JAGS**,
 2759 the **R2jags** (Su and Yajima, 2011) or **rjags** (Plummer, 2009) packages. **WinBUGS** and
 2760 **JAGS** are MCMC black boxes that take a pseudo-code description (i.e., written in the
 2761 **BUGS** language) of all of the relevant stochastic and deterministic elements of a model
 2762 and generate an MCMC algorithm for that model. But you never get to see the algorithm.
 2763 Instead, **WinBUGS/JAGS** will run the algorithm and return the Markov chain output
 2764 - the posterior samples of model parameters.

2765 The great thing about using the **BUGS** language is that it forces you to become
 2766 intimate with your statistical model - you have to write each element of the model down,
 2767 admit (explicitly) all of the various assumptions, understand what the actual probability
 2768 assumptions are and how data relate to latent variables and data and latent variables
 2769 relate to parameters, and how parameters relate to one another.

2770 While we normally use **WinBUGS**, we note that **OpenBUGS** is the current active
 2771 development tree of the **BUGS** project. See Kéry (2010) and Kéry and Schaub (2012,
 2772 especially Appendix 1) for more on practical analysis in **WinBUGS**. Those books should
 2773 be consulted for a more comprehensive introduction to using **WinBUGS**. Recently we
 2774 have migrated many of our analyses to **JAGS** (Plummer, 2009), which we adopt later in

2775 the book. You can refer to Hobbs (2011) for an ecological introduction to **JAGS**. Next,
 2776 we provide an example of a Bayesian analysis using **WinBUGS**.

2777 **3.4.1 Linear regression in WinBUGS**

2778 We provide a brief introductory example of a normal regression model using a small
 2779 simulated data set. The following commands are executed from within your **R** workspace.
 2780 First, simulate a covariate x and observations y having prescribed intercept, slope and
 2781 variance:

```
2782 > x <- rnorm(10)
2783 > mu <- -3.2 + 1.5*x
2784 > y <- rnorm(10, mu, sd=4)
```

2785 The **BUGS** model specification for a normal regression model is written within **R** as
 2786 a character string input to the command **cat()** and then dumped to a text file named
 2787 **normal.txt**:

```
2788 > cat("
2789   model{
2790     for (i in 1:10){
2791       y[i] ~ dnorm(mu[i],tau)      # the likelihood
2792       mu[i] <- beta0 + beta1*x[i]  # the linear predictor
2793     }
2794     beta0 ~ dnorm(0,.01)          # prior distributions
2795     beta1 ~ dnorm(0,.01)
2796     sigma ~ dunif(0,100)
2797     tau <- 1/(sigma*sigma)       # tau is the precision
2798   }                                # and a derived parameter
2799 ",file="normal.txt")
```

2800 Alternatively, you can write the model specifications directly within a text file and save it
 2801 in your current working directory, but we do not usually take that approach in this book.

2802 The **BUGS** dialects⁴ parameterize the normal distribution in terms of the mean and
 2803 inverse-variance, called the precision. Thus, **dnorm(0,.01)** implies a variance of 100.
 2804 We typically use diffuse normal priors for mean parameters, β_0 and β_1 in this case, but
 2805 sometimes we might use uniform priors with suitable bounds $-B$ and $+B$. Also, we
 2806 typically use a Uniform($0, B$) prior on standard deviation parameters (Gelman, 2006).
 2807 But sometimes we might use a gamma prior on the precision parameter τ . In a **BUGS**
 2808 model file, every variable referenced in the model description has to be either data, which
 2809 will be input (see below), a random variable which must have a probability distribution
 2810 associated with it using the tilde character “~” (a.k.a. “twiddle”) or it has to be a derived
 2811 parameter connected to variables and data using an assignment arrow: “<-”.

2812 To fit the model, we need to describe various data objects to **WinBUGS**. In particular,
 2813 we create an **R** list object called **data** which are the data objects identified in the **BUGS**
 2814 model file. In the example, the data consist of two objects which exist as y and x in the

⁴We use this to mean **WinBUGS**, **OpenBUGS** and **JAGS**

2815 **R** workspace and also in the **WinBUGS** model definition. We also create an **R** function
 2816 that produces a list of starting values, **inits**, that get sent to **WinBUGS**. In general,
 2817 starting values are optional. We recommend to always provide reasonable starting values
 2818 where possible, both for structural parameters and also random effects⁵. Finally, we
 2819 identify the names of the parameters (labeled correspondingly in the **WinBUGS** model
 2820 specification) that we want **WinBUGS** to save the MCMC output for. In this example,
 2821 we will “monitor” the parameters β_0 , β_1 , σ and τ . **WinBUGS** is executed using the
 2822 **R** command **bugs()**. We set the option **debug=TRUE** if we want the **WinBUGS** GUI to
 2823 stay open (useful for analyzing MCMC output and looking at the **WinBUGS** error log).
 2824 Also, we set **working.dir=getwd()** so that **WinBUGS** output files and the log file are
 2825 saved in the current **R** working directory (note that sometimes you will need to specify the
 2826 place where you installed **WinBUGS** within the **bugs()** call, using the **bugs.directory**
 2827 argument). All of these activities together look like this:

```
2828 > library(R2WinBUGS)      # "load" the R2WinBUGS package
2829 > data <- list( y=y, x=x)
2830 > inits <- function()
2831 > list ( beta1=rnorm(1),beta0=rnorm(1),sigma=runif(1,0,2) )
2832 > parameters <- c("beta0","beta1","sigma","tau")
2833 > out <- bugs(data, inits, parameters, "normal.txt", n.thin=1, n.chains=2,
2834   n.burnin=2000, n.iter=6000, debug=TRUE,working.dir=getwd())
```

2835 Note that the previously created objects defining data, initial values and parameters to
 2836 monitor are passed to the function **bugs()**. In addition, various other things are declared:
 2837 The number of parallel Markov chains (**n.chains**), the thinning rate (**n.thin**), the number
 2838 of burn-in iterations (**n.burnin**) and the total number of iterations (**n.iter**). To develop
 2839 a detailed understanding of the various parameters and settings used for MCMC, consult
 2840 a basic reference such as Kéry (2010). We also come back to these issues in the following
 2841 section (3.5) and in Chapt. 17. A common question is “how should my data be formatted?”
 2842 That depends on how you describe the model in the **BUGS** language, and how your data
 2843 are input into **R**. There is no unique way to describe any particular model and so you have
 2844 some flexibility. We talk about data format further in the context of capture-recapture
 2845 models and SCR models in Chapt. 5 and elsewhere.

2846 You should execute all of the commands given above and then close the **WinBUGS**
 2847 GUI, and the data will be read back into **R** (or specify **debug=FALSE** in the **bugs()** call).
 2848 We don’t want to give instructions on how to navigate and use the GUI – but you can
 2849 fire up **WinBUGS** and read the help files, or see Chapt. 4 from Kéry (2010) for a brief
 2850 introduction. The **print** command applied to the object **out** prints some basic summary
 2851 output (this is slightly edited):

```
2852 > print(out,digits=2)
2853 Inference for Bugs model at "normal.txt", fit using WinBUGS,
2854 2 chains, each with 6000 iterations (first 2000 discarded)
```

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

```

2855 n.sims = 8000 iterations saved
2856      mean   sd 2.5% 25% 50% 75% 97.5% Rhat n.eff
2857 beta0    -6.62 1.64 -9.77 -7.63 -6.64 -5.63 -3.29     1  4200
2858 beta1     0.81 1.20 -1.63  0.09  0.80  1.54  3.24     1  5100
2859 sigma     4.99 1.56  2.93  3.92  4.66  5.70  8.85     1  8000
2860 tau       0.05 0.03  0.01  0.03  0.05  0.07  0.12     1  8000
2861 deviance 58.72 3.21 55.06 56.35 57.85 60.26 67.15     1  6200
2862
2863 For each parameter, n.eff is a crude measure of effective sample size,
2864 and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
2865
2866 DIC info (using the rule, pD = Dbar-Dhat)
2867 pD = 2.5 and DIC = 61.3

```

2868 In the **WinBUGS** output you see a column called “Rhat”, as well as one called
2869 “n.eff”. These are convergence diagnostics (the \hat{R} or Brooks-Gelman-Rubin statistic
2870 and the effective sample size) and we will discuss those in the following section, 3.5.2.
2871 DIC is the deviance information criterion (Spiegelhalter et al. (2002), see section 3.9)
2872 which some people use in a manner similar to AIC although it is recognized to have some
2873 problems in hierarchical models (Millar, 2009). We consider use of DIC in the context of
2874 SCR models in Chapt. 8.

3.5 PRACTICAL BAYESIAN ANALYSIS AND MCMC

2875 The mere execution of a Bayesian analysis using the **BUGS** language, as demonstrated
2876 with the linear regression example, is fairly straight forward. There are, however, a number
2877 of really important practical issues to be considered in any Bayesian analysis and we cover
2878 some of these briefly here before we move on to implementing slightly more complex
2879 GL(M)Ms in a Bayesian framework.

2880 3.5.1 Choice of prior distributions

2881 Bayesian analysis requires that we choose prior distributions for all of the structural pa-
2882 rameters of the model (we use the term structural parameter to mean all parameters that
2883 aren’t customary thought of as latent variables). We will strive to use priors that are
2884 meant to express little or no prior information - default or customary “non-informative”
2885 or diffuse priors. This will be $\text{Uniform}(a, b)$ priors for parameters that have a natural
2886 bounded support and, for parameters that live on the real line we use either (1) diffuse
2887 normal priors, as we did in the linear regression example above; (2) improper uniform
2888 priors which have unbounded support, e.g., $[\theta] \propto 1$, or (3) sometimes even a bounded
2889 $\text{Uniform}(a, b)$ prior, if that greatly improves the performance of **WinBUGS** or other
2890 software doing the MCMC for us. In **WinBUGS** a prior with low precision, τ , where
2891 $\tau = 1/\sigma^2$, such as $\text{Normal}(0, .01)$ will typically be used. Of course $\tau = 0.01$ ($\sigma^2 = 100$)
2892 might be very informative for a regression parameter depending on its magnitude and
2893 scaling of x . Therefore, we recommend that predictor variables (covariates) *always* be
2894 standardized to have mean 0 and variance 1.

2895 **Lack of invariance of priors to transformation.** Clearly there are a lot of choices
 2896 for ostensibly non-informative priors, and the degree of non-informativeness depends on
 2897 the parameterization. For example, a natural non-informative prior for the intercept of a
 2898 logistic regression

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

2899 would be a very diffuse normal prior, $[\beta_0] = \text{Normal}(0, \text{Large})$ or even $\beta_0 \sim \text{Uniform}(-\text{Large}, \text{Large})$.
 2900 However, we might also use a prior on the parameter $p_0 = \text{logit}^{-1}(\beta_0)$, which is $\Pr(y=1)$
 2901 for the value $x=0$. Since p_0 is a probability a natural choice is $p_0 \sim \text{Uniform}(0, 1)$. These
 2902 priors are very different in their implications. For example, if we choose the normal prior
 2903 for β_0 with variance $\text{Large} = 5^2$ and look at the implied prior for p_0 we have the result
 shown in Fig. 3.2 which looks nothing like a $\text{Uniform}(0, 1)$ prior. These two priors can

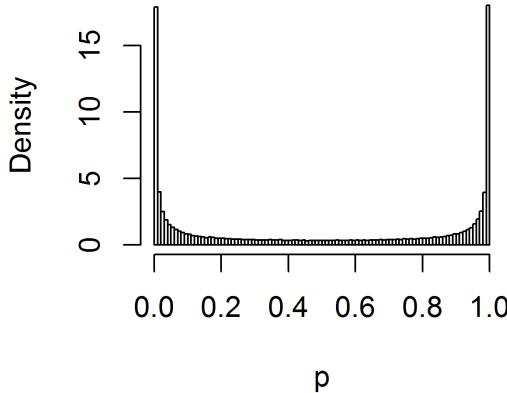


Figure 3.2. Implied prior for $p_0 = \exp(\beta_0)/(1 + \exp(\beta_0))$ if $\beta_0 \sim \text{Normal}(0, 5^2)$.

2904
 2905 affect results (see Sec. 4.4.2 for an illustration of this for a real data set), yet they are
 2906 both sensible non-informative priors. Despite this, it is often the case that priors will have
 2907 little or no impact on the results. Choice of priors and parameterization is very much
 2908 problem-specific and often largely subjective. Moreover, it also affects the behavior of
 2909 MCMC algorithms and therefore the analyst needs to pay some attention to this issue
 2910 and possibly try different things out. Most standard Bayesian analysis books address
 2911 issues related to specification and effect of prior distribution choice in some depth. Some
 2912 good references include Kass and Wasserman (1996), Gelman (2006) and Link and Barker
 2913 (2010).

2914 **3.5.2 Convergence and so-forth**

2915 Once we have carried out an analysis by MCMC, there are many other practical issues
 2916 that we have to confront. One characteristic of MCMC sampling is that Markov chains
 2917 take some time to converge to their stationary distribution - in our case the posterior
 2918 distribution for some parameter given data, $[\theta|y]$. Only when the Markov chain has
 2919 reached its stationary distribution, the generated samples can be used to characterize the
 2920 posterior distribution. Thus, one of the most important issues we need to address is “have
 2921 the chains converged?” Since we do not know what the stationary posterior distribution
 2922 of our Markov chain should look like (this is the whole point of doing an MCMC analysis),
 2923 we effectively have no means to assess whether or not it has truly converged to this desired
 2924 distribution. Most MCMC algorithms only guarantee that, eventually, the samples being
 2925 generated will be from the target posterior distribution, but no-one can tell us how long
 2926 this will take. Also, you only know the part of your posterior distribution that the Markov
 2927 chain has explored so far – for all you know the chain could be stuck in a local maximum,
 2928 while other maxima remain completely undiscovered. Acknowledging that there is truly
 2929 nothing we can do to ever prove convergence of our MCMC chains, there are several things
 2930 we can do to increase the degree of confidence we have about the convergence of our chains.
 2931 Some problems are easily detected using simple plots, such as a time-series plot, where
 2932 parameter values of each MCMC iteration are plotted against the number of iterations.
 2933 Fig. 3.3 shows the time series plots for the three parameters – β_0 , β_1 and σ – from our
 2934 linear regression example, taken from the **WinBUGS** GUI before closing it to return to
 2935 **R**.

2936 Typically a period of transience is observed in the early part of the MCMC algorithm,
 2937 and this is usually discarded as the “burn-in” period. In our linear regression example,
 2938 within the `bugs()` call we set the burn-in period as 2000 iterations so these are auto-
 2939 matically removed by **WinBUGS** and are not part of the output (but Fig. 3.6 shows a
 2940 time-series plot that starts at iteration 0 with a clearly visible burn-in period). The quick
 2941 diagnostic to whether convergence has been achieved is that your Markov chains look
 2942 “grassy” – this seems a reasonable statement for the plots in Fig. 3.3. Another way to
 2943 check convergence is to update the parameters some more and see if the posterior changes.
 2944 If the chains have converged to the posterior, the posterior mean, confidence intervals, and
 2945 other summaries should be relatively static as we continue to run the algorithm. Yet an-
 2946 other option, and one generally implemented in **WinBUGS**, is to run several Markov
 2947 chains and to start them off at different initial values that are over-dispersed relative to
 2948 the posterior distribution. Such initial values help to explore different areas of the param-
 2949 eter space simultaneously; if, after a while, all chains oscillate around the same average
 2950 value, chances are good that they indeed converged to the posterior distribution. Gelman
 2951 and Rubin came up with the so-called “R-hat” statistic (\hat{R}) or Brooks-Gelman-Rubin
 2952 statistic that essentially compares within-chain and between-chain variance to check for
 2953 convergence of multiple chains (Gelman et al., 1996). The R-hat statistic should be close
 2954 to 1 if the Markov chains have converged and sufficient posterior samples have been ob-
 2955 tained. For the linear regression example, we ran two parallel chains (also specified in the
 2956 `bugs()` call) and **WinBUGS** returns the \hat{R} statistic for us as part of the summary model
 2957 output. If you look back to Sec. 3.4.1 you see that $\hat{R} = 1$ for all parameters of the linear
 2958 model. In practice, $\hat{R} \leq 1.2$ may be good enough for some problems. For some models you
 2959 can’t actually realize a low \hat{R} . E.g., if the posterior is a discrete mixture of distributions

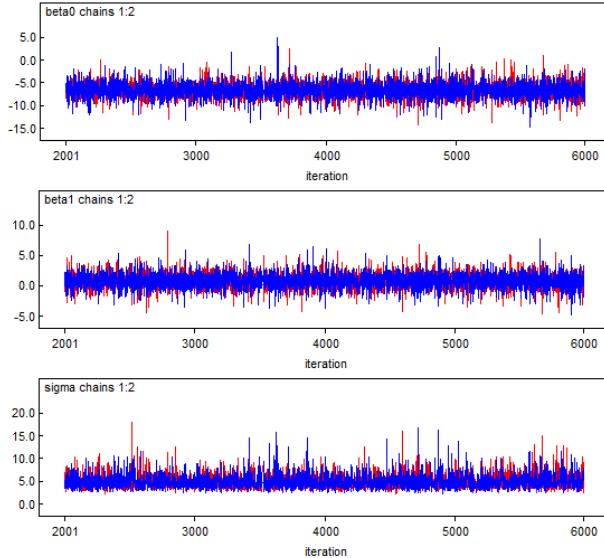


Figure 3.3. Time-series plots for parameters from a linear regression run in **WinBUGS** using two parallel Markov chains.

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

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

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

2980 (MacEachern and Berliner, 1994; Link and Eaton, 2011). Practical considerations might
 2981 necessitate thinning, even though it is statistically inefficient. For example, in models
 2982 with many parameters or other unknowns being tabulated, the output files might be
 2983 enormous and unwieldy to work with. In such cases, thinning is perfectly reasonable. In
 2984 many cases, how well the Markov chains mix is strongly influenced by parameterization,
 2985 standardization of covariates, and the prior distributions being used. Some things work
 2986 better than others, and the investigator should experiment with different settings and
 2987 remain calm when things don't work out perfectly.

2988 **Is the posterior sample large enough?** The subsequent samples generated from
 2989 a Markov chain are not *independent* samples from the posterior distribution, due to the
 2990 correlation among samples introduced by the Markov process⁶ and the sample size has
 2991 to be adjusted to account for the autocorrelation in subsequent samples (see Chapt. 8 in
 2992 Robert and Casella (2010) for more details). This adjusted sample size is referred to as the
 2993 effective sample size. Checking the degree of autocorrelation in your Markov chains and
 2994 estimating the effective sample size your chain has generated should be part of evaluating
 2995 your model output. **WinBUGS** will automatically return the effective sample size for
 2996 all monitored parameters, as we saw in our linear regression example (the “n.eff” column
 2997 of the summary output). If you find that your supposedly long Markov chain has only
 2998 generated a very short effective sample, you should consider a longer run. What exactly
 2999 constitutes a reasonable effective sample size is hard to say. A more palpable measure
 3000 of whether you've run your chain for enough iterations is the time-series or Monte Carlo
 3001 error - the “noise” introduced into your samples by the stochastic MCMC process. The
 3002 MC error is printed by default in summaries produced in the **WinBUGS** GUI, which
 3003 can be reproduced in **R** using `bugs.log('log.txt')$stats` (note that “log.txt” refers
 3004 to a model log file that **WinBUGS** automatically creates in the working directory; it is
 3005 overwritten with every new model you run unless you save it under a different name).

```
3006 > bugs.log('log.txt')$stats
3007 $stats
3008      mean      sd   Mcerror    2.5%   median   97.5% start sample
3009 beta0    -6.64700 1.60300 0.0179400 -9.7140 -6.70800 -3.2730 2001 8000
3010 beta1     0.82100 1.19000 0.0116800 -1.4900  0.82560  3.1800 2001 8000
3011 deviance  58.66000 3.08800 0.0506800 55.0700 57.93000 66.8400 2001 8000
3012 sigma      4.96800 1.52300 0.0248300  2.9350  4.68100  8.7410 2001 8000
3013 tau       0.05074 0.02677 0.0003651  0.0131  0.04564  0.1162 2001 8000
```

3014 When using **JAGS** the `summary` command will automatically produce the MC error
 3015 (which is called “Time-series SE” in **JAGS**). You want the MC error to be smallish relative
 3016 to the magnitude of the parameter and what smallish means will depend on the purpose
 3017 of the analysis. For a preliminary analysis you might settle for a few percent whereas
 3018 for a final analysis then certainly less than 1% is called for. You can run your MCMC
 3019 algorithm as long as it takes to achieve that. A consequence of the MC error is that even
 3020 for the exact same model, results will usually be slightly different. Thus, as a good rule of
 3021 thumb, you should avoid reporting MCMC results to more than 2 or 3 significant digits!

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

3022 **3.5.3 Bayesian confidence intervals**

3023 The 95% Bayesian confidence interval based on percentiles of the posterior is not a unique
3024 interval - there are many of them. The so-called “highest posterior density” (HPD) inter-
3025 val is an alternative, defined as the narrowest interval that contains *at least* 95% of the
3026 posterior mass. As a result (of the *at least* clause), for discrete parameters, the 95% HPD
3027 is not often exactly 95% but usually slightly more conservative than nominal.

3028 **3.5.4 Estimating functions of parameters**

3029 A benefit of analysis by MCMC is that we can seamlessly estimate functions of parameters
3030 by simply tabulating the desired function of the simulated posterior draws. For example,
3031 if θ is the parameter of interest and let $\theta^{(i)}$ for $i = 1, 2, \dots, M$ be the posterior samples
3032 of θ . Let $\eta = \exp(\theta)$, then a posterior sample of η can be obtained simply by computing
3033 $\exp(\theta^{(i)})$ for $i = 1, 2, \dots, M$. Almost all SCR models in this book involve at least 1 derived
3034 parameter. For example, density D is a derived parameter, being a function of population
3035 size N and the area A of the underlying state-space of the point process (see Chapt. 5).

3036 **Example: Finding the optimum value of a covariate.** As another example of
3037 estimating functions of model parameters, suppose that the normal regression model from
3038 Sec. 3.4.1 had a quadratic response function of the form

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

3039 Then the optimum value of x , i.e., that corresponding to the optimal expected response,
3040 can be found by setting the derivative of this function to 0 and solving for x . We find that

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

3041 yields that $x_{opt} = -\beta_1/(2 * \beta_2)$. We can just take our posterior draws for β_1 and β_2
3042 and obtain a posterior sample of x_{opt} by this simple calculation applied to the posterior
3043 output. As an exercise, take the normal model above and simulate a quadratic response
3044 and then describe the posterior distribution of x_{opt} .

3.6 POISSON GLMS

3045 The Poisson GLM (also known as “Poisson regression”) is probably the most relevant
3046 and important class of models in all of ecology. The basic model assumes observations
3047 $y_i; i = 1, 2, \dots, n$ follow a Poisson distribution with mean λ which we write

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

3048 Commonly y_i is a count of animals or plants at some point in space (“site”) i , and λ
3049 might vary over sites as well. For example, i might index point count locations in a
3050 forest, survey route centers, or sample quadrats, or similar, and we are interested in how
3051 λ depends on site characteristics such as habitat. If covariates are available it is typical to
3052 model them as linear effects on the log mean. If x_i is some measured covariate associated
3053 with observation i , then,

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

3054 While we only specify the mean of the Poisson model directly, the Poisson model (and
 3055 all GLMs) has a “built-in” variance which is directly related to the mean. In this case,
 3056 $\text{Var}(y) = \mathbb{E}(y) = \lambda$. Thus the model accommodates a linear increase in variance with the
 3057 mean.

3058 **3.6.1 Example: Breeding Bird Survey data**

3059 As an example we consider a classical situation in ecology where counts of an organism
 3060 are made at a collection of spatial locations. In this particular example, we have
 3061 mourning dove (*Zenaida macroura*) counts made along North American Breeding Bird
 3062 Survey (BBS) routes in Pennsylvania, USA. A route consists of 50 stops separated by
 3063 0.5 miles. For the purposes here we are defining y_i = route total count and the sample
 3064 location will be marked by the center point of the BBS route. The survey is run an-
 3065 nually and the data set we analyze is 1966-1998. BBS data can be obtained online at
 3066 <http://www.pwrc.usgs.gov/bbs/>, but the particular chunk of data we will be using here
 3067 is also included in the **scrbook** package (**data(bbsdata)**). We will make use of the whole
 3068 data set shortly but for now we’re going to focus on a specific year of counts (1990) for
 3069 the sake of building a simple model. In 1990 there were 77 active routes; this data set
 3070 contains rows which index the unique route, column 1 is the route ID, columns 2-3 are
 3071 the route coordinates (longitude/latitude), column 4 is a habitat covariate “forest cover”
 3072 (standardized, see below) and the remaining columns are the yearly counts. Years for
 3073 which a survey was not conducted on a route are coded as “NA” in the data matrix. We
 3074 imagine that this will be a typical format for many ecological studies, perhaps with more
 3075 columns representing covariates. To read in the data and display the first few elements of
 3076 the data frame containing the counts, do this:

```
3077 > data(bbsdata)           #  loads data frame 'bbs'  

3078 > bbsdata$counts[1:2,1:6]  

3079  

3080      X     lon     lat   habitat X66 X67  

3081 1 72002 -80.445 41.501 -0.3871372 NA  24  

3082 2 72003 -80.347 41.214 -1.0171629 NA  NA
```

3083 It is useful to display the spatial pattern in the observed counts. For that we use a
 3084 spatial dot plot – where we plot the coordinates of the observations and mark the color
 3085 of the plotting symbol based on the magnitude of the count. We have a special plotting
 3086 function for that which is called **spatial.plot()** and it is available with the supplemental
 3087 R package **scrbook**. Actually, what we want to do here is plot the log-counts (+1 of
 3088 course) which (Fig. 3.4) display a notable pattern that could be related to something.
 3089 The R commands for obtaining this figure are:

```
3090 > library(scrbook)  

3091 > data(bbsdata)  

3092 > library(maps)  

3093  

3094 > y <- bbsdata$counts[, "X90"] # Pick year 1990  

3095 > notna <- !is.na(y)
```

```

3096 > y <- y[notna]
3097 > locs <- bbsdata$counts[notna,c("lon","lat")]
3098 > sz <- y/max(y)
3099
3100 > par(mar=c(3,3,3,6))
3101 > plot(locs,pch=" ",axes=FALSE,xlim=range(locs[,1])+c(-.3,+.3),
3102       ylim=c(range(locs[,2]) + c(-.6,.6)), xlab=" ",ylab=" ")
3103 > map('state', regions='pennsylvania', add=TRUE, lwd=2)
3104 > spatial.plot(bbsdata$counts[notna,2:3], y, cx=1+sz*6, add=TRUE)

```

3105 We can ponder the potential effects that might lead to dove counts being high - corn
 3106 fields, telephone wires, barn roofs along with misidentification of pigeons, these could all
 3107 correlate reasonably well with the observed count of mourning doves. Unfortunately we
 3108 don't have any of that information. However, we do have a measure of forest cover (pro-

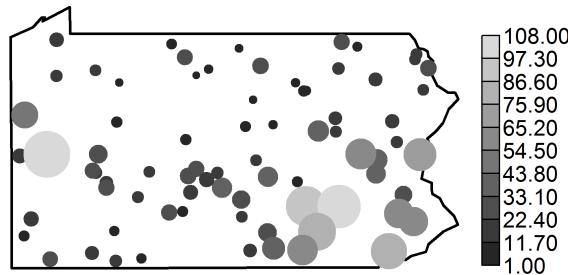


Figure 3.4. Mourning dove counts along North American Breeding Bird Survey routes in Pennsylvania (year = 1990). Plot symbol shading and circle size is proportional to raw count.

3108 vided in the data frame `bbsdata$habitat`) which can be plotted using the `spatial.plot`
 3109 function with the following **R** commands
 3110

```

3111 > habdata <- bbsdata$habitat
3112 > map('state',regions="penn",lwd=2)
3113 > I <- matrix(NA, nrow=30, ncol=40)
3114 > I <- matrix(habdata[,"dfor"], ncol=40, byrow=FALSE)
3115 > ux <- unique(habdata[,2])

```

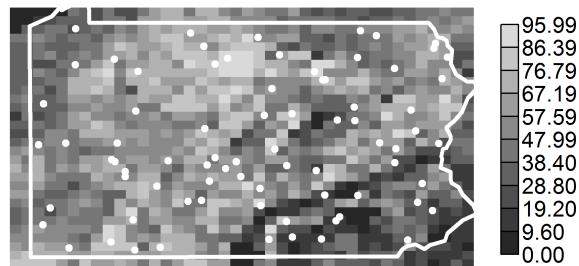


Figure 3.5. Forest cover (percent deciduous) in Pennsylvania. BBS route locations are shown by white dots.

```

3116 > uy <- sort(unique(habdata[,3]))
3117
3118 > par(mar=c(3,3,3,6))
3119 > plot(locs,pch=" ", axes=FALSE, xlim=range(locs[,1])+c(-.3,+.3),
3120   ylim=c(range(locs[,2]) + c(-.6,.6)), xlab=" ",ylab=" ")
3121 > image(ux,uy,rot(I), add=TRUE, col=gray(seq(3,17,,10)/20) )
3122 > map('state', regions='pennsylvania', add=TRUE, lwd=3, col="white")
3123 > image.scale(I, col=gray(seq(3,17,,10)/20) )
3124 > points(locs,pch=20, col="white")

```

3125 The result appears in Fig. 3.5. We see a prominent pattern that indicates high forest
 3126 coverage in the central part of the state and low forest cover in the SE. Inspecting the
 3127 previous figure of the raw counts suggests a relationship between counts and forest cover
 3128 which is perhaps not surprising.

3129 3.6.2 Doing it in WinBUGS

3130 Here we demonstrate how to fit a Poisson GLM in **WinBUGS** using the covariate $x_i =$
 3131 forest cover along BBS route i . It is advisable that x_i be standardized in most cases as
 3132 this will improve mixing of the Markov chains. We have pre-standardized the forest cover
 3133 covariate for the BBS route locations, and so we don't have to worry about that here. To
 3134 read the BBS data into **R** and get things set up for **WinBUGS** we issue the following
 3135 commands:

```

3136 > library(scrbook)
3137 > data(bbsdata)
3138
3139 > y <- bbsdata$counts[, "X90"] # Pick year 1990
3140 > notna <- !is.na(y)
3141 > y <- y[notna]
3142
3143 ## Forest cover already standardized here:
3144 > habitat <- bbsdata$counts[notna, "habitat"]
3145 > M <- length(y)
3146
3147 > library(R2WinBUGS) # Load R2WinBUGS
3148 > data <- list (y=y, M=M, habitat=habitat) # Bundle data for WinBUGS

```

3149 Now we write out the Poisson model specification in **WinBUGS** pseudo-code, provide
 3150 initial values, identify parameters to be monitored and then execute **WinBUGS**:

```

3151 > cat("
3152 model{
3153   for (i in 1:M){
3154     y[i] ~ dpois(lam[i])
3155     log(lam[i]) <- beta0+beta1*habitat[i]
3156   }
3157   beta0 ~ dunif(-5,5)
3158   beta1 ~ dunif(-5,5)
3159 }
3160 ",file="PoissonGLM.txt")

3161 > inits <- function() list ( beta0=rnorm(1),beta1=rnorm(1) )
3162 > parameters <- c("beta0","beta1")
3163 > out <- bugs(data, inits, parameters, "PoissonGLM.txt", n.thin=2,n.chains=2,
3164   n.burnin=2000,n.iter=6000,debug=TRUE,working.dir=getwd())

```

3165 The **WinBUGS** output can be viewed in **R** using the `print` command:

```

3166 print(out,digits=2)
3167 Inference for Bugs model at "PoissonGLM.txt", fit using WinBUGS,
3168 2 chains, each with 6000 iterations (first 2000 discarded), n.thin = 2
3169 n.sims = 4000 iterations saved
3170      mean    sd    2.5%    25%    50%    75%   97.5% Rhat n.eff
3171 beta0     3.15  0.02    3.10    3.13    3.15    3.17    3.20     1  4000
3172 beta1    -0.50  0.02   -0.54   -0.51   -0.50   -0.48   -0.46     1  4000
3173 deviance 1116.56 1.95 1115.00 1115.00 1116.00 1117.00 1122.00     1  4000

```

3174 3.6.3 Constructing your own MCMC algorithm

3175 At this point it might be helpful to suffer through an example building a custom MCMC
 3176 algorithm. Here, we develop an MCMC algorithm for the Poisson regression model, using

3177 a Metropolis-within-Gibbs sampling framework. Building MCMC algorithms is covered in
 3178 more detail in Chapt. 17 where you can also find step-by-step instructions for Metropolis-
 3179 within-Gibbs samplers, should the following section move through all this material too
 3180 quickly.

3181 We will assume that the two parameters, β_0 and β_1 , have diffuse normal priors, say
 3182 $[\beta_0] = \text{Normal}(0, 100)$ and $[\beta_1] = \text{Normal}(0, 100)$ where each has *standard deviation* 100
 3183 (recall that **WinBUGS** parameterizes the normal in terms of $1/\sigma^2$). We need to assem-
 3184 ble the relevant elements of the model which are these two prior distributions and the
 3185 likelihood $[\mathbf{y}|\beta_0, \beta_1] = \prod_i [y_i|\beta_0, \beta_1]$ which is, mathematically, the product of the Poisson
 3186 pmf evaluated at each y_i , given particular values of β_0 and β_1 . Next, we need to identify
 3187 the full conditionals $[\beta_0|\beta_1, \mathbf{y}]$ and $[\beta_1|\beta_0, \mathbf{y}]$. We use the all-purpose rule for constructing
 3188 full conditionals (section 3.3.2) to discover that:

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

3189 Mathematically, the full conditional is of the form

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

3190 which you can program as an **R** function with arguments β_0 , β_1 and \mathbf{y} without difficulty.

3191 The full-conditional for β_1 is:

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

3192 which has a similar mathematical representation except the prior is expressed in terms
 3193 of β_1 instead of β_0 . Remember, we could replace the “ \propto ” with “=” if we put $[y|\beta_1]$ or
 3194 $[y|\beta_0]$ in the denominator. But, in general, $[y|\beta_0]$ or $[y|\beta_1]$ will be quite a pain to compute
 3195 and, more importantly, it is a constant as far as the operative parameters (β_0 or β_1 ,
 3196 respectively) are concerned. Therefore, the MH acceptance probability will be the ratio
 3197 of the full-conditional evaluated at a candidate draw to that evaluated at the current
 3198 value, and so the denominator required to change \propto to $=$ winds up canceling from the
 3199 MH acceptance probability.

3200 Here we will use the so-called random walk candidate generator, which is a Normal
 3201 proposal distribution, so that, for example, $\beta_0^* \sim \text{Normal}(\beta_0^t, \delta)$ where δ is the standard-
 3202 deviation of the proposal distribution, which is just a tuning parameter that is set by
 3203 the user and adjusted to achieve efficient mixing of chains (see Sec. 17.3.2). We remark
 3204 also that calculations are often done on the log-scale to preserve numerical integrity of
 3205 things when quantities evaluate to small or large numbers, so keep in mind, for example,
 3206 $a * b = \exp(\log(a) + \log(b))$ for two positive numbers a and b . The “Metropolis within
 3207 Gibbs” algorithm for a Poisson regression turns out to be remarkably simple and is given
 3208 in Panel 3.1. It is also part of the **scrbook** package and you can run 1000 iterations of it
 3209 by calling `PoisGLMBBS(y=y, habitat=habitat, niter=1000)` (note that y = point count
 3210 data and `habitat` = forest cover have to be defined in your **R** workspace as shown in the
 3211 previous analysis of these data).

```

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

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

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

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

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

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

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

```

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

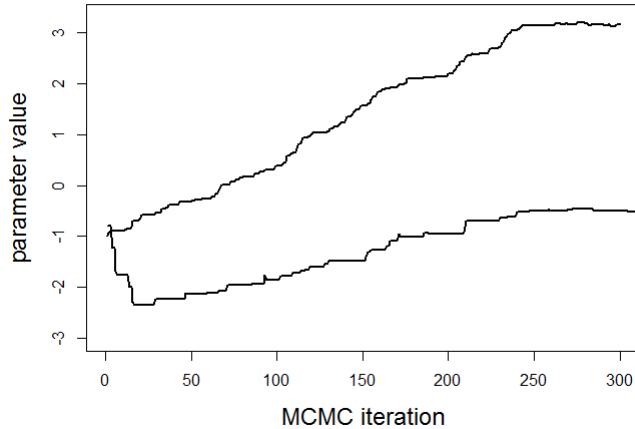


Figure 3.6. First 300 MCMC iterations for the Poisson GLM model parameters β_0 (top) and β_1 (bottom) using a Metropolis-Hastings tuning parameter of $\delta = 0.05$.

3212 The first 300 iterations of the MCMC history of each parameter are shown in Fig. 3.6.
 3213 These chains are not very appealing but a couple of things are evident: We see that the
 3214 burn-in takes about 250 iterations and that after that chains seem to mix reasonably well,
 3215 although this is not so clear given the scale of the y-axis, which we have chosen to get
 3216 both variables on the same graph. We generated 10,000 posterior samples, discarding the
 3217 first 500 as burn-in, and the result is shown in Fig. 3.7, this time on separate panels for
 3218 each parameter. The “grassy” look of the MCMC history is diagnostic of Markov chains
 3219 that are well-mixing and we would generally be very satisfied with results that look like
 3220 this.

3221 Note that we used a specific set of starting values for these simulations. It should be
 3222 clear that starting values closer to the mass of the posterior distribution might cause burn-
 3223 in to occur faster. Note also that we have used a different prior than in our **WinBUGS**
 3224 model specification given previously. We encourage you to evaluate whether this seems to
 3225 affect the result.

3.7 POISSON GLM WITH RANDOM EFFECTS

3226 In most of this book, we will be dealing with random effects in GLM-like models – similar
 3227 to what are usually referred to as generalized linear mixed models (GLMMs). We provide
 3228 a brief introduction of such a model by way of example, extending our Poisson regression
 3229 model to include a random effect.

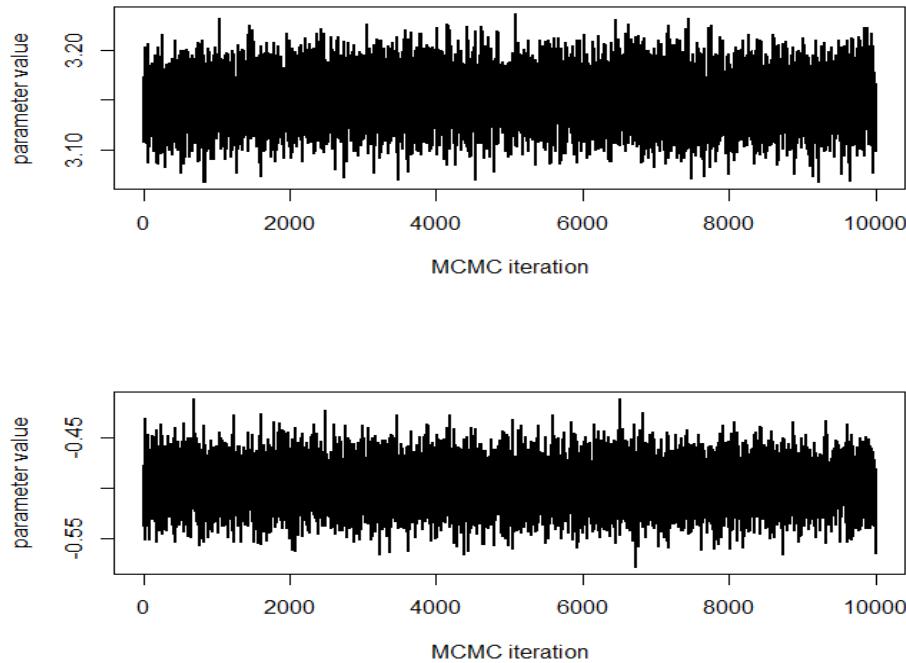


Figure 3.7. Nice grassy plots of 10,000 MCMC iterations for the Poisson GLM model parameters β_0 (top) and β_1 (bottom) using a Metropolis-Hastings tuning parameter of $\delta = 0.05$.

3230 **The Log-Normal mixture:** The classical situation involves a GLM with a normally
 3231 distributed random effect that is additive on the linear predictor. For the Poisson case,
 3232 we have:

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

3233 where $\eta_i \sim \text{Normal}(0, \sigma^2)$. In this context, η could represent an error term capturing
 3234 variation in λ_i not accounted for by the covariates, or overdispersion. It is really amazingly
 3235 simple to express this model in the **BUGS** language and have **WinBUGS** (or **JAGS**,
 3236 etc..) draw samples from the posterior distribution. The code for analysis of the BBS
 3237 dove counts is given as follows:

```
3238 > library(scrbook)
3239 ### Grab the BBS Data as before
3240 > data(bbsdata)
3241 ### Set random seed so that results are repeatable
```

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

Parameter	Mean	SD	2.5%	25%	50%	75%	97.5%	Rhat	n.eff
β_0	2.98	0.08	2.82	2.93	2.98	3.03	3.12	1.00	1400
β_1	-0.53	0.07	-0.68	-0.58	-0.53	-0.49	-0.38	1.01	350
σ	0.60	0.06	0.49	0.56	0.59	0.64	0.73	1.00	2000
τ	2.88	0.57	1.88	2.47	2.86	3.24	4.12	1.00	2000
deviance	445.94	12.18	424.00	437.40	445.20	453.90	471.50	1.00	4000

```

3242 > set.seed(2013)
3243 ### Dump the BUGS model into a file
3244 > cat("
3245 model{
3246   for (i in 1:M){ # Observation model, linear predictor, etc..
3247     y[i] ~ dpois(lam[i])
3248     log(lam[i]) <- beta0+ beta1*habitat[i] + eta[i]
3249     frog[i] <- beta1*habitat[i] + eta[i]
3250     eta[i] ~ dnorm(0,tau)
3251   }
3252   # Prior distributions:
3253   beta0 ~ dunif(-5,5)
3254   beta1 ~ dunif(-5,5)
3255   sigma ~ dunif(0,10)
3256   tau <- 1/(sigma*sigma)
3257 }
3258 ",file="model.txt")

3259 > data <- list ("y","M","habitat") # Define the data
3260 > inits <- function() # inits and parameters
3261   list ( beta0=rnorm(1), beta1=rnorm(1), sigma=runif(1,0,4))
3262 > parameters <- c("beta0","beta1","sigma","tau")
3263
3264 > library(R2WinBUGS)           # Load and run R2WinBUGS
3265 > out <- bugs (data, inits, parameters, "model.txt", n.thin=2,n.chains=2,
3266   n.burnin=1000, n.iter=5000, debug=TRUE)

```

3267 This produces the posterior summary statistics given in Table 3.1. One thing we notice
 3268 is that the posterior standard deviations of the regression parameters are much higher,
 3269 a result of the extra-Poisson variation allowed for by this model. We would also notice
 3270 much less precise predictions of hypothetical new observations.

3.8 BINOMIAL GLMS

3271 Another extremely important class of models in ecology are binomial models. We use
 3272 binomial models for count data whenever the observations are counts or frequencies and
 3273 it is natural to condition on a “sample size”, say K , the maximum frequency possible in
 3274 a sample. The random variable, $y \leq K$, is then the frequency of occurrences out of K
 3275 “trials”. The parameter of the binomial models is p , often called “success probability”
 3276 which is related to the expected value of y by $\mathbb{E}(y) = pK$. Usually we are interested
 3277 in modeling covariates that affect the parameter p , and such models are called binomial
 3278 GLMs, binomial regression models or logistic regression, although logistic regression re-
 3279 ally only applies when the logistic link is used to model the relationship between p and
 3280 covariates (see below).

3281 One of the most typical binomial GLMs occurs when the sample size equals 1 and
 3282 the outcome, y , is “presence” ($y = 1$) or “absence” ($y = 0$) of a species. In this case, y
 3283 has a Bernoulli distribution. This is a classical species distribution modeling situation. A
 3284 special situation occurs when presence/absence is observed with error (MacKenzie et al.,
 3285 2002; Tyre et al., 2003). In that case, $K > 1$ samples are usually needed for effective
 3286 estimation of model parameters.

3287 In standard binomial regression problems the sample size is fixed by design but in-
 3288 teresting models also arise when the sample size is itself a random variable. These are
 3289 the N -mixture models (Royle, 2004b; Kéry et al., 2005; Royle and Dorazio, 2008; Kéry,
 3290 2010) and related models (in this case, N being the sample size, which we labeled K
 3291 above)⁷. Another situation in which the binomial sample size is “fixed” is closed popula-
 3292 tion capture-recapture models in which a population of individuals is sampled K times.
 3293 The number of times each individual is encountered is a binomial outcome with parameter
 3294 (encounter probability) p , based on a sample of size K . In addition, the total number of
 3295 unique individuals observed, n , is also a binomial random variable based on population
 3296 size N . We consider such models in Chapt. 4.

3297 3.8.1 Binomial regression

3298 In binomial models, covariates are modeled on a suitable transformation (the link function)
 3299 of the binomial success probability, p . Let x_i denote some measured covariate for sample
 3300 unit i and let p_i be the success probability for unit or subject i . The standard choice is the
 3301 logit link function (3.1) but there are many other possible link functions. We sometimes use
 3302 the complementary log-log (= “cloglog”) link function in ecological applications because
 3303 it is natural in some cases when the response should scale in relation to area or effort
 3304 (Royle and Dorazio, 2008, p. 150). As an example, the “probability of observing a count
 3305 greater than 0” under a Poisson model is $\Pr(y > 0) = 1 - \exp(-\lambda)$. In that case, for the
 3306 i^{th} observation,

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

3307 so that if you have covariates in your linear predictor for $\mathbb{E}(y)$ under a Poisson model then
 3308 they are linear on the complementary log-log link of p . In models of species occurrence

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

3309 it seems natural to view occupancy as being derived from local abundance N (Royle
 3310 and Nichols, 2003; Royle and Dorazio, 2006; Dorazio, 2007). Therefore, models of local
 3311 abundance in which $N_i \sim \text{Poisson}(A_i \lambda_i)$ for a habitat patch of area A_i implies a model
 3312 for occupancy ψ_i of the form

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

3313 We will use the cloglog link in some analyses of SCR models in Chapt. 5 and elsewhere.

3314 **3.8.2 Example: waterfowl banding data**

3315 The standard binomial modeling problem in ecology is that of modeling species distri-
 3316 butions, where $K = 1$ and the outcome is occurrence ($y = 1$) or not ($y = 0$) of some
 3317 species. Such examples abound in books (e.g., Royle and Dorazio (2008, ch. 3); Kéry
 3318 (2010, ch. 21); Kéry and Schaub (2012, ch. 13)) and in the literature. Therefore, instead,
 3319 we will consider an example involving band returns of waterfowl in the upper great plains
 3320 including some Canadian provinces, which were analyzed by Royle and Dubovsky (2001).

3321 For these data, y_{it} is the number of mallard (*Anas platyrhynchos*) bands recovered out
 3322 of B_{it} birds banded at some location s_i in year t . In this case B_{it} is fixed. Thinking about
 3323 recovery rate as being proportional to harvest rate, we use these data to explore geographic
 3324 gradients in recovery rate resulting from variability in harvest pressure experienced by
 3325 different populations. As such, we fit a basic binomial GLM with a linear response to
 3326 geographic coordinates (including an interaction term). Here we provide the part of the
 3327 script for creating the model and fitting the model in **WinBUGS**. There are few structural
 3328 differences between this model and the Poisson GLM fitted previously. The main things
 3329 are due to the data structure (we have a matrix here instead of a vector) and otherwise
 3330 we change the distributional assumption to binomial (specified with `dbin`) and then use
 3331 the `logit` function to relate the parameter p_{it} to the covariates.

3332 **Dummy variables in BUGS:** In the mallard example, we model the band recovery
 3333 probability p_{it} not only as a linear function (on the logit scale) of geographic location, but
 3334 also allow for variation in p_{it} with year, t ; $t = 1, 2, \dots, T$. In this particular example there
 3335 are $T = 5$ years of data and we could describe the full mallard model with a formula in
 3336 terms of “dummy variables.” Dummy variables are binary variables, one variable for each
 3337 level of the categorical variable they describe, such that variable for level t takes on the
 3338 value 1 if the observation belongs with level t and 0 otherwise. So, the mallard model in
 3339 terms of dummy variables for “year” looks like this:

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

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

3340 Here, x_2 to x_5 are the dummy variable vectors of length T that take on the value of 1
 3341 when t corresponds to the respective year and 0 otherwise; β_0 is the common intercept
 3342 term and corresponds to $t = 1$; $\beta_1 - \beta_4$ describe the difference in p_{it} for each t relative to
 3343 $t = 1$.

3344 There is a more concise way of implementing such a model with a categorical covariate
 3345 in **BUGS**, namely, by using indexing instead of dummy variables⁸. Essentially, instead of
 3346 estimating the difference in p relative to category 1, we estimate a separate intercept term
 3347 for each category, so that we have 5 different β_0 parameters indexed by t . This reduces
 3348 the linear predictor to:

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

3349 The model can be implemented in the **BUGS** language for the mallard banding data
 3350 using the following **R** script, provided in the **scrbook** package (see `help(mallard)`):

```
3351 > library(scrbook)
3352 > data(mallard)      # Load mallard data
3353
3354 > cat("
3355 model{
3356   for(t in 1:5){
3357     for (i in 1:nobs){
3358       y[i,t] ~ dbin(p[i,t], B[i,t])
3359       pl[i,t] <- beta0[t]+beta1*X[i,1]+beta2*X[i,2]+beta3*X[i,1]*X[i,2]
3360       p[i,t] <- exp(pl[i,t])/(1+exp(pl[i,t]))
3361     }
3362   }
3363   beta1 ~ dnorm(0,.001)
3364   beta2 ~ dnorm(0,.001)
3365   beta3 ~ dnorm(0,.001)
3366   for(t in 1:5){
3367     beta0[t] ~ dnorm(0,.001)
3368   }
3369 }
3370 ",file="BinomialGLM.txt")

3371 > library(R2WinBUGS)
3372 > data <- list(B=mallard$bandings, y=mallard$recoveries,
3373   X=mallard$locs, nobs=nrow(mallard$locs))
3374 > inits <- function(){ list(beta0=rnorm(5),beta1=0,beta2=0,beta3=0) }
3375 > parms <- c('beta0','beta1','beta2','beta3')
3376 > out <- bugs(data, inits, parms,"BinomialGLM.txt", n.chains=3,
3377   n.iter=2000, n.burnin=1000, n.thin=2, debug=TRUE)
```

3378 Look at the posterior summaries of model parameters in Table 3.2. The basic result
 3379 suggests a negative east-west gradient and a positive south to north gradient of band
 3380 recovery probabilities, but no interaction. A map of the response surface is shown in Fig.
 3381 3.8.

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

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

Parameter	Mean	SD	2.5%	50%	97.5%	Rhat	n.eff
$\beta_0[1]$	-2.346	0.036	-2.417	-2.346	-2.277	1.001	1500
$\beta_0[2]$	-2.356	0.032	-2.420	-2.356	-2.292	1.001	1500
$\beta_0[3]$	-2.220	0.035	-2.291	-2.219	-2.153	1.001	1500
$\beta_0[4]$	-2.144	0.039	-2.225	-2.143	-2.068	1.000	1500
$\beta_0[5]$	-1.925	0.034	-1.990	-1.924	-1.856	1.004	570
β_1	-0.023	0.003	-0.028	-0.023	-0.018	1.001	1500
β_2	0.020	0.006	0.009	0.020	0.031	1.001	1500
β_3	0.000	0.001	-0.002	0.000	0.002	1.001	1500
deviance	1716.001	4.091	1710.000	1715.000	1726.000	1.001	1500

3.9 BAYESIAN MODEL CHECKING AND SELECTION

3382 In general terms, model checking – or assessing the adequacy of the model – and model
 3383 selection are quite thorny issues and, despite contrary and, sometimes, strongly held belief
 3384 among practitioners, there are not really definitive, general solutions to either problem.
 3385 We're against dogma on these issues and think people need to be open-minded about
 3386 such things and recognize that models can be useful whether or not they pass certain
 3387 statistical tests. Some models are intrinsically better than others because they make more
 3388 biological sense or foster understanding or achieve some objective that some bootstrap or
 3389 other goodness-of-fit test can't decide for you. That said, it gives you some confidence if
 3390 your model seems adequate in a purely statistical sense. We provide a very brief overview
 3391 of concepts here, but provide more detailed coverage in Chapt. 8. See also coverage of
 3392 these topics in Kéry (2010) and Link and Barker (2010) for specific context related to
 3393 Bayesian model checking and selection.

3394 3.9.1 Goodness-of-fit

3395 Goodness-of-fit testing is an important element of any analysis because our model re-
 3396 presents a general set of hypotheses about the ecological and observation processes that
 3397 generated our data. Thus, if our model “fits” in some statistical or scientific sense, then
 3398 we believe it to be consistent with the hypotheses that went into the model. More for-
 3399 mally, we would conclude that the data are *not inconsistent* with the hypotheses, or that
 3400 the model appears adequate. If we have enough data, then of course we will reject any
 3401 set of statistical hypotheses. Conversely, we can always come up with a model that fits
 3402 by making the model extremely complex. Despite this paradox, it seems to us that sim-
 3403 ple models that you can understand should usually be preferred even if they don't fit,
 3404 for example if they embody essential mechanisms central to our understanding of things,
 3405 or if we think that some contributing factors to lack-of-fit are minor or irrelevant to the
 3406 scientific context and intended use of the model. In other words, models can be useful

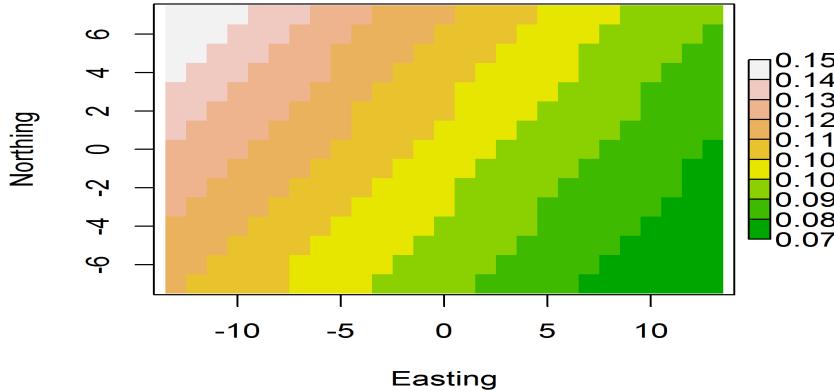


Figure 3.8. Predicted recovery rates of mallard bands in the upper great plains of North America. Note the negative gradient from the NW to the SE.

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

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

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

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

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

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

3424 MCMC iteration), the equivalent statistic is computed for a “new” data set, say \mathbf{y}^{new} ,
 3425 simulated using the current parameter values. From the new data set, we compute the
 3426 same fit statistic:

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

3427 and the Bayesian p-value is simply the posterior probability $\text{Pr}(T(\mathbf{y}^{new}) > T(\mathbf{y}))$ which
 3428 should be close to 0.50 for a good model – one that “fits” in the sense that the observed
 3429 data set is consistent with realizations simulated under the model being fitted to the
 3430 observed data. In practice we judge “close to 0.50” as being “not too close to 0 or 1” and,
 3431 as always, closeness is somewhat subjective. We’re happy with anything $> .1$ and $< .9$
 3432 but might settle for $> .05$ and < 0.95 . Another useful fit statistic is the Freeman-Tukey
 3433 statistic, in which

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

3434 (Brooks et al., 2000), where y_i is the observed value of observation i and $\mathbb{E}(y_i)$ its expected
 3435 value. In contrast to a Chi-square discrepancy, the Freeman-Tukey statistic removes the
 3436 need to pool cells with small expected values. In summary, you can see that the Bayesian
 3437 p-value is easy to compute, and it is widely used as a result.

3438 3.9.2 Model selection

3439 In ecology, scientific hypotheses are often manifest as different models or parameters of
 3440 a model, and so evaluating the importance of different models is fundamental to many
 3441 ecological studies. For Bayesian model selection we typically use three different methods:
 3442 First is, let’s say, common sense. If a variable should plausibly be relevant to explaining
 3443 the data-generating processes, and it has posterior mass concentrated away from 0, then it
 3444 seems like it should be regarded as important - that is, it is “significant.” This approach
 3445 seems to have fallen out of favor in ecology over the last 10 or 15 years but in many
 3446 situations it is a reasonable thing to do.

3447 For regression problems we sometimes use the indicator variable method of Kuo and
 3448 Mallick (1998), in which we introduce a set of binary variables w_k for variable k , and
 3449 express the model as, e.g., for a single covariate model:

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

3450 where w_1 is given a Bernoulli prior distribution with some prescribed probability. E.g.,
 3451 $w_1 \sim \text{Bernoulli}(0.50)$ to provide a prior probability of 0.50 that variable x should be an
 3452 element of the linear predictor. The posterior probability of the event $w_1 = 1$ is a gage of
 3453 the importance of the variable x . i.e., high values of $\text{Pr}(w_1 = 1)$ indicate stronger evidence
 3454 to support that “ x is in the model” whereas values of $\text{Pr}(w_1 = 1)$ close to 0 suggest that
 3455 x is less important. Expansion of the model to include the binary variable w_1 defines a
 3456 set of 2 distinct models for which we can directly compute the posterior probabilities for,
 3457 merely by tallying up the posterior frequency of w_1 . See Royle and Dorazio (2008, Chapt.
 3458 3) for an example in the context of logistic regression.

3459 This approach seems to even work sometimes with fairly complex hierarchical models
 3460 of a certain form. E.g., Royle (2008) applied it to a random effects model to evaluate the
 3461 importance of the random effect component of the model. The main problem, which is

3462 really a general problem in Bayesian model selection, is that its effectiveness and results
 3463 will typically be highly sensitive to the prior distribution on the structural parameters
 3464 (e.g., see Royle and Dorazio (2008, table 3.6)). The reason for this is obvious: If $w_1 = 0$
 3465 for the current iteration of the MCMC algorithm, so that β is sampled from the prior
 3466 distribution, and the prior distribution is very diffuse, then extreme values of β are likely.
 3467 Consequently, when the current value of β is far away from the mass of the posterior when
 3468 $w_1 = 1$, then the Markov chain may only jump from $w_1 = 0$ to $w_1 = 1$ infrequently. One
 3469 seemingly reasonable solution to this problem is to fit the full model to obtain posterior
 3470 distributions for all parameters, and then use those as prior distributions in a “model
 3471 selection” run of the MCMC algorithm (Aitkin, 1991). This seems preferable to more-or-
 3472 less arbitrary restriction of the prior support to improve the performance of the MCMC
 3473 algorithm.

3474 A third method that we advocate is subject-matter context. It seems that there are
 3475 some situations – some models – where one should not have to do model selection because a
 3476 specific model may be necessitated by the biological context of the problem, thus rendering
 3477 a formal hypothesis test pointless (Johnson, 1999). Certain aspects of SCR models are
 3478 such an example. In SCR models, we will see that “spatial location” of individuals is
 3479 an element of the model. The simpler, reduced, model is an ordinary capture-recapture
 3480 model which is not spatially explicit (i.e., Chapt. 4), but it seems silly and pointless to
 3481 think about actually using the reduced model even if we could concoct some statistical
 3482 test to refute the more complex model. The simpler model is manifestly wrong but, more
 3483 importantly, not even a plausible data-generating model! Other examples are when effort,
 3484 area or sample rate is used as a covariate. One might prefer to have such things in models
 3485 regardless of whether or not they pass some statistical litmus test.

3486 Many problems can be approached using one of these methods. In later chapters
 3487 (especially Chapt. 8) we will address model selection in specific contexts and we hope
 3488 those will prove useful for a majority of the situations you might encounter.

3.10 SUMMARY AND OUTLOOK

3489 GLMs and GLMMs are the most useful statistical methods in all of ecology. The prin-
 3490 ciples and procedures underlying these methods are relevant to nearly all modeling and
 3491 analysis problems in every branch of ecology. Therefore, understanding how to analyze
 3492 these models is an essential skill for the quantitative ecologist to possess. If you under-
 3493 stand and can conduct classical likelihood and Bayesian analysis of Poisson and binomial
 3494 GL(M)s, then you will be successful analyzing and understanding more complex classes
 3495 of models that arise. We will see shortly that spatial capture-recapture models are a
 3496 type of GL(M)M and thus having a basic understanding of the conceptual origins and
 3497 formulation of GL(M)s and their analysis is extremely useful.

3498 We note that GL(M)s are routinely analyzed by likelihood methods but we have
 3499 focused on Bayesian analysis here in order to develop the tools that are less familiar
 3500 to most ecologists, and that we will apply in much of the remainder of the book. In
 3501 particular, Bayesian analysis of models with random effects is relatively straightforward
 3502 because the models are easy to analyze conditional on the random effect, using MCMC.
 3503 Thus, we will often analyze SCR models in later chapters by MCMC, explicitly adopting a
 3504 Bayesian inference framework. In that regard, the various **BUGS** engines (**WinBUGS**,

3505 **OpenBUGS, JAGS**; see also Appendix 1) are enormously useful because they provide
3506 an accessible platform for carrying out analyses by MCMC by just describing the model,
3507 and not having to worry about how to actually build MCMC algorithms. That said, the
3508 **BUGS** language is more important than just to the extent that it enables one to do
3509 MCMC - it is useful as a modeling tool because it fosters understanding, in the sense
3510 that it forces you to become intimate with your model. You have to think about and
3511 write down all of the probability assumptions, and the relationships between observations
3512 and latent variables and parameters in a way that is ecologically sensible and statistically
3513 coherent. Because of this, it focuses your thinking on *model construction*, as M. Kéry says
3514 in his **WinBUGS** book (Kéry, 2010), “**WinBUGS** frees the modeler in you.”

3515 While we have emphasized Bayesian analysis in this chapter, and make primary use of
3516 it through the book, we will provide an introduction to likelihood analysis in Chapt. 6
3517 and use those methods also from time to time. Before getting to that, however, it will be
3518 useful to talk about more basic, conventional closed population capture-recapture models
3519 and such models are the topic of the next chapter.

3520
3521

4

3522

CLOSED POPULATION MODELS

3523 In this chapter we introduce ordinary *non-spatial* capture-recapture (CR) models for es-
3524 timating population size in closed populations. A closed population is one whose size, N ,
3525 does not change during the study. Two forms of closure are often discussed: demographic
3526 closure, meaning that no births or deaths occur, and geographic closure, which states
3527 that no individuals move onto or off of the sampled area during the study. Although few
3528 populations are actually closed except during very short time intervals, closed population
3529 CR models serve as the basis for the development of the rest of the models presented in
3530 this book, including the models for open populations discussed in Chapt. 16.

3531 We begin with the most basic capture-recapture model, colloquially referred to as
3532 “model M_0 ” (Otis et al., 1978), in which encounter probability is strictly constant in all
3533 respects (across individuals, and replicates). This allows us to highlight the basic structure
3534 of closed population models as binomial GLMs. We then consider some important exten-
3535 sions of ordinary closed population models that accommodate various types of “individual
3536 effects” — either in the form of explicit, observed covariates (sex, age, body mass) or
3537 unstructured “heterogeneity” in the form of an individual random effect, which represent
3538 unobserved or unmeasured covariates. A special type of individual covariate models is dis-
3539 tance sampling, which could be thought of as the most primitive spatial capture-recapture
3540 model. All of these different types of closed population models are closely related to bi-
3541 nomial (or logistic) regression-type models. In fact, when N is known, they are precisely
3542 logistic regression models.

3543 We emphasize Bayesian analysis of capture-recapture models and we accomplish this
3544 using a method related to classical “data augmentation” from the statistics literature (e.g.,
3545 Tanner and Wong, 1987). This is a general concept in statistics but, in the context of
3546 capture-recapture models where N is unknown, it has a consistent implementation across
3547 classes of capture-recapture models and one that is really convenient from the standpoint
3548 of doing MCMC (Royle et al., 2007; Royle and Dorazio, 2012). We use data augmentation
3549 throughout this book and thus emphasize its conceptual and technical origins and demon-
3550 strate applications to closed population models. We refer the reader to Kéry and Schaub
3551 (2012, ch. 6) for an accessible and complementary development of Bayesian analysis of

3552 ordinary, i.e., nonspatial closed population models.

4.1 THE SIMPLEST CLOSED POPULATION MODEL: MODEL M_0

3553 To start looking at the simplest capture-recapture model, let's suppose there exists a pop-
 3554 ulation of N individuals which we subject to repeated sampling, say over K "occasions",
 3555 such as trap nights, where individuals are captured, marked, released, and subsequently
 3556 recaptured. We suppose that individual encounter histories are obtained, and these are of
 3557 the form of a sequence of 0's and 1's indicating capture ($y = 1$) or not ($y = 0$) during any
 3558 sampling occasion. As an example, suppose $K = 5$ sampling occasions, then an individual
 3559 captured during occasion 2 and 3 but not otherwise would have an encounter history of
 3560 the form $\mathbf{y} = (0, 1, 1, 0, 0)$. Thus, the observation \mathbf{y}_i for each individual ($i = 1, 2, \dots, N$)
 3561 is a vector having elements denoted by y_{ik} for $k = 1, 2, \dots, K$. Usually this is organized
 3562 as a row of a matrix with elements y_{ik} , see Table 4.1. Except where noted explicitly,
 3563 we suppose that observations are independent within individuals and among individuals.
 3564 Formally, this allows us to say that y_{ik} are independent and identically distributed ("iid")
 3565 Bernoulli random variables and we may write $y_{ik} \sim \text{Bernoulli}(p)$. Consequently, for this
 3566 very simple model in which p is constant (i.e., there are no individual or temporal co-
 3567 variates that affect p) the original binary detection variables can be aggregated into the
 3568 total number of encounters for each individual¹, $y_{i\cdot} = \sum_k y_{ik}$, and the observation model
 3569 changes from a Bernoulli distribution to a binomial distribution based on a sample of size
 3570 K . That is

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

3571 for every individual in the population $i = 1, 2, \dots, N$, where N is the number of individuals
 3572 in the population (i.e., population size).

3573 We emphasize the central importance of the basic Bernoulli encounter model – an
 3574 individual is either encountered in a sample, or not – which forms the cornerstone of
 3575 almost all of classical capture-recapture models, including many spatial capture-recapture
 3576 models discussed in this book.

3577 Evidently, the basic capture-recapture model is a simplistic version of a logistic-
 3578 regression model with only an intercept term ($\text{logit}(p) = \text{constant}$). To say that all
 3579 capture-recapture models are just logistic regressions is a slight over-simplification. In
 3580 fact, we are proceeding here as if we knew N . In practice we don't, of course, and esti-
 3581 mating N is actually the central objective. But, by proceeding as if N were known, we
 3582 can specify a simple model and then deal with the fact that N is unknown using standard
 3583 methods that you are already familiar with (i.e., GLMs - see Chapt. 3).

3584 Assuming individuals in the population are encountered independently, the joint prob-
 3585 ability distribution of the observations is the product of N binomials

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

3586 We emphasize that this expression is conditional on N , in which case we get to observe
 3587 the $y_i = 0$ observations and the resulting data are just iid binomial counts. Because this

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

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

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

3588 is a binomial regression model of the variety described in Chapt. 3, fitting this model
 3589 using a **BUGS** engine poses no difficulty.

3590 Equation 4.1.1 can be simplified even further if we reformat the observations as en-
 3591 counter frequencies. Specifically, let n_k denote the number of individuals captured exactly
 3592 k times after K survey occasions, $n_k = \sum_{i=1}^N I(y_i = k)$ where $I()$ is the indicator func-
 3593 tion evaluating to 1 if its argument is true and 0 otherwise. For sake of illustration, we
 3594 converted the data from Table 4.1 to this format (Table 4.2). What is important to note
 3595 is that if we know N , then we know n_0 , i.e. the number of individuals not captured. In
 3596 this case, an alternative and equivalent expression to Eq. 4.1.1 is

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

3597 where $\pi_k = \Pr(y = k)$ under the binomial model with parameter p and sample size K .
 The essential problem in capture-recapture, however, is that N is *not* known because the

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

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

3598 number of uncaptured individuals (n_0) is unknown. Consequently, the observed capture
 3599 frequencies n_k are no longer independent because n_0 is a function of the other frequencies,
 3600 $n_0 = N - \sum_{k=1}^K n_k$. Hence, their joint distribution is multinomial (e.g., see Illian et al.
 3601 (2008, p. 61)):

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

3603 We gave a general overview of the multinomial distribution in Sec. 2.2. The multino-
 3604 mial distribution is the standard model for discrete responses that can fall into a fixed
 3605 number ($K + 1$ in this case) of possible categories. In the context of capture-recapture,

3606 the multinomial posits a population of N individuals with $K + 1$ possible outcomes de-
 3607 fined by the possible encounter frequencies: encountered $y = 1, 2, \dots, K$ times or not
 3608 encountered at all. These possible outcomes occur with probabilities π_k , which we refer
 3609 to as “cell probabilities” or in the specific context of capture-recapture, encounter history
 3610 probabilities.

3611 To fit the model in which N is *unknown*, we can regard n_0 as a parameter and maximize
 3612 the multinomial likelihood directly. Direct likelihood analysis of the multinomial model is
 3613 straightforward, but that is not always sufficiently useful in practice because we seldom
 3614 are concerned with models for the aggregated encounter history frequencies, which entail
 3615 that capture probabilities are the same for all individuals. In many instances, including
 3616 for spatial capture-recapture (SCR) models, we require a formulation of the model that
 3617 can accommodate individual-level covariates to account for differences in detection among
 3618 individuals, which we address subsequently in this chapter, and also in Chapt. 7.

3619 **4.1.1 The core capture-recapture assumptions**

3620 This basic capture-recapture model – model M_0 – comes with it a host of specific biological
 3621 and statistical assumptions. In addition to the basic assumption of population closure,
 3622 Otis et al. (1978) list the following:

- 3623 1. animals do not lose their marks during the experiment,
 3624 2. all marks are correctly noted and recorded at each trapping occasion, and
 3625 3. each animal has a constant and equal probability of capture on each trapping oc-
 3626 casion.

3627 The remainder of their classic work is dedicated to relaxing assumption 3. While assump-
 3628 tions 1 and 2 are undoubtedly necessary for inference from basic CR methods to be valid,
 3629 and while they are also assumed by most of the models we present in the following chap-
 3630 ters, we refrain from repeatedly making such statements. Our opinion is that all model
 3631 assumptions are apparent when a model is clearly specified, and it is both redundant and
 3632 impossible to list all the things not allowed by the model. For example, closed population
 3633 models also assume that other sources of error do not occur, but it is not necessary to
 3634 enumerate each possibility. Rather, it is necessary to make clear statements such as

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

3635 This simple model description carries a tremendous amount of information, and it leaves
 3636 very little left to say with respect to assumptions. Although we will not always show
 3637 the *iid* symbol, it will be assumed unless otherwise noted, and this assumption is critical
 3638 for valid inference. It implies that the encounter of one individual does not affect the
 3639 encounter of another individual, and encounter does not affect future encounter. Under
 3640 this assumption, it is easy to write down the likelihood of the parameters and obtain
 3641 parameter estimates; however, whether or not it is true depends upon biological and
 3642 sampling issues. If this assumption is deemed false, the model can be discarded in favor
 3643 of a more realistic alternative. However, once we have settled on our model, statistical
 3644 inference proceeds by assuming the model is truth—not an approximation to truth—but
 3645 actual truth.

3646 In spite of the fact that we assume that all models are truth, but we acknowledge that
 3647 all models are wrong due to their assumptions, assumptions should not be viewed as a
 3648 necessary evil. In fact, one way to view assumptions is as embodiments of our ecological
 3649 hypotheses. If we make these assumptions too complex or too specific, then we will never
 3650 be able to study general phenomena that hold true across space and time. Furthermore,
 3651 in practice, we will rarely have enough data to estimate the parameters of highly complex
 3652 models.

3653 4.1.2 Conditional likelihood

3654 We saw that the closed population model is a simple logistic regression model if N is known
 3655 and, when N is unknown, the model is multinomial with index or sample size parameter
 3656 N . This multinomial model, being conditional on N , is sometimes referred to as the “joint
 3657 likelihood” the “full likelihood” or the “unconditional likelihood” (sometimes “model” in
 3658 place of “likelihood”) (Sanathanan, 1972; Borchers et al., 2002). This formulation differs
 3659 from the so-called “conditional likelihood” approach in which the likelihood of the observed
 3660 encounter histories is devised conditional on the event that an individual is captured at
 3661 least once. To construct this likelihood, we have to recognize that individuals appear
 3662 or not in the sample based on the value of the random variable y_i , that is, if and only
 3663 if $y_i > 0$. The observation model is therefore based on $\Pr(y|y > 0)$. For the simple
 3664 case of model M_0 , the resulting conditional distribution is a “zero truncated” binomial
 3665 distribution which accounts for the fact that we cannot observe the value $y = 0$ in the data
 3666 set. Both the conditional and unconditional models are legitimate modes of analysis in
 3667 all capture-recapture types of studies. They provide equally valid descriptions of the data
 3668 and, for many practical purposes provide equivalent inferences, at least in large sample
 3669 sizes (Sanathanan, 1972).

3670 In this book we emphasize Bayesian analysis of capture-recapture models using data
 3671 augmentation (described in Sec. 4.2 below), which produces yet a third distinct formu-
 3672 lation of capture-recapture models based on the zero-*inflated* binomial distribution that
 3673 we describe in the next section. Thus, there are 3 distinct formulations of the model – or
 3674 modes of analysis – for analyzing all capture-recapture models based on the (1) binomial
 3675 model for the joint or unconditional specification; (2) zero-truncated binomial that arises
 3676 “conditional on n ”; and (3) the zero-inflated binomial that arises under data augmen-
 3677 tation. Each formulation has distinct model parameters (shown in Table 4.3 for model
 3678 M_0).

Table 4.3. Modes of analysis of capture-recapture models. Closed population models can be analyzed using the joint or “full likelihood” which contains N as an explicit parameter, the conditional likelihood which does not involve N , or by data augmentation which replaces N with ψ . 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, ψ	zero-inflated binomial

4.2 DATA AUGMENTATION

3679 We consider a method of analyzing closed population models using parameter-expanded
 3680 data augmentation (PX-DA), which we abbreviate to “data augmentation” or DA, which
 3681 is useful for Bayesian analysis and, in particular, analysis of models using the various
 3682 **BUGS** engines and other Bayesian model fitting software. Data augmentation is a general
 3683 statistical concept that is widely used in statistics in many different settings. The classical
 3684 reference is Tanner and Wong (1987), but see also Liu and Wu (1999). Data augmentation
 3685 can be adapted to provide a very generic framework for Bayesian analysis of capture-
 3686 recapture models with unknown N . This idea was introduced for closed populations by
 3687 Royle et al. (2007), and has subsequently been applied to a number of different contexts
 3688 including individual covariate models (Royle, 2009b), open population models (Royle and
 3689 Dorazio, 2008, 2012; Gardner et al., 2010a), spatial capture-recapture models (Royle and
 3690 Young, 2008; Royle et al., 2009a; Gardner et al., 2009), and many others. Kéry and Schaub
 3691 (2012, Chaps. 6 and 10) provide a good introduction to data augmentation in the context
 3692 of closed and open population models.

3693 Conceptually, the technique of data augmentation represents a reparameterization
 3694 of the “complete data” model – i.e., that conditional on N . The reparameterization
 3695 is achieved by embedding this data set into a larger data set having $M > N$ “rows”
 3696 (individuals) and re-expressing the model conditional on M instead of N . The great thing
 3697 about data augmentation is that we do not need to know N for this reparameterization.
 3698 Although this has a whiff of arbitrariness or even outright ad hockery to it, in the choice
 3699 of M , it is always possible, in practice, to choose M pretty easily for a given problem and
 3700 context and results will be insensitive to choice of M^2 . Then, under data augmentation,
 3701 analysis is focused on the “augmented data set.” That is, we analyze the bigger data set -
 3702 the one having M rows - with an appropriate model that accounts for the augmentation.
 3703 This is achieved by a Bernoulli sampling process that determines whether an individual
 3704 in M is also a member of N . Inference is focused directly on estimating the proportion
 3705 $\psi = E[N]/M$, instead of directly on N , where ψ is the “data augmentation parameter.”

3706 4.2.1 DA links occupancy models and closed population models

3707 There is a close correspondence between so-called “occupancy” models and closed popu-
 3708 lation models (see Royle and Dorazio, 2008, Sec. 5.6). In occupancy models (MacKenzie
 3709 et al., 2002; Tyre et al., 2003) the sampling situation is that M sites, or patches, are sam-
 3710 pled multiple times to assess whether a species occurs at the sites. This yields encounter
 3711 data such as that illustrated in the left panel of Table 4.4. The important problem is that
 3712 a species may occur at a site, but go undetected, yielding an all-zero encounter history for
 3713 the site, which in the case of occupancy studies, are *observed*. However, some of the zero
 3714 vectors will typically correspond to sites where the species in fact *does* occur. Thus, while
 3715 the zeros are observed, there are too many of them and, in a sense, the inference problem
 3716 is to partition the zeros into “structural” (fixed) and “sampling” (or stochastic) zeros,
 3717 where the former are associated with unoccupied sites and the latter with occupied sites
 3718 where the species went undetected. More formally, inference is focused on the parameter
 3719 ψ , the probability that a site is occupied.

²Unless the data set is sufficiently small that parameters are weakly identified

3720 In contrast to occupancy studies, in classical closed population studies, we observe a
 3721 data set as in the middle panel of Table 4.4 where *no* zeros are observed. The inference
 3722 problem is, essentially, to estimate how many sampling zeros there are – or should be – in
 3723 a “complete” data set. This objective (how many sampling zeros?) is precisely the same
 3724 for both types of problems if an upper limit M is specified for the closed population model.
 3725 The only distinction being that, in occupancy models, M is set by design (i.e., the number
 3726 of sites in the sample), whereas a natural choice of M for capture-recapture models may
 3727 not be obvious. However, the choice of M induces a uniform prior for N on the integers
 3728 $[0, M]$ (Royle et al., 2007). Then, one can analyze capture-recapture models by adding
 3729 $M - n$ all-zero encounter histories to the data set and regarding the augmented data
 3730 set, essentially, as a site-occupancy data set, where the occupancy or data augmentation
 3731 parameter (ψ) takes the place of the abundance parameter (N).

3732 Thus, the heuristic motivation of data augmentation is to fix the size of the data
 3733 set by adding *too many* all-zero encounter histories to create the data set shown in the
 3734 right panel of Table 4.4, and then analyze the augmented data set using an occupancy
 3735 type model which includes both “unoccupied sites” (in capture-recapture, augmented
 3736 individuals that are not members of the real population that was sampled) as well as
 3737 “occupied sites” (in capture-recapture, individuals that are members of the population
 3738 but that were undetected by sampling) at which detections did not occur. We call these
 3739 $M - n$ all-zero histories “potential individuals” because they exist to be recruited (in a
 3740 non-biological sense) into the population, for example during an analysis by MCMC.

3741 To analyze the augmented data set, we recognize that it is a zero-inflated version of
 3742 the known- N data set. That is, some of the augmented all-zero rows are sampling zeros
 3743 (corresponding to actual individuals that were missed) and some are “structural” zeros,
 3744 which do not correspond to individuals in the population. For a basic closed-population
 3745 model, the resulting likelihood under data augmentation – that is, for the data set of size
 3746 M – is a simple zero-inflated binomial likelihood. The zero-inflated binomial model can be
 3747 described “hierarchically”, by introducing a set of binary latent variables, z_1, z_2, \dots, z_M ,
 3748 to indicate whether each individual i is ($z_i = 1$) or is not ($z_i = 0$) a member of the
 3749 population of N individuals exposed to sampling. We assume that $z_i \sim \text{Bernoulli}(\psi)$
 3750 where ψ is the probability that an individual in the data set of size M is a member of
 3751 the sampled population – in the sense that $1 - \psi$ is the probability of a “structural zero”
 3752 in the augmented data set. The zero-inflated binomial model which arises under data
 3753 augmentation can be formally expressed by the following set of assumptions (we include
 3754 typical priors for a Bayesian analysis):

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

3755 for $i = 1, \dots, M$, where $I(y = 0)$ is a point mass at $y = 0$. It is sometimes convenient to
 3756 express the conditional-on- z observation model concisely in just one step:

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

3757 and we understand this to mean, if $z_i = 0$, then y_i is necessarily 0 because its success
 3758 probability is $z_i p = 0$.

3759 Note that, under data augmentation, N is no longer an explicit parameter of this
 3760 model. In its place, we estimate ψ and functions of the latent variables z . In particular,
 3761 under the assumptions of the zero-inflated model, $z_i \stackrel{iid}{\sim} \text{Bernoulli}(\psi)$; therefore, N is a
 3762 function of these latent variables:

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

3763 Further, we note that the latent z_i parameters *can be* removed from the model by inte-
 3764 gration, in which case the joint probability of the data is

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

3765 Interpreted as a likelihood, we can directly maximize this expression to obtain the MLEs of
 3766 the structural parameters ψ and p or those of other more complex models (e.g., see Royle,
 3767 2006). We could estimate these parameters and then use them to obtain an estimator of
 3768 N using the so-called “Best unbiased predictor” (see Royle and Dorazio, 2012). Normally,
 3769 however, we will analyze the model in its “conditional-on- z ” form using methods of MCMC
 3770 either in the **BUGS** engines or using our own MCMC algorithms (see Chapt. 17).

3771 4.2.2 Model M_0 in **BUGS**

3772 It is helpful to understand data augmentation by seeing what its effect is on implementing
 3773 model M_0 . For this model, in which we can aggregate the encounter data to individual-
 3774 specific encounter frequencies, the augmented data are given by the vector of frequencies
 3775 $(y_1, \dots, y_n, 0, 0, \dots, 0)$ where the augmented values of $y = 0$ represent the encounter fre-
 3776 quency for potential individuals y_{n+1}, \dots, y_M . The zero-inflated model of the augmented
 3777 data combines the model of the latent variables, $z_i \sim \text{Bernoulli}(\psi)$. The **BUGS** model
 3778 description of the closed population model M_0 is shown in Panel 4.1. The last line of the
 3779 model specification provides the expression for computing N from the data augmentation
 3780 variables z_i . Note that, to improve readability of code snippets (especially of large ones),
 3781 we will sometimes deviate from our standard notation a bit. In this case we use **nind**
 3782 for n (the number of encountered individuals), and $M = nind + nz$ is the total size of the
 3783 augmented data set. In other cases we might also use **nocc** in place of K and **ntraps**
 3784 in place of J . We find that word definitions make code easier to understand, especially
 3785 without having to read surrounding text.

3786 Specification of a more general model in terms of the individual encounter observations
 3787 y_{ik} is not much more difficult than for the individual encounter frequencies. We define
 3788 the observation model by a double loop and change the indexing of quantities accordingly,
 3789 i.e.,

```
3790 for(i in 1:(nind+nz)){
  3791   z[i] ~ dbern(psi)
  3792   for(k in 1:K){
    3793     mu[i,k] <- z[i]*p
```

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

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

```

3794     y[i,k] ~ dbin(mu[i,k],1)
3795   }
3796 }

```

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

4.2.3 Formal development of data augmentation (DA)

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

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

```

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

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

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

```

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

3807 The model assumptions, specifically the multinomial model (Eq. 4.1.3) and Eq. 4.2.2, may
 3808 be combined to yield a reparameterization of the conventional model that is appropriate
 3809 for the augmented data set of known size M :

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

3810 This expression arises by removing N from Eq. 4.1.3 by integrating over the binomial
 3811 prior distribution for N . Thus, the models we analyze under data augmentation arise
 3812 formally by removing the parameter N from the ordinary closed-population model, which
 3813 is conditional on N , by integrating over a binomial prior distribution for N .

3814 Note that the $M - n$ unobserved individuals in the augmented data set have probability
 3815 $\psi\pi(0) + (1 - \psi)$, indicating that these unobserved individuals are a mixture of individuals
 3816 that are sampling zeros ($\psi\pi_0$), and belong to the population of size N , and others that
 3817 are “structural zeros” (occurring in the augmented data set with probability $1 - \psi$). In
 3818 Eq. 4.2.3, N has been eliminated as a formal parameter of the model by marginalization
 3819 (integration) and replaced with the new parameter ψ , the data augmentation parameter.
 3820 However, the full likelihood containing both N and ψ can also be analyzed (see Royle
 3821 et al., 2007).

3822 4.2.4 Remarks on data augmentation

3823 Data augmentation may seem like a strange and mysterious black-box, and likely it is un-
 3824 familiar to most people, even to many of those with substantial experience with capture-

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

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

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

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

3868 4.2.5 Example: Black bear study on Fort Drum

3869 To illustrate the analysis of model M_0 using data augmentation, we use a data set collected
 3870 at Fort Drum Military Installation in upstate New York by P.D. Curtis and M.T Wegan of
 3871 Cornell University and their colleagues at the Fort Drum Military Installation. These data
 3872 have been analyzed in various forms by Wegan (2008); Gardner et al. (2009) and Gardner
 3873 et al. (2010b). The specific data used here are encounter histories on 47 individuals
 3874 obtained from an array of 38 baited “hair snares” (Fig. 4.1) during June and July 2006.
 3875 Barbed wire traps were baited and checked for hair samples each week for eight weeks,
 3876 thus we distinguished $K = 8$ weekly sample intervals. The data are provided in the **R**
 3877 package **scrbook**, can be loaded by typing `data(beardata)` at the **R** prompt, and the
 3878 analysis can be set up and run as follows (see `?beardata` for the commands to do the
 3879 analysis). Here, the data were augmented with 128 all-zero encounter histories, resulting
 3880 in a total sample size of $M = 175$.

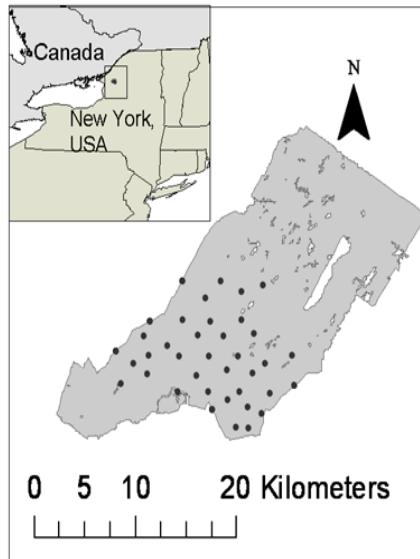


Figure 4.1. Fort Drum Black bear study area and the 38 baited hair snare locations operated for 8 weeks during June and July, 2006.

```

3881 > library(scrbook)
3882 > data(beardata)           # load the bear data and extract components
3883 > trapmat <- beardata$trapmat
3884 > nind <- dim(beardata$bearArray)[1]
3885 > K <- dim(beardata$bearArray)[3]
3886 > ntraps <- dim(beardata$bearArray)[2]
3887
  
```

```

3888 > M <- 175
3889 > nz <- M-nind
3890 > Yaug <- array(0, dim=c(M,ntraps,K))
3891
3892 > Yaug[1:nind,,] <- beardata$bearArray
3893 > y <- apply(Yaug,c(1,3),sum) # summarize by ind x rep
3894 > y[y>1] <- 1 # toss out multiple encounters per occasion
3895 # b/c traditional CR models ignore space

```

3896 The raw data object, `beardata$bearArray` is a 3-dimensional array $nind \times ntraps \times K$ 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 , see below), it is sufficient to reduce the data to individual encounter frequencies which we have re-labeled “y” above.

3897 The **BUGS** model file along with commands to fit the model are as follows:

```

3901 > set.seed(2013) # to obtain the same results each time
3902 > library(R2WinBUGS) # load R2WinBUGS, set-up:
3903 > data0 <- list(y=y, M=M, K=K) # data ....
3904 > params0 <- c('psi','p','N') # parameters ....
3905 > zst <- c(rep(1,nind),rbinom(M-nind, 1, .5)) # inits ....
3906 > inits <- function(){ list(z=zst, psi=runif(1), p=runif(1)) }
3907
3908 > cat("
3909 model{
3910
3911   psi ~ dunif(0, 1)
3912   p ~ dunif(0,1)
3913
3914   for (i in 1:M){
3915     z[i] ~ dbern(psi)
3916     for(k in 1:K){
3917       tmp[i,k] <- p*z[i]
3918       y[i,k] ~ dbin(tmp[i,k],1)
3919     }
3920   }
3921   N<-sum(z[1:M])
3922 }
3923 ",file="modelM0.txt")
3924
3925 ## Run the model:
3926 > fit0 <- bugs(data0, inits, params0, model.file="modelM0.txt",n.chains=3,
3927   n.iter=2000, n.burnin=1000, n.thin=1,debug=TRUE,working.directory=getwd())

```

3928 This produces the following posterior summary statistics:

```

3929 > print(fit0,digits=2)
3930 Inference for Bugs model at "modelM0.txt", fit using WinBUGS,

```

```

3931 3 chains, each with 2000 iterations (first 1000 discarded)
3932 n.sims = 3000 iterations saved
3933      mean    sd   2.5%   25%   50%   75% 97.5% Rhat n.eff
3934 psi     0.29  0.04  0.22  0.26  0.29  0.31  0.36    1 3000
3935 p      0.30  0.03  0.25  0.28  0.30  0.32  0.35    1 3000
3936 N      49.94 1.99 47.00 48.00 50.00 51.00 54.00    1 3000
3937 deviance 489.05 11.28 471.00 480.45 488.80 495.40 513.70    1 3000
3938
3939 [... some output deleted ...]

```

3940 WinBUGS did well in choosing an MCMC algorithm for this model – we have $\hat{R} = 1$
 3941 for each parameter, and an effective sample size of 3000, equal to the total number of
 3942 posterior samples³. We see that the posterior mean of N under this model is 49.94 and
 3943 a 95% posterior interval is (48, 54). We revisit these data later in the context of more
 3944 complex models.

3945 In order to obtain an estimate of density, D , we need an area to associate with the
 3946 estimate of N , and in Chapt. 1 we already went through a number of commonly used
 3947 procedures to conjure up such an area, including buffering the trap array by the home range
 3948 radius, often estimated by the mean maximum distance moved (MMDM) (Parmenter
 3949 et al., 2003), 1/2 MMDM (Dice, 1938) or directly from telemetry data (Wallace et al.,
 3950 2003). Typically, the trap array is defined by the convex hull around the trap locations,
 3951 and this is what we applied a buffer to. We computed the buffer by using a telemetry-based
 3952 estimate of the mean female home range radius (2.19 km) (Bales et al., 2005) instead of
 3953 using an estimate based on our relatively more sparse recapture data. For the Fort Drum
 3954 study, the convex hull has an area of 157.135 km², and the buffered convex hull has an
 3955 area of 277.011 km². To create this we used functions contained in the R package **rgeos**
 3956 and created a utility function **bcharea** which is in our R package **scrbook**. The commands
 3957 are as follows:

```

3958 > library(rgeos)
3959
3960 > bcharea <- function(buff,traplocs){
3961   p1 <- Polygon(rbind(traplocs,traplocs[1,]))
3962   p2 <- Polygons(list(p1=p1),ID=1)
3963   p3 <- SpatialPolygons(list(p2=p2))
3964   p1ch <- gConvexHull(p3)
3965   bp1 <- (gBuffer(p1ch, width=buff))
3966   plot(bp1, col='gray')
3967   plot(p1ch, border='black', lwd=2, add=TRUE)
3968   gArea(bp1)
3969 }
3970
3971 > bcharea(2.19,traplocs=trapmat)

```

3972 The resulting buffered convex hull is shown in Fig. 4.2.

3973 To conjure up a density estimate under model M_0 , we compute the appropriate pos-
 3974 terior summary of the ratio of N and the prescribed area (277.011 km²):

³This is even a little suspicious....

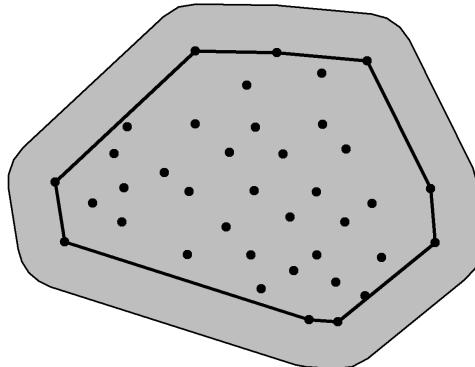


Figure 4.2. Convex hull of the bear hair snare array at Fort Drum, NY, buffered by mean female home range radius (2.19 km).

```

3975 > summary(fit0$sims.list$N/277.011)
3976   Min. 1st Qu. Median Mean 3rd Qu. Max.
3977 0.1697 0.1733 0.1805 0.1803 0.1841 0.2130
3978
3979 > quantile(fit0$sims.list$N/277.011,c(0.025,0.975))
3980      2.5%    97.5%
3981 0.1696684 0.1949381

```

3982 which yields a density estimate of about $0.18 \text{ ind}/\text{km}^2$, and a 95% Bayesian confidence
 3983 interval of $(0.170, 0.195)$. Our estimate of density should be reliable if we have faith in
 3984 our stated value of the “sampled area”. Clearly though this is largely subjective, and not
 3985 something we can formally evaluate (or estimate) from the data based on model M_0 .

4.3 TEMPORALLY VARYING AND BEHAVIORAL EFFECTS

3986 The purpose of this chapter is mainly to emphasize the central importance of the binomial
 3987 model in capture-recapture and so we have considered models for individual encounter
 3988 frequencies—the number of times individuals are captured out of K occasions. Sometimes
 3989 we can’t aggregate the encounter data for each individual, such as when encounter proba-
 3990 bility varies over time among samples. Time-varying responses that are relevant in many

3991 capture-recapture studies are “effort” such as amount of search time, number of observers,
 3992 or trap nights, or encounter probability varying over time, as a function of date or season
 3993 (Kéry et al., 2010) due to species behavior. A common situation in many animal studies
 3994 is that in which there exists a “behavioral response” to trapping (even if the animal is not
 3995 physically trapped).

3996 Behavioral response is an important concept in animal studies because individuals
 3997 might learn to come to baited traps or avoid traps due to trauma related to being encoun-
 3998 tered. There are a number of ways to parameterize a behavioral response to encounter.
 3999 The distinction between persistent and ephemeral was made by Yang and Chao (2005)
 4000 who considered a general behavioral response model of the form:

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

4001 where x_{ik} is a covariate indicator variable of previous capture (i.e., $x_{ik} = 1$ if captured
 4002 in any previous period). Therefore, encounter probability changes depending on whether
 4003 an individual was captured in the immediate previous period (a Markovian or ephemeral
 4004 behavioral response; (Yang and Chao, 2005)), described by the term $\alpha_1 y_{i,k-1}$ or in *any*
 4005 previous period (persistent behavioral response), described by the term $\alpha_2 x_{ik}$. Because
 4006 spatial capture-recapture models allow us to include trap-specific covariates, we can de-
 4007 scribe a 3rd type of behavioral response—a local behavioral response that is trap-specific
 4008 (Royle et al., 2011b). In this local behavioral response, the encounter probability is mod-
 4009 ified for an individual trap depending on previous capture in that trap. Models with
 4010 temporal effects are easy to describe and analyze in the **BUGS** language and we provide
 4011 a number of examples in Chapt. 7 and elsewhere.

4.4 MODELS WITH INDIVIDUAL HETEROGENEITY

4012 Models in which encounter probability varies by individual have a long history in capture-
 4013 recapture and, indeed, this so-called “model M_h ” is one of the elemental capture-recapture
 4014 models in (Otis et al., 1978). Conceptually, we imagine that the individual-specific en-
 4015 counter probability parameters, p_i , are random variables distributed according to some
 4016 probability distribution, $[\theta]$. We denote this basic model assumption as $p_i \sim [\theta]$. This
 4017 type of model is similar in concept to extending a GLM to a GLMM but in the capture-
 4018 recapture context N is unknown. The basic class of models is often referred to as “model
 4019 M_h ” (“h” for heterogeneity), but really this is a broad class of models, each being dis-
 4020 tinguished by the specific distribution assumed for p_i . There are many different varieties
 4021 of model M_h including parametric and various non-parametric approaches (Burnham and
 4022 Overton, 1978; Norris and Pollock, 1996; Pledger, 2004). One important practical matter
 4023 is that estimates of N can be extremely sensitive to the choice of heterogeneity model
 4024 (Fienberg et al., 1999; Dorazio and Royle, 2003; Link, 2003). Indeed, Link (2003) showed
 4025 that in some cases it’s possible to find models that yield precisely the same expected data,
 4026 yet produce wildly different estimates of N . In that sense, N for most practical pur-
 4027 poses is not identifiable across classes of different heterogeneity models, and this should
 4028 be understood before fitting any such model. One solution to this problem is to seek
 4029 to model explicit factors that contribute to heterogeneity, e.g., using individual covariate
 4030 models (See 4.5 below). Indeed, spatial capture-recapture models do just that, by mod-
 4031 eling heterogeneity due to the spatial organization of individuals in relation to traps or

4032 other encounter mechanism. For additional background and applications of model M_h see
 4033 Royle and Dorazio (2008, Chapt. 6) and Kéry and Schaub (2012, Chapt. 6).

4034 We will work with a specific type of model M_h here which is a natural extension of
 4035 the basic binomial observation model of model M_0 so that

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

4036 where μ is a fixed parameter (the mean) to be estimated, and η_i is an individual random
 4037 effect assumed to be normally distributed:

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

4038 We could as well combine these two steps and write $\text{logit}(p_i) \sim \text{Normal}(\mu, \sigma_p^2)$. This
 4039 “logit-normal mixture” was analyzed by Coull and Agresti (1999) and elsewhere. It is
 4040 a natural extension of the basic model with constant p , as a mixed GLMM, and similar
 4041 models occur throughout statistics. It is also natural to consider a beta prior distribution
 4042 for p_i (Dorazio and Royle, 2003) and so-called “finite-mixture” models are also popular
 4043 (Norris and Pollock, 1996; Pledger, 2004). In the latter, individuals are assumed to belong
 4044 to a finite number of latent classes, each of which has its own capture probability.

4045 Model M_h has important historical relevance to spatial capture-recapture situations
 4046 (Karanth, 1995) because investigators recognized that the juxtaposition of individuals with
 4047 the array of trap locations should yield heterogeneity in encounter probability, and thus it
 4048 became common to use some version of model M_h in spatial trapping arrays to estimate
 4049 N . While this doesn’t resolve the problem of not knowing the effective sample area, it
 4050 does yield an estimator that accommodates the heterogeneity in p induced by the spatial
 4051 aspect of capture-recapture studies. To see how this juxtaposition induces heterogeneity,
 4052 we have to understand the relevance of movement in capture-recapture models. Imagine a
 4053 quadrat that can be uniformly searched by a crew of biologists for some species of reptile
 4054 (see Royle and Young (2008)). Figure 4.3 shows a sample quadrat searched repeatedly
 4055 over a period of time. Further, suppose that the species exhibits some sense of spatial
 4056 fidelity in the form of a home range or territory, and individuals move about their home
 4057 range (home range centroids are given by the solid dots) in some kind of random fashion.
 4058 Heuristically, we imagine that each individual in the vicinity of the study area is liable
 4059 to experience variable exposure to encounter due to the overlap of its home range with
 4060 the sampled area - essentially the long-run proportion of times the individual is within
 4061 the sample plot boundaries, say ϕ . We might model the exposure or *availability* of an
 4062 individual to capture by supposing that $a_i = 1$ if individual i is available to be captured
 4063 (i.e., within the survey plot) during any sample, and 0 otherwise. Then, $\Pr(a_i = 1) = \phi$.
 4064 In the context of spatial studies, it is natural that ϕ should depend on *where* an individual
 4065 lives, i.e., it should be individual-specific ϕ_i (Chandler et al., 2011). This system describes,
 4066 precisely, that of “random temporary emigration” (Kendall et al., 1997) where ϕ_i is the
 4067 individual-specific probability of being “available” for capture.

4068 Conceptually, SCR models aim to deal with this problem of variable exposure to sam-
 4069 pling due to movement in the proximity of the trapping array explicitly and formally with
 4070 auxiliary spatial information. If individuals are detected with probability p_0 , *conditional*
 4071 on $a_i = 1$, then the marginal probability of detecting individual i is

$$p_i = p_0 \phi_i$$

4072 so we see clearly that individual heterogeneity in encounter probability is induced as a re-
 4073 sult of the juxtaposition of individuals (i.e., their home ranges) with the sample apparatus
 4074 and the movement of individuals about their home range.

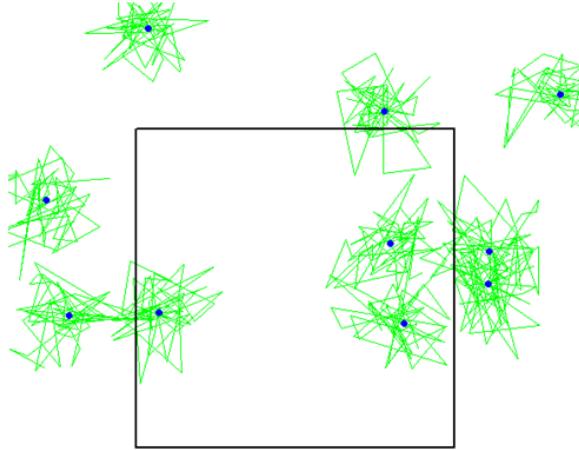


Figure 4.3. A quadrat searched for lizards over some period of time (simulated data). The locations of encounter for each of 10 lizards are connected by lines—the dots are activity centers.

4075 4.4.1 Analysis of model M_h

4076 If N is known, it is worth taking note of the essential simplicity of model M_h as a binomial
 4077 GLMM. This is a type of model that is widely applied throughout statistics using standard
 4078 methods of inference based either on integrated likelihood (Laird and Ware, 1982; Berger
 4079 et al., 1999), which we discuss in Chapt. 6, or standard Bayesian methods. However,
 4080 because N is not known, inference is somewhat more challenging. We address that here
 4081 using Bayesian analysis based on data augmentation. Although we use data augmentation
 4082 in the context of Bayesian methods here, we note that heterogeneity models formulated
 4083 under DA are easily analyzed by conventional likelihood methods as zero-inflated binomial
 4084 mixtures (Royle, 2006) and more traditional analysis of model M_h based on integrated
 4085 likelihood, without using data augmentation, has been considered by Coull and Agresti
 4086 (1999), Dorazio and Royle (2003), and others.

4087 As with model M_0 , we have the Bernoulli model for the zero-inflation variables: $z_i \sim$
 4088 Bernoulli(ψ) and the model of the observations expressed conditional on these latent

4089 variables z_i . For $z_i = 1$, we have a binomial model with individual-specific p_i :

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

4090 and otherwise $y_i | z_i = 0 \sim I(y = 0)$, i.e., a point mass at $y = 0$. Further, we prescribe a
4091 distribution for p_i . Here we assume

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

4092 For prior distributions we assume $p_0 = \text{logit}^{-1}(\mu) \sim \text{Uniform}(0, 1)$ and, for the standard
4093 deviation $\sigma \sim \text{Uniform}(0, B)$ for some large B . Another common default prior is to assume
4094 $\tau = 1/\sigma^2 \sim \text{Gamma}(1, 1)$, although we usually choose $\sigma \sim \text{Uniform}(0, B)$.

4095 4.4.2 Analysis of the Fort Drum data with model M_h

4096 Here we provide an analysis of the Fort Drum bear survey data using the logit-normal
4097 heterogeneity model, and we used data augmentation to produce a data set of $M = 700$
4098 individuals. We have so far mostly used **WinBUGS** but we are now transitioning to
4099 the use of **JAGS** run from within **R** using the useful packages **R2jags** or **rjags**. The
4100 function **jags** from the **R2jags** package runs essentially like the **bugs** function which we
4101 demonstrate here for setting up and running model M_h for the Fort Drum bear data:

```
4102 [...] get data as before ....]
4103
4104 > set.seed(2013)
4105
4106 > cat("
4107 model{
4108   p0 ~ dunif(0,1)           # prior distributions
4109   mup <- log(p0/(1-p0))
4110   sigmap ~ dunif(0,10)
4111   taup <- 1/(sigmap*sigmap)
4112   psi ~ dunif(0,1)
4113
4114   for(i in 1:(nind+nz)){
4115     z[i] ~ dbern(psi)        # zero inflation variables
4116     lp[i] ~ dnorm(mup,taup) # individual effect
4117     logit(p[i]) <- lp[i]
4118     mu[i] <- z[i]*p[i]
4119     y[i] ~ dbin(mu[i],K)    # observation model
4120   }
4121
4122   N<-sum(z[1:(nind+nz)])
4123 }
4124 ",file="modelMh.txt")
4125
4126 > data1 <- list(y=y, nz=nz, nind=nind, K=K)
4126 > params1 <- c('p0','sigmap','psi','N')
```

```

4127 > inits <- function(){ list(z=as.numeric(y>=1), psi=.6, p0=runif(1),
4128   sigmap=runif(1,.7,1.2),lp=rnorm(M,-2)) }
4129 > library(R2jags)
4130 > wbout <- jags(data1, inits, params1, model.file = "modelMh.txt", n.chains = 3,
4131   n.iter = 1010000, n.burnin = 10000, working.directory = getwd())

```

4132 We provide an **R** function `modelMhBUGS` in the package `scrbook` which will fit the
4133 model using either **JAGS** or **WinBUGS** as specified by the user. In addition, for fun,
4134 we construct our own MCMC algorithm using a Metropolis-within-Gibbs algorithm for
4135 model M_h in Chapt. 17, where we also develop MCMC algorithms for spatial capture-
4136 recapture models. Using `modelMhBUGS`, we ran 3 chains of 1 *million* iterations (mixing is
4137 poor for this model and this data set), which produced the posterior distribution for N
4138 shown in Fig. 4.4. Posterior summaries of parameters are given in Table 4.5.

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

Parameter	Mean	SD	2.5%	50 %	97.5%	Rhat	n.eff
p_0	0.072	0.056	0.002	0.060	0.203	1.008	540
σ_p	2.096	0.557	1.215	2.025	3.373	1.003	820
ψ	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

4139 We used $M = 700$ for this analysis and we note that while the posterior mass of N is
4140 concentrated away from this upper bound (Fig. 4.4), the posterior has an extremely long
4141 right tail, with some MCMC draws at the upper boundary $N = 700$, suggesting that an
4142 even higher value of M may be called for. To characterize the posterior distribution of
4143 density we produce the relevant summaries of the posterior distribution of $D = N/277.11$
4144 (recall the buffered area of the convex hull is 277.11 km^2):

```

4145 > summary(wbout$sims.list$N/277.11)
4146   Min. 1st Qu. Median Mean 3rd Qu. Max.
4147 0.1696 0.2959 0.3681 0.4428 0.4944 2.5260
4148
4149 > quantile(wbout$sims.list$N/277.11,c(0.025,0.50,0.975))
4150   2.5%      50%    97.5%
4151 0.2237379 0.3680849 1.1511674

```

4152 Therefore, the point estimate, characterized by the posterior median, is around 0.37 bears
4153 per square km and a 95% Bayesian credible interval is (0.224, 1.151).

4.4.3 Comparison with MLE

4155 The posterior of N is highly skewed; therefore, we see that the posterior mean ($N = 122.7$)
4156 is considerably higher than the posterior median ($N = 102$). Further, it may be surprising
4157 that these posterior summaries do not compare well with the MLE. We used the **R** code

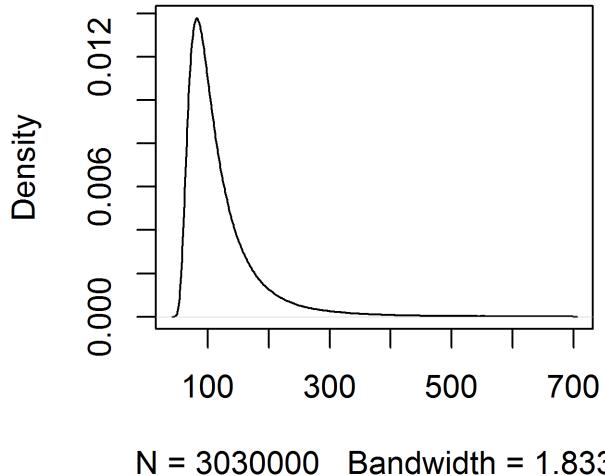


Figure 4.4. Posterior of N for Fort Drum bear study data under the logit-normal version of model M_h .

4158 contained in Panel 6.1 from Royle and Dorazio (2008) to obtain the MLE of $\log(n_0)$,
 4159 the logarithm of the number of uncaptured individuals, is $\widehat{\log(n_0)} = 3.86$ and therefore
 4160 $\hat{N} = \exp(3.86) + 47 = 94.47$, which is larger than the mode shown in Fig. 4.4. To see
 4161 this, we compute the posterior mode, by finding the posterior value of N with the highest
 4162 mass. Because N is discrete, we can use the `table()` function in **R** and find the most
 4163 frequent value⁴. If we want to smooth out some of the Monte Carlo error a bit, we can
 4164 use a smoother of some sort applied to the tabled posterior frequencies of N . Here we use
 4165 a smoothing spline (**R** function `smooth.spline`) with the degree of smoothing chosen by
 4166 cross-validation (the `cv=TRUE` argument):

```
4167 > N <- table(jout$BUGSoutput$sims.list$N)
4168 > xg <- as.numeric(names(N))
4169
4170 > sp <- smooth.spline(xg,N,cv=TRUE)
4171
4172 > sp
```

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

```

4173
4174 Call:
4175 smooth.spline(x = xg, y = N, cv = TRUE)
4176
4177 Smoothing Parameter spar= 0.09339815 lambda= 8.201724e-09 (17 iterations)
4178 Equivalent Degrees of Freedom (Df): 121.1825
4179 Penalized Criterion: 2544481
4180 PRESS: 5903.4

```

4181 We obtain the mode of the smoothed frequencies as follows:

```

4182 sp$x[sp$y==max(sp$y)]
4183 [1] 82

```

4184 We don't dwell too much on the difference between the MLE and features of the posterior, but we do note here that the posterior distribution for the parameters of this model, for the Fort Drum data set, are very sensitive to the prior distributions. In the present case, the use of a Uniform(0, 1) prior for $p_0 = \text{logit}^{-1}(\mu)$ is somewhat informative—in particular, it is not at all “flat” on the scale of μ , and this affects the posterior. We generally always recommend use of a Uniform(0, 1) prior for $\text{logit}^{-1}(\mu)$ in such models. That said, we were surprised at this result, and we experimented with other prior configurations including putting a flat prior on μ directly. This kind of small sample instability has been widely noted in model M_h (Fienberg et al., 1999; Dorazio and Royle, 2003), as has extreme sensitivity to the specific form of model M_h (Link, 2003). In summary, while the mode is well-defined, the data set is relatively sparse and hence inferences are poor and sensitive to model choice.

4.5 INDIVIDUAL COVARIATE MODELS: TOWARD SPATIAL CAPTURE-RECAPTURE

4196 A standard situation in capture-recapture models is when a covariate which is thought
 4197 to influence encounter probability is measured for each individual. These are often called
 4198 “individual covariate models” but, in keeping with the classical nomenclature on closed
 4199 population models, Kéry and Schaub (2012) referred to this class of models as “model
 4200 M_x ” (the x here being an explicit covariate). As with other closed population models, we
 4201 begin with the basic binomial observation model:

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

4202 To model the covariate, we use a logit model for encounter probability of the form:

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

4203 where x_i is the covariate value for individual i and the parameters $\boldsymbol{\alpha} = (\alpha_0, \alpha_1)$ are the
 4204 regression coefficients. Classical examples of covariates influencing detection probability
 4205 are type of animal (juvenile/adult or male/female), a continuous covariate such as body
 4206 mass, or a discrete covariate such as group or cluster size. For example, in models of aerial
 4207 survey data, it is natural to model the detection probability of a group as a function of the
 4208 observation-level individual covariate, “group size” (Royle, 2008; Langtimm et al., 2011).

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

Traditionally, estimation of N in model M_x is achieved using methods based on ideas of unequal probability sampling (i.e., Horvitz-Thompson estimation⁵; Huggins (1989), Alho (1990) and Borchers et al. (2002)). An estimator of N is

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

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

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

where p_i is a function of parameters α_0 and α_1 according to Eq. 4.5.1. In practice, parameters are estimated from the conditional-likelihood of the observed encounter histories which is, for observation y_i ,

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

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

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

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

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

⁵For a quick summary of the idea see:
http://en.wikipedia.org/wiki/Horvitz-Thompson_estimator

4242 form, for each $i = 1, 2, \dots, M$:

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

4243 Fully spatial capture-recapture models use this formulation with a latent covariate that
 4244 is directly related to the individual detection probability (see next section). As with
 4245 the previous models, implementation is trivial in the **BUGS** language. The **BUGS**
 4246 specification is very similar to that for model M_h , but we require the distribution of the
 4247 covariate to be specified, along with priors for the parameters of that distribution.

4248 **4.5.1 Example: Location of capture as a covariate**

4249 Here we consider a special type of model M_x that is especially relevant to spatial capture-
 4250 recapture. Intuitively, some measure of distance from home range center to traps for an
 4251 individual should be a reasonable covariate to explain heterogeneity in encounter probabilities
 4252 and vice versa. So we can imagine *estimating* such a quantity, say average distance from
 4253 home range center to “the trap array”, and then using it as an individual covariate in
 4254 capture-recapture models. A version of this idea was put forth by Boulanger and McLellan
 4255 (2001) (see also Ivan (2012)), but using the Huggins-Alho estimator and with covariate
 4256 “distance from home range center to edge” of the trapping array, where the home range
 4257 center is estimated by the average capture location. This is intuitively appealing because
 4258 we can imagine, in some kind of an ideal situation where we have a dense grid of traps
 4259 over some geographic region, that the average location of capture would be a decent esti-
 4260 mate (heuristically) of an individual’s home range center. We provide an example of this
 4261 type of approach using a fully model-based analysis of the version of model M_x described
 4262 above, analyzed by data augmentation. We take a slightly different approach than that
 4263 adopted by Boulanger and McLellan (2001). By analyzing the full likelihood and placing
 4264 a prior distribution on the individual covariate, we will resolve the problem of having an
 4265 ill-defined sample area. After you read later chapters of this book, it will be apparent that
 4266 SCR models represent a formalization of this heuristic procedure.

4267 For our purposes here, we define the scalar individual covariate x_i to be the distance
 4268 from the average encounter location of individual i , say \mathbf{s}_i , to the centroid of the trap
 4269 array, \mathbf{x}_0 : $x_i = \|\mathbf{s}_i - \mathbf{x}_0\|$. Note that $\|\mathbf{u}\|$ is standard notation for Euclidean norm or
 4270 magnitude of the vector \mathbf{u} , and we use it throughout the book. In practice, people have
 4271 used distance from edge of the trap array but that is less easy to quantify, as “edge” itself
 4272 is not precisely defined. Conceptually, individuals in the middle of the array should have
 4273 a higher probability of encounter and, as x_i increases, p_i should therefore decrease. We
 4274 note that we have defined \mathbf{s}_i in terms of a sample quantity—the observed mean encounter
 4275 location—which, while ad hoc, is consistent with the use of individual covariate models in
 4276 the literature. For an expansive, dense trapping grid we might expect the sample mean
 4277 encounter location to be a good estimate of home range center but, clearly this is biased
 4278 for individuals that live around the edge (or off) the trapping array.

4280 A key point is that s_i is missing for each individual that is not encountered and so
 4281 x_i is also missing. Therefore, it is a latent variable, and we need to specify a probability
 4282 distribution for it. As a measurement of distance we know it must be positive-valued, and
 4283 it seems sensible that an individual located extremely far from the array of traps would
 4284 not be captured. Therefore, let's assume that x_i is uniformly distributed from 0 to some
 4285 large number, say B , beyond which it would be difficult to imagine an individual being
 4286 captured by the trap array:

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

4287 where B is a specified constant, which we may choose to be arbitrarily large. For example,
 4288 B should be at least a home range diameter past the furthest trap from the centroid of
 4289 the array.

4290 4.5.2 Fort Drum bear study

4291 We have to do a little bit of data processing to fit this individual covariate model to the
 4292 Fort Drum data. We need to compute the individual covariate \mathbf{x}_i (distance from the
 4293 centroid of the trapping array) using the **R** function `spiderplot` provided in `scrbook`.
 4294 This function also produces the keen plot shown in Fig. 4.5 which we call a “spider plot”.
 4295 The **R** commands for obtaining the individual covariate “distance from trap centroid”
 4296 (the variable `xcent` returned by `spiderplot`) and making the spider plot are as follows:

```
4297 > library(scrbook)
4298 > data(beardata)
4299 > toad <- spiderplot(beardata$bearArray,beardata$trapmat)
4300 > xcent <- toad$xcent
```

4301 For the analysis of these data using the individual covariate “distance from centroid”
 4302 we used $x_i \sim \text{Uniform}(0, B)$ with $B = 11.5 \text{ km}^2$, which is about the distance from the
 4303 array center to the furthest trap. Once we choose a value for B , the direct implication is
 4304 that the population size parameter, N , applies to the area within 11.5 units of the trap
 4305 centroid. Therefore, the model associates a precise area within which the population of N
 4306 individuals resides. We will see shortly that N does, in fact, scale with our choice of B to
 4307 reflect the changing area over which the N individuals of the model reside. The **BUGS**
 4308 model specification and **R** commands to package the data and fit the model are as follows:

```
4309 cat("
4310 model{
4311   p0 ~ dunif(0,1)                                # Prior distributions
4312   alpha0 <- log(p0/(1-p0))
4313   psi ~ dunif(0,1)
4314   beta ~ dnorm(0,.01)

4315   for(i in 1:(nind+nz)){
4316     xcent[i] ~ dunif(0,B)
4317     z[i] ~ dbern(psi)                            # DA variables
4318     lp[i] <- alpha0 + beta*xcent[i] # Individual effect
4319     logit(p[i]) <- lp[i]
```

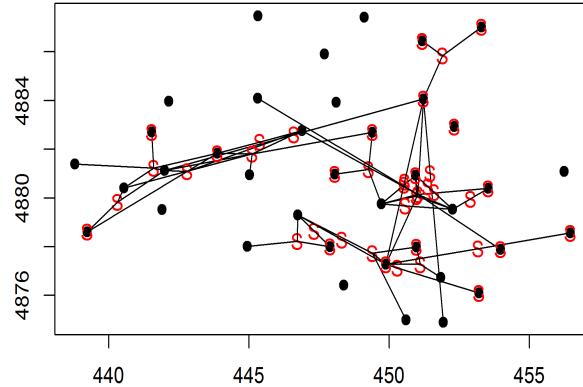


Figure 4.5. Spider plot of the Fort Drum study data. The black dots represent the 47 trap locations with the "S" symbols being the average capture location of each bear. i.e., its estimated home range center. All traps in which a bear was captured are connected to its estimated home range center with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

```

4321   mu[i] <- z[i]*p[i]
4322   y[i] ~ dbin(mu[i],K)           # Observation model
4323 }
4324
4325 N <- sum(z[1:(nind+nz)])
4326 }
4327 ",file='modelMcov.txt')"

4328 data2 <- list(y=y,nz=nz, nind=nind, K=K, xcent=xcent,B=11.5)
4329 params2 <- c('p0','psi','N','beta')
4330 inits <- function() {list(z=zst, psi=psi, p0=rnorm(1), beta=rnorm(1) ) }
4331 fit2 <- bugs(data2, inits, params2, model.file="modelMcov.txt",
4332                   n.chains=3, n.iter=11000, n.burnin=1000, n.thin=1)

```

4333 This produces the posterior summary statistics in Table 4.6.
 4334 We note that the estimated N is much lower than obtained by model M_h but there
 4335 is a good explanation for this which we discuss in the next section. That issue notwithstanding,
 4336 it is worth pondering how this model could be an improvement (conceptually or technically)
 4337 over some other model/estimator including M_0 and M_h considered previously. Well, for one, we have accounted formally for heterogeneity due to spatial location

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

Parameter	Mean	SD	2.5%	50 %	97.5%	Rhat	n.eff
p_0	0.54	0.07	0.40	0.54	0.67	1	1100
ψ	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
β	-0.25	0.06	-0.36	-0.25	-0.12	1	780

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

4.5.3 Extension of the model

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

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

```
4368 > u <- runif(10000,-1,1)
4369 > v <- runif(10000,-1,1)
4370 > d <- sqrt(u*u+v*v)
```

```

4371 > hist(d[d<1])
4372 > hist(d[d<1],100)
4373 > hist(d[d<1],100,probability=TRUE)
4374 > abline(0,2)

```

4375 It would be useful if we could describe this distribution directly in **BUGS** but there
 4376 is not a built-in way to do so. However, we can implement a discrete version of the pdf⁶.
 4377 To do this, we break B into L distance classes of width δ , with probabilities proportional
 4378 to $2 * x$. In particular, if we denote the cut-points by $g_1 = 0, g_2, \dots, g_{L+1} = B$ and the
 4379 interval midpoints are $m_i = g_{i+1} - \delta$. Then the interval probabilities are, approximately⁷,
 4380 $p_i = \delta(2m_i/B^2)$, which we can compute once and then pass them to **BUGS** as data. The
 4381 **R** commands for doing all of this (noting that we have already loaded and processed the
 4382 Fort Drum bear data) are given in the following **R/BUGS** script:

```

4383 > delta <- .2
4384 > xbin <- xcent%/%delta + 1                      # Put x in bins
4385 > midpts <- seq(delta,Dmax,delta)
4386 > xprobs <- delta*(2*midpts/(B*B))
4387 > xprobs <- xprobs/sum(xprobs)

4388
4389 > cat("
4390 model{
4391 p0 ~ dunif(0,1)                                # Prior distributions
4392 alpha0 <- log(p0/(1-p0))
4393 psi ~ dunif(0,1)
4394 beta ~ dnorm(0,.01)

4395 for(i in 1:(nind+nz)){
4396   xbin[i] ~ dcat(xprobs[])
4397   z[i] ~ dbern(psi)                               # DA variables
4398   lp[i] <- alpha0 + beta*xbin[i]*delta          # Individual covariate model
4399   logit(p[i]) <- lp[i]
4400   mu[i] <- z[i]*p[i]
4401   y[i] ~ dbin(mu[i],K)                          # Observation model
4402 }
4403

4404 N <- sum(z[1:(nind+nz)])                      # N is derived
4405 }
4406 ",file="modelMcov.txt")

```

4408 In the model description, the variable x (observed distance from centroid of the trap
 4409 array) has been rounded or binned (placed into a distance bin) so that the discrete version
 4410 of the pdf of x can be used, as described previously. The new variable labeled **xbin** is
 4411 then the *integer category* in units of δ from 0. Thus, to convert back to distance in the

⁶We 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.

⁷This is just length \times width, the area of small rectangles approximating the integral.

4412 expression for `lp[i]`, `xbin[i]` has to be multiplied by δ . To fit the model, keeping in
 4413 mind that the data objects required below have been defined in previous analyses of this
 4414 chapter, we do this:

```
4415 > data2 <- list(y=y, nz=nz, nind=nind, K=K, xbin=xbin, xprobs=xprobs,  

4416   delta=delta)  

4417 > params2 <- c('p0','psi','N','beta')  

4418 > inits <- function() {list(z=z, psi=psi, p0=runif(1),beta=rnorm(1) ) }  

4419 > fit <- bugs(data2, inits, params2, model.file="modelMcov.txt",  

4420   working.directory=getwd(), debug=FALSE, n.chains=3,  

4421   n.iter=11000, n.burnin=1000, n.thin=2)
```

4422 By specification of B , this model induces a clear definition of area in which the popu-
 4423 lation of N individuals reside. The parameter N of the model is the population size that
 4424 applies to the particular value of B and, as such, we will see that N scales with our choice
 4425 of B . This might be disconcerting to some—we can get whatever value of N we want
 4426 by changing B ! However, it is intuitively reasonable that, as we increase the area under
 4427 consideration, there should be more individuals in it. Fortunately, we find empirically,
 4428 that while N is highly sensitive to the prescribed value of B , density appears invariant to
 4429 B as long as B is sufficiently large. We fit the model for a set of values of B from $B = 12$
 4430 (restricting values of x to be in close proximity to the trap array) on up to 20. The results
 4431 are given in Table 4.7.

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

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

4432 We see that the posterior mean and SD of density (individuals per square km) appear
 4433 insensitive to choice of B once we reach about $B = 17$ or so. The estimated density of
 4434 0.25 per km² is actually quite a bit lower than we reported using model M_h for which no
 4435 relevant “area” quantity is explicit in the model (and so we had to make it up). Using
 4436 MLEs of N in conjunction with buffer strips (see Tab. 1.1) our estimates were in the
 4437 range of 0.32 – 0.43 and see Sec. 4.4 above. On the other hand our estimate of $\hat{D} = 0.25$
 4438 here (based on the posterior mean) is higher than that reported from model M_0 using
 4439 the buffered area ($\hat{D} = 0.18$). There is no basis really for comparing or contrasting
 4440 these various estimates. In particular, application of models M_0 and M_h are distinctly
 4441 *not* spatially explicit models—the area within which the population resides is not defined
 4442 under either model. There is therefore no reason at all to think that the estimates produced
 4443 under either closed population model, based on a buffered “trap area”, are justifiable by
 4444 any theory. In fact, we would get exactly the same estimate of N no matter what we declare

4445 the area to be. On the other hand, the individual covariate model uses an explicit model
 4446 for “distance from centroid” that is a reasonable and standard null model—it posits, in the
 4447 absence of direct information, that individual home range centers are randomly distributed
 4448 in space and that probability of detection depends on the distance between home range
 4449 center and the centroid of the trap array. Under this definition of the system, we see that
 4450 density is invariant to the choice of area, which seems like a desirable feature.

4451 **4.5.4 Invariance of density to B**

4452 Under model M_x , and also under models that we consider in later chapters, a general
 4453 property of the estimators is that while N increases with the prescribed area of the model
 4454 (defined by B in this model), we expect that density estimators should be invariant to this
 4455 area. In the model used above, we note that $\text{Area}(B) = \pi B^2$ and $\mathbb{E}(N(B)) = \lambda \text{Area}(B)$
 4456 and thus $\mathbb{E}(\text{Density}(B)) = \lambda$, i.e., constant. This should be interpreted as the *prior*
 4457 density. Absent data, then realizations under the model will have density λ regardless
 4458 of what B is prescribed to be. As we verified empirically above, posterior summaries of
 4459 density are also invariant to B as long as the prescribed area is sufficiently large.

4460 **4.5.5 Toward fully spatial capture-recapture models**

4461 While the use of an individual covariate model resolves two important problems inherent
 4462 in almost all capture-recapture studies (induced heterogeneity and absence of a precise
 4463 relationship between N and area), is not ideal for all purposes because it does not make
 4464 full use of the spatial information in the data set, i.e., the trap locations and the locations
 4465 of each individual encounter, so that we cannot use this model to model trap-specific
 4466 effects (e.g., trap effort or type). Moreover, we applied this model for “data” being the
 4467 average observed encounter location, and equated that summary to the home range center
 4468 s_i . Intuitively, taking the average encounter location as an estimate of home range center
 4469 makes sense but more so when the trapping grid is dense and expansive relative to typical
 4470 home range sizes which might not be reasonable in practice. Moreover, this approach
 4471 also ignored the variable precision with which each s_i is estimated. Finally, it ignores
 4472 that estimates of s_i around the “edge” (however we define that) are biased because the
 4473 observations are truncated—we can only observe locations interior to the array.

4474 However, there is hope to extend this model in order to resolve these remaining defi-
 4475 ciencies. In the next chapter we provide a further extension of this individual covariate
 4476 model that definitively resolves the *ad hoc* nature of the approach we took here. In that
 4477 chapter we build a model in which s_i are regarded as latent variables and the observation
 4478 locations (i.e., trap specific encounters) are linked to those latent variables with an explicit
 4479 model. We note that the model fitted previously could be adapted easily to deal with s_i
 4480 as a latent variable, simply by adding a prior distribution for s_i . This is actually easier,
 4481 and less ad hoc in a number of respects, and you should try it out.

4.6 DISTANCE SAMPLING: A PRIMITIVE SCR MODEL

4482 Distance sampling is a class of methods for estimating animal density from measurements
 4483 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 α_0 , is identifiable and such models are usually called "capture-recapture distance sampling" (Alpizar-Jara and Pollock, 1996; Borchers et al., 1998).

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

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

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

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

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

along with

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

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

```

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

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

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

```

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

4524 Basic math can be used to argue that if individuals have a uniform distribution in space,
 4525 then the distribution of Euclidean distance is also uniform. In particular, if a transect of
 4526 length L is used and x is distance to the transect then $F(x) = \Pr(X \leq x) = L*x/L*B =$
 4527 x/B and $f(x) = dF/dx = (1/B)$. For measurements of radial distance, we provided the
 4528 analogous argument in the previous section.

4529 The preceding paragraph makes it clear that distance sampling is a special case of
 4530 spatial capture-recapture models, such as those derived from model M_x of the previous
 4531 section, where the encounter probability is related directly to *distance*, which is a reduced
 4532 information summary of *location*, \mathbf{u} . Some intermediate forms of SCR/DS models can
 4533 be described (Royle et al., 2011a). In the context of our general characterization of SCR
 4534 models (Chapt. 2.6), we suggested that every SCR model can be described, conceptually,
 4535 by a hierarchical model of the form:

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

4536 Distance sampling ignores the part of the model pertaining to \mathbf{s} , and deals only with the
 4537 model components for the observed data \mathbf{u} ⁸. Thus, we are left with a hierarchical model
 4538 of the form

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

4539 In contrast, as we will see in the next chapters, many SCR models (Chapt. 5) ignore \mathbf{u}
 4540 and condition on \mathbf{s} , which is not observed:

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

4541 Since $[\mathbf{u}]$ and $[\mathbf{s}]$ are both assumed to be uniformly distributed, these are equivalent models!
 4542 The main differences have to do with interpretation of model components and whether or
 4543 not the latent variables are observable (in distance sampling they are).

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

4544 So why bother with SCR models when distance sampling yields density estimates and
4545 accounts for spatial heterogeneity in detection? For one, imagine trying to collect distance
4546 sampling data on species such as jaguars or tigers! Clearly, distance sampling requires
4547 that one can collect large quantities of distance data, which is not always possible. For
4548 tigers, it is much easier, efficient, and safer to employ camera traps or track plates and
4549 then apply SCR models. Furthermore, as we will see in Chapt. 15, SCR models can make
4550 use of distance data, allowing us to study distribution, movement, and density. Thus,
4551 SCR models are more general and versatile than distance sampling models (which clearly
4552 are a special case), and can accommodate data from virtually all animal survey designs.

4553 **4.6.1 Example: Sonoran desert tortoise study**

4554 We illustrate the application of distance sampling models using data on the Sonoran desert
4555 tortoise (*Gopherus agassizii*), shown in Fig. 4.6, collected along transects in southern
4556 Arizona (see Zylstra et al. (2010) for details). The data are from 120 square transects
4557 having four 250-m sides, although we ignore this detail in our analysis here and regard
4558 them as 1 km transects, and we pooled the detection data from all 120 transects. The
4559 histogram of encounter distances from the 65 encountered individuals is shown in Fig. 4.7



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

4560
4561 Commands for reading in and organizing the data for analysis using **WinBUGS** are
4562 given in the help file `?tortoise` provided with the `scrbook` package. To compute density,

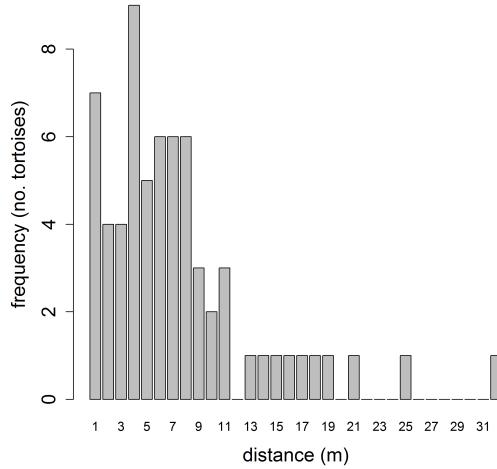


Figure 4.7. Distance histogram of $n = 65$ Sonoran desert tortoise detections from a total of 120 km of survey transect.

4563 the total sampled area of the transects `striparea` is input as data, and computed as:
 4564 120 (transects) multiplied by the length (1000 m) and half-width ($B = 40$ m), then
 4565 multiplied by 2, and divided by 10000 to convert to units of individuals per ha. We also
 4566 provide commands for analyzing the data with `unmarked` (Fiske and Chandler, 2011) using
 4567 hierarchical distance sampling models (Royle et al., 2004).

4568 Posterior summaries for the tortoise data are given in Tab. 4.8. Estimated density
 4569 (posterior mean) is 0.54 individuals per ha and the estimated scale parameter of the
 4570 distance function (posterior mean) is $\sigma = 9.12$ meters. The R-hat statistics of around 1.02
 4571 suggest that slightly longer MCMC simulations might be called for. The posterior mass
 4572 of the data augmentation parameter ψ is located away from the upper bound $\psi = 1$ and
 4573 so the degree of data augmentation appears sufficient.

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

Parameter	Mean	SD	2.5%	50 %	97.5%	Rhat	n.eff
N	516.67	54.71	415.00	516.00	632.00	1.02	100
D	0.54	0.06	0.43	0.54	0.66	1.02	100
α_1	0.01	0.00	0.00	0.01	0.01	1.02	130
σ	9.12	0.77	7.77	9.07	10.77	1.02	130
ψ	0.61	0.07	0.49	0.61	0.75	1.02	96

4.7 SUMMARY AND OUTLOOK

4574 Traditional closed population capture-recapture models are closely related to binomial
4575 generalized linear models. Indeed, the only real distinction is that in capture-recapture
4576 models, the population size parameter N (corresponding also to the size of a hypothetical
4577 “complete” data set) is unknown. This requires special consideration in the analysis of
4578 capture-recapture models. The classical approach to inference recognizes that the observa-
4579 tions don’t have a standard binomial distribution but, rather, a truncated binomial (from
4580 which which the so-called *conditional likelihood* derives) since we only have encounter fre-
4581 quency data on observed individuals. If instead we analyze the models using data augmen-
4582 tation, which arises under a $\text{Uniform}(0, M)$ prior for N , the observations can be modeled
4583 using a zero-inflated binomial distribution. When we deal with the unknown- N problem
4584 using data augmentation then we are left with zero-inflated GLMs and GLMMs instead
4585 of ordinary GLMs or GLMMs. The analysis of such zero-inflated models is practically
4586 convenient, especially using the **BUGS** variants.

4587 Spatial capture-recapture models that we will consider in the rest of the chapters
4588 of this book are closely related to individual covariate models (model M_x). Naturally,
4589 spatial capture-recapture models arise by defining individual covariates based on observed
4590 locations of individuals—we can think of using some function of mean encounter location as
4591 an individual covariate. We did this in a novel way, by using distance to the centroid of the
4592 trapping array as a covariate. We analyzed the *full likelihood* using data augmentation,
4593 and placed a prior distribution on the individual covariate which was derived from an
4594 assumption that individual locations are, *a priori*, uniformly distributed in space. This
4595 assumption provides for invariance of the density estimator to the choice of population
4596 size area (induced by maximum distance from the centroid of the trap array). The model
4597 addressed some important problems in the use of closed population models: it allows for
4598 heterogeneity in encounter probability due to the spatial juxtaposition of individuals with
4599 the array of traps, and it also provides a direct estimate of density because area is a
4600 feature of the model (via the prior on the individual covariate). The model is still not
4601 completely general, however, because it does not make full use of the spatial encounter
4602 histories, which provide direct information about the locations and density of individuals.

4603 A specific individual covariate model that is in widespread use is classical distance
4604 sampling. The model underlying distance sampling is precisely a special kind of SCR
4605 model—but one without replicate samples. Understanding distance sampling and individ-
4606 ual covariate models more broadly provides a solid basis for understanding and analyzing
4607 spatial capture-recapture models. In fact if, instead of placing an explicit model on *dis-*
4608 *tance* in the classical distance sampling model, we were to place the prior distribution on
4609 *location*, s , of each individual, then the form of the distance sampling model more closely
4610 resembles the SCR model we introduce in the next chapter.

4611

Part II

4612

4613

Basic SCR Models

5

FULLY SPATIAL CAPTURE-RECAPTURE MODELS

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

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

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

4645 considered in this chapter is the formulation of a model for encounter probability that is
 4646 a function of distance between individual home range center and trap locations. We also
 4647 discuss how encounter probability models are related to explicit models of space usage
 4648 or “home range area.” Understanding this allows us to compute, for example, the area
 4649 used by an individual during some prescribed time. While it is intuitive that SCR models
 4650 should be related to some model of space usage, this has not been discussed much in the
 4651 literature (but see Royle et al. (2012a) which we address further in Chapt. 13).

5.1 SAMPLING DESIGN AND DATA STRUCTURE

4652 In our development here, we will assume a standard sampling design in which an array
 4653 of J traps is operated for K sample occasions (say, nights) producing encounters of n
 4654 individuals. Because sampling occurs by traps and also over time, the most general data
 4655 structure yields temporally *and* spatially indexed encounter histories for *each individual*.
 4656 Thus a typical data set will include an encounter history *matrix* for each individual indicating
 4657 which trap the individual was captured, during each sample occasion. For example,
 4658 suppose we sample at 4 traps over 3 nights. A plausible data set for a single individual
 4659 captured one time in trap 1 on the first night and one time in trap 3 on the 3rd night is:

```
4660      night1 night2 night3
4661 trap1    1    0    0
4662 trap2    0    0    0
4663 trap3    0    0    1
4664 trap4    0    0    0
```

4665 This data structure would be obtained for *each* of the $i = 1, 2, \dots, n$ captured individuals.

4666 We develop models in this chapter for passive detection devices such as “hair snares” or
 4667 other DNA sampling methods (Kéry et al., 2010; Gardner et al., 2010b) and related types of
 4668 sampling devices in which (i) devices (“traps”) may capture any number of individuals (i.e.,
 4669 they don’t fill up); (ii) an individual may be captured in more than one trap during each
 4670 occasion but (iii) individuals can be encountered at most 1 time by each trap during any
 4671 occasion. Hair snares for sampling DNA from bears and other species function according
 4672 to these rules. An individual bear wandering about its territory might come into contact
 4673 with > 1 devices; a device may encounter multiple bears; however, in practice, it will
 4674 often not be possible to attribute multiple visits of the same individual during a single
 4675 occasion (e.g., night) to distinct encounter events. Thus, an individual may be captured
 4676 at most 1 time in each trap during any occasion. While this model, which we refer to
 4677 as SCR0, is most directly relevant to hair snares and other DNA sampling methods for
 4678 which multiple detections of an individual are not distinguishable, we will also make use
 4679 of the model for data that arise from camera-trapping studies. In practice, with camera
 4680 trapping, individuals might be photographed several times in a night but it is common to
 4681 distill such data into a single binary encounter event for reasons discussed later in Chapt.
 4682 9.

4683 The statistical assumptions we make to build a model for these data are that individual
 4684 encounters within and among traps are independent, and this allows us to regard
 4685 individual- and trap-specific encounters as *independent* Bernoulli trials (see next section).
 4686 These basic (but admittedly at this point somewhat imprecise) assumptions define the

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

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

4687 basic spatial capture-recapture model, SCR0. We will make things more precise as we
 4688 develop a formal statistical definition of the model shortly.

5.2 THE BINOMIAL OBSERVATION MODEL

4689 We begin by considering the simple model in which there are no time-varying covariates
 4690 that influence encounter, there are no explicit individual-specific covariates, and there are
 4691 no covariates that influence density. In this case, we can aggregate the binary encounters
 4692 over the K sample occasions and record the total number of encounters out of K . We will
 4693 denote these individual- and trap-specific encounter frequencies by y_{ij} for $i = 1, 2, \dots, n$
 4694 captured individuals and $j = 1, 2, \dots, J$ traps. For example, suppose we observe 6 individuals
 4695 in sampling at 4 traps over 3 nights of sampling then a plausible data set is the 6×4
 4696 matrix of encounters (out of 3 sampling occasions) shown in Table 5.1. We assume that
 4697 y_{ij} are mutually independent outcomes of a binomial random variable which we express
 4698 as:

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

4699 This is the basic model underlying standard closed population models (Chapt. 4) except
 4700 that, in the present case, the encounter frequencies are individual- *and* trap-specific, and
 4701 encounter probability p_{ij} depends on both individual *and* trap.

4702 As we did in Sec. 4.5, we will make explicit the notion that p_{ij} is defined conditional
 4703 on *where* individual i lives. Naturally, we think about defining an individual home range
 4704 and then relating p_{ij} explicitly to a summary of its location relative to each trap. For
 4705 example, the centroid of the individuals home range, or its center of activity (Efford, 2004;
 4706 Borchers and Efford, 2008; Royle and Young, 2008). In what follows, we define \mathbf{s}_i , a two-
 4707 dimensional spatial coordinate, to be the home range or activity center of individual i .
 4708 Then, the SCR model postulates that encounter probability, p_{ij} , is a decreasing function
 4709 of distance between \mathbf{s}_i and the location of trap j , \mathbf{x}_j (also a two-dimensional spatial
 4710 coordinate). A standard model for modeling binomial counts is the logistic regression,
 4711 where we model the dependence of p_{ij} on distance according to:

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

4712 where, here, $\|\mathbf{x}_j - \mathbf{s}_i\|$ is the distance between \mathbf{s}_i and \mathbf{x}_j . We sometimes write $\|\mathbf{x}_j - \mathbf{s}_i\| =$
 4713 $\text{dist}(\mathbf{x}_j, \mathbf{s}_i) = d_{ij}$. Alternatively, a popular model is

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

4714 which is similar to the “half-normal” model in distance sampling, except with an intercept
 4715 $p_0 \leq 1$ which can be estimated in SCR studies. Because it is the kernel of a bivariate
 4716 normal, or Gaussian, probability density function for the random variable “individual
 4717 location” we will refer to it as the “(bivariate) normal” or “Gaussian” model although
 4718 the distance sampling term “half-normal” is widely used. In the context of 2-dimensional
 4719 space, the model is clearly interpretable as a primitive model of movement outcomes or
 4720 space usage (we discuss this in Sec. 5.4).

4721 There are a large number of standard detection models commonly used (see Chapt. 7).
 4722 All other standard models that relate encounter probability to \mathbf{s} will also have a parameter
 4723 that multiplies distance in some non-linear function. To be consistent with parameter
 4724 naming across models, we will sometimes parameterize any encounter probability model
 4725 so that the coefficient on distance (or distance squared) is α_1 . So, for the Gaussian model,
 4726 $\alpha_1 = 1/(2\sigma^2)$. A characteristic of the common parametric forms is they are monotone de-
 4727 creasing with distance, but vary in their characteristic behavior as they approach distance
 4728 = 0. We show the standard Gaussian, Gaussian hazard, negative exponential and logistic
 4729 models in Fig. 5.1. The negative exponential model has $p_{ij} = p_0 \exp(-\alpha_1 \|\mathbf{x}_j - \mathbf{s}_i\|)$ and
 4730 the Gaussian hazard model has $p_{ij} = 1 - \exp(-\lambda_0 k(\mathbf{x}_j, \mathbf{s}_i))$ where $k(\mathbf{x}_j, \mathbf{s}_i)$ is the Gaussian
 kernel. Whatever model we choose for encounter probability, we should always keep in

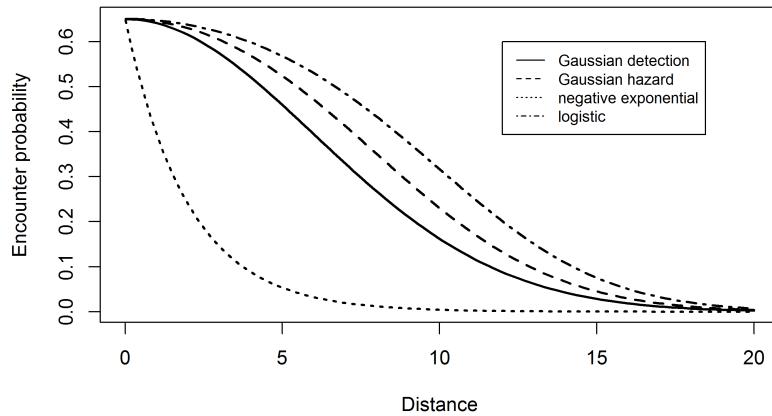


Figure 5.1. Some common encounter probability models showing the characteristic monotone decrease of encounter probability with distance between activity center and trap location.

4731
 4732 mind that the activity center for individual i , \mathbf{s}_i , is an unobserved random variable. To
 4733 be precise about this in the model, we should express the observation model as

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

4734 but sometimes, for notational simplicity, we abbreviate this by omitting some of the
 4735 arguments to p .

4736 **5.2.1 Definition of home range center**

4737 We define an individual's home range as *the area used by an organism during some time*
 4738 *period* which has a clear meaning for most species regardless of their biology. We therefore
 4739 define the home range center (or activity center) to be the center of the space that individ-
 4740 *ual was occupying (or using) during the period in which traps were active.* Thinking about
 4741 it in that way, it could even be observable (almost) as the centroid of a very large number
 4742 of radio fixes over the course of a survey period or a season. Thus, this practical version
 4743 of a home range center in terms of space usage is a well-defined construct regardless of
 4744 whether one thinks the home range itself is a meaningful concept. We use the terms home
 4745 range center and activity center interchangeably, and we recognize that this is a transient
 4746 thing which applies only to a well-defined period of study.

4747 **5.2.2 Distance as a latent variable**

4748 If we knew precisely every \mathbf{s}_i in the population (and population size N), then the model
 4749 specified by Eqs. 5.2.1 and 5.2.2 would be just an ordinary logistic regression-type of
 4750 a model (with covariate d_{ij}) which we learned how to fit using **WinBUGS** previously
 4751 (Chapt. 3). However, the activity centers are unobservable even in the best possible
 4752 circumstances. In that case, d_{ij} is an unobserved variable, analogous to the situation in
 4753 classical random effects models. We need to therefore extend the model to accommodate
 4754 these random variables with an additional model component – the random effects dis-
 4755 tribution. The customary assumption is the so-called “uniformity assumption,” which is
 4756 to assume that the \mathbf{s}_i are uniformly distributed over space (the obvious next question:
 4757 “which space?” is addressed below). This uniformity assumption amounts to a uniform
 4758 prior distribution on \mathbf{s}_i , i.e., the pdf of \mathbf{s}_i is constant, which we may express

$$\Pr(\mathbf{s}_i) \propto \text{constant} \quad (5.2.4)$$

4759 As it turns out, this assumption is usually not precise enough to fit SCR models in practice
 4760 for reasons we discuss shortly. We will give another way to represent this prior distribution
 4761 that is more concrete, but depends on specifying the “state-space” of the random variable
 4762 \mathbf{s}_i . The term state-space is a technical way of saying “the space of all possible outcomes”
 4763 of the random variable.

5.3 THE BINOMIAL POINT PROCESS MODEL

4764 In the SCR model, the individual activity centers are unobserved and thus we treat them
 4765 as random effects. Specifically, the collection of individual activity centers $\mathbf{s}_1, \dots, \mathbf{s}_N$
 4766 represents a realization of a *binomial point process* (Illian et al., 2008, p. 61). The
 4767 binomial point process (BPP) is analogous to a Poisson point process in the sense that it
 4768 represents a “random scatter” of points in space – except that the total number of points
 4769 is *fixed*, whereas, in a Poisson point process, it is random (having a Poisson distribution).

4770 As an example, we show in Fig. 5.2 locations of 20 individual activity centers (black
 4771 dots) in relation to a grid of 25 traps. For a Poisson point process the number of such
 4772 points in the prescribed state-space would be random whereas often we will simulate fixed
 4773 numbers of points, e.g., for evaluating the performance of procedures, e.g., how well does
 our estimator perform when $N = 50$?

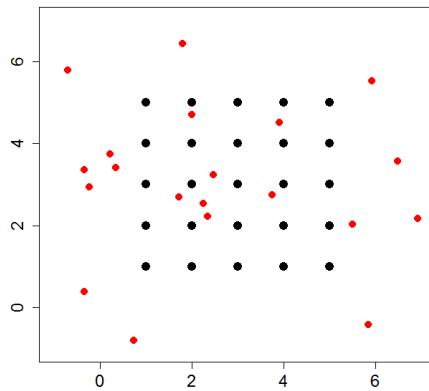


Figure 5.2. Realization (small dots) of a binomial point process with $N = 20$. The large dots represent trap locations.

4774
 4775 It is natural to consider a binomial point process in the context of capture-recapture
 4776 models because it preserves N in the model and thus preserves the linkage directly with
 4777 closed population models. In fact, under the binomial point process model, model M_0
 4778 and other closed models are simple limiting cases of SCR models, i.e., they arise as the
 4779 coefficient on distance (α_1 above) tends to 0.

4780 While we often will express SCR models “conditional-on- N ”, it will sometimes be
 4781 convenient to impose specific prior distributions on N . By assuming N has a binomial
 4782 distribution, we can make use of data augmentation, our preferred tool, for Bayesian
 4783 analysis of the models as in Chapt. 4, thus yielding a methodologically coherent approach
 4784 to analyzing the different classes of models. We might also assume that N has a Poisson
 4785 distribution in some cases (see Chapt. 14). Of course, the two assumptions are closely
 4786 related in the usual limiting sense.

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

one of spatial independence among individual activity centers because each activity center is distributed independently of each other activity center. Despite this independence we see in Fig. 5.2 that *realizations* of randomly distributed points will typically exhibit distinct non-uniformity. Thus, independent, uniformly distributed points will almost never appear regularly, uniformly or systematically distributed. For this reason, the basic binomial (or Poisson) point process models are enormously useful in practical settings since they allow for a range of distribution patterns without violating the assumption of spatial randomness. More relevant for SCR models is that we actually have a little bit of data for some individuals and thus the resulting posterior point pattern can deviate strongly from uniformity, a point we come back to repeatedly in this book. The uniformity hypothesis is only a *prior* distribution which is directly affected by the quantity and quality of the observed data, to produce a posterior distribution which may appear distinctly non-uniform. In addition, we can build more flexible models for the point process, which we take up in Chapt. 11.

5.3.1 The state-space of the point process

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

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

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

Prescribing the state-space

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

For our simulated system developed previously in this chapter, we defined the state-space to be a square within which our trap array was centered. For many practical

4835 situations this might be an acceptable approach to defining the state-space, i.e., just a
 4836 rectangle around the trap array. Although defining the state-space to be a regular polygon
 4837 has computational advantages (e.g., we can implement this more efficiently in **BUGS** and
 4838 cannot for irregular polygons), a regular polygon induces an apparent problem of admitting
 4839 into the state-space regions that are distinctly non-habitat (e.g., oceans, large lakes, ice
 4840 fields, etc.). It is difficult to describe complex regions in mathematical terms that can
 4841 be used in **BUGS**. As an alternative, we can provide a representation of the state-space
 4842 as a discrete set of points which the **R** package **secr** (Efford, 2011a) permits (**secr** uses
 4843 the term “mask” for what we call the state-space). Defining the state-space by a discrete
 4844 set of points is handy because it allows specific points to be deleted or not, depending on
 4845 whether they represent available or suitable habitat (see Sec. 5.10). We can also define
 4846 the state-space as an arbitrary collection of polygons stored as a GIS shapefile which can
 4847 be analyzed easily by MCMC in **R** (see Sec. 17.7), but not so easily in the **BUGS** engines.
 4848 In Sec. 5.10, we provide an analysis of the wolverine camera trapping data, in which we
 4849 define the state-space to be a regular continuous polygon (a rectangle).

4850 **Invariance to the state-space**

4851 We will assert for all models we consider in this book that density is invariant to the size
 4852 and extent of \mathcal{S} , if \mathcal{S} is sufficiently large, and as long as our model relating p_{ij} to \mathbf{s}_i is a
 4853 decreasing function of distance. We can prove this easily by drawing an analogy with a 1-d
 4854 case involving distance sampling. Let y_j be the number of individuals captured in some
 4855 interval $[d_{j-1}, d_j)$, and define $d_J = B$ for some large value of B . The observations from a
 4856 survey are y_1, \dots, y_J and the likelihood is a multinomial likelihood, so the log-likelihood
 4857 is of the form

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

4858 where π_j is the probability of detecting an individual in distance class j , which depends on
 4859 parameters of the detection function (the manner of which is not relevant for the present
 4860 discussion). Choosing B sufficiently large guarantees that $\mathbb{E}(y_J) = 0$ and therefore the
 4861 observed frequency in the “last cell” contributes nothing to the likelihood, in regular
 4862 situations in which the detection function decays monotonically with distance and prior
 4863 density is constant. We can think of B as being related to the state-space in an SCR
 4864 model, as the width of a rectangular state-space with area $B \times L$, L being the length
 4865 of the transect. Thus, if we choose B large enough, then we ensure that the expected
 4866 trap-frequencies beyond B will be 0, and thus contribute nothing to the likelihood.

4867 Sometimes our estimate of density can be affected by choosing \mathcal{S} too small. However,
 4868 this might be sensible if \mathcal{S} is naturally well-defined. As we discussed in Chapt. 1, \mathcal{S} is
 4869 *part of the model*, and thus it is sensible that estimates of density might be sensitive to
 4870 its definition in problems where it is natural to restrict \mathcal{S} . One could imagine, however,
 4871 in specific cases, e.g., a small population with well-defined habitat preferences, that a
 4872 problem could arise because changing the state-space based on differing opinions, and
 4873 GIS layers, might have substantial affects on the density estimate. But this is a real
 4874 biological problem, and a natural consequence of the spatial formalization of capture-
 4875 recapture models – a feature, not a bug or some statistical artifact – and it should be
 4876 resolved with better information, research, and thinking. For situations where there is not
 4877 a natural choice of \mathcal{S} , we should default to choosing \mathcal{S} to be very large in order to achieve

invariance or, otherwise, evaluate sensitivity of density estimates by trying a couple of different choices of \mathcal{S} . This is a standard “sensitivity to prior” argument that Bayesians always have to be conscious of. We demonstrate this in our analysis of Sec. 5.9 below. As an additional practical consideration, we note that the area of the state-space \mathcal{S} affects data augmentation. If you increase the size of \mathcal{S} , then there are more individuals to account for and therefore the size of the augmented data set M must increase. This has computational implications.

5.3.2 Connection to model M_h and distance sampling

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

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

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

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

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

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

It is also worth re-emphasizing that the basic SCR encounter model is a binomial encounter model in which distance is a covariate. As such, it is strikingly similar to classical distance sampling models (Buckland et al., 2001). Both have distance as a covariate but, in classical distance sampling problems, the focus is on the distance between the observer and the animal at an instant in time, not the distance between a trap and an animal’s home range center. As a practical matter, in distance sampling, “distance” is *observed* for those individuals that appear in the sample. Conversely, in SCR problems, it is only imperfectly observed (we have partial information in the form of trap observations). Clearly, it is preferable to observe distance if possible, but distance sampling requires field methods that are not practical in many situations, e.g. when studying carnivores such as

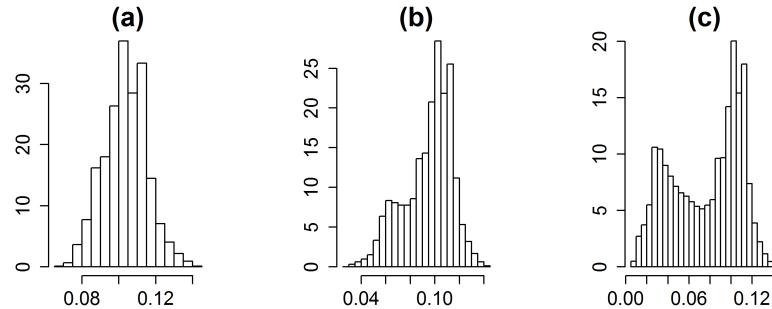


Figure 5.3. Implied distribution of π_i for a population of individuals as a function of the size of the state-space buffer around the trap array. The state-space buffer is 0.2, 0.5 and 1.0 for panels (a), (b), (c), respectively. In each case, the trap array is fixed and centered within a square state-space.

bears or large cats. Furthermore, SCR models allow us to relax many of the assumptions made in classical distance sampling, such as perfect detection at distance zero, and SCR models allow for estimates of quantities other than density, such as home range size, and space usage (see Chaps. 12 and 13).

5.4 THE IMPLIED MODEL OF SPACE USAGE

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

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

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

In practice it might make sense to think about the first component, i.e., $\Pr(\text{encounter}|\text{usage of } \mathbf{x}, \mathbf{s})$ as being a constant (e.g., if traps are located within arbitrarily small grid cells) and then, in that case, the encounter probability model is directly

4939 proportional to this model for individual movements about their home range center deter-
 4940 mining the use frequency of each \mathbf{x} . This is a sensible heuristic model for what ecologists
 4941 would call a central place forager although, as we have stated previously, it may be mean-
 4942 ingful as a description of transient space usage as well (that is, the space usage during the
 4943 period of sampling).

4944 To motivate a specific model for space usage, imagine the area we are interested in
 4945 consists of some large number of small pixels (i.e. we're looking at a discrete representation
 4946 of space), and that we have some kind of perfect observation device (e.g., continuous
 4947 telemetry) so that we observe every time an individual moves into a pixel. After a long
 4948 period of time, we observe an enormous sample size of \mathbf{x} values. We tally those up into
 4949 each pixel, producing the frequency $m(\mathbf{x}, \mathbf{s})$, which is something like the "true" usage of
 4950 pixel \mathbf{x} by individual with activity center \mathbf{s} . So, then, the usage model should be regarded
 4951 as a probability mass function for these counts and, naturally, we regard the counts $m(\mathbf{x}, \mathbf{s})$
 4952 as a multinomial observation with probabilities $\pi(\mathbf{x}|\mathbf{s})$, and prescribe a suitable model for
 4953 $\pi(\mathbf{x}|\mathbf{s})$ that describes how use events should accumulate in space. A natural null model
 4954 for $\pi(\mathbf{x}|\mathbf{s})$ has a decreasing probability of use as \mathbf{x} gets far away from \mathbf{s} ; i.e., animals spend
 4955 more time close to their activity centers than far away. We can regard points used by
 4956 the individual with activity center \mathbf{s} as the realization of a point process with conditional
 4957 intensity:

$$\pi(\mathbf{x}|\mathbf{s}) = \frac{k(\mathbf{x}, \mathbf{s})}{\sum_x k(\mathbf{x}, \mathbf{s})} \quad (5.4.1)$$

4958 where $k(\mathbf{x}, \mathbf{s})$ is any positive function. In continuous space, the equivalent representation
 4959 would be:

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

4960 Clearly the space used by an individual will be proportional to whatever kernel, $k(\mathbf{x}, \mathbf{s})$,
 4961 we plug-in here. If we use a negative exponential function, then this produces a standard
 4962 resource selection function (RSF) model (e.g., Manly et al., 2002, Chapt. 8). But, here
 4963 we use a Gaussian kernel, i.e.,

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

4964 so that contours of the probability of space usage resemble a bivariate normal or Gaussian
 4965 probability distribution function.

4966 To apply this model of space-usage to SCR problems we allow for imperfect detection
 4967 by introducing a non-uniform "thinning rate" of the true counts $m(\mathbf{x}, \mathbf{s})$. This yields,
 4968 precisely, our Gaussian encounter probability model where the thinning rate is our baseline
 4969 encounter probability p_0 for each pixel where we place a trap, and $p = 0$ in each pixel
 4970 where we don't place a trap.

4971 The main take-away point here is that underlying most SCR models is some kind of
 4972 model of space-usage, implied by the specific choice of $k(\mathbf{x}, \mathbf{s})$. Whether or not we have
 4973 perfect sampling devices, the function we use in the encounter probability model equates
 4974 to some conditional distribution of points, a utilization distribution, as in Eq. 5.4.1, from
 4975 which we can compute effective home range area, i.e., the area that contains some percent
 4976 of the mass of a probability distribution proportional to $k(\mathbf{x}, \mathbf{s})$; e.g., 95% of all space used
 4977 by an individual with activity center \mathbf{s} .

4978 **5.4.1 Bivariate normal case**

4979 One encounter model that allows direct analytic computation of home range area is the
 4980 Gaussian encounter probability model

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

4981 For this model, encounter probability is proportional to the kernel of a bivariate normal
 4982 (Gaussian) pdf and so the natural interpretation is that in which movement outcomes (or
 4983 successive locations of an individual) are draws from a bivariate normal distribution with
 4984 standard deviation σ . We say that use of this model implies a bivariate normal model of
 4985 space usage. Under this model we can compute precisely the effective home range area. In
 4986 particular, if use outcomes are bivariate normal, then $\|\mathbf{x} - \mathbf{s}\|^2$ has a chi-square distribution
 4987 with 2 d.f. and the quantity $B(\alpha)$ that encloses $(1 - \alpha)\%$ of all realized distances i.e.,
 4988 $\Pr(d \leq B(\alpha)) = 1 - \alpha$, is $B(\alpha) = \sigma * \sqrt{q(\alpha, 2)}$ where $q(\alpha, 2)$ is the 0.05 chi-square
 4989 critical value on 2 df. For example, to compute $q(.05, 2)$ in R we execute the command
 4990 `qchisq(.95, 2)` which is $q(2, \alpha) = 5.99$. Then, for $\sigma = 1$, $B(\alpha) = 1 * \sqrt{5.99} = 2.447$.
 4991 Therefore 95% of the points used will be within 2.447 (standard deviation) units of the
 4992 home range center. So, in practice, we can estimate σ by fitting the bivariate normal
 4993 encounter probability model to some SCR data, and then use the estimated σ to compute
 4994 the “95% radius”, say $r_{.95} = \sigma\sqrt{5.99}$, and convert this to the 95% use area – the area
 4995 around \mathbf{s} which contains 95% of the movement outcomes – according to $A_{.95} = \pi r_{.95}^2$.

4996 An alternative bivariate normal model is the bivariate normal hazard rate model:

$$p(\mathbf{x}, \mathbf{s}) = 1 - \exp(-\lambda_0 * \exp(-\frac{1}{2\sigma^2} \|\mathbf{x} - \mathbf{s}\|^2)) \quad (5.4.2)$$

4997 We use λ_0 here because this parameter, the baseline encounter rate, can be > 1 . This arises
 4998 by assuming the latent “use frequency” $m(\mathbf{x}, \mathbf{s})$ is a Poisson random variable with intensity
 4999 $\lambda_0 k(\mathbf{x}, \mathbf{s})$. The model is distinct from our Gaussian encounter model $p(\mathbf{x}, \mathbf{s}) = p_0 k(\mathbf{x}, \mathbf{s})$
 5000 used previously, although we find that they produce similar results in terms of estimates
 5001 of density or 95% use area, as long as baseline encounter probability is low. We discuss
 5002 these two formulations of the bivariate normal model further in Chapt. 9.

5003 **5.4.2 Empirical analysis**

5004 For any encounter model we can compute space usage quantiles empirically by taking a fine
 5005 grid of points and either simulating movement outcomes with probabilities proportional to
 5006 $p(\mathbf{x}, \mathbf{s})$ and accumulating area around \mathbf{s} , or else we can do this precisely by varying $B(\alpha)$
 5007 to find that value within which 95% of all movements are concentrated, i.e., the set of all
 5008 \mathbf{x} such that $\|\mathbf{x} - \mathbf{s}\| \leq B(q)$. Under any detection model, movement outcomes will occur
 5009 in proportion to $p(\mathbf{x}, \mathbf{s})$, as long as the probability of encounter is constant, *conditional on*
 5010 use, and so we can define our space usage distribution according to:

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

5011 Given the probabilities $\pi(\mathbf{x}, \mathbf{s})$ for all \mathbf{x} we can find the value of $B(q)$, for any q , such that

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

5012 (here, we use \ni to mean “such that”). We have a function called `hra` in the `scrbook`
 5013 package that computes the home range area for any encounter model and prescribed
 5014 parameter values. The help file for `hra` has an example of simulating some data. The
 5015 following commands illustrate this calculation for two different bivariate normal models
 5016 of space usage:

```

5017 ##  

5018 ## Define encounter probability model as R function  

5019 ##  

5020 > pGauss2 <- function(parms,Dmat){  

5021   a0 <- parms[1]  

5022   sigma <- parms[2]  

5023   lp <- parms[1] -(1/(2*parms[2]*parms[2]))*Dmat*Dmat  

5024   p <- 1-exp(-exp(lp))  

5025   p  

5026 }  

5027  

5028 > pGauss1 <- function(parms,Dmat){  

5029   a0 <- parms[1]  

5030   sigma <- parms[2]  

5031   p <- plogis(parms[1])*exp( -(1/(2*parms[2]*parms[2]))*Dmat*Dmat )  

5032   p  

5033 }  

5034  

5035 ##  

5036 ## Execute hra with sigma = .3993  

5037 ##  

5038 > hra(pGauss1,parms=c(-2,.3993),plot=FALSE,xlim=c(0,6),ylim=c(0,6),  

5039   ng=500,tol=.0005)  

5040  

5041 [1] 0.9784019  

5042 radius to achieve 95% of area: 0.9784019  

5043 home range area: 3.007353  

5044 [1] 3.007353  

5045  

5046  

5047 ## Analytic solution:  

5048 ##      true sigma that produces area of 3  

5049 > sqrt(3/pi)/sqrt(5.99)  

5050 [1] 0.3992751

```

5051 What this means is that $B(q) = 0.978$ is the radius that encloses about 95% of all
 5052 movements under the standard bivariate normal encounter model. Therefore, the area is
 5053 about $\pi * .978^2 = 3.007$ spatial units. You can change the intercept of the model and find
 5054 that it has no effect. The true (analytic) value of σ that produces a home range area of 3.0
 5055 is 0.3993 which is the value we initially plugged in to the `hra` function. We can improve
 5056 on the numerical approximation to home range area (get it closer to 3.0) by increasing the
 5057 resolution of our spatial grid (increase the `ng` argument) along with the `tol` argument.

5058 We can also reverse this process, and find, for any detection model, the parameter
 5059 values that produce a certain $(1 - q)\%$ home range area, which we imagine would be
 5060 useful for doing simulation studies. The function `hra` will compute the value of the scale
 5061 parameter that achieves a certain target $(1 - q)\%$ home range area, by simply providing a
 5062 non-null value of the variable `target.area`. Here we use `target.area = 3.00735` (from
 5063 above) to obtain a close approximation to the value σ we started with (the parameter
 5064 argument is meaningless here):

```
5065 > hra(pGauss1,parms=c(-2,.3993),plot=FALSE,xlim,ylim,ng=500,  

5066   target.area=3.00735,tol=.0005)  

5067  

5068 Value of parm[2] to achieve 95% home range area of 3.00735: 0.3993674
```

5069 **5.4.3 Relevance of understanding space usage**

5070 One important reason that we need to be able to deduce “home range area” from a
 5071 detection model is so that we can compare different models with respect to a common
 5072 biological currency. Many encounter probability models have some “scale parameter”,
 5073 which we might call σ no matter the model, but this relates to 95% area in a different
 5074 manner under each model. Therefore, we want to be able to convert different models
 5075 to the same currency. Another reason to understand the relationship between models of
 5076 encounter probability and space usage is that it opens the door to combining traditional
 5077 resource selection data from telemetry with spatial capture-recapture data. In Chapt. 13
 5078 we consider this problem, for the case in which a sample of individuals produces encounter
 5079 history data suitable for SCR models and, in addition, we have telemetry relocations on a
 5080 sample of individuals. This is achieved by regarding the two sources of data as resulting
 5081 from the same underlying process of space usage but telemetry data produce “perfect”
 5082 observations, like always-on camera traps blanketing a landscape. We use this idea to
 5083 model the effect of a measured covariate at each pixel, say $C(\mathbf{x})$, on home range size and
 5084 geometry and, hence, the probability of encounter in traps.

5085 **5.4.4 Contamination due to behavioral response**

5086 Interpretation of encounter probability models as models of animal home range and space
 5087 usage can be complicated by a number of factors, including whether traps are baited or
 5088 not. In the case of baited traps, this might lead to a behavioral response (Sec. 7.2.3)
 5089 which could affect animal space usage. For example, if traps attract animals from a long
 5090 distance, it could make typical home ranges appear larger than normal. More likely, in our
 5091 view, it wouldn’t change the typical size of a range but would change how individuals use
 5092 their range e.g., by moving from baited trap to baited trap, so that observed movement
 5093 distances of individuals are typically larger than normal.

5094 In other cases, the reliance on Euclidean distance in models for encounter probability
 5095 might be unrealistic, and can lead to biased estimates of density (Royle et al., 2013).
 5096 For example, animals might concentrate their movements along trails, roads, or other
 5097 landscape features. In this case, models that accommodate other distance metrics can be
 5098 considered. We present models based on least-cost path in Chapt. 12.

5.5 SIMULATING SCR DATA

5099 It is always useful to simulate data because it allows you to understand the system that
 5100 you're modeling and also calibrate your understanding with specific values of the model
 5101 parameters. That is, you can simulate data using different parameter values until you
 5102 obtain data that "look right" based on your knowledge of the specific situation that
 5103 you're interested in. Here we provide a simple script to illustrate how to simulate spatial
 5104 encounter history data. In this exercise we simulate data for 100 individuals and a 25 trap
 5105 array laid out in a 5×5 grid of unit spacing. The specific encounter model is the Gaussian
 5106 model given above and we used this code to simulate data used in subsequent analyses.
 5107 The 100 activity centers were simulated on a state-space defined by a 8×8 square within
 5108 which the trap array was centered (thus the trap array is buffered by 2 units). Therefore,
 5109 the density of individuals in this system is fixed at 100/64.

```

5110 > set.seed(2013)
5111 # Create 5 x 5 grid of trap locations with unit spacing
5112 > traplocs <- cbind(sort(rep(1:5,5)),rep(1:5,5))
5113 > ntraps <- nrow(traplocs)
5114 # Compute distance matrix:
5115 > Dmat <- e2dist(traplocs,traplocs)
5116
5117
5118 # Define state-space of point process. (i.e., where animals live).
5119 # "buffer" just adds a fixed buffer to the outer extent of the traps.
5120 #
5121 > buffer <- 2
5122 > xlim <- c(min(traplocs[,1] - buffer),max(traplocs[,1] + buffer))
5123 > ylim <- c(min(traplocs[,2] - buffer),max(traplocs[,2] + buffer))
5124
5125 > N <- 100    # population size
5126 > K <- 20    # number nights of effort
5127
5128 > sx <- runif(N,xlim[1],xlim[2])    # simulate activity centers
5129 > sy <- runif(N,ylim[1],ylim[2])
5130 > S <- cbind(sx,sy)
5131 # Compute distance matrix:
5132 > D <- e2dist(S,traplocs) # distance of each individual from each trap
5133
5134 > alpha0 <- -2.5      # define parameters of encounter probability
5135 > sigma <- 0.5        # scale parameter of half-normal
5136 > alpha1 <- 1/(2*sigma*sigma) # convert to coefficient on distance
5137
5138 # Compute Probability of encounter:
5139 #
5140 > probcap <- plogis(-2.5)*exp( - alpha1*D*D)
5141
5142 # Generate the encounters of every individual in every trap

```

```

5143 > Y <- matrix(NA,nrow=N,ncol=ntraps)
5144 > for(i in 1:nrow(Y)){
5145   Y[i,] <- rbinom(ntraps,K,probcap[i,])
5146 }

```

5147 We remind the reader that, in presenting **R** or other code snippets throughout the
5148 book, we will deviate from our standard variable expressions for some quantities. In
5149 particular, we sometimes substitute words for integer variable designations: **nind** (for n),
5150 **ntraps** (for J), and **nocc** (for K). In our opinion this leaves less to be inferred by the
5151 reader in trying to understand code snippets.

5152 Subsequently we will generate data using this code packaged in an **R** function called
5153 **simSCRO** in the package **scrbook** which takes a number of arguments including **discard0**
5154 which, if TRUE, will return only the encounter histories for captured individuals. A second
5155 argument is **array3d** which, if TRUE, returns the 3-dimensional encounter history array
5156 instead of the aggregated **nind** \times **ntraps** encounter frequencies (see below). Finally we
5157 provide a random number seed, **rnd** = 2013 to ensure repeatability of the analysis here.
5158 We obtain a data set as above using the following command:

```

5159 > data <- simSCRO(discard0=TRUE, array3d=FALSE, rnd=2013)

```

5160 The **R** object **data** is a list, so let's take a look at what's in the list and then harvest some
5161 of its elements for further analysis below.

```

5162 > names(data)
5163 [1] "Y"      "traplocs" "xlim"      "ylim"      "N"       "alpha0"    "beta"
5164 [8] "sigma"   "K"
5165
5166 ## Grab encounter histories from simulated data list
5167 > Y <- data$Y
5168 ## Grab the trap locations
5169 > traplocs <- data$traplocs

```

5170 5.5.1 Formatting and manipulating real data sets

5171 Conventional capture-recapture data are easily stored and manipulated as a 2-dimensional
5172 array, an **nind** \times **K** (individuals by sample occasions) matrix, which is maximally informative
5173 for any conventional capture-recapture model, but not for spatial capture-recapture
5174 models. For SCR models we must preserve the spatial information in the encounter history
5175 information. We will routinely analyze data from 3 standard formats:

- 5176 (1) The basic 2-dimensional data format, which is an **nind** \times **ntraps** encounter frequency
5177 matrix such as that simulated previously. These are the total number of encounters in
5178 each trap, summed over the K sample occasions.
- 5179 (2) The maximally informative 3-dimensional array, for which we establish here the con-
5180 convention that it has dimensions **nind** \times **ntraps** \times **K**.
- 5181 (3) We use a compact format – the “encounter data file” – which we describe below in
5182 Sec. 5.9.

5183 To simulate data in the most informative format - the “3-d array” - we can use the **R**
 5184 commands given previously but replace the last 4 lines with the following:

```
5185 > Y <- array(NA,dim=c(N,ntraps,K))
5186
5187 > for(i in 1:nrow(Y)){
5188   for(j in 1:ntraps){
5189     Y[i,j,1:K] <- rbinom(K,1,probcap[i,j])
5190   }
5191 }
```

5192 We see that a collection of K binary encounter events are generated for *each* individual
 5193 and for *each* trap. The probabilities of those Bernoulli trials are computed based on the
 5194 distance from each individual’s home range center and the trap (see calculation above),
 5195 and those are housed in the matrix `probcap`. Our data simulator function `simSRC0` will
 5196 return the full 3-d array if `array3d=TRUE` is specified in the function call. To recover the
 5197 2-d matrix from the 3-d array, and subset the 3-d array to individuals that were captured,
 5198 we do this:

```
5199 # Sum over the ‘‘sample occasions’’ dimension (3rd margin of the array)
5200 > Y2d <- apply(Y,c(1,2),sum)
5201
5202 # Compute how many times each individual was captured
5203 > ncaps <- apply(Y2d,1,sum)
5204
5205 # Keep those individuals that were captured
5206 > Y <- Y[ncaps>0,,]
```

5.6 FITTING MODEL SCR0 IN BUGS

5207 Clearly if we somehow knew the value of N then we could fit this model directly because,
 5208 in that case, it is a special kind of logistic regression model, one with a random effect (`s`)
 5209 that enters into the model in a peculiar fashion, and also with a distribution (uniform)
 5210 which we don’t usually think of as standard for random effects models. So our aim here is
 5211 to analyze the known- N problem, using our simulated data, as an incremental step in our
 5212 progress toward fitting more generally useful models. To begin, we use our simulator to
 5213 grab a data set and then harvest the elements of the resulting object for further analysis.

```
5214 > data <- simSRC0(discard0=FALSE,rnd=2013)
5215 > y <- data$Y
5216 > traplocs <- data$traplocs
5217
5218 # In this case nind=N because we’re doing the known-N problem
5219 #
5220 > nind <- nrow(y)
5221 > X <- data$traplocs
5222 > J <- nrow(X)    # number of traps
5223 > K <- data$K
```

```
5224 > xlim <- data$xlim
5225 > ylim <- data$ylim
```

5226 Note that we specify `discard0 = FALSE` so that we have a “complete” data set, i.e.,
 5227 one with the all-zero encounter histories corresponding to uncaptured individuals. Now,
 5228 within an **R** session, we can create the **BUGS** model file and fit the model using the
 5229 following commands.

```
5230 cat("
5231   model{
5232     alpha0 ~ dnorm(0,.1)
5233     logit(p0) <- alpha0
5234     alpha1 ~ dnorm(0,.1)
5235     sigma <- sqrt(1/(2*alpha1))
5236     for(i in 1:N){ # note N here -- N is KNOWN in this example
5237       s[i,1] ~ dunif(xlim[1],xlim[2])
5238       s[i,2] ~ dunif(ylim[1],ylim[2])
5239       for(j in 1:J){
5240         d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
5241         y[i,j] ~ dbin(p[i,j],K)
5242         p[i,j] <- p0*exp(- alpha1*d[i,j]*d[i,j])
5243       }
5244     }
5245   }
5246   ",file = "SCR0a.txt")
```

5247 This model describes the Gaussian encounter probability model, but it would be trivial
 5248 to modify that to various others including the logistic described above. One consequence
 5249 of using the half-normal is that we have to constrain the encounter probability to be in
 5250 $[0, 1]$ which we do here by defining `alpha0` to be the logit of the intercept parameter `p0`.
 5251 Note that the distance covariate is computed within the **BUGS** model specification given
 5252 the matrix of trap locations, `X`, which is provided to **WinBUGS** as data.

5253 Next we do a number of organizational activities including bundling the data for **Win-**
 5254 **BUGS**, defining some initial values, the parameters to monitor and some basic MCMC
 5255 settings. We choose initial values for the activity centers `s` by generating uniform random
 5256 numbers in the state-space but, for the observed individuals, we replace those values by
 5257 each individual’s mean trap coordinate for all encounters

```
5258 ### Starting values for activity centers, s
5259 > sst <- cbind(runif(nind,xlim[1],xlim[2]),runif(nind,ylim[1],ylim[2]))
5260 > for(i in 1:nind){
5261   if(sum(y[i,])==0) next
5262   sst[i,1] <- mean( X[y[i,>0,1] )
5263   sst[i,2] <- mean( X[y[i,>0,2] )
5264 }
5265
5266 > data <- list (y=y, X=X, K=K, N=nind, J=J, xlim=xlim, ylim=ylim)
5267 > inits <- function(){
```

```

5268     list (alpha0=rnorm(1,-4,.4), alpha1=runif(1,1,2), s=sst)
5269   }
5270
5271 > library(R2WinBUGS)
5272 > parameters <- c("alpha0","alpha1","sigma")
5273 > out <- bugs (data, inits, parameters, "SCR0a.txt", n.thin=1, n.chains=3,
5274                 n.burnin=1000,n.iter=2000,debug=TRUE,working.dir=getwd())

```

5275 There is little to say about the preceding operations other than to suggest that you might
 5276 explore the output and investigate additional analyses by running the `simSCR0` script
 5277 provided in the **R** package `scrbook`.

5278 For purposes here, we ran 1000 burn-in and 1000 post-burn-in iterations, and 3 chains,
 5279 to obtain 3000 posterior samples. Because we know N for this particular data set we only
 5280 have 2 parameters of the detection model to summarize (`alpha0` and `alpha1`), along with
 5281 the derived parameter σ , the scale parameter of the Gaussian kernel, i.e., $\sigma = \sqrt{1/(2\alpha_1)}$.
 5282 When the object `out` is produced we print a summary of the results as follows:

```

5283 > print(out,digits=2)
5284 Inference for Bugs model at "SCR0a.txt", fit using WinBUGS,
5285   3 chains, each with 2000 iterations (first 1000 discarded)
5286   n.sims = 3000 iterations saved
5287     mean    sd   2.5%   25%   50%   75% 97.5% Rhat n.eff
5288 alpha0    -2.50  0.22  -2.95  -2.65  -2.48  -2.34  -2.09  1.01   190
5289 alpha1     2.44  0.42   1.64   2.15   2.44   2.72   3.30  1.00   530
5290 sigma      0.46  0.04   0.39   0.43   0.45   0.48   0.55  1.00   530
5291 deviance  292.80 21.16 255.60 277.50 291.90 306.00 339.30 1.01   380
5292
5293
5294 [...some output deleted...]
5295

```

5296 We know the data were generated with `alpha0 = -2.5` and `alpha1 = 2`. The estimates
 5297 look reasonably close to those data-generating values and we probably feel pretty good
 5298 about the performance of the Bayesian analysis and MCMC algorithm that **WinBUGS**
 5299 cooked-up based on our sample size of 1 data set. It is worth noting that the `Rhat`
 5300 statistics indicate reasonable convergence but, as a practical matter, we might choose to
 5301 run the MCMC algorithm for additional time to bring these closer to 1.0 and to increase
 5302 the effective posterior sample size (`n.eff`). Other summary output includes “deviance”
 5303 and related things including the deviance information criterion (DIC). We discuss general
 5304 issues of convergence and other MCMC considerations in Chapt. 17, and DIC and model
 5305 selection in Chapt. 8.

5.7 UNKNOWN N

5306 In all real applications N is unknown. We handled this important issue in Chapt. 4
 5307 using the method of data augmentation (DA) which we apply here to achieve a realistic
 5308 analysis of model SCR0. As with the basic closed population models considered previously,

5309 we formulate the problem by augmenting our observed data set with a number of “all-
 5310 zero” encounter histories - what we referred to in Chapt. 4 as potential individuals. If
 5311 n is the number of observed individuals, then let $M - n$ be the number of potential
 5312 individuals in the data set. For the 2-dimensional y_{ij} data structure (n individual $\times J$
 5313 traps encounter frequencies) we simply add additional rows of all-zero observations to
 5314 that data set. Because such “individuals” are unobserved, they therefore necessarily have
 5315 $y_{ij} = 0$ for all j . A data set, say with 4 traps and 6 individuals, augmented with 4
 5316 pseudo-individuals therefore might look like this:

```
5317      trap1 trap2 trap3 trap4
5318 [1,]    1    0    0    0
5319 [2,]    0    2    0    0
5320 [3,]    0    0    0    1
5321 [4,]    0    1    0    0
5322 [5,]    0    0    1    1
5323 [6,]    1    0    1    0
5324 [7,]    0    0    0    0
5325 [8,]    0    0    0    0
5326 [9,]    0    0    0    0
5327 [10,]   0    0    0    0
```

5328 We typically have more than 4 traps and, if we’re fortunate, many more individuals in
 5329 our data set.

5330 For the augmented data set, we introduce a set of binary latent variables (the data
 5331 augmentation variables), z_i , and the model is extended to describe $\Pr(z_i = 1)$ which is, in
 5332 the context of this problem, the probability that an individual in the augmented data set
 5333 is a member of the population of size N that was exposed to sampling. In other words,
 5334 if $z_i = 1$ for one of the all-zero encounter histories, this is implied to be a sampling zero
 5335 whereas observations for which $z_i = 0$ are “structural zeros” under the model. Under DA,
 5336 we also express the binomial observation model *conditional on z_i* as follows:

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

5337 where we see that the binomial probability evaluates to 0 if $z_i = 0$ (so y_{ij} is a fixed 0 in
 5338 that case) and evaluates to p_{ij} if $z_i = 1$.

5339 How big does the augmented data set have to be? We discussed this issue in Chapt. 4
 5340 where we noted that the size of the data set is equivalent to the upper limit of a uniform
 5341 prior distribution on N . Practically speaking, it should be sufficiently large so that the
 5342 posterior distribution for N is not truncated. On the other hand, if it is too large then
 5343 unnecessary calculations are being done. An approach to choosing M by trial-and-error
 5344 is indicated. Do a short MCMC run and then consider whether you need to increase M .
 5345 See Chapt. 17 for an example of this. Kéry and Schaub (2012, Chapt. 6) provide an
 5346 assessment of choosing M in closed population models. The useful thing about DA is that
 5347 it removes N as an explicit parameter of the model. Instead, N is a derived parameter,
 5348 computed by $N = \sum_{i=1}^M z_i$. Similarly, *density*, D , is also a derived parameter computed
 5349 as $D = N/\text{area}(\mathcal{S})$.

5350 5.7.1 Analysis using data augmentation in WinBUGS

5351 We provide a complete **R** script for simulating and organizing a data set, and analyzing
 5352 the data in **WinBUGS**. As before we begin by obtaining a data set using our `simSCR0`
 5353 function and then harvesting the required data objects from the resulting data list. Note
 5354 that we use the `discard0=TRUE` option this time so that we get a “real looking” data set
 5355 with no all-zero encounter histories:

```
5356 ##  
5357 ## Simulate the data and extract the required objects  
5358 ##  
5359 > data <- simSCR0(discard0=TRUE, rnd=2013)  
5360 > y <- data$Y  
5361 > nind <- nrow(y)  
5362 > X <- data$traplocs  
5363 > K <- data$K  
5364 > J <- nrow(X)  
5365 > xlim <- data$xlim  
5366 > ylim <- data$ylim
```

5367 After harvesting the data we augment the data matrix y with $M - n$ all-zero encounter
 5368 histories, and create starting values for the variables z_i and also the activity centers s_i
 5369 of which, for each, we require M values. One thing to take care of in using the **BUGS**
 5370 engines is the starting values for the activity centers. It is usually helpful to start the s_i
 5371 for each observed individual at or near the trap(s) it was captured. All of this happens as
 5372 follows:

```
5373 ## Data augmentation  
5374 > M <- 200  
5375 > y <- rbind(y,matrix(0,nrow=M-nind,ncol=ncol(y)))  
5376 > z <- c(rep(1,nind),rep(0,M-nind))  
5377  
5378 ## Starting values for s  
5379 > sst <- cbind(runif(M,xlim[1],xlim[2]),runif(M,ylim[1],ylim[2]))  
5380 > for(i in 1:nind){  
5381   sst[i,1] <- mean( X[y[i,]>0,1] )  
5382   sst[i,2] <- mean( X[y[i,]>0,2] )  
5383 }
```

5384 Next, we write out the **BUGS** model specification and save it to an external file
 5385 called `SCR0b.txt`. The model specification now includes M encounter histories including
 5386 the augmented potential individuals, the data augmentation parameters z_i , and the data
 5387 augmentation parameter ψ :

```
5388 > cat("model{  
5389   alpha0 ~ dnorm(0,.1)  
5390   logit(p0) <- alpha0
```

```

5392 alpha1 ~ dnorm(0,.1)
5393 sigma <- sqrt(1/(2*alpha1))
5394 psi ~ dunif(0,1)
5395
5396 for(i in 1:M){
5397   z[i] ~ dbern(psi)
5398   s[i,1] ~ dunif(xlim[1],xlim[2])
5399   s[i,2] ~ dunif(ylim[1],ylim[2])
5400   for(j in 1:J){
5401     d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
5402     y[i,j] ~ dbin(p[i,j],K)
5403     p[i,j] <- z[i]*p0*exp(- alpha1*d[i,j]*d[i,j])
5404   }
5405 }
5406 N <- sum(z[])
5407 D <- N/64
5408 }
5409 ",file = "SCR0b.txt")

```

5410 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:

```

5413 > data <- list (y=y, X=X, K=K, M=M, J=J, xlim=xlim, ylim=ylim)
5414 > inits <- function(){
5415   list (alpha0=rnorm(1,-4,.4), alpha1=runif(1,1,2), s=sst, z=z)
5416 }
5417
5418 > library(R2WinBUGS)
5419 > parameters <- c("alpha0","alpha1","sigma","N","D")
5420 > out <- bugs (data, inits, parameters, "SCR0b.txt", n.thin=1,n.chains=3,
5421   n.burnin=1000,n.iter=2000,debug=TRUE,working.dir=getwd())

```

5422 Note the differences in this new **WinBUGS** model with that appearing in the known-
5423 N version – there are not many! The loop over individuals goes up to M now, and there is a
5424 model component for the DA variables z . We are also computing some derived parameters:
5425 population size $N(\mathcal{S})$ is computed by summing up all of the data augmentation variables
5426 z_i (as we've done previously in Chapt. 4) and density, D , is also a derived parameter,
5427 being a function of N . The input data has changed slightly too, as the augmented data
5428 set has more rows to include excess all-zero encounter histories. Previously we knew that
5429 $N = 100$ but in this analysis we pretend not to know N , but think that $N = 200$ is a
5430 good upper bound. This analysis can be run directly using the **SCR0bayes** function once
5431 the **scrbook** package is loaded, by issuing the following commands:

```

5432 > library(scrbook)
5433 > data <- simSCR0(discard0=TRUE,rnd=2013)
5434 > out1 <- SCR0bayes(data,M=200,engine="winbugs",ni=2000,nb=1000)

```

5435 Summarizing the output from **WinBUGS** produces:

```

5436 > print(out1,digits=2)
5437 Inference for Bugs model at "SCR0b.txt", fit using WinBUGS,
5438   3 chains, each with 2000 iterations (first 1000 discarded)
5439   n.sims = 3000 iterations saved
5440     mean    sd   2.5%   25%   50%   75% 97.5% Rhat n.eff
5441 alpha0   -2.57  0.23  -3.04  -2.72  -2.56  -2.41  -2.15 1.01   320
5442 alpha1    2.46  0.42   1.63   2.16   2.46   2.73   3.33 1.02   120
5443 sigma     0.46  0.04   0.39   0.43   0.45   0.48   0.55 1.02   120
5444 N        113.62 15.73  86.00 102.00 113.00 124.00 147.00 1.01   260
5445 D         1.78  0.25   1.34   1.59   1.77   1.94   2.30 1.01   260
5446 deviance 302.60 23.67 261.19 285.47 301.50 317.90 354.91 1.00 1400
5447
5448 [...some output deleted...]
5449

```

5450 The **Rhat** statistic (discussed in Secs. 3.5.2 and 17.6.4) for this analysis indicates
 5451 satisfactory convergence. We see that the estimated parameters (α_0 and α_1) are comparable
 5452 to the previous results obtained for the known- N case, and also not too different
 5453 from the data-generating values. The posterior of N overlaps the data-generating value
 5454 substantially.

5455 **Use of other BUGS engines: JAGS**

5456 There are two other popular BUGS engines in widespread use: **OpenBUGS** (Thomas
 5457 et al., 2006) and **JAGS** (Plummer, 2003). Both of these are easily called from **R**. **Open-**
 5458 **BUGS** can be used instead of **WinBUGS** by changing the package option in the **bugs**
 5459 call to `package='OpenBUGS'`. **JAGS** can be called using the function **jags()** in package
 5460 **R2jags** which has nearly the same arguments as **bugs()**. Or, it can be executed from the
 5461 **R** package **rjags** (Plummer, 2011) which has a slightly different implementation that we
 5462 demonstrate here as we reanalyze the simulated data set in the previous section (note:
 5463 the same **R** commands are used to generate the data and package the data, inits and
 5464 parameters to monitor). The function **jags.model** is used to initialize the model and run
 5465 the MCMC algorithm for an adaptive period during which tuning of the MCMC algorithm
 5466 might take place. These samples cannot be used for inference. Then the Markov chains
 5467 are updated using **coda.samples()** to obtain posterior samples for analysis, as follows:

```

5468 > jinit <- jags.model("SCR0b.txt", data=data, inits=inits,
5469   n.chains=3, n.adapt=1000)
5470 > jout <- coda.samples(jinit, parameters, n.iter=1000, thin=1)

```

5471 These commands can be executed using the function **SCR0bayes** provided with the **R**
 5472 package **scrbook**. Hobbs (2011) provides a good introduction to ecological modeling with
 5473 **JAGS** which we recommend.

5474 **5.7.2 Implied home range area**

5475 Here we apply the method described in Sec. 5.4 to compute the effective home range
 5476 area under different encounter probability models fit to simulated data. We simulated a
 5477 data set from the Gaussian kernel model as in Sec. 5.7 and then we fitted 4 models to it:

Table 5.2. Posterior mean of model parameters for 4 different models fitted to a single simulated data set, and the effective home range area under each detection model.

	Gaussian	Cloglog	Exponential	Logit
N	113.62	114.16	119.69	118.29
D	1.78	1.78	1.87	1.85
α_0	-2.57	-2.60	-1.51	-0.47
α_1	2.46	2.56	3.59	3.86
hra	3.85	3.78	5.51	2.64

(1) the true data-generating Gaussian encounter probability model; (2) the “hazard” or complementary log-log link model (Eq. 5.4.2); (3) the negative exponential model and (4) the logit model (Eq. 5.2.2). We modified the function **SCR0bayes** for this purpose which you should be able to do with little difficulty. We fit each model to the same simulated data set using **WinBUGS**, based only on 1000 post-burn-in samples and 3 chains, which produced the posterior summaries given in Table 5.2. The main thing we see is that, while the implied home range area can vary substantially, there are smaller differences in the estimated N and hence D .

5.7.3 Realized and expected density

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

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

There is a third type of inference objective that is relevant in practice and that is prediction of N for a population that was not sampled – i.e., a “new” population. To elaborate on this, consider a situation in which we are concerned about the tiger population in 2 distinct reserves in India. We do a camera trapping study on one of the reserves to estimate N_1 and we think the reserves are similar and homogeneous so we’re willing to apply a density estimate based on N_1 to the 2nd reserve. For the 2nd reserve, do we want a prediction of the realized population size, N_2 , or do we want an estimate of its expected

5512 value? We believe the former is the proper quantity for inference about the population
 5513 size in the 2nd reserve. An estimate of N_2 should include the uncertainty with which
 5514 the mean is estimated (from reserve 1) and it should also include “process variation” for
 5515 making the prediction of the latent variable N_2 .

5516 As a practical matter, to do a Bayesian analysis of this you could just define the state-
 5517 space to be the union of the two state-spaces, increase M so that the posterior of the
 5518 total population size is not truncated, and then have MCMC generate a posterior sample
 5519 of individuals on the joint state-space. You can tally-up the ones that are on \mathcal{S}_2 as an
 5520 estimate of N_2 . Alternatively, we can define $\mu = \psi M/A_1$ and then simulate posterior
 5521 samples of $N_s \sim \text{Binomial}(M, \mu A_2/M)$ for the new state-space area, A_2 .

5522 To carry out a classical likelihood analysis of this 2nd type of problem, what should we
 5523 do? The argument for making a prediction of a new value of N would go something like
 5524 this: If you obtain an MLE of N , say \hat{N} , then the inference procedure tells us the variance
 5525 of this *conditional* on N . i.e., $\text{Var}(\hat{N}|N)$. This is fine, if we care about the specific value
 5526 of N that generated our data set. However, if we don’t care about the specific one in
 5527 question then we want to “uncondition” on N to introduce a new variance component.
 5528 Law of total variance says:

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

5529 If \hat{N} is unbiased then we say the unconditional variance is

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

5530 The first part is estimation error and the 2nd component is the “process variance.” If
 5531 you do Bayesian analysis, then you don’t have to worry too much about how to compute
 5532 variances properly. You decide if you care about N , or its expected value, or predictions
 5533 of some “new” N , and you tabulate the correct posterior distribution from your MCMC
 5534 output.

5535 The considerations for estimating density are the same. Density can be N/A where
 5536 N is the realized population, which we understand it to be unless we put an expectation
 5537 operator around the N like $\mathbb{E}(N)/A$. Classically, density is thought of as being defined as
 5538 the expected value of N but this might not always be meaningful because the context of
 5539 whether we mean realized density, of an actual population, or expected density for some
 5540 hypothetical unspecified population, should matter. The formula for obtaining “expected
 5541 density” is slightly different depending on whether we assume N has a Poisson distribution
 5542 or whether we assume a binomial distribution (under data augmentation). In the latter
 5543 case ψ is related to the point process intensity (see Chapt. 11) in the sense that, under
 5544 the binomial prior:

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

5545 so, what we think of as “density”, D , is $D = M\psi/A$. Under the Poisson point process
 5546 model we have:

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

5547 In summary, there are 3 basic inference problems that relate to estimating population
 5548 size (or density):

5549 (1) What is the value of N for some population that was sampled. This is what Efford
 5550 and Fewster call “realized N” In general, we want the uncertainty to reflect having to
 5551 estimate n_0 , the part of the population not seen.

- 5552 (2) We need to estimate N for some population that we didn't sample but it is "similar"
5553 to the population that we have information on. In this case, we have to account for
5554 both variation in having to estimate parameters of the distribution of N and we have
5555 to account for process variation in N (i.e., due to the stochastic model of N).
5556 (3) In some extremely limited cases we might care about estimating the expected value of
5557 N , $\mathbb{E}(N)$. This is only useful as a hypothetical statement that we might use, e.g., if we
5558 were to establish a new million ha refuge somewhere, then we might say its expected
5559 population size is 200 tigers.

5.8 THE CORE SCR ASSUMPTIONS

5560 It's always a good idea to sit down and reflect on the meaning of any particular model,
5561 its various assumptions, and what they mean in a specific context. From the statistician's
5562 point of view, the basic assumption, the omnibus assumption, as in all of statistics, and
5563 for every statistical model, is that "the model is correctly specified". So, naturally, that
5564 precludes everything that isn't explicitly addressed by the model. To point this out to
5565 someone seems to cause a lot of anxiety, so we enumerate here what we think are the most
5566 important statistical assumptions of the basic SCR0 model:

- 5567 • **Demographic closure.** The model does not allow for demographic processes. There
5568 is no recruitment or entry into the sampled population. There is no mortality or exit
5569 from the sampled population.
- 5570 • **Geographic closure.** We assume no permanent emigration or immigration from the
5571 state-space. However, we allow for "temporary" movements around the state-space
5572 and variable exposure to encounter as a result. The whole point of SCR models is to
5573 accommodate this dynamic. In ordinary capture-recapture models we have to assume
5574 geographic closure to interpret N in a meaningful way.
- 5575 • **Activity centers are randomly distributed.** That is, uniformity and independence
5576 of the underlying point process s_1, \dots, s_N (see next section).
- 5577 • **Detection is a function of distance.** A detection model that describes how encounter
5578 probability declines as a function of distance from an individual's home range center.
- 5579 • **Independence of encounters** among individuals. Encounter of any individual is
5580 independent of encounter of each other individual.
- 5581 • **Independence of encounters** of the same individual. Encounter of an individual
5582 in any trap is independent of its encounter in any other trap, and subsequent sample
5583 occasion.

5584 It's easy to get worried and question the whole SCR enterprise just on the grounds that
5585 these assumptions combine to form such a simplistic model, one that surely can't describe
5586 the complexity of real populations. On this sentiment, a few points are worth making.
5587 First, you don't have inherently fewer assumptions by using an ordinary capture-recapture
5588 model but, rather, the SCR model relaxes a number of important assumptions compared
5589 to the non-spatial counterpart. For one, here, we're not assuming that p is constant for all
5590 individuals but rather that p varies substantially as a matter of the spatial juxtaposition of
5591 individuals with traps. So maybe the manner in which p varies isn't quite right, but that's
5592 not an argument that supports doing less modeling. Fundamentally a distance-based
5593 model for p has some basic biological justification in virtually every capture-recapture

study. Secondly, for some of these core assumptions such as uniformity, and independence of individuals and of encounters, we expect a fair amount of robustness to departures. They function primarily to allow us to build a model and an estimation scheme and we don't usually think they represent real populations (of course, no model does!). Third, we can extend these assumptions in many different ways and we do that to varying extents in this book, and more work remains to be done in this regard. Forth, we can also evaluate the reasonableness of the assumptions formally in some cases using standard methods of assessing model fit (Chapt. 8).

Finally, we return back to our sentiment about the omnibus assumptions which is that the model is properly specified. This precludes *everything* that isn't in the model. Sometimes you see in capture-recapture literature statements like "we assume no marks are lost", "marks are correctly identified" and similar things. We might as well also assume that, a shopping mall is not built, or a meteor does not crash down into our study area, the sun does not go super-nova, and so forth. Our point is that we should separate statistical assumptions about model parameters or aspects of the probability model from what are essentially logistical or operational assumptions about how we interpret our data, or based on our ability to conduct the study. It is pointless to enumerate all of the possible explanations for apparent *departures*, because there are an infinity of such cases.

5.9 WOLVERINE CAMERA TRAPPING STUDY

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

5.9.1 Practical data organization

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

- The encounter data file (EDF) containing a record of which traps and when each individual encounter occurred.
- The trap deployment file (TDF) which contains the coordinates of each trap, along with information indicating which sample occasions each trap was operating.

Encounter Data File (EDF) – We store the encounter data in the an efficient file format which is easily manipulated in **R** and easy to create in Excel and other spreadsheets which are widely used for data management. The file structure is a simple matrix with 4 columns, those being: (1) **session ID**: the trap *session* which usually corresponds to a year or a primary period in the context of a Robust Design situation, but it could also correspond to a distinct spatial unit (see Sec. 6.5.4 and Chapt. 14). For a single-year study (as considered here) this should be an integer that is the same for all records;



Figure 5.4. Wolverine camera trap locations (black dots) from a study that took place in SE Alaska. See Magoun et al. (2011) for details.

5634 (2) **individual ID:** the individual identity, being an integer from 1 to n (repeated for
 5635 multiple captures of the same individual) indicating which individual the record (row) of
 5636 the matrix belongs to; (3) **occasion ID:** The integer sample occasion which generated
 5637 the record, and (4) **trap ID:** the trap identity, an integer from 1 to J , the number of
 5638 traps. The structure of the EDF is the same as used in the **secr** package (Efford, 2011a)
 5639 and similar to that used in the **SPACECAP** (Gopalaswamy et al., 2012a), and **SCRbayes**
 5640 (Russell et al., 2012) packages, both of which have a 3-column format (**trapID**, **indID**,
 5641 **sampID**). We note that the naming of the columns is irrelevant as far as anything we do in
 5642 this book, although **secr** and other software may have requirements on variable naming.
 5643 To illustrate this format, the wolverine data are available in the package **scrbook** by
 5644 typing:

```
5645 > data(wolverine)
```

5646 which contains a list having elements `wcaps` (the EDF) and `wtraps` (the TDF). We see
 5647 that `wcaps` has 115 rows, each representing a unique encounter event including the trap
 5648 identity, the individual identity and the sample occasion index (`sample`). The first 5 rows
 5649 of `wcaps` are:

```
5650 > wolverine$wcaps[1:5,]
5651   year individual day trap
5652 [1,]    1         2 127   1
5653 [2,]    1         2 128   1
5654 [3,]    1         2 129   1
5655 [4,]    1        18 130   1
5656 [5,]    1         3 106   2
```

5657 The 1st column here, labeled `year`, is an integer indicating the year or session of the
 5658 encounter. All these data come from a single year (2008) and so `year` is set to 1. Variable
 5659 `individual` is an integer identity of each individual captured, `day` is the sample occasion of
 5660 capture (in this case, the sample occasions correspond to days), and `trap` is the integer trap
 5661 identity. The variable `trapid` will have to correspond to the row of a matrix containing
 5662 the trap coordinates - in this case the TDF file `wtraps` which we describe further below.

5663 Note that the information provided in this encounter data file `wcaps` does not represent
 5664 a completely informative summary of the data. For example, if no individuals were
 5665 captured in a certain trap or during a certain period, then this compact data format will
 5666 have no record. Thus we will need to know J , the number of traps, and K , the number of
 5667 sample occasions when reformatting this SCR data format into a 2-d encounter frequency
 5668 matrix or 3-d array. In addition, the encounter data file does not provide information
 5669 about which periods each trap was operated. This additional information is also necessary
 5670 as the trap-specific sample sizes must be passed to **BUGS** as data. We provide this
 5671 information along with trap coordinates, in the “trap deployment file” (TDF) which is
 5672 described below.

5673 For our purposes, we need to convert the `wcaps` file into the $n \times J$ array of binomial
 5674 encounter frequencies, although more general models might require an encounter-history
 5675 formulation of the model which requires a full 3-d array. To obtain our encounter frequency
 5676 matrix, we do this the hard way by first converting the encounter data file into a 3-d array
 5677 and then summarize to trap totals. We have a handy function `SCR23darray` which takes
 5678 the compact encounter data file, and converts it to a 3-d array, and then we use the **R**
 5679 function `apply` to summarize over the sample occasion dimension (by convention here,
 5680 this is the 2nd dimension). To apply this to the wolverine data in order to compute the
 5681 3-d array we do this:

```
5682 > y3d <- SCR23darray(wolverine$wcaps,wolverine$wtraps)
5683 > y <- apply(y3d,c(1,2),sum)
```

5684 See the help file for more information on `SCR23darray`. The 3-d array is necessary to
 5685 fit certain types of models (e.g., behavioral response) and this is why we sometimes will
 5686 require this maximally informative 3-d data format but, here, we analyze the summarized
 5687 data.

5688 **Trap Deployment File (TDF)** – The other important information needed to fit SCR
 5689 models is the “trap deployment file” (TDF) which provides additional information not

5690 contained in the encounter data file. The traps file has $K + 3$ columns. The first column is
 5691 assumed to be a trap identifier, columns 2 and 3 are the easting and northing coordinates
 5692 (assumed to be in a Euclidean coordinate system), and columns 4 to $K + 3$ are binary
 5693 indicators of whether each trap was operational during each sample occasion. The first 10
 5694 rows (out of 37) and 10 columns (out of 167) of the trap deployment file for the wolverine
 5695 data are shown as follows:

```
5696 > wolverine$wtraps[1:10,1:10]
5697
5698     Easting Northing 1 2 3 4 5 6 7 8
5699 1   632538  6316012 0 0 0 0 0 0 0 0
5700 2   634822  6316568 1 1 1 1 1 1 1 1
5701 3   638455  6309781 0 0 0 0 0 0 0 0
5702 4   634649  6320016 0 0 0 0 0 0 0 0
5703 5   637738  6313994 0 0 0 0 0 0 0 0
5704 6   625278  6318386 0 0 0 0 0 0 0 0
5705 7   631690  6325157 0 0 0 0 0 0 0 0
5706 8   632631  6316609 0 0 0 0 0 0 0 0
5707 9   631374  6331273 0 0 0 0 0 0 0 0
5708 10  634068  6328575 0 0 0 0 0 0 0 0
```

5709 This tells us that trap 2 was operated during occasions (days) 1-7 but the other traps
 5710 were not operational during those periods. It is extremely important to recognize that
 5711 each trap was operated for a variable period of time and thus the binomial “sample size”
 5712 is different for each, and this needs to be accounted for in the **BUGS** model specification.
 5713 To compute the vector of sample sizes K , and extract the trap locations, we do this:

```
5714 > traps <- wolverine$wtraps
5715 > traplocs <- traps[,1:2]
5716 > K <- apply(traps[,3:ncol(traps)],1,sum)
```

5717 This results in a matrix `traplocs` which contains the coordinates of each trap and a vector
 5718 `K` containing the number of days that each trap was operational. We now have all the
 5719 information required to fit a basic SCR model in **BUGS**.

5720 Summarizing the data for the wolverine study, we see that 21 unique individuals were
 5721 captured a total of 115 times. Most individuals were captured 1-6 times, with 4, 1, 4, 3, 1,
 5722 and 2 individuals captured 1-6 times, respectively. In addition, 1 individual was captured
 5723 each 8 and 14 times and 2 individuals each were captured 10 and 13 times. The number
 5724 of unique traps that captured a particular individual ranged from 1-6, with 5, 10, 3, 1, 1,
 5725 and 1 individual captured in each of 1 to 6 different traps, respectively, for a total of 50
 5726 unique wolverine-trap encounters. These numbers might be hard to get your mind around
 5727 whereas some tabular summary is often more convenient. For that it seems natural to
 5728 tabulate individuals by trap and total encounter frequencies. The spatial information in
 5729 SCR data is based on multi-trap captures, and so, it is informative to understand how
 5730 many unique traps each individual is captured in, and the total number of encounters.
 5731 For the wolverine data, we reproduce Table 1 from Royle et al. (2011b) as Table 5.3.

Table 5.3. Individual frequencies of capture for wolverines captured in camera traps in Southeast Alaska in 2008. Rows index unique traps of capture for each individual and columns represent total number of captures (e.g., we captured 4 individuals 1 time, necessarily in only 1 trap; we captured 3 individuals 3 times but in 2 different traps).

No. of traps	No. of captures									
	1	2	3	4	5	6	8	10	13	14
1	4	1	0	0	0	0	0	0	0	0
2	0	0	3	2	0	2	1	2	0	0
3	0	0	1	1	0	0	0	0	0	1
4	0	0	0	0	0	0	0	0	1	0
5	0	0	0	0	1	0	0	0	0	0
6	0	0	0	0	0	0	0	0	1	0

5.9.2 Fitting the model in WinBUGS

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

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

We assume customary flat priors on the structural (hyper-) parameters of the model, $\alpha_0 = \text{logit}(p_0)$, α_1 and ψ .

It remains to define the state-space \mathcal{S} . For this, we nested the trap array (Fig. 5.4) in a rectangular state-space extending 20 km beyond the traps in each cardinal direction. We scaled the coordinate system so that a unit distance was equal to 10 km, producing a rectangular state-space of dimension 9.88×10.5 units ($\text{area} = 10374 \text{ km}^2$) within which the trap array was nested. As a general rule, we recommend scaling the state-space so that it is defined near the origin $(x, y) = (0, 0)$. While the scaling of the coordinate system is theoretically irrelevant, a poorly scaled coordinate system can produce Markov chains that mix poorly. The buffer of the state space should be large enough so that individuals beyond the state-space boundary are not likely to be encountered (Sec. 5.3.1). To evaluate this, we fit models for various choices of a rectangular state-space based on buffers from 1.0 to 5.0 units (10 km to 50 km). In the **R** package **scrbook** we provide a function **wolvSCR0** which will fit model SCR0. For example, to fit the model in **WinBUGS** using data augmentation with $M = 300$ potential individuals, using 3 Markov chains each of 12000 total iterations, discarding the first 2000 as burn-in, we execute the following **R** commands:

```

5757 > library(scrbook)
5758 > data(wolverine)
5759 > traps <- wolverine$wtraps
5760 > y3d <- SCR23darray(wolverine$wcaps,wolverine$wtraps)
5761 > wolv <- wolvSCR0(y3d,traps,nb=2000,ni=12000,buffer=1,M=300)

```

Table 5.4. Posterior summaries of SCR model parameters for the wolverine camera trapping data from SE Alaska, using state-space buffers from 10 up to 50 km. Each analysis was based on 3 chains, 12000 iterations, 2000 burn-in, for a total of 30000 posterior samples.

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

Table 5.5. Posterior summaries of SCR model parameters for the wolverine camera trapping data from SE Alaska. The model was run with the trap array centered in a state-space with a 20 km rectangular buffer.

Parameter	Mean	SD	2.5%	25%	50%	75%	97.5%	Rhat
N	59.84	11.89	40.00	51.00	59.00	67.00	86.00	1
D	5.77	1.15	3.86	4.92	5.69	6.46	8.29	1
α_1	1.26	0.21	0.87	1.11	1.25	1.40	1.71	1
p_0	0.06	0.01	0.04	0.05	0.06	0.06	0.08	1
σ	0.64	0.06	0.54	0.60	0.63	0.67	0.76	1
ψ	0.20	0.05	0.12	0.17	0.20	0.23	0.30	1

5762 The argument `buffer` determines the buffer size of the state-space in the scaled units
 5763 (i.e., 10 km). Note that this analysis takes between 1-2 hours on many machines (in 2013)
 5764 so we recommend testing it with lower values of M and fewer iterations. The posterior
 5765 summaries are shown in Table 5.9.2.

5766 5.9.3 Summary of the wolverine analysis

5767 We see that the estimated density is roughly consistent as we increase the state-space
 5768 buffer from 15 to 55 km. We do note that the data augmentation parameter ψ (and,
 5769 correspondingly, N) increase with the size of the state space in accordance with the deter-
 5770 ministic relationship $N = D * A$. However, density is more or less constant as we increase
 5771 the size of the state-space beyond a certain point. For the 10 km state-space buffer, we see
 5772 a slight effect on the posterior distribution of D because the state-space is not sufficiently
 5773 large. The full results from the analysis based on 20 km state-space buffer are given in
 5774 Table 5.5.

5775 Our point estimate of wolverine density from this study, using the posterior mean from
 5776 the state-space based on the 20 km buffer, is approximately 5.77 individuals/1000 km²

with a 95% posterior interval of [3.86, 8.29]. Density is estimated imprecisely which might not be surprising given the low sample size ($n = 21$ individuals!). This seems to be a 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.

5.9.4 Wolverine space usage

The parameter α_1 is related to the home range radius (Sec. 5.4). For the Gaussian model we interpret the scale parameter σ , related to α_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 $\pi \times 15.66^2 = 770.4 \text{ km}^2$. Using our handy function `hra` we do this:

```
5801 hra(pGauss1,parms=c(-2,1/(2*.64*.64)),xlim=c(-1,7),ylim=c(-1,7))
5802
5803 [1] 7.731408
```

which is in units of 100 km², 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 km² (Whitman et al., 1986).

Royle et al. (2011b) reported estimates for σ in the range 6.3 – 9.8 km depending on the model, which isn't too different than here¹. 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

¹ Royle et al. (2011b) expressed the model as $\text{cloglog}(p_{ij}) = \alpha_0 - (1/\sigma^2) * d_{ij}^2$, but the estimates of σ 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 σ 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.

5815 biological interpretations of home range area based on σ , although SCR models can be
5816 extended to handle more general, non-Euclidean, patterns of space usage. See Chaps. 12
5817 and 13.

5818 We can calibrate the desired size of the state-space by looking at the estimated home
5819 range radius of the species. We should target a buffer of width 2 to $3 \times \sigma$ in order that
5820 the probability of encountering an individual is very close to 0 beyond the prescribed
5821 state-space. Essentially, by specifying a state-space, we're setting $p = 0$ for individuals
5822 beyond the prescribed state-space. For the wolverine data, with σ in the range of 6-9 km,
5823 a state-space buffer of 20 km is sufficiently large.

5.10 USING A DISCRETE HABITAT MASK

5824 The SCR model developed previously in this chapter assumes that individual activity
5825 centers are distributed uniformly over the prescribed state-space. Clearly this will not
5826 always be a reasonable assumption. In Chapt. 11, we develop models that allow explicitly
5827 for non-uniformity of the activity centers by modeling covariate effects on density. A
5828 simplistic method of affecting the distribution of activity centers, which we address here,
5829 is to modify the shape and organization of the state-space explicitly. For example, we
5830 might be able to classify the state-space into distinct blocks of habitat and non-habitat.
5831 In that case we can remove the non-habitat from the state-space and assume uniformity of
5832 the activity centers over the remaining portions judged to be suitable habitat. There are
5833 several ways to approach this: We can use a grid of points to represent the state-space, i.e.,
5834 by the set of coordinates s_1, \dots, s_G , and assign equal probabilities to each possible value.
5835 Alternatively, we can retain the continuous formulation of the state-space but attempt
5836 to describe constraints analytically, or we can use polygon clipping methods to enforce
5837 constraints on the state-space in the MCMC analysis. We focus here on the formulation of
5838 the basic SCR model in terms of a discrete state-space but in Chapt. 17 we demonstrate
5839 the latter approach based on using polygon operations to define an irregular state-space.
5840 Use of a discrete state-space can be computationally expensive in **WinBUGS**. That said,
5841 it isn't too difficult to perform the MCMC calculations in **R** (discussed in Chapt. 17).
5842 The **R** package **SPACECAP** (Gopalaswamy et al., 2012a) arose from the **R** implementation
5843 of the SCR model in Royle et al. (2009a).

5844 While clipping out non-habitat seems like a good idea, we think investigators should
5845 go about this very cautiously. We might prefer to do it when non-habitat represents a
5846 clear-cut restriction on the state-space such as a reserve boundary or a lake, ocean or
5847 river. But, having the capability to do this also causes people to start defining "habitat"
5848 vs. "non-habitat" based on their understanding of the system whereas it can't be known
5849 whether the animal being studied has the same understanding. Moreover, differentiating
5850 the landscape by habitat or habitat quality must affect the geometry and morphology of
5851 home ranges (see Chapt. 13) much more so than the plausible locations of activity centers.
5852 That is, a home range centroid could, in actual fact, occur in a shopping mall parking lot
5853 if there is pretty good habitat around the shopping mall, so there is probably no sense
5854 preclude it as the location for an activity center. It would generally be better to include
5855 some definition of habitat quality in the model for the detection probability (Royle et al.,
5856 2013) which we address in Chaps. 12 and 13.

5.10.1 Evaluation of coarseness of habitat mask

5857 The coarseness of the state-space should not really have much of an effect on estimates
 5859 if the grain is sufficiently fine relative to typical animal home range sizes. Why is this?
 5860 We have two analogies that can help us understand. First is the relationship to model
 5861 M_h . As noted in Sec. 5.3.2 above, we can think about SCR models as a type of finite
 5862 mixture (Norris and Pollock, 1996; Pledger, 2004) where we are fortunate to be able to
 5863 obtain direct information about which group individuals belong to (group being location
 5864 of activity center). In the standard finite mixture models we typically find that a small
 5865 number of groups (e.g., 2 or 3 at the most) can explain high levels of heterogeneity and
 5866 are adequate for most data sets of small to moderate sample sizes. We therefore expect a
 5867 similar effect in SCR models when we discretize the state-space. We can also think about
 5868 discretizing the state-space as being related to numerical integration where we find (see
 5869 Chapt. 6) that we don't need a very fine grid of support points to evaluate the integral to
 5870 a reasonable level of accuracy. We demonstrate this here by reanalyzing simulated data
 5871 using a state-space defined by a different number of support points. We provide an **R**
 5872 script called **SCR0bayesDss** in the **R** package **scrbook**. We note that for this comparison
 5873 we generated the actual activity centers as a continuous random variable and thus the
 5874 discrete state-space is, strictly speaking, an approximation to truth. That said, we regard
 5875 all state-space specifications as approximations to truth in the sense that they represent
 5876 a component of the SCR model.

5877 As with our **R** function **SCR0bayes**, the modification **SCR0bayesDss** will use either
 5878 **WinBUGS** or **JAGS**. In addition, it requires a grid resolution argument (**ng**) which
 5879 is the dimension of 1 side of a square state-space. To execute this function we do, for
 5880 example:

```
5881 > library(scrbook)
5882 > data <- simSCR0(discard0=TRUE,rnd=2013)    # Generate data set
5883
5884 # Run with JAGS
5885 > out1 <- SCR0bayesDss(data,ng=8,M=200,engine="jags",ni=2000,nb=1000)
5886
5887 # Run with WinBUGS
5888 > out2 <- SCR0bayesDss(data,ng=8,M=200,engine="winbugs",ni=2000,nb=1000)
```

5889 We fit this model to the same simulated data set for 6×6 , 9×9 , 12×12 , 15×15
 5890 state-space grids. For **WinBUGS**, we used 3 chains of 5000 total length with 1000 burn-in,
 5891 which yields 12000 total posterior samples. Summary results are shown in Table 5.6.
 5892 The results are broadly consistent except for the 6×6 case. We see that the run time
 5893 increases with the size of the state-space grid (not unexpected), such that we imagine it
 5894 would be impractical to run models with more than a few hundred state-space grid points.
 5895 We found (not shown here) that the runtime of **JAGS** is much faster and, furthermore,
 5896 relatively *constant* as we increase the grid size. We suspect that **WinBUGS** is evaluating
 5897 the full-conditional for each activity center at all G possible values whereas it may be
 5898 that **JAGS** is evaluating the full-conditional only at a subset of values or perhaps using
 5899 previous calculations more effectively. While this might suggest that one should always
 5900 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

Table 5.6. Comparison of the effect of state-space grid coarseness on estimates of N for a simulated data set. Posterior summaries and run time are given. Results obtained using **WinBUGS** run from **R2WinBUGS**.

Grid Size	Mean	SD	NaiveSE	Time-seriesSE	runtime (sec)
6 × 6	111.6699	16.61414	0.1516657	0.682008	2274
9 × 9	114.2294	17.99109	0.1642355	0.833291	4300
12 × 12	115.9806	17.3843	0.1586964	0.762756	7100
15 × 15	115.379	17.93721	0.1637436	0.832483	13010

5902 often simply do not work for some problems, so be careful when using **JAGS**. To improve
 5903 its performance, always start the latent activity centers at values near where individuals
 5904 were captured. The performance of either should improve if we compute the full distance
 5905 matrix outside of **BUGS** and pass it as data, although we haven't fully evaluated this
 5906 approach.

5907 5.10.2 Analysis of the wolverine camera trapping data

5908 We reanalyzed the wolverine data using discrete state-space grids with points spaced by
 5909 2, 4 and 8 km (see Fig. 5.5). These were constructed from a 40 km buffered state-space,
 5910 and deleting the points over water (see Royle et al., 2011b). Our interest in doing this
 5911 was to evaluate the relative influence of grid resolution on estimated density because the
 5912 coarser grids will be more efficient from a computational stand-point and so we would
 5913 prefer to use them, but only if there is no strong influence on estimated density. The
 5914 posterior summaries for the 3 habitat grids are given in Table 5.7. We see that the
 5915 density estimates are quite a bit larger than obtained in our analysis (Table 5.9.2) based
 5916 on a rectangular, continuous state-space. We also see that there are slight differences
 5917 depending on the resolution of the state-space grid. Interestingly, the effectiveness of the
 5918 MCMC algorithms, as measured by effective sample size (**n.eff**) is pretty remarkably
 5919 different. Furthermore, the finest grid resolution (2 km spacing) took about 6 days to run
 5920 and thus it would not be practical for large problems or with many models.

5.11 SUMMARIZING DENSITY AND ACTIVITY CENTER LOCATIONS

5921 One of the most useful aspects of SCR models is that they are parameterized in terms of
 5922 individual locations – i.e., *where* each individual lives – and, thus, we can compute many
 5923 useful and interesting summaries of the activity centers using output from an MCMC sim-
 5924 ulation, including maps of density (the number of activity centers per unit area), estimates
 5925 of N for any well-defined polygon, or estimates of where the activity centers for specific
 5926 individuals reside. In Bayesian analysis by MCMC, obtaining such summaries entails no
 5927 added calculations, because we need only post-process the output for the individual ac-
 5928 tivity centers to obtain the desired summaries. We demonstrate that in this section. Note
 5929 that you have to be sure to retain the MCMC history for the **s** variables and also the data
 5930 augmentation variables z in order to do the following analyses.

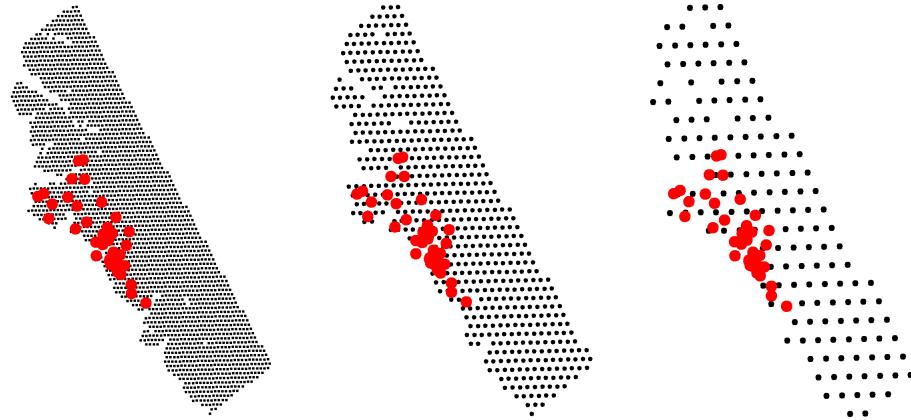


Figure 5.5. Three habitat mask grids used in the comparison of the effect of pixel size on the estimated density surface of wolverines. The 3 cases are 2 (left) (center) and 8 (right) km spacing of state-space points, extending 40 km from the vicinity of the trap array.

5.11.1 Constructing density maps

Because SCR models are spatially-explicit, it is natural to want to summarize the results of fitting a model by producing a map of density. Using Bayesian analysis by MCMC, it is most easy to make a map of *realized* density. We can do this by tallying up the number of activity centers \mathbf{s}_i in pixels of arbitrary size and then producing a nice multi-color spatial plot of the result. Specifically, let $B(\mathbf{x})$ indicate a pixel centered at \mathbf{x} then

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

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

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

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

Table 5.7. Posterior summaries for the wolverine camera trapping data, using model SCR0, with a Gaussian hazard encounter probability model, and a discrete habitat mask of 3 different resolutions: 2, 4 and 8 km. Parameters are λ_0 = baseline encounter rate, $p_0 = 1 - \exp(-\lambda_0)$, σ is the scale parameter of the Gaussian kernel, ψ is the data augmentation parameter, N and D are population size and density, respectively. Models fitted using **WinBUGS**, 3 chains, each with 11000 iterations (first 1000 discarded) producing 30000 posterior samples.

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

5946 Thus, $N(\mathbf{x}, 1), N(\mathbf{x}, 2), \dots$, is the Markov chain for parameter $N(\mathbf{x})$. In what follows we
 5947 will provide a set of **R** commands for doing this calculation and making a basic image
 5948 plot from the MCMC output.

5949 **Step 1:** Define the center points of each pixel $B(\mathbf{x})$, or point at which local density will
 5950 be estimated:

```
5951 > xg <- seq(xlim[1], xlim[2], , 50)
  5952 > yg <- seq(ylim[1], ylim[2], , 50)
```

5953 **Step 2:** Extract the MCMC histories for the activity centers and the data augmentation
 5954 variables. Note that these are each $N \times \text{niter}$ matrices. Here we do this assuming that
 5955 **WinBUGS** was run producing the **R** object named **out**:

```
5956 > Sxout <- out$sims.list$s[, , 1]
  5957 > Syout <- out$sims.list$s[, , 2]
  5958 > z <- out$sims.list$z
```

5959 **Step 3:** We associate each coordinate with the proper pixel using the **R** command `cut()`.
 5960 Note that we keep only the activity centers for which $z = 1$ (i.e., individuals that belong
 5961 to the population of size N):

```
5962 > Sxout <- cut(Sxout[z==1], breaks=xg, include.lowest=TRUE)
5963 > Syout <- cut(Syout[z==1], breaks=yg, include.lowest=TRUE)
```

5964 **Step 4:** Use the `table()` command to tally up how many activity centers are in each
 5965 $B(\mathbf{x})$:

```
5966 > Dn <- table(Sxout, Syout)
```

5967 **Step 5:** Use the `image()` command to display the resulting matrix.

```
5968 > image(xg, yg, Dn/nrow(z), col=topo.colors(10))
```

5969 It is worth emphasizing here that density maps will not usually appear uniform despite
 5970 that we have assumed that activity centers are uniformly distributed. This is because
 5971 the observed encounters of individuals provide direct information about the location of
 5972 the $i = 1, 2, \dots, n$ activity centers and thus their “estimated” locations will be affected
 5973 by the observations. In a limiting sense, were we to sample space intensely enough,
 5974 every individual would be captured a number of times and we would have considerable
 5975 information about all N point locations. Consequently, the uniform prior would have
 5976 almost no influence at all on the estimated density surface in this limiting situation.
 5977 Thus, in practice, the influence of the uniformity assumption decreases as the fraction of
 5978 the population encountered, and the total number of encounters per individual, increases.

5979 **On the non-intuitiveness of `image()`** – the **R** function `image()`, invoked for a
 5980 matrix M by `image(M)`, might not be very intuitive to some – it plots $M[1, 1]$ in the lower
 5981 left corner. If you want $M[]$ to be plotted “as you look at it” then $M[1, 1]$ should be in the
 5982 upper left corner. We have a function `rot()` which does that. If you do `image(rot(M))`
 5983 then it puts it on the monitor as if it was a map you were looking at. You can always
 5984 specify the x - and y -labels explicitly as we did above.

5985 **Spatial dot plots** – A cruder version of the density map can be made using our
 5986 “spatial dot map” function `spatial.plot` (in `scrbook`). This function requires, as input,
 5987 point locations and the value to be displayed. A simplified version of this function is as
 5988 follows:

```
5989 > spatial.plot <- function(x,y){
  5990   nc <- as.numeric(cut(y,20))
  5991   plot(x,pch=" ")
  5992   points(x,pch=20,col=topo.colors(20)[nc],cex=2)
  5993   image.scale(y,col=topo.colors(20))
  5994 }
  5995 #
  5996 # To execute the function do this:
  5997 #
  5998 > spatial.plot(cbind(xg,yg), Dn/nrow(z))
```

5.11.2 Example: Wolverine density map

6000 We return to the wolverine study which took place in 2008 in SE Alaska (Fig. 5.4) and
 6001 we produce a density map of wolverines from that analysis. We include the function
 6002 **SCRdensity** which requires a specific data structure as shown below. In particular, we
 6003 have to package up the MCMC history for the activity centers and the data augmentation
 6004 variables z into a list. This also requires that we add those variables to the parameters-
 6005 to-be-monitored list when we pass things to **BUGS**.

6006 We used the posterior output from the wolverine model fitted previously to compute
 6007 a relatively coarse version of a density map, using 100 pixels in a 10×10 grid (Fig. 5.6
 6008 top panel) and using 900 pixels arranged in a 30×30 grid (Fig. 5.6 lower panel) for a
 6009 fine-scale map. The **R** commands for producing such a plot (for a short MCMC run) are
 6010 as follows:

```
6011 > library(scrbook)
6012 > data(wolverine)
6013 > traps <- wolverine$wtraps
6014 > y3d <- SCR23darray(wolverine$wcaps,wolverine$wtraps)
6015
6016 # This takes 341 seconds on a standard CPU circa 2011
6017 > out <- wolvSCRO(y3d,traps,nb=1000,ni=2000,buffer=1,M=100,keepz=TRUE)
6018
6019 > Sx <- out$sims.list$s[,,1]
6020 > Sy <- out$sims.list$s[,,2]
6021 > z <- out$sims.list$z
6022 > obj <- list(Sx=Sx,Sy=Sy,z=z)
6023 > tmp <- SCRdensity(obj,nx=10,ny=10,scalein=100,scaleout=100)
```

6024 In these figures density is expressed in units of individuals per 100 km^2 , while the area of
 6025 the pixels is about 103.7 km^2 and 11.5 km^2 , respectively. That calculation is based on:

```
6026 > total.area <- (ylim[2]-ylim[1])*(xlim[2]-xlim[1])*100
6027 > total.area/(10*10)
6028 [1] 103.7427
6029 > total.area/(30*30)
6030 [1] 11.52697
```

6031 A couple of things are worth noting: First is that as we move away from “where the
 6032 data live” – away from the trap array – we see that the density approaches the mean
 6033 density. This is a property of the estimator as long as the detection function decreases
 6034 sufficiently rapidly as a function of distance. Relatedly, it is also a property of statistical
 6035 smoothers such as splines, kernel smoothers, and regression smoothers – predictions tend
 6036 toward the global mean as the influence of data diminishes. Another way to think of it is
 6037 that it is a consequence of the prior, which imposes uniformity, and as you get far away
 6038 from the data, the predictions tend to the expected constant density under the prior.
 6039 Another thing to note about this map is that density is not 0 over water (although the
 6040 coastline is not shown). This might be perplexing to some who are fairly certain that
 6041 wolverines do not like water. However, there is nothing about the model that recognizes

6042 water from non-water and so the model predicts over water *as if* it were habitat similar to
 6043 that within which the array is nested. But, all of this is OK as far as estimating density
 6044 goes and, furthermore, we can compute valid estimates of N over any well-defined region
 6045 which presumably wouldn't include water if we so wished. Alternatively, areas covered by
 6046 water could be masked out, which we discuss in the next section.

6047 5.11.3 Predicting where an individual lives

6048 The density maps in the previous section show the expected number of individuals per
 6049 unit area. A closely related problem is that of producing a map of the probable location
 6050 of a specific individual's activity center. For any observed encounter history, we can easily
 6051 generate a posterior distribution of \mathbf{s}_i for individual i . In addition, for an individual that
 6052 is *not* captured, we can use the MCMC output to produce a corresponding plot of where
 6053 such an individual might live, say \mathbf{s}_{n+1} . Obviously, all such uncaptured individuals (for
 6054 $i = n + 1, \dots, N$) should have the same posterior distribution. To illustrate, we show the
 6055 posterior distribution of \mathbf{s}_1 , the activity center for the individual labeled 1 in the data
 6056 set, in Fig. 5.7. This individual was captured a single time at trap 30 which is circled
 6057 in Fig. 5.7. We see that the posterior distribution is affected by traps of capture *and*
 6058 traps of non-capture in fairly intuitive ways. In particular, because there are other traps
 6059 in close proximity to trap 30, in which individual 1 was *not* captured, the model pushes
 6060 its activity center away from the trap array. The help file for `SCRdensity` shows how to
 6061 calculate Fig. 5.7.

5.12 EFFECTIVE SAMPLE AREA

6062 One of the key issues in using ordinary capture recapture models which we've brought up
 6063 over and over again is this issue that the area which is sampled by a trapping array is
 6064 unknown – in other words, the N that is estimated by capture-recapture models does not
 6065 have an explicit region of space associated with it. Classically this has been addressed in
 6066 the ad hoc way of prescribing an area that contains the trap array, usually by adding a
 6067 buffer of some width, which is not estimated as part of the capture-recapture model. In
 6068 SCR models we avoid the problem of not having an explicit linkage between N and “area”,
 6069 by prescribing explicitly the area within which the underlying point process is defined – the
 6070 state-space of the point process. This state-space is *not* the effective sample (or sampled)
 6071 area (ESA) – it is desirable that it be somewhat larger than the ESA, whatever that may
 6072 be, in the sense that individuals at the edge of the state-space have no probability of being
 6073 captured, but as part of the SCR model we don't need to try to estimate or otherwise
 6074 characterize the ESA explicitly.

6075 However, it is possible to provide a characterization of effective sampled area under
 6076 any SCR model. This is directly analogous to the calculation of “effective strip width” in
 6077 distance sampling (Buckland et al., 2001; Borchers et al., 2002). The conceptual definition
 6078 of ESA follows from equating density to “apparent density” – ESA is the magic number
 6079 that satisfies that equivalence:

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

6080 In other words, the ratio of N to the area of the state-space should be equal to the ratio
 6081 of the observed sample size n to this number ESA. Both of these should equal density.

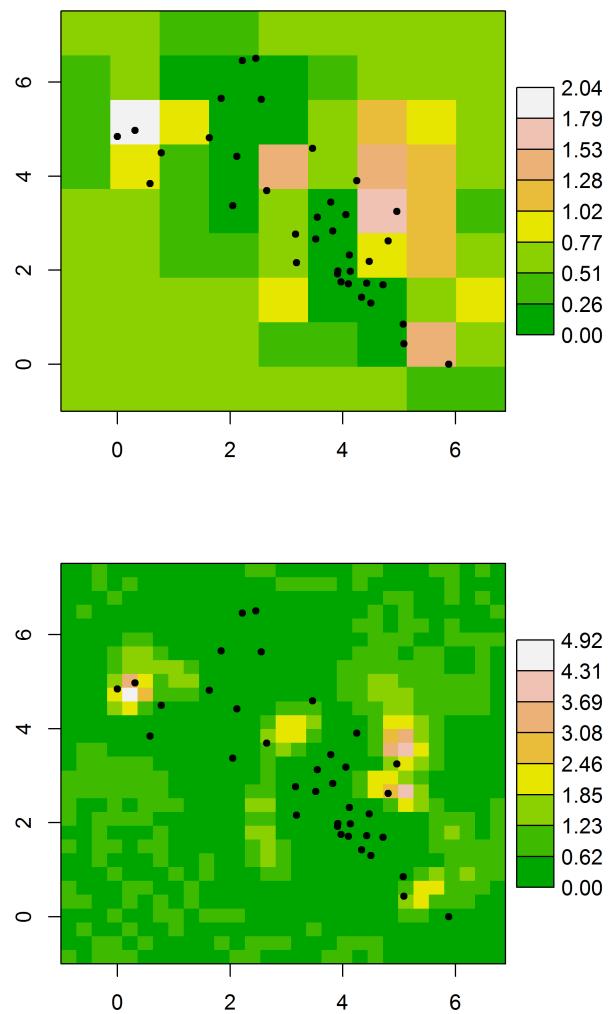


Figure 5.6. Density of wolverines (individuals per 100 km²) in SE Alaska in 2007 based on model SCR0. Map grid cells are about 103.7 km² (top panel) and 11.5 km² (bottom panel) in area. Dots are the trap locations.

6082 So, to compute ESA for a model, we substitute $\mathbb{E}(n)$ for n into the above equation, and
 6083 solve for ESA , to get:

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

6084 Our following development assumes that D is constant, but these calculations can be
 6085 generalized to allow for D to vary spatially. Imagine our habitat mask for the wolverine
 6086 data, or the bins we just used to produce a density map, then we can write $\mathbb{E}(n)$ according
 6087 to

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

6088 where if we prefer to think of this more conceptually we could replace the summation with
 6089 an integration (which, in practice, we would just replace with a summation, and so we
 6090 just begin there). In this expression note that $\mathbb{E}(N(\mathbf{s}))$ is the expected population size at
 6091 pixel \mathbf{s} which is the density times the area of the pixel, i.e., $\mathbb{E}(N(\mathbf{s})) = D \times a$. Therefore

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

6092 and (plugging this into the expression above for ESA)

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

6093 We see that D cancels and we have $ESA = a \times \sum_s \Pr(\text{encounter}|\mathbf{s})$ So what you have to
 6094 do here is substitute in $\Pr(\text{encounter}|\mathbf{s})$ and just sum them up over all pixels. For the
 6095 Bernoulli model of model SCR0

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

6096 with slight modifications when encounter probability depends on covariates. Thus,

$$ESA = a \sum_s 1 - (1 - p(\mathbf{s}))^K \tag{5.12.1}$$

6097 Clearly the calculation of ESA is affected by the use of a habitat mask, because the
 6098 summation in Eq. 5.12.1 only occurs over pixels that define the state-space.

6099 For the wolverine camera trapping data, we used the 2×2 km habitat mask and the
 6100 posterior means of p_0 and σ (see Sec. 5.10.2) to compute the probability of encounter for
 6101 each \mathbf{s} of the mask points. The result is shown graphically in Fig. 5.8. The ESA is the
 6102 sum of the values plotted in that figure multiplied by 4, the area of each pixel. For the
 6103 wolverine study, the result is 2507.152 km 2 . We note that the probability of encounter
 6104 declines rapidly to 0 as we move away from the periphery of the camera traps, indicating
 6105 the state-space constructed from a 40 km buffered trap array was indeed sufficient for the
 6106 analysis of these data. An R script for producing this figure is in the `wolvESA` function of
 6107 the `scrbook` package.

5.13 SUMMARY AND OUTLOOK

6108 In this chapter, we introduced the simplest SCR model – “model SCR0” – which is an ordi-
 6109 nary capture-recapture model like model M_0 , but augmented with a set of latent individual

6110 effects, s_i , which relate encounter probability to some sense of individual location using a
6111 covariate, “distance”, from s_i to each trap location. Thus, individuals in close proximity
6112 to a trap will have a higher probability of encounter, and *vice versa*. The explicit modeling
6113 of individual locations and distance in this fashion resolves classical problems related to
6114 estimating density: unknown sample area, and heterogeneous encounter probability due
6115 to variable exposure to traps.

6116 SCR models are closely related to classical individual covariate models (“model M_x ”,
6117 as introduced in Chapt. 4), but with imperfect information about the individual covari-
6118 ate. Therefore, they are also not too dissimilar from standard GLMMs used throughout
6119 statistics and, as a result, we find that they are easy to analyze using standard MCMC
6120 methods encased in black boxes such as **WinBUGS** or **JAGS**. We will also see that they
6121 are easy to analyze using likelihood methods, which we address in Chapt. 6.

6122 Formal consideration of the collection of individual locations (s_1, \dots, s_N) is funda-
6123 mental to all models considered in this book. In statistical terminology, we think of the
6124 collection of points $\{s_i\}$ as a realization of a point process. Because SCR models formally
6125 link individual encounter history data to an underlying point process, we can obtain for-
6126 mal inferences about the point process. For example, we showed how to produce a density
6127 map (Fig. 5.6), or even a probability map for an individual’s home range center (Fig.
6128 5.7). We can also use SCR models as the basis for doing more traditional point process
6129 analyses, such as testing for “complete spatial randomness” (CSR) (see Chapt. 8), and
6130 computing other point process summaries (Illian et al., 2008).

6131 Part of the promise, and ongoing challenge, of SCR models is to develop models that
6132 reflect interesting biological processes, for example interactions among points or temporal
6133 dynamics in point locations. In this chapter we considered the simplest possible point
6134 process model in which points are independent and uniformly (“randomly”) distributed
6135 over space. Despite the simplicity of this model, it should suffice in many applications of
6136 SCR models, although we do address generalizations in later chapters. Moreover, even
6137 though the *prior* distribution on the point locations is uniform, the realized pattern may
6138 deviate markedly from uniformity as the observed encounter data provide information to
6139 impart deviations from uniformity. Thus, estimated density maps will typically appear
6140 distinctly non-uniform (as we saw in the wolverine example). In applications of the basic
6141 SCR model, we find that this simple *a priori* model can effectively reflect or adapt to
6142 complex realizations of the underlying point process. For example, if individuals are
6143 highly territorial then the data should indicate this in the form of individuals not being
6144 encountered in the same trap – the resulting posterior distribution of point locations should
6145 therefore reflect non-independence. Obviously the complexity of posterior estimates of the
6146 point pattern will depend on the quantity of data, both number of individuals and captures
6147 per individual. Because the point process is such an integral component of SCR models,
6148 the state-space of the point process plays an important role in developing SCR models.
6149 As we emphasized in this chapter, the state-space is part of the model. It can have an
6150 influence on parameter estimates and other inferences, such as model selection (see chapter
6151 8).

6152 One concept we introduced in this chapter, which has not been discussed much in
6153 the literature on SCR models, is the manner in which the encounter probability model
6154 relates to a model of space usage by individuals. The standard SCR models of encounter
6155 probability can all be motivated as simplistic models of space usage and movement, in

6156 which individuals make random use decisions from a probability distribution proportional
6157 to the encounter probability model. This both clarifies the simplicity of the underlying
6158 model of space usage and also suggests a direct extension to produce more realistic models,
6159 which we discuss in Chapt. 13. We consider some other important extensions of the basic
6160 SCR model in later chapters. For example, we consider models that include covariates that
6161 vary by individual, trap, or over time (Chapt. 7), spatial covariates on density (Chapt.
6162 11), open populations (Chapt. 16), and methods for model assessment and selection
6163 (Chapt. 8) among other topics. We also consider technical details of maximum likelihood
6164 (Chapt. 6) and Bayesian (Chapt. 17) estimation, so that the interested reader can develop
6165 or extend methods to suit their own needs.

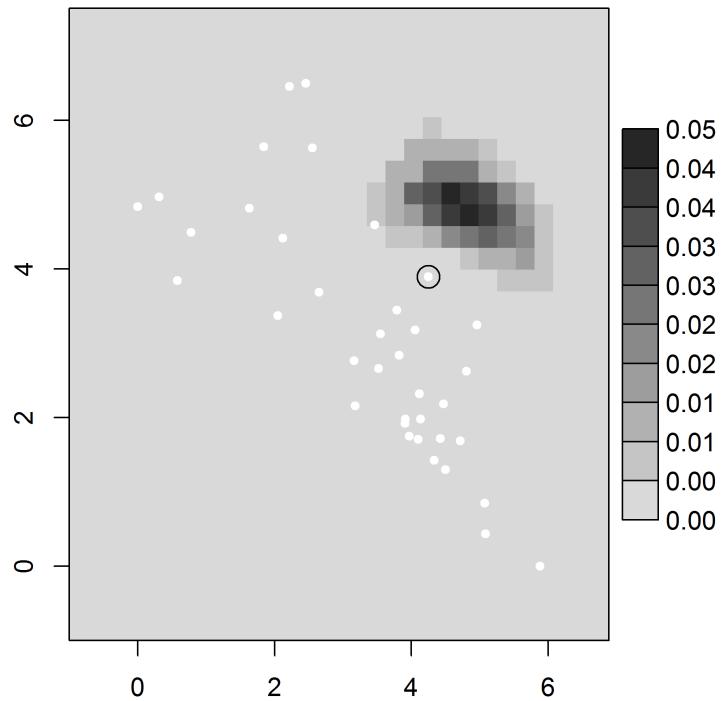


Figure 5.7. Posterior probability distribution of s_1 , the activity center for individual 1 in the wolverine data set. This individual was captured a single time in one trap (trap 30) which is circled. White dots are trap locations.

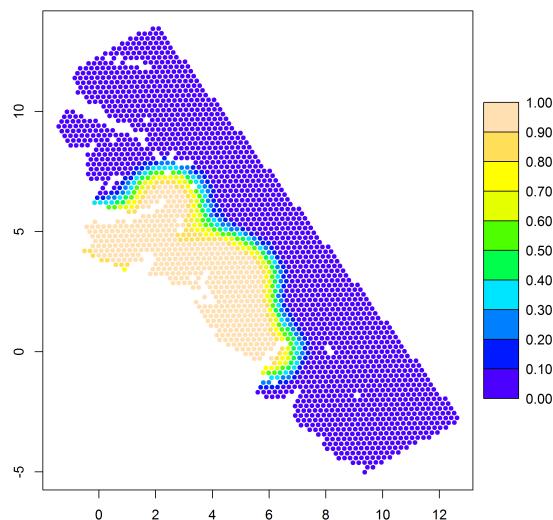


Figure 5.8. Probability of encounter used in computing effective sampled area for the wolverine camera trapping array, using the parameter estimates (posterior means) for the 2×2 km habitat mask.

6166

6167

6168

6169

6

LIKELIHOOD ANALYSIS OF SPATIAL CAPTURE-RECAPTURE MODELS

6170 We have so far mainly focused on Bayesian analysis of spatial capture-recapture models.
6171 And, in the previous chapters we learned how to fit some basic spatial capture-recapture
6172 models using a Bayesian formulation of the models analyzed in **BUGS** engines including
6173 **WinBUGS** and **JAGS**. Despite our focus on Bayesian analysis, it is instructive to de-
6174 velop the basic concepts and ideas behind classical analysis based on likelihood methods
6175 and frequentist inference for SCR models. We recognized earlier (Chapt. 5) that SCR
6176 models are versions of binomial (or other) GLMs, but with random effects (i.e., GLMMs).
6177 Throughout statistics, such models are routinely analyzed by likelihood methods. In par-
6178 ticular, likelihood analysis is based on the integrated or marginal likelihood in which the
6179 random effects are removed, by integration, from the conditional-on-s likelihood (s being
6180 the individual activity center). This has been the approach taken by Borchers and Ef-
6181 ford (2008); Dawson and Efford (2009) and related papers. Therefore, in this chapter, we
6182 provide some conceptual and technical foundation for likelihood-based analysis of spatial
6183 capture-recapture models.

6184 We will show here that it is straightforward to compute the maximum likelihood esti-
6185 mates (MLE) for SCR models by integrated likelihood. We develop the MLE framework
6186 using **R**, and we also provide a basic introduction to the **R** package **secr** (Efford, 2011a)
6187 which does likelihood analysis of SCR models (see also the stand-alone program **DEN-**
6188 **SITY** (Efford et al., 2004)). To set the context for likelihood analysis of SCR models,
6189 we first analyze the SCR model when N is known because, in that case, analysis is no
6190 different at all than a standard GLMM. We generalize the model to allow for unknown N
6191 using both conventional ideas based on the “full likelihood” (e.g., Borchers et al., 2002)
6192 and also using a formulation based on data augmentation. We obtain the MLEs for the
6193 SCR model from the wolverine camera trapping study (Magoun et al., 2011) analyzed in
6194 previous chapters to compare/contrast the results.

6.1 MLE WITH KNOWN N

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

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

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

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

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

$$y_{ij}|\mathbf{s}_i \sim \text{Binomial}(K, p_{\boldsymbol{\alpha}}(\mathbf{x}_j, \mathbf{s}_i)) \quad (6.1.1)$$

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

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

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

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

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

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

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

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

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

6224 We emphasize that two independence assumptions are explicit in this development: independence
 6225 of trap-specific encounters within individuals and also independence among
 6226 individuals. In particular, this would only be valid when individuals are not physically
 6227 restrained or removed upon capture, and when traps do not “fill up.”

6228 The key operation for computing the likelihood is solving a 2-dimensional integration
 6229 problem. There are some general purpose **R** packages that implement a number of multi-
 6230 dimensional integration routines including **adapt** (Genz et al., 2007) and **R2cuba** (Hahn
 6231 et al., 2010). In practice, we won’t rely on these extraneous **R** packages (except see Chapt.
 6232 11 for an application of **R2cuba**) but instead will use perhaps less efficient methods in which
 6233 we replace the integral with a summation over an equal area mesh of points on the state-
 6234 space \mathcal{S} and explicitly evaluate the integrand at each point. We invoke the rectangular
 6235 rule for integration here¹ in which we evaluate the integrand on a regular grid of points
 6236 of equal area and compute the average of the integrand over that grid of points. Let
 6237 $u = 1, 2, \dots, nG$ index a grid of nG points, \mathbf{s}_u , where the area of grid cells is constant, say
 6238 A . In this case, the integrand, i.e., the marginal pmf of \mathbf{y}_i , is approximated by

$$[\mathbf{y}_i | \boldsymbol{\alpha}] = \frac{1}{nG} \sum_{u=1}^{nG} [\mathbf{y}_i | \mathbf{s}_u, \boldsymbol{\alpha}] \quad (6.1.2)$$

6239 This is a specific case of the general expression that could be used for approximating
 6240 the integral for any arbitrary distribution $[\mathbf{s}]$. The general case is

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

6241 Under the uniformity assumption, $[\mathbf{s}] = 1/A(\mathcal{S})$ and thus the grid-cell area cancels in the
 6242 above expression to yield Eq. 6.1.2. The rectangular rule for integration can be seen as
 6243 an application of the Law of Total Probability for a discrete random variable \mathbf{s} , having
 6244 nG unique values with equal probabilities $1/nG$.

6245 6.1.1 Implementation (simulated data)

6246 Here we will illustrate how to carry out this integration and optimization based on the
 6247 integrated likelihood using simulated data (i.e., see Sec. 5.5). Using **simSCR0** we simulate
 6248 data for 100 individuals and an array of 25 traps laid out in a 5×5 grid of traps having unit
 6249 spacing. The specific encounter model is the Gaussian model. The 100 activity centers
 6250 were simulated on a state-space defined by an 8×8 square within which the trap array was
 6251 centered (thus the trap array is buffered by 2 units). Therefore, the density of individuals
 6252 in this system is fixed at 100/64. In the following set of **R** commands we generate the
 6253 data and then harvest the required data objects:

```
6254 ## simulate a complete data set (perfect detection)
6255 > data <- simSCR0(discard0=FALSE, rnd=2013)
6256     ## extract the objects that we need for analysis
6257 > y <- data$Y
```

¹e.g., http://en.wikipedia.org/wiki/Rectangle_method

```

6258 > traplocs <- data$traplocs
6259 > nind <- nrow(y) ## in this case nind=N
6260 > J <- nrow(traplocs)
6261 > K <- data$K
6262 > xlim <- data$xlim
6263 > ylim <- data$ylim

```

6264 Now, we need to define the integration grid, say **G**, which we do with the following set of
 6265 **R** commands (here, **delta** is the grid spacing):

```

6266 > delta <- .2
6267 > xg <- seq(xlim[1]+delta/2,xlim[2]-delta/2,by=delta)
6268 > yg <- seq(ylim[1]+delta/2,ylim[2]-delta/2,by=delta)
6269 > npix <- length(xg)           # valid for square state-space only
6270 > G <- cbind(rep(xg,npix),sort(rep(yg,npix)))
6271 > nG <- nrow(G)

```

6272 In this case, the integration grid is set up as a grid with spacing $\delta = 0.2$ which produces,
 6273 for our example, a 40×40 grid of points for evaluating the integrand if the state-space
 6274 buffer is set at 2. We note that the integration grid is set-up here to correspond exactly
 6275 to the state-space used in simulating the data. However, in practice, we wouldn't know
 6276 this, and our estimate of N (for the unknown case, see below) would be sensitive to choice
 6277 of the extent of the integration grid. As we've discussed previously, density, which is N
 6278 standardized by the area of the state-space, will not be so sensitive in most cases.

6279 We are now ready to compute the conditional-on-s likelihood and carry out the
 6280 marginalization described by Eq. 6.1.2. We need to do this by defining an **R** function
 6281 that computes the likelihood for the integration grid, as a function of the data objects
 6282 **y** and **traplocs** which were created above. However, it is a bit untidy to store the grid
 6283 information in your workspace, and define the likelihood function in a way that depends
 6284 on these things that exist in your workspace. Therefore, we build the **R** function so that
 6285 it computes the integration grid *within* the function, thereby avoiding potential problems
 6286 if our trapping grid locations change, or if we want to modify the state-space buffer easily.
 6287 We therefore define the function, called **intlik1**, to which we pass the data objects and
 6288 other information necessary to compute the marginal likelihood. This function is available
 6289 in the **scrbook** package (use **?intlik1** at the **R** prompt). The code is reproduced here:

```

6290 intlik1 <- function(parm,y=y,X=traplocs, delta=.2, ssbuffer=2){
6291
6292   Xl <- min(X[,1]) - ssbuffer ## These lines of code are setting up the
6293   Xu <- max(X[,1]) + ssbuffer ## support for the integration which is
6294   Yu <- max(X[,2]) + ssbuffer ## the same as the state-space of "s"
6295   Yl <- min(X[,2]) - ssbuffer
6296   xg <- seq(Xl+delta/2,Xu-delta/2,,length=npix)
6297   yg <- seq(Yl+delta/2,Yu-delta/2,,length=npix)
6298   npix<- length(xg)
6299
6300   G <- cbind(rep(xg,npix),sort(rep(yg,npix)))

```

```

6301   nG <- nrow(G)
6302   D <- e2dist(X,G)
6303
6304   alpha0 <- parm[1]
6305   alpha1 <- exp(parm[2]) # alpha1 restricted to be positive here
6306
6307   probcap <- plogis(alpha0)*exp(-alpha1*D*D)
6308   Pm <- matrix(NA,nrow=nrow(probcap),ncol=ncol(probcap))
6309       # Frequency of all-zero encounter histories
6310   n0 <- sum(apply(y,1,sum)==0)
6311       # Encounter histories with at least 1 detection
6312   ymat <- y[apply(y,1,sum)>0,]
6313   ymat <- rbind(ymat,rep(0,ncol(ymat)))
6314   lik.marg <- rep(NA,nrow(ymat))
6315
6316   for(i in 1:nrow(ymat)){
6317       ## Next line: log conditional likelihood for ALL possible values of s
6318       Pm[1:length(Pm)] <- dbinom(rep(ymat[i,],nG),K,probcap[1:length(Pm)],
6319           log=TRUE)
6320       ## Next line: sum the log conditional likelihoods, exp() result
6321       ## same as taking the product
6322       lik.cond <- exp(colSums(Pm))
6323       ## Take the average value == computing marginal
6324       lik.marg[i] <- sum(lik.cond*(1/nG))
6325   }
6326   ## n0 = number of all-0 encounter histories
6327   nv <- c(rep(1,length(lik.marg)-1),n0)
6328   return( -1*(sum(nv*log(lik.marg)) ) )
6329 }
```

6330 We emphasize that this function (and subsequent) are not meant to be general-purpose
6331 routines for solving all of your SCR problems but, rather, they are meant for illustrative
6332 purposes – so you can see how the integrated likelihood is constructed and how we connect
6333 it to data and other information that is needed.

6334 The function `intlik1` accepts as input the encounter history matrix, y , the trap locations,
6335 X , and the state-space buffer. This allows us to vary the state-space buffer and easily
6336 evaluate the sensitivity of the MLE to the size of the state-space. Note that we have a
6337 peculiar handling of the encounter history matrix y . In particular, we remove the all-zero
6338 encounter histories from the matrix and tack-on a single all-zero encounter history as the
6339 last row which then gets weighted by the number of such encounter histories ($n0$). This is
6340 a bit long-winded and strictly unnecessary when N is known, but we did it this way be-
6341 cause the extension to the unknown- N case is now transparent (as we demonstrate in the
6342 following section). The matrix Pm holds the log-likelihood contributions of each encounter
6343 frequency for each possible state-space location of the individual. The log contribu-
6344 tions are summed up and the result exponentiated on the next line, producing `lik.cond`, the
6345 conditional-on- s likelihood (Eq. 6.1.1 above). The marginal likelihood (`lik.marg`) sums
6346 up the conditional elements weighted by the probabilities [s] (Eq. 6.1.2 above).

6347 This is a fairly primitive function which doesn't allow much flexibility in the data
 6348 structure. For example, it assumes that K , the number of replicates, is constant for each
 6349 trap. Further, it assumes that the state-space is a square. We generalize this to some
 6350 extent later in this chapter.

6351 Here is the **R** command for maximizing the likelihood using **nlm** (the function **optim**
 6352 could also be used) and saving the results into an object called **frog**. The output is a list
 6353 of the following structure and these specific estimates are produced using the simulated
 6354 data set:

```
6355 # should take 15-30 seconds
6356
6357 > starts <- c(-2,2)
6358 > frog <- nlm(intlik1,starts,y=y,X=traplocs,delta=.1,ssbuffer=2,hessian=TRUE)
6359 > frog
6360
6361 $minimum
6362 [1] 297.1896
6363
6364 $estimate
6365 [1] -2.504824  2.373343
6366
6367 $gradient
6368 [1] -2.069654e-05  1.968754e-05
6369
6370 $hessian
6371      [,1]      [,2]
6372 [1,]  48.67898 -19.25750
6373 [2,] -19.25750  13.34114
6374
6375 $code
6376 [1] 1
6377
6378 $iterations
6379 [1] 11
```

6380 Details about this output can be found on the help page for **nlm**. We note briefly that
 6381 **frog\$minimum** is the negative log-likelihood value at the MLEs, which are stored in the
 6382 **frog\$estimate** component of the list. The order of the parameters is as they are defined
 6383 in the likelihood function so, in this case, the first element (value = -2.504824) is the
 6384 logit transform of p_0 and the second element (value = 2.373343) is the value of α_1 the
 6385 “coefficient” on distance-squared. The Hessian is the observed Fisher information matrix,
 6386 which can be inverted to obtain the variance-covariance matrix using the command:

```
6387 > solve(frog$hessian)
```

6388 It is worth drawing attention to the fact that the estimates are slightly different than
 6389 the Bayesian estimates reported previously in Sec. 5.6. There are several reasons for this.
 6390 First Bayesian inference is based on the posterior distribution and it is not generally the

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

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

6.2 MLE WHEN N IS UNKNOWN

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

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

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

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

6428 (2) We compute the probability of an all-0 encounter history:

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

6429 (3) The combinatorial term: $\binom{N}{n}$. Then, the marginal likelihood has this form:

$$\mathcal{L}(\boldsymbol{\alpha}, n_0 | \mathbf{y}) = \frac{N!}{n! n_0!} \left\{ \prod_{i=1}^n [\mathbf{y}_i | \boldsymbol{\alpha}] \right\} \pi_0^{n_0}. \quad (6.2.1)$$

6430 This is discussed in Borchers and Efford (2008, p. 379) as the conditional-on- N form of the
6431 likelihood – we also call it the “binomial form” of the likelihood because of its appearance.

6432 Operationally, things proceed much as before: We compute the marginal probability
6433 of each observed \mathbf{y}_i , i.e., by removing the latent \mathbf{s}_i by integration. In addition, we com-
6434 pute the marginal probability of the “all-zero” encounter history \mathbf{y}_{n+1} , and make sure to
6435 weight it n_0 times. We accomplish this by “padding” the data set with a single encounter
6436 history having $y_{n+1,j} = 0$ for all traps $j = 1, 2, \dots, J$. Then we be sure to include the
6437 combinatorial term in the likelihood or log-likelihood computation. We demonstrate this
6438 shortly. To analyze a specific case, we’ll simulate our fake data set (simulated using the
6439 parameters given above). To set some things up in our workspace we do this:

```
6440 ## Obtain a simulated data set
6441 > data <- simSCRO(discard0=TRUE, rnd=2013)
6442
6443 ## Extract the items we need for analysis
6444 > y <- data$Y
6445 > nind <- nrow(y)
6446 > traplocs <- data$traplocs
6447 > J <- nrow(traplocs)
6448 > K <- data$K
```

6449 Recall that these data are simulated by default with $N = 100$, on an 8×8 unit state-
6450 space representing the trap locations buffered by 2 units, although you can modify the
6451 simulation script easily.

6452 As before, the likelihood is defined in the **R** workspace as an **R** function, **intlik2**,
6453 which takes an argument being the unknown parameters of the model and additional
6454 arguments as prescribed. In particular, we provide the encounter history matrix **y**, the
6455 trap locations **traplocs**, the spacing of the integration grid (argument **delta**) and the
6456 state-space buffer. Here is the new likelihood function:

```
6457 intlik2 <- function(parm,y=y,X=traplocs,delta=.3,ssbuffer=2){
6458
6459   Xl <- min(X[,1]) - ssbuffer
6460   Xu <- max(X[,1]) + ssbuffer
6461   Yu <- max(X[,2]) + ssbuffer
6462   Yl <- min(X[,2]) - ssbuffer
6463
6464   xg <- seq(Xl+delta/2,Xu-delta/2,delta)
```

```

6465   yg <- seq(Yl+delta/2,Yu-delta/2,delta)
6466   npix.x <- length(xg)
6467   npix.y <- plength(yg)
6468   area <- (Xu-Xl)*(Yu-Yl)/((npix.x)*(npix.y))
6469   G <- cbind(rep(xg,npix.y),sort(rep(yg,npix.x)))
6470   nG <- nrow(G)
6471   D <- e2dist(X,G)
6472   # extract the parameters from the input vector
6473   alpha0 <- parm[1]
6474   alpha1 <- exp(parm[2])
6475   n0 <- exp(parm[3]) # note parm[3] lives on the real line
6476   probcap <- plogis(alpha0)*exp(-alpha1*D*D)
6477   Pm <- matrix(NA,nrow=nrow(probcap),ncol=ncol(probcap))
6478   ymat <- rbind(y,rep(0,ncol(y)))
6479
6480   lik.marg <- rep(NA,nrow(ymat))
6481   for(i in 1:nrow(ymat)){
6482     Pm[1:length(Pm)] <- (dbinom(rep(ymat[i,],nG),K,probcap[1:length(Pm)],
6483                                     log=TRUE))
6484     lik.cond <- exp(colSums(Pm))
6485     lik.marg[i] <- sum(lik.cond*(1/nG) )
6486   }
6487   nv <- c(rep(1,length(lik.marg)-1),n0)
6488   ## part1 here is the combinatorial term.
6489   ## math: log(factorial(N)) = lgamma(N+1)
6490   part1 <- lgamma(nrow(y)+n0+1) - lgamma(n0+1)
6491   part2 <- sum(nv*log(lik.marg))
6492   return( -1*(part1+ part2) )
6493 }
```

6494 To execute this function for the data that we created with `simSCR0`, we execute the
 6495 following command (saving the result in our friend `frog`). This results in the usual output,
 6496 including the parameter estimates, the gradient, and the numerical Hessian which is useful
 6497 for obtaining asymptotic standard errors (see below):

```

6498 > starts <- c(-2.5,0,4)
6499 > frog <- nlm(intlik2,starts,hessian=TRUE,y=y,X=traplocs,delta=.2,ssbuffer=2)
6500
6501 Warning message:
6502 In nlm(intlik2, starts, hessian = TRUE, y = y, X = traplocs, delta = 0.2, :
6503 NA/Inf replaced by maximum positive value
6504
6505 > frog
6506 $minimum
6507 [1] 113.5004
6508
6509 $estimate
```

```
6510 [1] -2.538333 0.902807 4.232810
6511
6512 [... additional output deleted ...]
```

6513 Executing `nlm` here usually produces one or more **R** warnings due to numerical calculations
 6514 happening on extremely small or large numbers (calculation of p near the edge of the
 6515 state-space), and they also happen if a poor parameterization is used which produces
 6516 evaluations of the objective function beyond the boundary of the parameter space (e.g.,
 6517 $n_0 < 0$). Such numerical warnings can often be minimized or avoided altogether by picking
 6518 judicious starting values of parameters or properly transforming or scaling the parameters
 6519 but, in general, they can be ignored. You will see from the `nlm` output that the algorithm
 6520 performed satisfactory in minimizing the objective function. The estimate of population
 6521 size, \hat{N} , for the state-space (using the default state-space buffer) is

```
6522 > Nhat <- nrow(y) + exp(4.2328) #### This is n + MLE of n0
6523 > Nhat
6524 [1] 110.9099
```

6525 Which differs from the data-generating value ($N = 100$), as we might expect for a single
 6526 realization. We usually will present an estimate of uncertainty associated with this MLE
 6527 which we can obtain by inverting the Hessian. Note that $\text{Var}(\hat{N}) = n + \text{Var}(\hat{n}_0)$. Since
 6528 we have parameterized the model in terms of $\log(n_0)$ we use the delta method described
 6529 in Williams et al. (2002, Appendix F4) (see also Ver Hoef, 2012) to obtain the variance
 6530 on the scale of n_0 as follows:

```
6531 > (exp(4.2328)^2)*solve(frog$hessian)[3,3]
6532 [1] 260.2033
6533
6534 > sqrt(260)
6535 [1] 16.12452
```

6536 Therefore, the asymptotic “Wald-type” confidence interval for N is $110.91 \pm 1.96 \times 16.125 =$
 6537 $(79.305, 142.515)$. To report this in terms of density, we scale appropriately by the area
 6538 of the prescribed state-space which is 64 units of area (i.e., an 8×8 square). Our MLE
 6539 of D is $\hat{D} = 110.91/64 = 1.733$ individuals per square unit. To get the standard error
 6540 for \hat{D} we need to divide the SE for \hat{N} by the area of the state-space, and so $\text{SE}(\hat{D}) =$
 6541 $(1/64) * 16.12452 = 0.252$.

6542 **6.2.1 Integrated likelihood under data augmentation**

6543 The likelihood analysis developed in the previous sections is based on the likelihood in
 6544 which N (or n_0) is an explicit parameter. This is usually called the “full likelihood” or
 6545 sometimes “unconditional likelihood” (Borchers et al., 2002) because it is the likelihood
 6546 for all individuals in the population, not just those which have been captured, i.e., not that
 6547 which is *conditional on capture*. It is also possible to express an alternative unconditional
 6548 likelihood using data augmentation, replacing the parameter N with ψ (e.g., see Sec. 7.1.6
 6549 Royle and Dorazio, 2008, for an example). We don’t go into detail here, but we note that
 6550 the likelihood under data augmentation is a zero-inflated binomial mixture – precisely an

occupancy type model (Royle, 2006). Thus, while it is possible to carry out likelihood analysis of models under data augmentation, we primarily advocate data augmentation for Bayesian analysis.

6.2.2 Extensions

We have only considered basic SCR models with no additional covariates. However, in practice, we are interested in covariate effects including “behavioral response”, sex-specificity of parameters, and potentially others. Some of these can be added directly to the likelihood if the covariate is fixed and known for all individuals captured or not. An example is a behavioral response, which amounts to having a covariate $x_{ik} = 1$ if individual i was captured prior to occasion k and $x_{ik} = 0$ otherwise. For uncaptured individuals, $x_{ik} = 0$ for all k . Royle et al. (2011b) called this a global behavioral response because the covariate is defined for all traps, no matter the trap in which an individual was captured. We could also define a *local* behavioral response which occurs at the level of the trap, i.e., $x_{ijk} = 1$ if individual i was captured in trap j prior to occasion k , etc... Trap-specific covariates such as trap type or status, or time-specific covariates such as date, are easily accommodated as well. As an example, Kéry et al. (2010) develop a model for the European wildcat *Felis silvestris* in which traps are either baited or not (a trap-specific covariate with only 2 values), and also encounter probability varies over time in the form of a quadratic seasonal response. We consider models with behavioral response or fixed covariates in Chapt. 7. The integrated likelihood routines we provided above can be modified directly for such cases, which we leave to the interested reader to investigate.

Sex-specificity is more difficult to deal with since sex is not known for uncaptured individuals (and sometimes not even for all captured individuals). To analyze such models, we do Bayesian analysis of the joint likelihood using data augmentation (Gardner et al., 2010b; Russell et al., 2012), discussed further in Chapt. 7. For such covariates (i.e., that are not fixed and known for all individuals), it is somewhat more challenging to do MLE based on the joint likelihood as we have developed above. Instead it is more conventional to use what is colloquially referred to as the “Huggins-Alho” type model which is one of the approaches taken in the software package **secr** (Efford, 2011a). We introduce the **secr** package in Sec. 6.5 below.

6.3 CLASSICAL MODEL SELECTION AND ASSESSMENT

In most analyses, one is interested in choosing from among various potential models, or ranking models, or something else to do with assessing the relative merits of a set of models. A good thing about classical analysis based on likelihood is we can apply Akaike Information Criterion (AIC) methods (Burnham and Anderson, 2002) without difficulty. AIC is convenient for assessing the relative merits of these different models although if there are only a few models it is not objectionable to use hypothesis tests or confidence intervals to determine importance of effects. A second model selection context has to do with choosing among various detection models, although, as a general rule, we don't recommend this application of model selection. This is because there is hardly ever (if at all) a rational subject-matter based reason motivating specific distance functions. As a result, we believe that doing too much model selection will invariably lead to over-fitting

and thus over-statement of precision. This is the main reason that we haven't loaded you down with a basket of models for detection probability so far, although we discuss many possibilities in Chapt. 7.

Goodness-of-fit or model-checking – For many standard capture-recapture models, it is possible to identify goodness-of-fit statistics based on the multinomial likelihood, (Cooch and White, 2006, Chapt. 5), and evaluate model adequacy using formal statistical tests. Similar strategies can be applied to SCR models using expected cell-frequencies based on the marginal distribution of the observations. Also, because computing MLEs is somewhat more efficient in many cases compared to Bayesian analysis, it is sometimes feasible to use bootstrap methods. At the present time, we don't know of any applications of goodness-of-fit testing for SCR models based on likelihood inference, although we discuss the use of Bayesian p-values for assessing model fit in Chapt. 8. An important practical problem in trying to evaluate goodness-of-fit is that, in realistic sample sizes, fit tests often lack the power to detect departures from the model under consideration and so they may not be generally useful in practice.

6.4 LIKELIHOOD ANALYSIS OF THE WOLVERINE CAMERA TRAPPING DATA

Here we compute the MLEs for the wolverine data using an expanded version of the function we developed in the previous section. To accommodate that each trap might be operational a variable number of nights, we provided an additional argument to the likelihood function (allowing for a vector $\mathbf{K} = (K_1, \dots, K_J)$), which requires also a modification to the construction of the likelihood. In addition, we accommodate the state-space is a general rectangle, and we included a line in the code to compute the state-space area which we apply below for computing density. The more general function (`intlik3`) is given in the **R** package `scrbook`. Incidentally, this function also returns the area of the state-space for a given set of parameter values, as an attribute to the function value, which will be used in converting \hat{N} to \hat{D} . To use this function to obtain the MLEs for the wolverine camera trap study, we execute the following commands (note: these are in the help file and will execute if you type `example(intlik3)`):

```

6619 > library(scrbook)
6620 > data(wolverine)
6621
6622 > traps <- wolverine$traps
6623 > traplocs <- traps[,2:3]/10000
6624 > K.wolv <- apply(traps[,4:ncol(traps)],1,sum)
6625
6626 > y3d <- SCR23darray(wolverine$wcaps,traps)
6627 > y2d <- apply(y3d,c(1,2),sum)
6628
6629 > starts <- c(-1.5,0,3)
6630
6631 > wolv <- nlm(intlik3,starts,hessian=TRUE,y=y2d,K=K.wolv,X=traplocs,
6632           delta=.2,ssbuffer=2)
6633

```

```

6634 > wolv
6635 $minimum
6636 [1] 220.4313
6637
6638 $estimate
6639 [1] -2.8176120 0.2269395 3.5836875
6640
6641 [.... output deleted ....]

```

6642 Of course we're interested in obtaining an estimate of population size for the prescribed
 6643 state-space, or density, and associated measures of uncertainty which we do using the delta
 6644 method (Williams et al., 2002, Appendix F4). To do all of that we need to manipulate the
 6645 output of `nlm` since we have our estimate in terms of $\log(n_0)$. We execute the following
 6646 commands:

```

6647 > wolv <- nlm(intlik3,starts,hessian=TRUE,y=y2d,K=K.wolv,X=traplocs,delta=.2,
6648           ssbuffer=2)
6649 > Nhat <- nrow(y2d)+exp(wolv$estimate[3])
6650 > area <- attr(intlik3(starts,y=y2d,K=K.wolv,X=traplocs,delta=.2,ssbuffer=2),
6651           "SSarea")
6652 > Dhat <- Nhat/area
6653
6654 > Dhat
6655 [1] 0.5494947
6656
6657 > SE <- (1/area)*exp(wolv$estimate[3])*sqrt(solve(wolv$hessian)[3,3])
6658
6659 > SE
6660 [1] 0.1087073

```

6661 Our estimate of density is 0.55 individuals per “standardized unit” which is 100 km^2 ,
 6662 because we divided UTM coordinates by 10000. So this is about 5.5 individuals per 1000
 6663 km^2 , with a SE of around 1.09 individuals. This compares closely with 5.77 reported in
 6664 Sec. 5.9 based on Bayesian analysis of the model.

6665 6.4.1 Sensitivity to integration grid and state-space buffer

6666 The effect of approximating the integral by a discrete mesh of points is that it induces
 6667 some numerical error in evaluation of the integral and, further, that error increases as the
 6668 coarseness of the mesh increases. To evaluate the effect (or sensitivity) of the integration
 6669 grid spacing, we obtained the MLEs for a state-space buffer of 2 (standardized units) and
 6670 for integration grid with spacing $\delta = .3, .2, .1, .05$. The MLEs for these 4 cases including
 6671 the relative runtime are given in Table 6.1. We see the results change only slightly as the
 6672 integration grid changes. Conversely, the runtime on the platform of the day for the 4 cases
 6673 increases rapidly. These runtimes could be regarded in relative terms, across platforms,
 6674 for gaging the decrease in speed as the fineness of the integration grid increases.

6675 We studied the effect of the state-space buffer on the MLEs, using a fixed $\delta = .2$ for
 6676 all analyses. We used state-space buffers of 1 to 4 units stepped by .5. As we can see

Table 6.1. Runtime and MLEs for different integration grid resolutions for the wolverine camera trapping data.

δ	Estimates			
	runtime (s)	$\hat{\alpha}_0$	$\hat{\alpha}_1$	$\log(\hat{n}_0)$
0.30	9.9	-2.819786	1.258468	3.569731
0.20	32.3	-2.817610	1.254757	3.583690
0.10	115.1	-2.817570	1.255112	3.599040
0.05	407.3	-2.817559	1.255281	3.607158

(Table 6.2), the estimates of D stabilize rapidly and the incremental difference is within the numerical error associated with approximating the integral.

Table 6.2. Results of the effect of the state-space buffer on the MLE. Given here are the state-space buffer, area of the state-space (area), the MLE of N (\hat{N}) for the prescribed state-space and the corresponding MLE of density (\hat{D}).

Buffer	Area	\hat{N}	\hat{D}
1.0	66.98212	37.73338	0.5633352
1.5	84.36242	46.21008	0.5477567
2.0	103.74272	57.00617	0.5494956
2.5	125.12302	69.03616	0.5517463
3.0	148.50332	82.17550	0.5533580
3.5	173.88362	96.44018	0.5546249
4.0	201.26392	111.83524	0.5556646

6.4.2 Using a habitat mask (Restricted state-space)

In Sec. 5.10 we used a discrete representation of the state-space in order to have control over its extent and shape. This makes it easy to do things like clip out non-habitat, or create a *habitat mask* which defines suitable habitat. Clearly that formulation of the model is relevant to the calculation of the marginal likelihood in the sense that the discrete state-space is equivalent to the integration grid. Thus, for example, we could easily compute the MLE of parameters under some model with a restricted state-space merely by creating the required state-space at whatever grid resolution is desired, and then inputting that state-space into the likelihood function above, instead of computing it within the function. We can easily create an explicit state-space grid for integration from arbitrary polygons or GIS shapefiles which we demonstrate here. Our approach is to create the integration grid (or state-space grid) outside of the likelihood evaluation, and then determine which points of the grid lie in the polygon defined by the shapefile using functions in the **R** packages **sp** 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 polygon², identifying such points

²We 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

6694 with a value of `mask=1` and `mask=0` for points that are *not* in the polygon. We load the
 6695 shapefile which originates by an application of the `readShapeSpatial` function. We have
 6696 saved the result into an **R** data object called `SSp` which is in the `scrbook` package. Here
 6697 are the **R** commands for doing this (see the helpfile `?intlik4`):

```
6698 > library(mapproj)
6699 > library(sp)
6700 > library(scrbook)
6701
6702 ##### If we have the .shp file in place, we would use this command:
6703 ##### SSp <- readShapeSpatial('Sim_Polygon.shp')
6704 ##### The object SSp is in data(fakeshapefile)
6705 > data(fakeshapefile)
6706 > Pcoord <- SpatialPoints(G)
6707 > PinPoly <- over(Pcoord,SSp) #### determine if each point is in polygon
6708 > mask <- as.numeric(!is.na(PinPoly[,1])) ## convert to binary 0/1
6709 > G <- G[mask==1,]
```

6710 We created the function `intlik4` which accepts the integration grid as an explicit argument,
 6711 and this function is also available in the package `scrbook`.

6712 We apply this modification to the wolverine camera trapping study. Royle et al.
 6713 (2011b) created 2, 4 and 8 km state-space grids so as to remove “non-habitat” (mostly
 6714 ocean, bays, and large lakes). We previously analyzed the model using **JAGS** and **Win-**
6715 BUGS in Chapt. 5. To set up the wolverine data and fit the model using maximum
 6716 likelihood we execute the following commands:

```
6717 > library(scrbook)
6718 > data(wolverine)
6719
6720 > traps <- wolverine$wtraps
6721 > traplocs <- traps[,2:3]/10000
6722 > K.wolv <- apply(traps[,4:ncol(traps)],1,sum)
6723
6724 > y3d <- SCR23darray(wolverine$wcaps,traps)
6725 > y2d <- apply(y3d,c(1,2),sum)
6726 > G <- wolverine$grid2/10000
6727
6728 > starts <- c(-1.5,0,3)
6729 > wolv <- nlm(intlik4, starts, y=y2d, K=K.wolv, X=traplocs, G=G)
6730
6731 > wolv
```

can hold additional information for each polygon, and the output value of the function differs slightly for these two classes: if using a “`SpatialPolygons`” object, the function returns a vector of length equal to the number of points (e.g., in the example above), but if using a “`SpatialPolygonsDataFrame`” it returns a data frame (e.g., see Sec. 17.7 in Chapt. 17). If you use the `over` function, make sure you know the class of your second argument so that when processing the function output you index it correctly.

Table 6.3. Maximum likelihood estimates (MLEs) and asymptotic standard errors (SE) for the wolverine camera trapping data using 2, 4 and 8 km state-space grids.

Grid	α_0	α_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

```

6732
6733 $minimum
6734 [1] 225.8355
6735
6736 $estimate
6737 [1] -2.9955424 0.2350885 4.1104757
6738
6739 [... some output deleted ...]

```

6740 Next we convert the parameter estimates to estimates of total population size for the
6741 prescribed state-space, and then obtain an estimate of density (per 1000 km²) using the
6742 area computed as the number of pixels in the state-space grid, G, multiplied by the area
6743 per grid cell. In the present case (the calculation above) we used a state-space grid with 2
6744 km × 2 km pixels. Finally, we compute a standard errors using the delta approximation:

```

6745 > area <- nrow(G)*4
6746 # Nhat = n (observed) + MLE of n0 (not observed)
6747 > Nhat <- 21 + exp(wolv$estimate[3])
6748 > SE <- exp(wolv$estimate[3])*sqrt(solve(wolv$hessian)[3,3])
6749 > D <- (Nhat/(nrow(G)*area))*1000
6750 > SE.D <- (SE/(nrow(G)*area))*1000

```

6751 We did this for each the 2 km, 4 km and 8 km state-space grids which produced the
6752 estimates summarized in Table 6.3. These estimates compare with the 8.6 (2 km grid)
6753 and 8.2 (8 km grid) reported in Royle et al. (2011b) based on a clipped state-space as
6754 described in Sec. 5.10.

6.5 DENSITY AND THE R PACKAGE SECR

6755 **DENSITY** is a software program developed by Efford (2004) for fitting spatial capture-
6756 recapture models based mostly on classical maximum likelihood estimation and related
6757 inference methods. Efford (2011a) has also released an **R** package called **secr**, that con-
6758 tains much of the functionality of **DENSITY** but also incorporates new models and
6759 features. Here, we briefly introduce the **secr** package which we prefer to use over **DEN-**
6760 **SITY**, because it allows us to remain in the **R** environment for data processing and
6761 summarization. We provide a brief introduction to **secr** and some of its capabilities here,
6762 and we also use it for doing some analysis in other parts of this book. We believe that **secr**
6763 will be sufficient for many (if not most) of the SCR problems that one might encounter.
6764 It provides a flexible analysis platform, with a large number of summary features, and

6765 “publication ready” output. Its user-interface is clean and intuitive to **R** users, and it has
 6766 been stable, efficient and reliable in the (fairly extensive) evaluations that we have done.

6767 To install and run models in **secr**, you must download the package and load it in **R**.

```
6768 > install.packages("secr")
6769 > library(secr)
```

6770 **secr** allows the user to simulate data and fit a suite of models with various detection func-
 6771 tions and covariate responses. It also contains a number of helpful constructor functions
 6772 for creating objects of the proper class that are recognized by other **secr** functions. We
 6773 provide a brief overview of the capabilities here, but the **secr** help manual can be accessed
 6774 with the command:

```
6775 > RShowDoc("secr-manual", package = "secr")
```

6776 We note that **secr** has many capabilities that we will not cover or do so only sparingly.
 6777 We encourage you to read through the manual, the extensive documentation, and the
 6778 vignettes, in order to get a better understanding of what the package is capable of. We
 6779 also cover certain capabilities of **secr** in other chapters.

6780 The main model-fitting function in **secr** is called **secr.fit**, which makes use of the
 6781 standard **R** model specification framework with tildes. As an example, the equivalent of
 6782 the basic model SCR0 is fitted as follows:

```
6783 > secr.fit(capturedata, model = list(D ~ 1, g0 ~ 1, sigma ~ 1),
6784   buffer = 20000)
```

6785 where **capturedata** is the object created by **secr** containing the encounter history data
 6786 and the trap information, and the model expression $g0^1$ indicates the intercept-only (i.e.,
 6787 constant) model. Note that we use p_0 for the baseline encounter probability parameter,
 6788 which is g_0 in **secr** notation. A number of possible models for encounter probability can
 6789 be fitted including both pre-defined variables (e.g., **t** and **b** corresponding to “time” and
 6790 “behavior”), and user-defined covariates of several kinds. For example, to include a global
 6791 behavioral response, this would be written as $g0^1b$. The discussion of this (global versus
 6792 local trap-specific behavioral response) and other covariates is developed more in Chapt.
 6793 7. We can also model covariates on density in **secr**, which we discuss in Chapt. 11. It
 6794 is important to note that **secr** requires the buffer distance to be defined in meters and
 6795 density will be returned as number of animals per hectare. Thus to make comparisons
 6796 between **secr** and output from other programs, we will often have to convert the density
 6797 to the same units.

6798 Before we can fit the models, the data must first be packaged properly for **secr**.
 6799 We require data files that contain two types of information: trap layout (location and
 6800 identification information for each trap), which is equivalent to the trap deployment file
 6801 (TDF) described in Sec. 5.9 and the capture data file containing sampling *session*, animal
 6802 identification, trap occasion, and trap location, equivalent in information content to the
 6803 encounter data file (EDF). Sample session can be thought of as primary period identifier
 6804 in a robust design like framework – it could represent a yearly sample or multiple sample
 6805 periods within a year, each of them producing data on a closed population. We discuss
 6806 “multi-session” models in more detail below, in Sec. 6.5.4 and Chapt. 14.

6807 There are three important constructor functions that help package-up your data for
 6808 use in **secr**: **read.traps**, **make.capthist** and **read.mask**. We provide a brief description
 6809 of each here, but apply them to our wolverine camera trapping data in the next section:

6810 (1) **read.traps**: This function points to an external file or **R** data object containing the
 6811 trap coordinates, and other information, and also requires specification of the type of
 6812 encounter devices (described in the next section). A typical application of this function
 6813 looks like the following, invoking the **data=** option when there is an existing **R** object
 6814 containing the trap information:

6815 > trapfile <- **read.traps**(**data=traps**, **detector="proximity"**)

6816 (2) **make.capthist**: This function takes the EDF and combines it with trap information,
 6817 and the number of sampling occasions. A typical application looks like this:

6818 > capturedata <- **make.capthist**(**enc.data**, **trapfile**, **fmt="trapID"**,
 6819 **noccasions=165**)

6820 See **?make.capthist** for definition of distinct file formats. Specifying **fmt = trapID** is
 6821 equivalent to our EDF format.

6822 (3) **read.mask**: If there is a habitat mask available (as described in sec. 6.4.2), then this
 6823 function will organize it so that **secr.fit** knows what to do with it. The function
 6824 accepts either an external file name (see **?read.mask** for details of the structure) or a
 6825 $NG \times 2$ **R** object, say **mask.coords**, containing the coordinates of the mask. A typical
 6826 application looks like the following:

6827 > grid <- **read.mask**(**data=mask.coords**)

6828 These constructor functions produce output that can then be used in the fitting of models
 6829 using **secr.fit**.

6830 6.5.1 Encounter device types and detection models

6831 The **secr** package requires that you specify the type of encounter device. Instead of
 6832 describing models by their statistical distribution (Bernoulli, Poisson, etc..), **secr** uses
 6833 certain operational classifications of detector types including ‘proximity’, ‘multi’, ‘single’,
 6834 ‘polygon’ and ‘signal’. For camera trapping/hair snares we might consider ‘proximity’
 6835 detectors or ‘count’ detectors. The ‘proximity’ detector type allows, at most, one detection
 6836 of each individual at a particular detector on any occasion (i.e., it is equivalent to what
 6837 we call the Bernoulli or binomial encounter process model, or model SCR0). The ‘count’
 6838 detector designation allows repeat encounters of each individual at a particular detector
 6839 on any occasion. There are other detector types that one can select such as: ‘polygon’
 6840 detector type which allows for a trap to be a sampled polygon (Royle and Young, 2008)
 6841 which we discuss further in Chapt. 15, and ‘signal’ detector which allows for traps that
 6842 have a strength indicator, e.g., acoustic arrays (Dawson and Efford, 2009). The detector
 6843 types ‘single’ and ‘multi’ refer to traps that retain individuals, thus precluding the ability
 6844 for animals to be captured in other traps during the sampling occasion. The ‘single’ type
 6845 indicates trap that can only catch one animal at a time (single-catch traps), while ‘multi’
 6846 indicates traps that may catch more than one animal at a time (multi-catch). These are
 6847 both variations of the multinomial encounter models described in Chapt. 9.

6848 As with all SCR models, **secr** fits an encounter probability model (“detection function”
 6849 in **secr** terminology relating the probability of encounter to the distance of a detector from
 6850 an individual activity center. **secr** allows the user to specify one of a variety of detection
 6851 functions including the commonly used half-normal (“Gaussian”), hazard rate (“Gaussian
 6852 hazard”), and (negative) exponential models. There are 12 different functions as of version
 6853 2.3.1 (see Table 7.1 in Chapt. 7), but some are only available for simulating data. The
 6854 different detection functions are defined in the **secr** manual and can be found by calling
 6855 the help function for the detection function:

6856 > ?detectfn

6857 Most of the detection functions available in **secr** contain some kind of a scale parameter
 6858 which is usually labeled σ . The units of this parameter default to meters in the **secr**
 6859 output. We caution that the meaning of this parameter depends on the specific detection
 6860 model being used, and it should not be directly compared as a measure of home-range size
 6861 across models. Instead, as we noted in Sec. 5.4 most encounter probability models imply
 6862 a model of space-usage and fitted encounter models should be converted to a common
 6863 currency such as “area used.”

6864 6.5.2 Analysis using the **secr** package

6865 To demonstrate the use of the **secr** package, we will show how to do the same analysis on
 6866 the wolverine study as shown in Sec. 5.9. To use the **secr** package, the data need to be
 6867 formatted in a similar but slightly different manner than we use in **WinBUGS**.

6868 For example, in Sec. 5.9 we introduced a standard data format for the encounter data
 6869 file (EDF) and trap deployment file (TDF). The EDF shares the same format as that used
 6870 by the **secr** package with 1 row for every encounter observation and 4 columns representing
 6871 trap session (‘Session’), individual identity (‘ID’), sample occasion (‘Occasion’), and trap
 6872 identity (‘trapID’). For a standard closed population study that takes place during a single
 6873 season, the ‘Session’ column in our case is all 1’s, to indicate a single primary sampling
 6874 occasion. In addition to providing the encounter data file (EDF), we must tell **secr** infor-
 6875 mation about the traps, which is formated as a matrix with column labels ‘trapID’, ‘x’ and
 6876 ‘y’, the last two being the coordinates of each trap, with additional columns representing
 6877 the operational state of each trap during each occasion (1=operational, 0=not).

6878 We demonstrate these differences now by walking through an analysis of the wolverine
 6879 camera trapping data using **secr**. To read in the trap locations and other related infor-
 6880 mation, we make use of the constructor function **read.traps** which also requires that we
 6881 specify the detector type. The detector type is important because it will determine the
 6882 likelihood that **secr** will use to fit the model. Here, we have selected “proximity” which
 6883 corresponds to the Bernoulli encounter model in which individuals are captured at most
 6884 once in each trap during each sampling occasion:

```
6885 > library(secr)
6886 > library(scrbook)
6887 > data(wolverine)
6888
6889 > traps <- as.matrix(wolverine$wtraps)
```

```

6890 > dimnames(traps) <- list(NULL,c("trapID","x","y",paste("day",1:165,sep="")))
6891 > traps1 <- as.data.frame(traps[,1:3])
6892 > trapfile1 <- read.traps(data=traps1,detector="proximity")

```

6893 Here we note that trap coordinates are extracted from the wolverine data but we do
6894 *not* scale them. This is because **secr** defaults to coordinate scaling of meters which is
6895 the extant scaling of the wolverine trap coordinates. Note that we add a 'trapID' column
6896 to the trap coordinates and provide appropriate column labels to the 'traps' matrix. An
6897 important aspect of the wolverine study is that while the camera traps were operated over
6898 a 165 day period, each trap was operational during only a portion of that period. We need
6899 to provide the trap operation information which is contained in the columns to the right
6900 of the trap coordinates in our standard trap deployment file (TDF). Unfortunately, this is
6901 less easy to do in **secr**³, which requires an external file with a single long string of 1's and
6902 0's indicating the days in which each trap was operational (1) or not (0). The **read.traps**
6903 function will not allow for this information on trap operation if the data exists as an **R**
6904 object – instead, we can create this external file and then read it back in with **read.traps**
6905 using these commands:

```

6906 > hold <- rep(NA,nrow(traps))
6907 > for(i in 1:nrow(traps)){
6908 >   hold[i] <- paste(traps[i,4:ncol(traps)],collapse="")
6909 > }
6910 > traps1 <- cbind(traps[,1:3],"usage"=hold)
6911
6912 > write.table(traps1, "traps.txt", row.names=FALSE, col.names=FALSE)
6913 > trapfile2 <- read.traps("traps.txt",detector="proximity")

```

6914 These operations can be accomplished using the function **scr2secr** which is provided in
6915 the **R** package **scrbook**.

6916 After reading in the trap data, we now need to create the encounter matrix or array
6917 using the **make.capthist** command, where we provide the capture histories in EDF format,
6918 which is the existing format of the data input file **wcaps**. In creating the capture history,
6919 we provide also the trapfile created previously, the format (e.g., here EDF format is
6920 **fmt=** ‘‘**trapID**’’), and finally, we provide the number of occasions.

```

6921 #
6922 # Grab the encounter data file and format it:
6923 #
6924 wolv.dat <- wolverine$wcaps
6925 dimnames(wolv.dat) <- list(NULL,c("Session","ID","Occasion","trapID"))
6926 wolv.dat <- as.data.frame(wolv.dat)
6927 wolvcapt2 <- make.capthist(wolv.dat,trapfile2,fmt="trapID",noccasions=165)

```

6928 We also set up a habitat mask using the 2×2 km grid which we used previously in the
6929 analysis of the wolverine data and then pass the relevant objects to **secr.fit** as follows:

³as of v. 2.3.1

```

6930 #
6931 # Grab the habitat mask (2 x 2 km) and format it:
6932 #
6933 gr2 <- (as.matrix(wolverine$grid2))
6934 dimnames(gr2) <- list(NULL,c("x","y"))
6935 gr2 <- read.mask(data=gr2)
6936 #
6937 # To fit the model we use secr.fit:
6938 #
6939 wolv.secr2 <- secr.fit(wolvcapt2,model=list(D ~ 1, g0 ~ 1, sigma ~ 1),
6940                         buffer=20000,mask=gr2)

```

6941 We are using the “proximity detector” model (SCR0), so we do not need to make any
 6942 specifications in the command line because we have specified the detector type using the
 6943 constructor function `read.traps`, except to provide the buffer size (in meters). To specify
 6944 different models, you can change the default model `D~1`, `g0~1`, `sigma~1`. We provide all
 6945 of these commands and additional analyses in the `scrbook` package with the function called
 6946 `secr_wolverine`. Printing the output object produces the following (slightly edited):

```

6947 > wolv.secr2
6948
6949 secr 2.3.1, 15:52:45 29 Aug 2012
6950
6951 Detector type      proximity
6952 Detector number     37
6953 Average spacing    4415.693 m
6954 x-range             593498 652294 m
6955 y-range             6296796 6361803 m
6956 N animals          : 21
6957 N detections        : 115
6958 N occasions         : 165
6959 Mask area           : 987828.1 ha
6960
6961 Model               : D ~ 1 g0 ~ 1 sigma ~ 1
6962 Fixed (real)         : none
6963 Detection fn         : halfnormal
6964 Distribution          : poisson
6965 N parameters         : 3
6966 Log likelihood       : -602.9207
6967 AIC                  : 1211.841
6968 AICc                 : 1213.253
6969
6970 Beta parameters (coefficients)
6971          beta   SE.beta      lcl      ucl
6972 D      -9.390124 0.22636698 -9.833795 -8.946452
6973 g0     -2.995611 0.16891982 -3.326688 -2.664535
6974 sigma   8.745547 0.07664648  8.595323  8.895772

```

```

6975
6976 Variance-covariance matrix of beta parameters
6977          D      g0      sigma
6978 D      0.0512420110 -0.0004113326 -0.003945371
6979 g0     -0.0004113326  0.0285339045 -0.006269477
6980 sigma  -0.0039453711 -0.0062694767  0.005874683
6981
6982 Fitted (real) parameters evaluated at base levels of covariates
6983      link   estimate   SE.estimate      lcl      ucl
6984 D      log 8.354513e-05 1.915674e-05 5.360894e-05 1.301982e-04
6985 g0    logit 4.762453e-02 7.661601e-03 3.466689e-02 6.509881e-02
6986 sigma  log 6.282651e+03 4.822512e+02 5.406315e+03 7.301037e+03

```

6987 The object returned by `secr.fit` provides extensive default output when printed.
6988 Much of this is basic descriptive information about the model, the traps, or the encounter
6989 data. We focus here on the parameter estimates. Under the fitted (real) parameters, we
6990 find D , the density, given in units of individuals/hectare (1 hectare = 10000 m^2). To
6991 convert this into individuals/1000 km², we multiply by 100000, thus our density estimate
6992 is 8.35 individuals/1000 km². The parameter σ is given in units of meters, and so this
6993 corresponds to 6.283 km. Both of these estimates are very similar to those obtained in
6994 our likelihood analysis summarized in Table 6.3 which, for the 2 × 2 km grid, we obtained
6995 $\hat{D} = 8.31$ with a SE of $100000 \times 1.915674e - 05 = 1.9156$ and, accounting for the scale
6996 difference (1 unit = 10000 m in the previous analysis), $\hat{\sigma} = \sqrt{1/(2\hat{\alpha}_1)} * 10000 = 6.289$
6997 km. The difference in the MLE between Table 6.3 and those produced by `secr` could be
6998 due to subtle differences in internal tuning of optimization algorithms, starting values or
6999 other numerical settings. In addition, the likelihood is based on a Poisson prior for N (see
7000 the next section). On the other hand, the SE is slightly larger based on `secr` which is due
7001 to a subtle difference in the interpretation of D under the `secr` model (See below).

7002 6.5.3 Likelihood analysis in the `secr` package

7003 The `secr` package does likelihood analysis of SCR models for most classes of models
7004 as developed by Borchers and Efford (2008). Their formulation deviates slightly from
7005 the binomial form we presented in Sec. 6.2 above (though Borchers and Efford (2008)
7006 also mention the binomial form). Specifically, the likelihood that `secr` implements is that
7007 based on removing N from the likelihood by integrating the binomial likelihood (Eq. 6.2.1
7008 above) over a Poisson prior for N – what we will call the *Poisson-integrated likelihood* as
7009 opposed to the conditional-on- N (*binomial-form*) considered previously.

7010 To develop the Poisson-integrated likelihood we compute the marginal probability of
7011 each \mathbf{y}_i and the probability of an all-0 encounter history, π_0 , as before, to arrive at the
7012 marginal likelihood in the binomial-form:

$$\mathcal{L}(\boldsymbol{\alpha}, n_0 | \mathbf{y}) = \frac{N!}{n! n_0!} \left\{ \prod_i [\mathbf{y}_i | \boldsymbol{\alpha}] \right\} \pi_0^{n_0}$$

7013 Now, what Borchers and Efford (2008) do is assume that $N \sim \text{Poisson}(\Lambda)$ and they do a

7014 further level of marginalization over this prior distribution:

$$\sum_{n_0=0}^{\infty} \frac{N!}{n_0! n_0!} \left\{ \prod_i [\mathbf{y}_i | \boldsymbol{\alpha}] \right\} \pi_0^{n_0} \frac{\exp(-\Lambda) \Lambda^N}{N!}$$

7015 In Chapt. 11 we write $\Lambda = \mu ||\mathcal{S}||$ where $||\mathcal{S}||$ is the area of the state-space, and μ is the
 7016 density (“intensity”) of the point process. Carrying out the summation above produces
 7017 exactly this marginal likelihood:

$$\mathcal{L}_2(\boldsymbol{\alpha}, \Lambda | \mathbf{y}) = \left\{ \prod_i [\mathbf{y}_i | \boldsymbol{\alpha}] \right\} \Lambda^n \exp(-\Lambda(1 - \pi_0))$$

7018 which is Eq. 2 of Borchers and Efford (2008) except for notational differences. It also
 7019 resembles the binomial-form of the likelihood in Eq. 6.2.1 with $\Lambda^n \exp(-\Lambda\pi_0)$ replacing
 7020 the combinatorial term and the $\pi_0^{n_0}$ term. We emphasize there are two marginalizations
 7021 going on here: (1) the integration to remove the latent variables \mathbf{s} ; and, (2) summation
 7022 to remove the parameter N . We provide a function for computing this in the **scrbook**
 7023 package called **intlik3Poisson**. The help file for that function shows how to conduct a
 7024 small simulation study to compare the MLE under the Poisson-integrated likelihood with
 7025 that from the binomial form.

7026 The essential distinction between our MLE and Borchers and Efford as implemented in
 7027 **secr** is whether you keep N in the model or remove it by integration over a Poisson prior.
 7028 If you have prescribed a state-space explicitly with a sufficiently large buffer, then we
 7029 imagine there should be hardly any difference at all between the MLEs obtained by either
 7030 the Poisson-integrated likelihood or the binomial-form of the likelihood which retains N .
 7031 There is a subtle distinction in the sense that under the binomial form, we estimate the
 7032 realized population size N for the state-space whereas, for the Poisson-integrated form we
 7033 estimate the *prior* expected value which would apply to a hypothetical new study of a
 7034 similar population (see Sec. 5.7.3).

7035 Both models (likelihoods) assume \mathbf{s} is uniformly distributed over space, but for the
 7036 binomial model we make no additional assumption about N whereas we assume N is
 7037 Poisson using the formulation in **secr** from (Borchers and Efford, 2008). Using data
 7038 augmentation we could do a similar kind of integration but integrate N over a binomial
 7039 (M, ψ) prior – which we referred to as the binomial-integrated likelihood in Sec. 4.2.4.
 7040 So obviously the two approaches (data augmentation and Poisson-integrated likelihood)
 7041 are approximately the same as M gets large. However, doing a Bayesian analysis by
 7042 MCMC, we obtain an estimate of both N , the *realized population size*, and the parameter
 7043 controlling its expected value ψ which are, in fact, both identifiable from the data even
 7044 using likelihood analysis (Royle et al., 2007). That said we can integrate N out completely
 7045 and just estimate ψ as we noted in Sec. 6.2.1 above.

7046 6.5.4 Multi-session models in **secr**

7047 In practice we will often deal with SCR data that have some meaningful stratification or
 7048 group structure. For example, we might conduct mist-netting of birds on K consecutive
 7049 days, repeated, say, T times during a year, or perhaps over T years. Or we might collect
 7050 data from R distinct trapping grids. In these cases, we have T or R groups which we might

7051 reasonably regard as being samples of independent populations. While the groups might
 7052 be distinct sites, year, or periods within years, they could also be other biological groups
 7053 such as sex or age. Conveniently, **secr** fits a specific model for stratified populations –
 7054 referred to as *multi-session* models. These models build on the Poisson assumption which
 7055 underlies the integrated likelihood used in **secr** (as described in the previous section). To
 7056 understand the technical framework, let N_g be the population size of group g and *assume*

$$N_g \sim \text{Poisson}(\Lambda_g).$$

7057 Naturally, we model group-specific covariates on Λ_g :

$$\log(\Lambda_g) = \beta_0 + \beta_1 z_g$$

7058 where z_g is some group-specific covariate such as a categorical index to the group, or a
 7059 trend variable, or a spatial covariate, such as treatment effect or habitat structure, if the
 7060 groups represent spatial units. Under this model, we can marginalize *all* N_g parameters
 7061 out of the likelihood to concentrate the likelihood on the parameters β_0 and β_1 precisely
 7062 as discussed in the previous section. This Poisson hierarchical model is the basis of the
 7063 multi-session models in **secr**.

7064 To implement a multi-session model (or stratified population model) in **secr**, we pro-
 7065 vide the relevant stratification information in the ‘Session’ variable of the input encounter
 7066 data file (EDF). If ‘Session’ has multiple values then a “multi-session” object is created
 7067 by default and session-specific variables can be described in the model. For example, if
 7068 the session has 2 values for males and females then we have sex-specific densities , and
 7069 baseline encounter probability p_0 (g_0 in **secr**) by just doing this (see Chapt. 8 for the **R**
 7070 code to set this up):

```
7071 > out <- secr.fit(capdata, model=list(D ~ session, g0 ~ session, sigma^ 1),  

  7072           buffer=20000)
```

7073 More detailed analysis is given in Sec. 8.1 where we fit a number of different models and
 7074 apply methods of model selection to obtain model-averaged estimates of density.

7075 We can also easily implement stratified population models in the various **BUGS** en-
 7076 gines using data augmentation (Converse and Royle, 2012; Royle and Converse, in review)
 7077 which we discuss, with examples, in Chapt. 14.

7078 6.5.5 Some additional capabilities of **secr**

7079 The **secr** package has capabilities to do a complete analysis of SCR data sets, including
 7080 model fitting, selection, and many summary analyses. In the previous sections, we’ve
 7081 given a basic overview, and we do more in later chapters of this book. Here we mention a
 7082 few of these other capabilities that you should know about as you use **secr**. Of course, you
 7083 should skim through the associated documentation (**?secr**) to see more of what’s available.

7084 Alternative observation models

7085 **secr** fits a wide range of alternative observation models besides the Bernoulli encounter
 7086 model, including multinomial encounter models for “multi-catch” and “single catch” traps,
 7087 models for sound attenuation from acoustic detection devices, and many others. We
 7088 discuss many of these other methods in Chapt. 9 and elsewhere in the book.

7089 Summary statistics

7090 **secr** provides a useful default summary of the data, but it also has summary statistics
 7091 about animal movement including mean-maximum distance moved (the function **MMDM**).
 7092 For example, see the help page **?MMDM** which lists a number of other summary functions
 7093 which take a **capthist** object:

```
7094 > moves(capthist)
7095 > dbar(capthist)
7096 > RPSV(capthist)
7097 > MMDM(capthist, min.recapt = 1, full = FALSE)
7098 > ARL(capthist, min.recapt = 1, plt = FALSE, full = FALSE)
```

7099 The function **moves** returns the observed distances moved, **dbar** returns the average dis-
 7100 tance moved, **RPSV** produces a measure of dispersion about the home-range center, and
 7101 **ARL** gives the *Asymptotic Range Length* which is the asymptote of an exponential model
 7102 fit to the observed range length vs. the number of detections of each individual (Jett and
 7103 Nichols, 1987).

7104 State-space buffer

7105 **secr** will produce a warning if the state-space buffer is chosen too small. For example,
 7106 in fitting the wolverine data as in Sec. 6.5.2 but with a 1000 m buffer, and we see the
 7107 following warning message:

```
7108 Warning message:
7109 In secr.fit(wolvcapt2, model=list(D ~ 1, g0 ~ 1, sigma ~ 1), buffer=1000):
7110   predicted relative bias exceeds 0.01 with buffer = 1000
```

7111 This should cause you to contemplate modifying the state-space buffer if that is a reason-
 7112 able thing to do in the specific application.

7113 Model selection and averaging

7114 **secr** does likelihood ratio tests to compare nested models using the function **LR.test**.
 7115 You can create model selection tables based on AIC or AICc, using the function **AIC**,
 7116 and obtain model-averaged parameter estimates using the function **model.average** (See
 7117 Chapt. 8 for examples).

7118 Population closure test

7119 **secr** has a population closure test with the function **closure.test** which implements the
 7120 tests of Stanley and Burnham (1999) or Otis et al. (1978). The function is used like this:
 7121 **closure.test(object, SB = FALSE)**. Here **object** is a **capthist** object and **SB** is a logical
 7122 variable that, if TRUE, produces the Stanley and Burnham (1999) test.

7123 Density mapping and effective sample area

7124 **secr** produces likelihood versions of the various summaries of posterior density and effec-
 7125 tive sample area that we discussed in Chapt. 5. For example, while **secr** reports estimates
 7126 of the expected value of N or density directly in the summary output from fitting a model,
 7127 you can use the function **region.N** to produce estimates of N for any given region. In
 7128 addition, **secr** has functions for creating maps of detection contours for individuals traps,
 7129 or for the entire trap array. See the function **pdot.contour**, and also **fxi.contour** for

7130 computing the 2-dimensional pdf of the locations of one or more individual activity cen-
 7131 ters (as in Sec. 5.11.3). In the context of likelihood analysis, estimation of a random effect
 7132 **s** is based on a plug-in application of Bayes' Rule. When **s** has a uniform distribution, and
 7133 we use a discrete evaluation of the integral, it can be computed simply by renormalizing
 7134 the likelihood:

$$[s|y, \theta] = \frac{[y|s, \theta]}{\sum_s [y|s, \theta]}.$$

7135 Any of the **intlik** functions given previously in this chapter can be easily modified to
 7136 return the posterior distribution of **s** for any, or all, individuals, or an individual that is
 7137 not encountered.

7138 Effective sample area (see Sec. 5.12) can be calculated in **secr** using the functions **esa**
 7139 and **esa.plot**).

7140 Covariate models

7141 **secr** has many capabilities for modeling covariates. It has a number of built-in models
 7142 that allow certain covariates on encounter probability, which we cover to a large extent
 7143 in Chapt. 7, and also see Chapt. 8 for more examples. **secr** also allows covariates to be
 7144 built into the density model (see Chapt. 11). It has some built in response surface models,
 7145 allowing for the fitting of linear or quadratic response surfaces. This is done by modifying
 7146 the density model in **secr.fit**. For example, $D \sim 1$ is a constant density surface, and
 7147 $D \sim x + y$ fits a linear response surface, etc.. See the manual **secr-densitysurfaces.pdf**
 7148 for the details.

7149 There are a number of ways to model your own "custom" covariates (as opposed to
 7150 pre-specified models). One way is to use the **addCovariates** function and supply it a
 7151 **mask** or **traps** object along with some "spatialdata." Or, if you have covariates at each
 7152 trap location then it will extrapolate to all points on the habitat mask. There's also a
 7153 method by which the user can create a function of geographic coordinates, **userDfn**, which
 7154 seems to provide additional flexibility, although we haven't used this method. There is a
 7155 handy function **predictDsurface** for producing density maps under the specified model
 7156 for density.

6.6 SUMMARY AND OUTLOOK

7157 In this chapter, we discussed basic concepts related to classical analysis of SCR models
 7158 based on likelihood methods. Analysis is based on the so-called integrated or marginal
 7159 likelihood in which the individual activity centers (random effects) are removed from the
 7160 conditional-on-**s** likelihood by integration. We showed how to construct the integrated
 7161 likelihood and fit some simple models in the **R** programming language. In addition,
 7162 likelihood analysis for some broad classes of SCR models can be accomplished using the
 7163 **R** library **secr** (Efford, 2011a) which we provided a brief introduction to. In later chapters
 7164 we provide more detailed analyses of SCR data using likelihood methods and the **secr**
 7165 package.

7166 Why or why not use likelihood inference exclusively? For certain specific models, it
 7167 is may be more computationally efficient to produce MLEs (for an example see Chapt.
 7168 12). And, likelihood analysis makes it easy to do model-selection by AIC and compute
 7169 standard errors or confidence intervals. However, **BUGS** is extremely flexible in terms
 7170 of describing models and we can devise models in the **BUGS** language easily that we

7171 cannot fit in **secr**. For example, in Chapt 16 we consider open population models which
7172 are straightforward to develop in **BUGS** but, so far, there is no available platform for
7173 doing MLE of such models. We can also fit models in **BUGS** that accommodate missing
7174 covariates in complete generality (e.g., unobserved sex of individuals), and we can adopt
7175 SCR models to include auxiliary data types. For example, we might have camera trapping
7176 and genetic data and we can describe the models directly in **BUGS** and fit a joint model
7177 (Gopalaswamy et al., 2012b). To do maximum likelihood estimation, we have to write a
7178 custom new piece of code for each model⁴ or hope someone has done it for us. You should
7179 have some capability to develop your own MLE routines with the tools we provided in
7180 this chapter.

⁴Although we may be able to handle multiple survey methods together in **secr** using the multi-session models.

7

MODELING VARIATION IN ENCOUNTER PROBABILITY

In previous chapters we showed how to fit basic spatial capture-recapture models using Bayesian analysis (in **WinBUGS** or **JAGS**; Chapt. 5) or by classical likelihood methods (Chapt. 6 or using **secr**). We mostly focused on a specific observation model, the Bernoulli or binomial model for devices such as “proximity detectors” (although we extend this model to Poisson and multinomial type observation models in Chapt. 9). We have not, however, described a general framework for modeling covariates that might influence encounter probability of individuals, traps or over time. In practice, investigators are invariably concerned with explicit factors or covariates that might influence variation in parameters. Such covariates include time (e.g., day of year, or season), behavior (e.g., is there an effect of trapping on subsequent capture probabilities), sex of the individual, and trap type (e.g., various camera types, or different constructions for hair snares). Traditionally, in the non-spatial capture recapture literature, such models were called “model M_t ”, “model M_h ”, or “model M_b ”, identifying models that account for variation in detection probability as a function of time, “individual heterogeneity” or “behavior”, where behavior describes whether or not an individual had been previously captured. In SCR models, more complex covariate models are possible because we might also have trap-specific covariates, or covariates that vary spatially over the landscape, and because we generally have more than one parameter describing the detection function: Most encounter probability functions include a baseline encounter rate (λ_0) or probability (p_0) parameter, and a scale parameter (σ), which takes on different interpretations depending on the specific encounter probability function under consideration.

In this chapter, we generalize the basic SCR model to accommodate both alternative detection functions as well as many different kinds of covariates. We focus on the binomial observation model used throughout Chaps. 5 and 6 and the Gaussian encounter model (also called the “half-normal” model in the distance sampling literature), 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

7212 three distinct types of covariates – those which are fixed, partially observed or completely
 7213 unobserved (latent). Fixed covariates are those that are fully observed; for example, the
 7214 date of all sampling occasions. Partially observed covariates are those which are not known
 7215 for all observations; for example, the sex of an individual cannot always be determined
 7216 from photos taken during camera trapping. Even if we are able to observe the sex of all
 7217 individuals sampled, we cannot know it for those individuals never observed during the
 7218 study. And finally, unobserved covariates are those which we cannot observe at all, for
 7219 example, the home range size of individuals, or unstructured random “individual effects”.

7220 We will see that models containing these different types of covariates are relatively easy
 7221 to describe in **WinBUGS** or **JAGS**, and therefore to analyze using Bayesian analysis
 7222 of the joint likelihood based on data augmentation thus providing a coherent and flexible
 7223 framework for inference for all classes of SCR models. Throughout the chapter, we will
 7224 continue to develop the analysis of the black bear study introduced in Chapt. 4, using the
 7225 software **JAGS**. We also consider the likelihood analysis of many of these models; to do so,
 7226 we continue to use the **R** package **secr**, and we introduce some ideas of model comparison
 7227 using AIC (Sec. 7.4 at the end of the chapter). There are other types of covariates that
 7228 we do *not* cover in this chapter; for example, covariates that vary across the landscape
 7229 might affect density, and we consider these covariates in Chapt. 11. Alternatively, these
 7230 landscape covariates might affect the way individuals use space. There are probably very
 7231 few circumstances under which animals use all space uniformly and we develop more
 7232 realistic models of encounter probability in which covariates affect space usage in Chapt.
 7233 12.

7.1 ENCOUNTER PROBABILITY MODELS

7234 In Chapt. 5, we developed a basic spatial capture recapture model using a standard
 7235 encounter probability function based on the kernel of a normal (Gaussian) probability
 7236 distribution:

$$p_{ij} = p_0 \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

7237 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
 7238 Sec. 5.4) that one can view this model as corresponding to an explicit model of space
 7239 usage – namely, that individual locations are draws from a bivariate normal distribution.
 7240 We also mentioned that other detection models are possible, including a logit model of
 7241 the form:

$$\text{logit}(p_{ij}) = \alpha_0 + \alpha_1 ||\mathbf{x}_j - \mathbf{s}_i||. \quad (7.1.1)$$

7242 However, there's nothing preventing us from constructing a myriad of other models for
 7243 encounter probability as a function of distance. The most commonly used detection prob-
 7244 ability models are also those used in the distance sampling literature: the half-normal
 7245 (Gaussian), the hazard, and the negative exponential. The negative exponential model is:

$$p_{ij} = p_0 * \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||)$$

7246 where we define $\alpha_1 = 1/\sigma$. We could use the general power model (Russell et al., 2012):

$$p_{ij} = p_0 * \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^\theta)$$

7247 of which the Gaussian and exponential models are special cases. Another model that could
 7248 be considered is the Gaussian hazard rate model (Hayes and Buckland, 1983):

$$p_{ij} = 1 - \exp(-\lambda_0 * \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2))$$

7249 which was previously discussed in Sec. .

7250 In each of the cases, the relationship of α_1 to σ varies and must be properly spec-
 7251 ified. The **R** package **secr** allows the user to access 12 different encounter probability
 7252 models (termed “distance functions” in **secr**), of which some are only used for simulating
 7253 data (see Table 7.1). These encounter probability models can also be implemented in **R**,
 7254 **WinBUGS**, **JAGS** etc..

Table 7.1. Basic encounter probability models (“distance functions”) available in **secr**. (Table taken from the **secr** help files). Notation deviates from that used in the text. In this table g_0 is the baseline encounter rate or probability parameter used in **secr** which is equivalent to our p_0 or λ_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, σ	$g(d) = g_0 e^{-d^2/(2\sigma^2)}$
1	hazard rate	g_0, σ, z	$g(d) = g_0(1 - e^{-(d/\sigma)^{-z}})$
2	exponential	g_0, σ	$g(d) = g_0 e^{-d/\sigma}$
3	compound half-normal	g_0, σ, z	$g(d) = g_0[1 - \{1 - e^{-d^2/(2\sigma^2)}\}^z]$
4	uniform	g_0, σ	$g(d) = g_0, d \leq \sigma;$ $g(d) = 0, \text{ otherwise}$
5	w exponential	g_0, σ, w	$g(d) = g_0, d < w;$ $g(d) = g_0 e^{(-(d-w)/\sigma)}, \text{ otherwise}$
6	annular normal	g_0, σ, w	$g(d) = g_0 e^{(-(d-w)^2/(2\sigma^2))}$
7	cumulative lognormal	g_0, σ, z	$g(d) = g_0[1 - F(d - \mu)/s)]$
8	cumulative gamma	g_0, σ, 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	β_0, β_1, S	$g(d) = 1 - F[\{c - (\beta_0 + \beta_1 d)\}/S]$
11	signal strength spherical	β_0, β_1, S	$g(d) = 1 - F[\{c - (\beta_0 + \beta_1(d-1) - 10 * \log_{10}(d^2))\}/S]$

7255 Insofar as all these encounter probability models are symmetric and stationary, they
 7256 are pretty crude descriptions of space usage by real animals. This is not to say they are
 7257 inadequate descriptions of the data and, as we discuss in Chaps. 13 and 12, we can use
 7258 them as the basis for producing more realistic models of space usage.

7259 By changing the encounter probability model and the specification of α_1 , we can
 7260 basically create any function of distance for the data. It is important to note that σ is not
 7261 comparable under these different encounter probability models and should not be regarded
 7262 as “home range radius” in general. While there is generally a relationship between σ and
 7263 home range size, that relationship varies depending on the model under consideration. We
 7264 demonstrate how to fit different encounter probability models in the Bayesian framework
 7265 here, and then provide information on the likelihood analysis (in **secr**) in a separate
 7266 section below.

7267 **7.1.1 Bayesian analysis with bear.JAGS**

7268 To demonstrate how to incorporate various types of covariates into models for encounter
 7269 probability using **JAGS**, we return to the data collected during the Fort Drum bear study.
 7270 This data set was first introduced in Chapt. 4, but, to refresh your memory, there were
 7271 38 baited hair snares that were operated between June and July 2006. The snares were
 7272 checked each week for a total for $K = 8$ sample occasions and $n = 47$ individual bears
 7273 were encountered at least once. The data are provided in the **R** package **scrbook** and an
 7274 **R** function called **bear.JAGS** allows the user to easily pick which model to analyze. The
 7275 function **bear.JAGS** will set up the data, write the model, define the MCMC specifications
 7276 (e.g., initial values, etc.) and, finally, run the selected model in **JAGS**. In addition to
 7277 choosing which model to run, the user can also specify the number of chains, iterations and
 7278 length of the burn-in phase. Calling the function will provide all the code to implement
 7279 the models independently as well. In the following sections we will present the model code
 7280 and output for the most commonly employed models; for all analyses we ran 3 chains with
 7281 a burn-in of 500 iterations and 20000 saved iterations.

7282 **7.1.2 Bayesian analysis of encounter probability models**

7283 In Panel 7.1, we present the basic SCR model and show how to specify the negative exponential
 7284 encounter probability model. To call each of these from the function **bear.JAGS** set
 7285 **model='SCRO'** or **model='SCRexp'** in the function call, respectively. To reduce repetition
 7286 of the R coding, we include the basic code here and then only show modifications when
 7287 necessary throughout the chapter. All of the R coding can be found within the **bear.JAGS**
 7288 function as well. The function begins by loading the required **R** libraries as well as the
 7289 Ft. Drum bear data set. This data set includes a 3-d data array (called **bearArray** in our
 7290 code), with dimensions **nind** \times **ntraps** \times **nreps** representing the capture histories of **nind**
 7291 captured individuals at **ntraps** trap locations. In the Bayesian analysis, data augmentation
 7292 is used to estimate N and therefore the **bearArray** data must be augmented with
 7293 $M - nind$ all zero encounter histories. In models without time dependence, the augmented
 7294 **bearArray** (called **Yaug** in the code) will be reduced to a 2 dimensional array (denoted **y**
 7295 in the code) that has dimensions **M** \times **ntraps**.

```
7296 > library(rjags) # Load the necessary libraries
7297 > library(scrbook)
7298
7299 > data(beardata) # Attach the bear data for Ft. Drum
7300 > ymat <- beardata$bearArray
7301 > trapmat <- beardata$trapmat
7302 > nind <- dim(beardata$bearArray)[1]
7303 > K <- dim(beardata$bearArray)[3]
7304 > ntraps <- dim(beardata$bearArray)[2]
7305 > M <- 650
7306 > nz <- M-nind
7307
7308 # Create augmented array
7309 > Yaug <- array(0, dim=c(M,ntraps,K))
```

```

7310 > Yaug[1:nind,,] <- ymat
7311 > y <- apply(Yaug,1:2, sum)

```

7312 The function `bear.JAGS` also establishes the upper and lower limits on the state space
 7313 by centering the trap array coordinates (which are imported with the `beardata` and saved
 7314 in the code above as `trapmat`) and then buffering by 20km.

```

model{
  alpha0 ~ dnorm(0,.1)                               # Prior distributions
  logit(p0) <- alpha0
  alpha1 <- 1/(2*sigma*sigma)
  sigma ~ dunif(0, 15)
  psi ~ dunif(0,1)

  for(i in 1:M){
    z[i] ~ dbern(psi)
    s[i,1] ~ dunif(xlim[1],xlim[2])
    s[i,2] ~ dunif(ylim[1],ylim[2])
    for(j in 1:J){
      d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
      y[i,j] ~ dbin(p[i,j],K)
      p[i,j] <- z[i]*p0*exp(-alpha1*d[i,j]*d[i,j]) # Gaussian model
      #p[i,j] <- z[i]*p0*exp(-alpha1*d[i,j])        # exponential model
    }
  }
  N <- sum(z[])
  D <- N/area
}

```

Panel 7.1: **JAGS** model specification for a basic SCR model with Gaussian encounter probability function and the alternative exponential encounter probability function.

7315 Applying the SCR model with Gaussian encounter probability model provides an
 7316 estimate (posterior mean) of $D = 0.167$ bears per km^2 and with the negative exponential
 7317 encounter probability model the posterior mean is virtually the same $D = 0.167$. In
 7318 distance sampling, the use of different encounter probability models often results in very
 7319 different estimates of density (especially when using the negative exponential model).
 7320 There are two main reasons why the different models may have less of an impact on the
 7321 density estimates under the SCR models. First, we can estimate the baseline encounter
 7322 probability parameter (p_0). In most distance sampling models, detection at distance 0
 7323 is set to 1. In Table 7.2, the posterior mean of p_0 is 0.11 under the Gaussian model
 7324 and 0.34 under the negative exponential model. The larger baseline encounter probability

under the negative exponential model reduces the impact of the quick decline in detection as a function of distance. Secondly, the detection probability function here is governing 'movement' of individuals (which we have more information on than in distance sampling), not the whole detection process, so the shape of the detection probability function does not impact the density estimation as much.

In all analyses it is important to check that the size of the augmented data set (M) is sufficiently large and does not impact the estimate of N . Here, the 97.5% percentile for N is 628 (Table 7.2), thus not reaching our $M = 650$ value. We could also increase M and compare the posterior of N under the different scenarios as another check that the data augmentation is sufficient.

Table 7.2. Posterior summaries of SCR model parameters having different encounter probability models, for the Fort Drum black bear data.

Parameter	Mean	SD	2.5%	97.5%
Gaussian				
N	500.63	66.652	371.000	628.000
D	0.17	0.022	0.122	0.207
p_0	0.11	0.014	0.081	0.135
σ	1.99	0.131	1.762	2.275
ψ	0.77	0.104	0.566	0.966
Exponential				
N	512.06	65.771	382.000	634.000
D	0.17	0.022	0.130	0.210
p_0	0.34	0.056	0.246	0.465
σ	1.12	0.095	0.951	1.323
ψ	0.79	0.102	0.584	0.974

A very important consideration when using different detection probability functions is the interpretation of σ . The estimate (posterior mean) of σ under the negative exponential model is 1.12, which is distinct from our estimate of σ under the Gaussian model, $\sigma = 1.996$. The interpretation of σ in the two models is really quite distinct. In the normal model it can be interpreted as the standard deviation of a bivariate normal movement model whereas the manner in which σ relates to "area used" for the negative exponential model has nothing to do with a bivariate normal model of movement. This highlights that it is important for the user to know what detection probability function is used and what the interpretation of σ might be in relation to the home range size. This relationship was discussed in Sec. 5.4.

We now move onto incorporating covariates into the model using the **JAGS** language. For this part, we will stick with the Gaussian encounter probability model shown in Panel 7.1 above.

7.2 MODELING COVARIATE EFFECTS

The basic strategy for modeling covariate effects is to include them on the baseline encounter rate or probability parameter, p_0 (or λ_0), or the scale parameter of the encounter model, σ , or in some cases, both parameters.

Broadly speaking, we recognize (here) 3 types of covariates. Fixed covariates are fully observable and might vary by trap alone (e.g., type of trap, baited or not, disturbance regime, even habitat), sample occasion (e.g., day of season or weather conditions), or both (e.g., behavior, weather - if over a large region). Another class of covariates are those which vary at the level of the individual (and possibly also over time). As a technical matter, and as noted before, these are different from fixed covariates because we cannot see all of the individuals and the covariates are almost always incompletely observed (if at all). The lone exception is the effect of previous capture, used to model a behavioral response to capture, which is known for all individuals, captured or not (an animal never captured/observed has never been captured before). We noted many times before that space itself (i.e., the activity centers) is a type of individual covariate and this notion actually helped us derive the fully spatial capture-recapture model from the traditional, non-spatial model (Chapt. 4). We do not get to observe the activity center for any individuals, but for individuals that are encountered we get to observe some information about it in the form of which traps the individual was encountered in. And finally, we have completely unobserved covariates such as heterogeneity in home range size. We consider heterogeneity in a separate section below since there are a suite of models for describing latent heterogeneity.

Table 7.3. Examples of different types of covariates in SCR models.

Covariate type	Examples
individual	sex, age, home range
trap	baited/not, habitat (see also Chapter 13)
time	season, shedding, weather
individual x time	global behavioral response
trap x time	trap failures
individual x trap x time	local behavioral response

To develop covariate models, we assume a standard sampling design in which an array of J traps is operated for K sample occasions, which produces encounter histories for n individuals. For the null model, there are no time-varying covariates that influence encounter, there are no explicit individual-specific covariates, and there are no covariates that influence density. For fixed effects, those which we observe fully, we can easily incorporate these into the encounter probability model, just as we would do in any standard GLM or GLMM, on some suitable scale for the encounter probability, p_{ijk} . For example,

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * C_{ijk}$$

$$p_{ijk} = p_{0,ijk} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

where C_{ijk} is some covariate that varies (potentially) by individual (i), trap (j) and occasions (k), and α_2 is the coefficient to be estimated. How we define specific covariates (e.g., trap specific versus individual specific) will influence exactly how we include them in the model. Table 7.3 shows examples of covariates by type – trap, individual, and time – and also gives examples of some combined types. These are the types of covariates we will specifically address in this chapter, demonstrating how to analyze the different types in the following sections.

7383 **7.2.1 Date and time**

7384 Often, researchers are interested in modeling the effect of date or chronological time on
 7385 encounter probability. For example, in a long term hair snare study, we may expect that
 7386 seasonal shedding (Wegan et al., 2012) will influence encounter probabilities directly. Or,
 7387 we may expect behaviors such as denning, mating, etc., to influence the encounter of
 7388 certain species at certain times of year (Kéry et al., 2011). There are two common ways
 7389 to incorporate date or time information into a model for encounter probability. For cases
 7390 with a small number of sampling occasions we can fit a time-specific intercept (analogous
 7391 to “model M_t ” in classical capture-recapture (Otis et al., 1978)). In this model, there are
 7392 K sampling occasion-specific parameters to reflect potential variation in sampling effort
 7393 or other factors that might vary across samples. Alternatively, we can model parametric
 7394 functions of date or time such as polynomial or sinusoidal functions.

7395 In the first case, we allow each sampling occasion, k , to have its own baseline encounter
 7396 probability, e.g.,

$$\text{logit}(p_{0,k}) = \alpha_{0,k}$$

7397 so that

$$p_{ijk} = p_{0,k} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2).$$

7398 This description of the model includes k occasion-specific baseline encounter probabilities.
 7399 Thus, if there are 4 sampling occasions, then there are 4 different baseline encounter
 7400 probabilities. We imagine that complete time-specificity of p_0 (i.e., one distinct value
 7401 for each sample occasion) would be most useful in situations where there are just a few
 7402 sampling occasions (if there are many, this formulation will dramatically increase the
 7403 number of parameters to be estimated) or we do not expect systematic patterns over time
 7404 (e.g., explainable by a polynomial function or time-varying covariates).

7405 To implement this in **JAGS**, α_0 has to be estimated for each time period k either
 7406 using an index vector or dummy variables (as described in Chapt. 2 and Sec. 4.3) and this
 7407 can be done by only changing only a few lines in Panel 7.1:

```
7408 alpha0[k] ~ dnorm(0,.1)
7409 logit(p0[k]) <- alpha0[k]
7410 .....
7411 .....
7412 y[i,j,k] ~ dbin(p[i,j,k],K)
7413 p[i,j,k] <- z[i]*p0[k]*exp(- alpha1*d[i,j]*d[i,j])
```

7414 Since the model contains a parameter for each time period, the encounter histories
 7415 must be time-dependent. Thus, a 3-d data array (called **bearArray** in our code), with
 7416 dimensions **nind** \times **ntraps** \times **nreps** is required (recall that we use the 3-d augmented array
 7417 called **Yaug** with dimensions **M** \times **ntraps** \times **nreps** for the Bayesian analysis). In addition
 7418 to using the 3-d data array, the initial values must be updated so that there are K values
 7419 generated for α_0 . And finally, this means that another nested *for loop* is needed in the
 7420 code to account for the K sample occasions. A side note: the computation time will
 7421 increase quite a bit (this model for the bear data may take up to 15 hours or more on
 7422 your machine to obtain a sufficient posterior sample).

7423 Running this model with the function **bear.JAGS** by setting **model=SCRt**, returns esti-
 7424 mates of density similar to those from the model without covariates (see Table 7.4), but

now we have a characterization of variation in encounter probability over time. Encounter probability seems to increase for the first few time periods before stabilizing around 0.14, dropping off again at the end of the study. The differences in encounter probability from the first time periods to the others might actually be due to something like a behavioral response (see below) or possibly seasonal differences in the efficiency of the sampling technique. Researchers have found that hair snares are more effective at different times of the year (even within season) due to shedding (Wegan et al., 2012). In this particular example, our density estimates (posterior means) are similar to the base model, likely because the differences in encounter probability between occasion were not that large. In a longer term study or in one with greater variation in the encounter probability, the implication of such differences might have a bigger impact on the estimates of density and σ .

Table 7.4. Posterior summaries of parameter estimates from a SCR model with time-dependent baseline encounter probability for the Ft. Drum black bear data set.

Parameter	Mean	SD	2.5%	97.5%
N	509.24	66.13	381	632
D	0.17	0.02	0.13	0.21
$p_0(t = 1)$	0.06	0.02	0.03	0.10
$p_0(t = 2)$	0.05	0.02	0.02	0.09
$p_0(t = 3)$	0.15	0.03	0.09	0.22
$p_0(t = 4)$	0.14	0.03	0.09	0.21
$p_0(t = 5)$	0.15	0.03	0.09	0.22
$p_0(t = 6)$	0.12	0.03	0.07	0.19
$p_0(t = 7)$	0.15	0.03	0.09	0.22
$p_0(t = 8)$	0.08	0.02	0.04	0.13
σ	1.96	0.12	1.73	2.22
ψ	0.78	0.10	0.58	0.97

The occasion specific intercepts (baseline encounter probability) model might not be the most appropriate for all scenarios and could require the estimation of many parameters if we had many sampling occasions, take the wolverine example from Chapt. 5.9 where there were 165 daily sampling occasions. Particularly in such a case as the wolverine study, variation in the encounter process over time is to be expected. For example, if a camera trap study is conducted for an entire year, it is expected that there would be behavioral patterns in individuals due to mating or denning. Instead of fitting a model with K baseline encounter probabilities, we can include date as a linear (or quadratic, ...) effect. An example can be found in Kéry et al. (2011) who incorporated a day-of-year covariate, both as a linear and a quadratic effect, into their SCR model of European wildcats; the data had been collected over a year-long period and cat behavior was expected to vary seasonally thus influencing the probability of encounter. In these cases, we would specifically incorporate day of year (variable “Date”) as a numeric covariate as:

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * \text{Date}_k$$

$$p_{ijk} = p_{0,ijk} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

7449 or a quadratic effect of day-of-year:

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * \text{Date}_k + \alpha_3 * \text{Date}_k^2$$

$$p_{ijk} = p_{0,ijk} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

7450 where the variable **Date** is an integer coding of day-of-year, indexed to some arbitrary
7451 start point in time.

7452 7.2.2 Trap-specific covariates

7453 In some studies it makes sense to model encounter probability as a function of local or trap-
7454 specific covariates. These can be one of two types: genuine trap covariates that describe
7455 the trap or encounter site, such as whether a trap is baited or not, or how many traps were
7456 set at a sampling location, or what kind of bait was used, etc., or local covariates that
7457 describe the likelihood that an animal would use the habitat in the vicinity of the trap
7458 (see Chapt. 13 for more on this situation). We imagine that these covariates, of either
7459 type, should affect baseline encounter probability. For example, Sollmann et al. (2011)
7460 found a large difference in the encounter probability of jaguars due to traps being located
7461 on roads, which the animals were using to travel along, as opposed to traps placed off
7462 of roads. In this case, the trap type is a binary variable – on/off road, (another binary
7463 variable could be baited/non-baited). We can write this as:

$$\text{logit}(p_{0,j}) = \alpha_{0,type_j}$$

$$p_{ijk} = p_{0,j} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2).$$

7464 Here, we use an index variable, “type”, an integer value for the trap-specific covariate.
7465 Thus for our example of on/off road, we would have $type_j = 1$ if trap j is on a road and
7466 $type_j = 2$ otherwise, and we would estimate two separate α_0 parameters – one for on-road
7467 and one for off-road cameras. An alternative way to express the 2-category model, using
7468 dummy variables, requires that we specify our “type” vector as $Type_j = 0$ if trap j is on
7469 a road and $Type_j = 1$ otherwise, and write the model as

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * Type_j.$$

7470 Now, α_0 is the baseline encounter probability (on the logit scale) for traps on a road
7471 ($Type_j = 0$) and α_2 is the effect on baseline encounter probability of a trap being of
7472 $Type = 1$. This general set up also allows for more than 2 categories, say if 4 different
7473 camera models were used in a study, we would use a set of 3 binary dummy variables
7474 to allow for estimation of the different encounter rates (i.e., the intercept). While these
7475 models are equivalent, and should yield identical results, sometimes one parameterization
7476 might work better than the other in **WinBUGS** or **JAGS** (Kéry, 2010).

7477 7.2.3 Behavior or trap response by individual

7478 One of the most basic of encounter models is that which accommodates a change in
7479 encounter probability as a result of initial encounter. This is colloquially referred to as
7480 “trap happiness” or “trap shyness”, or in other words, a behavioral response of individuals

7481 to being captured (Otis et al., 1978). If a trap is baited with a food source, an individual
 7482 might come back for more. On the other hand, if being captured is traumatic then an
 7483 individual might learn to avoid traps. Both of these types of responses can occur in
 7484 most species depending on the type of encounter mechanisms being employed. Moreover,
 7485 behavioral response can be either global (Gardner et al., 2010b) or local (Royle et al.,
 7486 2011b). The local response is a trap-specific response while a global response suggests that
 7487 initial capture provides a net increase or decrease in subsequent probabilities of capture
 7488 (across all traps). A behavioral response does not need to be enduring (i.e., persist for
 7489 the entire study after the individual has been captured/observed for the first time) but
 7490 can also be ephemeral, if, for example, an animal only avoids a trap on the occasion
 7491 immediately after it was captured (Yang and Chao, 2005; Royle, 2008). While we will
 7492 focus the examples in this chapter on enduring behavioral effects, extending such a model
 7493 to the case of an ephemeral response should not pose any difficulties.

7494 To describe these behavioral models we need to create a binary matrix that indicates
 7495 if an individual has been captured previously. For the global behavioral response, define
 7496 the $n \times K$ matrix, \mathbf{C} , where $C_{ik} = 1$ if individual i was captured at least once prior to
 7497 session k , otherwise $C_{ik} = 0$.

$$\text{logit}(p_{0,ik}) = \alpha_0 + \alpha_2 * C_{ik}$$

$$p_{ijk} = p_{0,ik} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2).$$

7498 For the local behavioral response, which is trap specific, we create an array, C_{ijk} , that
 7499 indicates if an individual i has been previously captured in trap j at time k . (For the
 7500 augmented individuals, the entries are all 0 since the animals were never captured.) We
 7501 then include this in the model in the exact same form as above (with the sole difference
 7502 that both C and p are now also indexed by k):

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * C_{i,j,k}$$

$$p_{ijk} = p_{0,ijk} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2).$$

7503 Since the behavioral response is occasion specific, to implement either the local or
 7504 global response model in **JAGS**, we will have to use the 3-d array of the augmented
 7505 capture histories ($M \times ntraps \times nreps$) as we did for the time-varying encounter probability
 7506 model above. The code must loop over each sampling occasion, but otherwise, the model
 7507 varies only a little from the basic SCR model shown in Panel 7.1. Here is the specification
 7508 of the the occasion specific (k) loop:

```
7509 for(k in 1:K){  

  7510   logit(p0[i,j,k]) <- alpha0 + alpha2*C[i,j,k]  

  7511   y[i,j,k] ~ dbin(p[i,j,k],1)  

  7512   p[i,j,k] <- z[i]*p0[i,j,k]*exp(- alpha1*d[i,j]*d[i,j]).  

  7513 }
```

7514 Despite only minor changes to the **BUGS** code, this model can require quite a bit
 7515 of time and computational effort. Implementing the behavioral models with the function
 7516 **bear.JAGS** by setting **model=SCRb** or **model=SCRb** for the local or global model respec-
 7517 tively, returns the results shown in Table 7.5. There is a strong global behavioral response
 7518 suggested by the posterior mean of $\alpha_2 = 0.90$. The estimate of N and subsequently D are

7519 larger than under the model without a behavioral response; here we estimate the posterior
 7520 mean of $N = 577.56$, whereas in the SCR0 model, we estimated the posterior mean as
 7521 $N = 500$. This makes sense given the large estimate of α_2 , which suggests that bears
 7522 are trap happy. In situations where animals are trap happy, the null model tends to over
 7523 estimate encounter probability (i.e., the bears that are never observed have a lower en-
 7524 counter probability than those that have been captured in the study) and thereby reduce
 7525 the estimate of N . We do not include the results here, but the estimates were similar
 7526 under the local behavioral response model.

Table 7.5. Posterior summaries of parameter estimates from the SCR model with a global behavioral response in encounter for the Fort Drum black bear data set.

Parameter	Mean	SD	2.5%	97.5%
N	577.56	54.30	452	648
D	0.19	0.02	0.15	0.21
α_0	-2.81	0.24	-2.91	-2.36
α_2	0.90	0.23	0.45	1.35
σ	2.00	0.13	1.77	2.28
ψ	0.88	0.08	0.69	0.99

7527 7.2.4 Individual covariates

7528 Individual covariates are those which are measured (or measurable) on individuals, so
 7529 we get to observe them only for the captured individuals. Sex is a simple example of an
 7530 individual covariate, but one of the most commonly used in capture-recapture studies. The
 7531 sex of an individual can influence many aspects of its ecology and behavior, including for
 7532 example, the frequency of movement, seasonal behavior, and its home range size. This is
 7533 common in studies of carnivores where females often have smaller home ranges than males
 7534 (Gardner et al., 2010b; Sollmann et al., 2011). Additionally, we may find differences in
 7535 the baseline encounter probability between males and females because females may move
 7536 around less frequently, or possibly because they are less likely to use landscape structures
 7537 that researchers may target with sampling devices in order to increase sample size, such
 7538 as roads (e.g. Salom-Pérez et al., 2007). Therefore, we can imagine that sex may impact
 7539 both the baseline encounter probability α_0 and the typical home range size, so that α_1
 7540 might also be sex-specific also. The fully sex-specific model is:

$$\text{logit}(p_{0,i}) = \alpha_{0,sex_i}$$

$$p_{ijk} = p_{0,i} \exp(-\alpha_{1,sex_i} * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

7541 where sex_i is a vector indicating the sex of each individual (1 = male, 2 = female). While
 7542 we might know the sex of all individuals observed in the study, we will never know the sex
 7543 of individuals that are not observed (Gardner et al., 2010b). It is also possible that we
 7544 may not be able to determine the sex of individuals that are observed during the study.
 7545 For example photographic captures do not necessarily result in pictures that allow the sex
 7546 to be absolutely determined, thus sometimes resulting in missing values of this covariate
 7547 for animals captured in the study. We deal with this slightly differently depending on

7548 the inference framework that we adopt (Bayesian or likelihood). Here we demonstrate
 7549 the Bayesian implementation and we discuss the likelihood approach using **secr** in detail
 7550 below in Sec. 7.4.2. Before proceeding with that, we note that it would be possible also to
 7551 model covariates directly on the parameter σ (or its logarithm), e.g., $\log(\sigma_i) = \theta_1 + \theta_2 \text{sex}_i$
 7552 (see Sec. 8.1). One or the other (or perhaps *some* other) parameterization may yield a
 7553 better performing MCMC algorithm or provide a more natural or preferred interpretation.
 7554 In the context of Bayesian analysis, given that priors are not invariant to transformation of
 7555 the parameters, this may be a consideration in choosing the particular parameterization.

7556 Specifying a fully sex-specific model for **JAGS** is similar to the time-specific model
 7557 shown above. We need to use an index or dummy variable to let α_0 and/or α_1 be defined
 7558 separately for males and females. The main difference in this specification is that we do
 7559 not observe sex for the augmented individuals. Therefore, we have missing observations
 7560 of the covariate for those individuals. As a result, sex is regarded as a random variable
 7561 and so the missing values can be estimated along with the other structural parameters of
 7562 the model.

7563 Because we are regarding sex as a random variable, we have to specify a distribution for
 7564 it. With only two possible outcomes, it is natural to suppose that $\text{Sex}_i \sim \text{Bernoulli}(\psi_{\text{sex}})$
 7565 where the parameter ψ_{sex} is the sex ratio of the population. We assume our default non-
 7566 informative prior for this parameter: $\psi_{\text{sex}} \sim \text{Uniform}(0, 1)$. The model specification in
 7567 Panel 7.2 demonstrates how to incorporate a partially observed covariate (i.e., “sex”). It
 7568 is important to note that in the previous equation, sex_i is a vector with two categories
 7569 indicating the sex of each individual (e.g., 1 = male, 2 = female). This corresponds
 7570 directly to having a binary indicator of sex (e.g., $\text{Sex}_i = 1$ if individual i is female, and 0
 7571 otherwise). In the Bayesian formulation of the model, we use both the binary indicator
 7572 (**Sex**) and a categorical indicator ($\text{Sex2} = \text{Sex} + 1$). The former (termed **Sex** in Panel
 7573 7.2) allows us to specify the Bernoulli distribution for the random variable, and the latter
 7574 (termed **Sex2**) allows us to use the dummy or indicator variable specification in the model.

7575 In both **JAGS** or **BUGS** missing data are indicated by **NA** in the data objects passed
 7576 to the program through the **bugs** or **jags** functions in **R**. To set up the data, we need to
 7577 create a vector of length M with the first n elements being 0 if individual i is a female, or
 7578 1 if i is a male (for the Fort Drum black bear data the function **bear.JAGS** extracts this
 7579 information automatically from the **beardata** object), and the subsequent $M - n$ elements
 7580 being **NA**. It is generally a good idea to provide starting values for the missing data, but we
 7581 cannot provide starting values for observed data; in this case where one vector (or other
 7582 object) contains both observed and missing data, initial values for the observed data have
 7583 to be specified as **NA**. The code snippet below shows you how to set up the data including
 7584 the **Sex** vector and the initial values function (the remainder of the code is identical to
 7585 what we've shown before).

```
7586 > sex <- beardata$sex #the sex data for captured individual
7587 > Sex <- c(sex-1, rep(NA, nz)) #sex enters as 1/2, this recodes it to 0/1
7588                                #so we can use Bernoulli distribution
7589
7590 > data <- list(y=y,Sex=Sex, M=M,K=K, J=ntraps, xlim=xlim, ylim=ylim,area=areaX)
7591 > params <- c('psi','p0','N', 'D', 'sigma', 'psi.sex')
7592 > inits <- function() { list(z=c(rep(1,nind), rbinom(nz,1,0.5)),psi=rnorm(1),
7593                           s=cbind(rnorm(M, xlim[1],xlim[2]), rnorm(M,ylim[1],ylim[2])),
```

```
7594     psi.sex=runif(1,Sex=c(rep(NA, nind), rbinom(nz,1,0.5)),
7595     sigma=runif(2,2,3),alpha0=runif(2)) }
```

7596 The **BUGS** model specification is shown in Panel 7.2.

```
model{

psi ~ dunif(0,1)                                # Prior distributions
psi.sex ~ dunif(0,1)
for(t in 1:2){
  alpha0[t] ~ dnorm(0,.1)
  logit(p0[t]) <- alpha0[t]
  alphai[t] <- 1/(2*sigma[t]*sigma[t])
  sigma[t] ~ dunif(0, 15)
}

for(i in 1:M){
  z[i] ~ dbern(psi)
  Sex[i] ~ dbern(psi.sex)                      # Sex is binary
  Sex2[i] <- Sex[i] + 1                         # Convert to categorical
  s[i,1] ~ dunif(xlim[1],xlim[2])
  s[i,2] ~ dunif(ylim[1],ylim[2])

  for(j in 1:J){
    d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
    y[i,j] ~ dbin(p[i,j],K)
    p[i,j] <- z[i]*p0[Sex2[i]]*exp(-alphai[Sex2[i]]*d[i,j]*d[i,j])
  }
}
N <- sum(z[])
D <- N/area
}
```

Panel 7.2: **JAGS** model specification for an SCR model with sex-specific encounter probability parameters.

7597 Our estimate of density under the fully sex-specific model is still very similar to the
 7598 previous models (Table 7.6), and while the baseline detection was not very different be-
 7599 tween males and females, we can see that they had very different σ estimates (note that
 7600 the BCIs do not overlap). As usual, you can reproduce this analysis by calling the function
 7601 `bear.JAGS` and set `model='SCRsex'`.

Table 7.6. Posterior summaries of parameter estimates from sex-specific SCR models for the Fort Drum black bear data set.

Parameter	Mean	SD	2.5%	97.5%
N	509.982	66.355	376	631
D	0.168	0.022	0.12	0.21
$p_{0,female}$	0.136	0.025	0.09	0.19
$p_{0,male}$	0.092	0.017	0.06	0.13
σ_{female}	1.542	0.132	1.31	1.83
σ_{male}	2.682	0.389	2.09	3.62
ψ_{sex}	0.310	0.068	0.19	0.45
ψ	0.784	0.103	0.58	0.97

7.3 INDIVIDUAL HETEROGENEITY

7602 Here we consider SCR models with individual heterogeneity. Capture-recapture models
 7603 with individual heterogeneity in detection probability, so-called model M_h , have a long
 7604 history in classical capture recapture models and they have special relevance to SCR (Sec.
 7605 4.4). While the advent of SCR models may appear to have rendered the use of classical
 7606 model M_h obsolete (because one major source of heterogeneity, namely exposure to the
 7607 trap array is being accounted for explicitly) we may still wish to consider heterogeneity
 7608 models for other biological reasons. It is reasonable to expect in real populations that there
 7609 exists heterogeneity in home range size and so we think that α_1 could exhibit heterogeneity
 7610 among individuals. As we noted previously, it may be advantageous or desirable in some
 7611 cases to model heterogeneity directly in terms of the scale parameter of the encounter
 7612 probability function, σ , or some other transformation of the “distance coefficient”, perhaps
 7613 even 95% home range area.

7614 In this section, we describe a class of spatial capture-recapture models to allow for
 7615 individual heterogeneity in encounter probability. In particular, one class of models we
 7616 propose explicitly admits individual heterogeneity in home range *size*. In addition, we con-
 7617 sider a standard representation for heterogeneity in which an additive individual-specific
 7618 random effect is included in the linear predictor for baseline encounter probability.

7619 7.3.1 Models of heterogeneity

7620 An obvious extension to the SCR model is to include an additive individual effect, analo-
 7621 gous to classical “model M_h ”. We’ll call this model “SCR+Mh”:

$$\begin{aligned} \text{logit}(p_{0,i}) &= \alpha_0 + \eta_i \\ p_{ijk} &= p_{0,i} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2) \end{aligned}$$

7622 where η_i is an individual random effect having distribution $[\eta|\sigma_p]$. A popular class of
 7623 models arises by assuming $\eta_i \sim \text{Normal}(0, \sigma_p^2)$ (Coull and Agresti, 1999; Dorazio and
 7624 Royle, 2003). We show how to implement this specific SCR + Mh model in Panel 7.3,
 7625 and this model can be used to analyze the Ft. Drum bear data by calling the function
 7626 `bear.JAGS` and setting `model='SCRh'`. While we show one possible implementation here,
 7627 many other random effects distributions are possible. A popular one is the finite-mixture

7628 of point masses (Norris and Pollock, 1996; Pledger, 2004) which we demonstrate how to
 7629 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[])
  D <- N/area                                         # N, D are derived
}
```

Panel 7.3: **JAGS** model specification for the SCR + Mh model with Gaussian encounter probability model and additive normal random effect.

7630 7.3.2 Heterogeneity induced by variation in home range size

7631 An alternative heterogeneity model, one that has more of a direct biological motivation and
 7632 interpretation, describes heterogeneity in home range size among individuals. To model
 7633 heterogeneity in home range area, we can assume a distribution for a transformation of
 7634 the scale parameter of the encounter probability model such as σ^2 , or $\log(\sigma^2)$, etc.. We
 7635 call this “model SCR + Ah” (Ah here for area-induced heterogeneity).

7636 Consider the following log-normal model for the individual scale parameter of the
 7637 Gaussian encounter probability model, σ_i^2 :

$$\log(\sigma_i^2) \sim \text{Normal}(\mu_{hra}, \tau_{hra}^2)$$

7638 then the 95% home range area has a scaled log-normal distribution with mean

$$6\pi \exp(\mu_{hra} + \tau_{hra}^2/2).$$

7639 The variance is slightly more complicated, but you can look up the variance of a log-normal
 7640 distribution and combine it with the 95% home range area calculation in Sec. 5.4 to work
 7641 out the implied variance of home range area under this model. We show two examples of
 7642 the implied *population* distribution of home range area under this log-normal model that
 7643 indicates a mean home range area of about 6.9 area units (Figure 7.1). The left panel
 7644 shows a standard deviation in home range area of 2.88 units and the right panel shows
 7645 a standard deviation in home range area of 0.70 units. The two cases were generated by
 7646 tweaking the μ_{hra} and τ_{hra}^2 parameters of the log-normal distribution to achieve a constant
 7647 expected value of home range area, but modify the standard deviation.

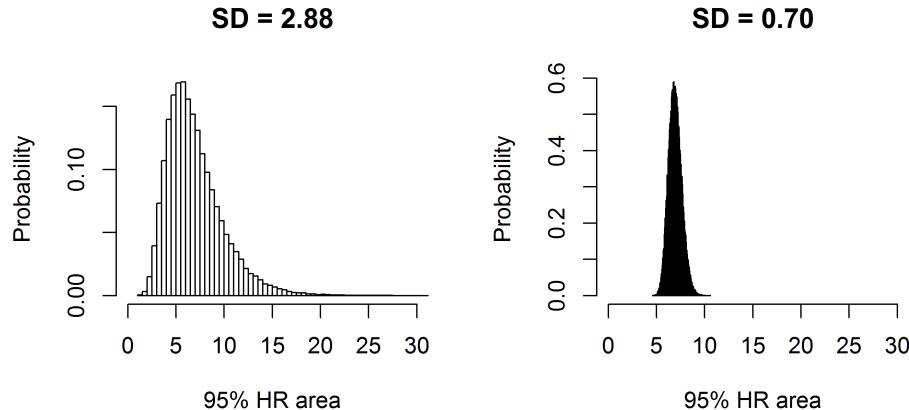


Figure 7.1. Population distribution of home range area for a model in which $\log(\sigma^2)$ has a normal distribution with mean μ_{hra} and variance τ_{hra}^2 . The parameters were chosen to yield a constant expected value of about 6.9 units of area, but to produce two different levels of heterogeneity: A population standard deviation of 2.88 units (left panel) and 0.70 units (right panel).

7.4 LIKELIHOOD ANALYSIS IN SECR

7648 Previously, in Chapt. 6, we introduced the R package **secr** and described the likelihood
 7649 based inference approach taken by that package (see Sec. 6.5.3). Here we discuss how
 7650 to implement some standard covariate models in **secr** and provide an example of model
 7651 selection using AIC. As we saw in Chapt. 6, **secr** uses the standard R model specifi-
 7652 cation syntax, defining the dependent and independent variable relationship using tildes

7653 (e.g., `y ~ x`). Thus, in `secr` we might have `g0 ~ behavior` or `sigma ~ time`; when left
 7654 unspecified or set to 1 (e.g., `g0 ~ 1`), this will default to a model with no covariates (i.e.,
 7655 constant parameter values). A number of default model formulas for the baseline and
 7656 scale parameter of the encounter probability model are available in `secr`. Additionally,
 7657 `secr` allows us to specify covariates on density (we cover this in Chapt. 11), which are
 7658 set for example as `D ~ habitat`.

7659 To demonstrate models with various types of covariates using `secr`, we continue using
 7660 the Fort Drum black bear data. We include in the `scrbook` package a function called
 7661 `secr.bear` that will format the data (see Chapt. 6 for the `secr` data format) and then fit
 7662 and compare 8 models (details shown in Panel 7.4). We have described all of these models
 7663 in the previous sections, so we only briefly comment here on how to fit certain models in
 7664 `secr` and compare them using AIC, and give a few helpful notes.

7665 7.4.1 Notes for fitting standard models

7666 In the `secr` package, the encounter probability model is called the “detection func-
 7667 tion” and it is specified by changing the “`detectfn`” option (an integer code) within the
 7668 `secr.fit` command. Table 7.1 shows the possible encounter probability models that `secr`
 7669 allows; the default is that based on the kernel of a bivariate normal probability distribu-
 7670 tion function (hence we call this the Gaussian model, but it is referred to as “half-normal”
 7671 in `secr`) and the (negative) exponential is `detectfn = 2`. See model 2 in Panel 7.4 for
 7672 how to fit the exponential model to the Fort Drum bear data set.

7673 The `secr` package easily fits a range of SCR equivalents of standard capture-recapture
 7674 models. The package has pre-defined versions of the classic model M_t where each oc-
 7675 casion has its own encounter probability, as well as a linear trend in baseline encounter
 7676 probability over occasions (in a spatial modeling framework σ could also be an occasion
 7677 specific parameter, but having encounter probability change with time seems like the more
 7678 common case). For the classical time-effects type of model with K distinct parameters
 7679 `secr` uses ‘t’ to denote this in the model specification formula (see model 3 in panel 7.4);
 7680 whereas, for a linear trend over occasions `secr` uses ‘T’.

7681 The global trap response model (what we called model M_B), or a local trap-specific
 7682 behavioral response (model M_b) can be fitted in `secr` using formulae with “b” for the
 7683 global response model and “bk” for the local trap response model (see models 4 and 5 in
 7684 Panel 7.4; note that to fit the trap specific behavioral response model you need version
 7685 2.3.1 or newer of `secr`).

7686 7.4.2 Sex effects

7687 Incorporating sex effects into models with `secr` can be done a few different ways, but
 7688 there are not pre-defined models for this. A limitation of fitting models with sex effects
 7689 in `secr` is that it does not accommodate missing values of the sex variable. Thus, in all
 7690 cases, individuals that are of unknown sex must be removed from the data set (recall that
 7691 in a Bayesian framework we can keep these individuals in the data set by specifying a
 7692 distribution for the individual covariate “sex”). In `secr`, the easiest way to include sex
 7693 effects is to code sex as a “session” variable using the multi-session models (see Sec. 6.5.4
 7694 for a description of the multi-session models), providing two sessions, one representing

```
1. null model with a bivariate normal encounter probability model  
bear_0=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ 1, sigma ~ 1))  
  
2. null model with an exponential encounter probability model  
bear_0exp=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ 1, sigma ~ 1),  
                    detectfn=2)  
  
3. model with fixed time effects  
bear_t=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ t, sigma ~ 1))  
  
4. global behavioral model  
bear_B=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ b, sigma ~ 1))  
  
5. trap specific behavioral response  
bear_b=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ bk, sigma ~ 1))  
  
6. global behavior model with fixed time effects  
bear_bt=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ b+t, sigma ~ 1))  
  
7. sex-specific model  
bear_sex=secr.fit(bear.cap, model=list(D ~ session, g0 ~ session,  
                                         sigma ~ session))  
  
8. heterogeneity model  
bear_h2=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ h2, sigma ~ h2))
```

Panel 7.4: Models called from **secr.bear** function. All models use **buffer = 20000**

7695 males and one for females (see model 7 in Panel 7.4). This method provides two separate
 7696 density estimates, which can then be combined into a total density.

7697 **7.4.3 Individual heterogeneity**

7698 To incorporate heterogeneity, **secr** fits a set of finite mixture models (Norris and Pollock,
 7699 1996; Pledger, 2004). These are expensive in terms of parameters but they have been
 7700 widely adopted because they are easy to analyze using likelihood methods, as the marginal
 7701 distribution of the data is just a sum of a small number of components. Using **secr**,
 7702 individual heterogeneity can be incorporated into the encounter probability model using
 7703 default models for either a 2- or 3-component finite mixture model using the “**h2**” or “**h3**”
 7704 model terms. The 2-part mixture is shown in model 8 of panel 7.4 and the 3-part mixture
 7705 can easily be fit by substituting **h3** for **h2**. We only showed the SCR + Mh logit-normal
 7706 mixture in the version above (see Sec. 7.3.1), but finite-mixture models can also be fit in
 7707 **JAGS** or **BUGS**.

7708 **7.4.4 Model selection in **secr** using AIC**

7709 One practical advantage to using the **secr** package, or likelihood inference in general, is
 7710 the convenience of automatic model selection using AIC (Burnham and Anderson, 2002).
 7711 The **secr** package has a number of convenient functions for computing AIC and producing
 7712 model selection tables, or doing model-averaging (as described in Chapt. 8). Running the
 7713 function **secr.bear**, which calls all of the models we have described, will return, in addition
 7714 to all model results, an AIC table with all of the summarized results including the AIC
 7715 values, delta AIC, and model weights (see Table 7.7 or reproduce results in R using **out<-**
 7716 **secr.bear()**; **out\$AIC.tab**).

7717 It is important to note that AIC is not comparable between a multi-session model and
 7718 a model that is not a multi-session model. Therefore, to compare the sex-specific model
 7719 (which uses “sessions”) with all the other models including the null, time, and behavioral
 7720 models, we coded the data set as a multi-session design when first loading it to **secr**. This
 7721 results in all the model outputs listing separate parameter estimates for each session, even
 7722 the null model with no covariates; however, the estimates are the same for both “sessions”
 7723 in all but the sex-specific model (in other words, we don’t specify any effect of session on
 7724 parameters, except in the sex specific model).

7725 The results from this AIC analysis are straightforward to interpret; the model with
 7726 a local trap response of encounter probability, “**bk**”, has a model weight of 1 and thus,
 7727 according to AIC, 100% support compared with the other models in this model set. The
 7728 2-part finite mixture model for g_0 and σ has the second lowest AIC, but considering the
 7729 large dAICc compared to the local trap response model we would probably not consider
 7730 it any further.

7.5 SUMMARY AND OUTLOOK

7731 There are endless covariates and encounter probability models that can be defined and our
 7732 goal in this chapter was to introduce basic types of covariate models and demonstrate how
 7733 to implement them in **BUGS** and **secr**. Essentially, SCR’s are GLMMs and therefore

Table 7.7. Log-likelihood, AIC, deltaAIC and AIC weight for several models run in secr for the Fort Drum black bear data set.

model	logLik	AIC	AICc	dAICc	AICwt
bear.b	-641.7215	1291.443	1292.395	0.000	1
bear.h2	-653.8382	1319.676	1321.776	29.381	0
bear.0exp	-663.9152	1333.830	1334.389	41.994	0
bear.B	-677.6175	1363.235	1364.187	71.792	0
bear.bt	-668.3044	1358.609	1366.152	73.757	0
bear.sex	-677.7151	1367.430	1369.530	77.135	0
bear.t	-674.4134	1368.827	1374.938	82.543	0
bear.0	-686.2455	1378.491	1379.049	86.654	0

7734 we develop covariate models in much the same way, using a suitable transformation (link
 7735 function) of the parameter(s). In SCR models, we typically have 2 parameters of the
 7736 encounter probability model for which we might specify covariate models – the baseline
 7737 encounter probability (or rate) parameter, and a scale parameter that is related in many
 7738 cases to the home range size of the species. A few examples of different covariate models
 7739 are given in Table 7.3. We can also consider covariates by their classification as fixed,
 7740 partially observed, or unobserved (see Table 7.8). This classification of covariate types
 7741 can be important because the MLE and Bayesian approaches to dealing with partially
 7742 and unobserved covariates is often different. This was seen above in how the covariate **Sex**
 7743 was handled in the two frameworks.

Table 7.8. Examples of different covariate classifications.

Covariate class	Examples
Fixed	baited, weather, habitat
Partially observed	sex, age,
Unobserved	home range size, ind. effects

7744 While the move to spatially explicit models in capture-recapture studies has largely
 7745 rendered the basic CR models (Otis et al., 1978) obsolete, we continue to find this clas-
 7746 sification useful for categorizing the *spatial* extensions of these standard CR models. The
 7747 extended models include the standard M_0 , M_t , M_b , and M_h , but also new models that
 7748 allow for trap-specific information such as "baited/not-baited" or "on/off road". In addi-
 7749 tion, in Chaps. 12, 13 and 11, we explore models for explaining variation in encounter
 7750 probability and density based on spatial covariates that describe variation in landscape or
 7751 habitat conditions.

7752
7753
7754

8

MODEL SELECTION AND ASSESSMENT

7755 Our purpose in life is to analyze models. By that, we mean one or more of the following
7756 basic 4 tasks: (1) estimate parameters, (2) make predictions of unobserved random vari-
7757 ables, (3) evaluate the relative merits of different models or choosing a best model (model
7758 selection), and (4) checking whether a specific model appears to provide a reasonable de-
7759 scription of the data or not (model checking, assessment, or “goodness-of-fit”). In previous
7760 chapters we addressed the problems of estimation of model parameters, and also making
7761 predictions of latent variables, s or z , or functions of these variables such as density or
7762 population size. In this chapter, we focus on the last two of these basic inference tasks:
7763 model selection (which model or models should be favored), and model assessment (do
7764 the data appear to be consistent with a particular model).

7765 In this chapter we review basic strategies of model selection using both likelihood
7766 methods (as implemented in the `secr` package) and Bayesian analysis. Specifically, we
7767 review a number of standard methods of model selection that apply to “variable selection”
7768 problems, when our set of models consists of distinct covariate effects and they represent
7769 constraints of some larger model. For classical analysis based on likelihood, model selection
7770 by Akaike Information Criterion (AIC) is the standard approach (Burnham and Anderson,
7771 2002). For Bayesian analysis we rely on a number of different methods. We demonstrate
7772 the use of the deviance information criterion (DIC) (Spiegelhalter et al., 2002) for variable
7773 selection problems although it has deficiencies when applied to hierarchical models in some
7774 cases (Millar, 2009). We use the Kuo and Mallick indicator variable selection approach
7775 (Kuo and Mallick, 1998) which produces direct statements of posterior model probabilities
7776 which we think are the most useful, and leads directly to model-averaged estimates of
7777 density. There is a good review paper recently by O’Hara and Sillanpää (2009) that
7778 discusses these and many other related ideas for variable selection. In addition to O’Hara
7779 and Sillanpää (2009) we also recommend Link and Barker (2010, Chapt. 7) for general
7780 information on model selection and assessment.

7781 To check model adequacy in a Bayesian framework, or whether a specific model pro-
7782 vides a satisfactory description of our data set, we rely exclusively on the Bayesian p-value
7783 framework (Gelman et al., 1996). For assessing fit of SCR models, part of the challenge

7784 is coming up with good measures of model fit, and there does not appear much definitive
 7785 guidance in the literature on this point. Following Royle et al. (2011a), we break the prob-
 7786 lem up into 2 components which we attack separately: (1) Conditional on the underlying
 7787 point process, does the encounter model fit? (2) Do the uniformity and independence
 7788 assumptions appear adequate for the point process model of activity centers? The latter
 7789 component of model fit has a considerable precedence in the ecological literature as it
 7790 is analogous to the classical problem of testing “complete spatial randomness” (Cressie,
 7791 1991; Illian et al., 2008).

7792 We apply some of these methods to the wolverine camera trapping data first introduced
 7793 in Chapt. 5 to investigate sex specificity of model parameters and whether there is a
 7794 behavioral response to encounter. We note that individuals are drawn to the camera
 7795 trap devices by bait and therefore it stands to reason that once an individual discovers a
 7796 trap, it might be more likely to return subsequently, a response termed “trap happiness”.
 7797 We evaluate whether certain models for encounter probability appear to be adequate
 7798 descriptions of the data, and we evaluate the uniformity assumption for the underlying
 7799 point process.

8.1 MODEL SELECTION BY AIC

7800 Using classical analysis based on likelihood, model selection is easily accomplished using
 7801 AIC (Burnham and Anderson, 2002) which we demonstrate below. The AIC of a model is
 7802 simply twice the negative log-likelihood evaluated at the MLE, penalized by the number
 7803 of parameters (np) in the model:

$$\text{AIC} = -2\log L(\hat{\theta}|\mathbf{y}) + 2np$$

7804 Models with small values of AIC are preferred. It is common to use a modified (“cor-
 7805 rected”) AIC referred to as AIC_c for small sample sizes which is

$$\text{AIC}_c = -2\log L(\hat{\theta}|\mathbf{y}) + \frac{2np(np+1)}{n-np-1}$$

7806 where n is the sample size. Two important problems with the use of AIC and AIC_c are
 7807 that they don’t apply directly to hierarchical models that contain random effects, unless
 7808 they are computed directly from the marginal likelihood (for SCR models we can do this,
 7809 see Chapt. 6). Moreover, it is not clear what should be the effective sample size n in
 7810 calculation of AIC_c , as there can be covariates that affect individuals, that vary over
 7811 time, or space. We do not offer strict guidelines as to when to use a small sample size
 7812 adjustment.

7813 The R package **secr** computes and outputs AIC automatically for each model fitted
 7814 and it provides some capabilities for producing a model selection table (function **AIC**) and
 7815 also doing model-averaging (function **model.average**), which we recommend for obtaining
 7816 estimates of density from multiple models.

8.1.1 AIC analysis of the wolverine data

7817 We provide an example of model selection for the wolverine camera trapping data using
 7818 **secr**. We consider a model set with distinct models to accommodate various types of sex
 7819 specificity of model parameters:

7821 Model 0: model SCR0 with constant density and constant encounter model parameters;
 7822 Model 1: model SCR0 with constant parameter values for both male and female wolverines but with sex-specific density only;
 7823 Model 2: Sex-specific density, sex-specific p_0 but constant σ ;
 7824 Model 3: Sex-specific density, sex-specific σ but constant p_0 ;
 7825 Model 4: Sex-specific density, sex-specific p_0 and sex-specific σ .

7827 To model sex-specific abundance (density), we use the multi-session models provided
 7828 by **secr** (introduced in Sec. 6.5.4), which allow one to model session-specific effects on
 7829 density, baseline encounter probability, p_0 (labeled g_0 in **secr**), and also the scale parameter
 7830 σ of the encounter probability model. Using this formulation, we define the “Session”
 7831 variable to be a *categorical* sex code having value 1 or 2 (demonstrated below) and thus
 7832 *session*-specific parameters represent *sex*-specific parameters. For example, if we model
 7833 session-specific density, D , then this corresponds to Model 1 in our list above. We note
 7834 that “Model 0” in our list corresponds to a model where all of the encounter histories
 7835 have the same session ID. This model is one of constant density, which implies that the
 7836 population sex ratio is fixed at 0.5, i.e., $\psi_{\text{sex}} = 0.5$.

7837 Although **secr** also uses the logit/log linear predictors as the default for modeling
 7838 covariates on baseline encounter probability and the scale parameter, respectively, **secr**
 7839 does something different with the multi-session models. It reports estimates in a *session*
 7840 *mean* parameterization (equivalent to, in **BUGS**, using an index variable instead of a set
 7841 of dummy variables), and not the *session effect* (i.e., deviation from the intercept) which
 7842 arises from the use of dummy variables. We show this **BUGS** model description in Sec.
 7843 8.2.2.

7844 To fit these models using **secr**, we load the wolverine data and do a slight bit of
 7845 formatting to prepare the data objects for analysis by **secr**. The key difference from our
 7846 analysis in Chapt. 6 is, here, we use the wolverine sex information (**wolverine\$wsex**)
 7847 which is a binary 0/1 variable (1=male) and we add 1 so that we can define a categorical
 7848 “Session” variable (having values 1 or 2). We also have a function **scr2secr** which converts
 7849 a standard trap-deployment file (TDF) matrix into a **secr** object of class “traps.” The
 7850 **R** commands are as follows (contained in the help file **?secr_wolverine**):

```

7851
7852 > library(secr)
7853 > library(scrbook)
7854 > data(wolverine)
7855 > traps <- as.matrix(wolverine$wtraps)

7856 ## Name variables as required by secr
7857 > dimnames(traps) <- list(NULL,c("trapID","x","y",paste("day",1:165,sep="")))
7858 ## Convert trap information to a secr "traps" object
7859 > trapfile <- scr2secr(scrtraps=traps,type="proximity")

7860 ## Grab the wolverine state-space grid (2km here)
7861 > gr <- as.matrix(wolverine$grid2)
7862 > dimnames(gr) <- list(NULL,c("x","y"))
7863 > gr2 <- read.mask(data=gr)

```

```

7866
7867 ## Grab the encounter data, and re-name variables
7868 > wolv.dat <- wolverine$wcaps
7869 > dimnames(wolv.dat) <- list(NULL,c("Session","ID","Occasion","trapID"))
7870
7871 ## Convert binary 0/1 sex variable to categorical 1/2 for "session"
7872 > wolv.dat[,1] <- wolverine$wsex[wolv.dat[,2]]+1
7873 > wolv.dat <- as.data.frame(wolv.dat)
7874
7875 ## Convert to capthist object
7876 > wolvcapt <- make.capthist(wolv.dat,trapfile,fmt="trapID",noccasions=165)

```

Once the data have been prepared in this way, we use the `secr` model fitting function `secr.fit` to fit the different models, and then the function `AIC` to package the models together and summarize them in the form of an AIC table, with rows of the table ordered from best to worst. The function `model.average` performs AIC-based model-averaging of the parameters specified by the `realnames` variable (below this is demonstrated for the parameter density, D). Because this function defaults to averaging by AIC_c , we slightly modified this function (called `model.average2`) to do model averaging by either AIC or AIC_c as specified by the user. The model fitting commands look like this (for Model 0 and Model 1):

```

7886 > model0 <- secr.fit(wolvcapt, model=list(D~1, g0~1, sigma~1),
7887                         buffer=20000)
7888 > model1 <- secr.fit(wolvcapt, model=list(D~session, g0~1, sigma~1),
7889                         buffer=20000)

```

Next we use the function `AIC`, passing the fit objects from all 5 models, and that produces the following output (abbreviated horizontally to fit on the page):

```

7892 > AIC (model0,model1,model2,model3,model4)
7893           model      ... npar logLik   AIC    AICc dAICc  AICwt
7894 model0  D~1 g0~1 sigma~1  ...  3 -627.2603 1260.521 1261.932 0.000 0.5831
7895 model2      ..      ...  5 -624.9051 1259.810 1263.810 1.878 0.2280
7896 model1      ..      ...  4 -627.2365 1262.473 1264.973 3.041 0.1275
7897 model4      ..      ...  6 -624.6632 1261.326 1267.326 5.394 0.0393
7898 model3      ..      ...  5 -627.2358 1264.472 1268.472 6.540 0.0222

```

Model averaging the results is done as follows:

```

7900 > model.average (model0,model1,model2,model3,model4,realnames="D")
7901           estimate  SE.estimate      lcl      ucl
7902 session=1 2.707190e-05 7.913577e-06 1.544474e-05 4.745224e-05
7903 session=2 2.927423e-05 8.270402e-06 1.700631e-05 5.039193e-05

```

As usual, estimates and standard errors of the individual model parameters can be obtained from the `secr.fit` summary output of any of the `modelX` objects shown above. The default output of estimated density is in individuals per ha, so we have to scale this up to something more reasonable. To get into units of per 1000 km², we need to first

7908 multiply by 100 to get to units of km^2 and then multiply by 1000. This produces an
 7909 estimated density of about 2.71 for `session=1` (females) and 2.93 for `session=2` (males).
 7910 We can use the generic **R** function `predict` applied to the `secr.fit` output to obtain
 7911 specific information about the MLEs on the natural scale.

7912 We don't necessarily agree with the use of AIC_c here and think its better to use AIC,
 7913 in general. This is because, as noted previously, it is not clear what the effective sample
 7914 size is for most capture-recapture problems. While we have 21 individuals in the data
 7915 set, most of the model structure has to do with encounter probability samples and for
 7916 that there are hundreds of observations. We do note that the AIC and AIC_c results are
 7917 not entirely consistent. By looking at the best model by AIC (Table 8.1), we find that
 7918 the model with sex specific density and sex-specific baseline encounter probability, p_0 , is
 7919 preferred (Model 2). This is just slightly better than the null model (Model 0) with no
 7920 sex effects at all and hence an implied fixed sex ratio of $\psi_{\text{sex}} = 0.50$.

Table 8.1. Model selection results for the wolverine models of sex specificity, with/without habitat mask. Fitting was done using `secr` with a half-normal (Gaussian) encounter probability model. Models are ordered by *AIC*. Density, *D*, is reported in units of individuals per 1000 km^2 . Model abbreviations indicate which parameters are sex-specific in order $D/p_0/\sigma$.

NO HABITAT MASK										
model	npar	Female			Male			D	p_0	σ
		AIC	AICc	D	p_0	σ				
2: sex/sex/1	5	1259.8	1263.8	2.45	0.08	6435.51	3.16	0.04	6435.51	
0: 1/1/1	3	1260.5	1261.9	2.83	0.06	6298.66	2.83	0.06	6298.66	
4: sex/sex/sex	6	1261.3	1267.3	2.59	0.08	6080.70	2.99	0.04	6833.16	
1: sex/1/1	4	1262.5	1265.0	2.69	0.06	6298.69	2.96	0.06	6298.69	
3: sex/1/sex	5	1264.5	1268.5	2.70	0.06	6280.49	2.95	0.06	6319.03	
WITH HABITAT MASK										
model	npar	Female			Male			D	p_0	σ
		AIC	AICc	D	p_0	σ				
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	

7921 We fit the same models but now using a modified state-space which excludes the ocean
 7922 (this is a habitat mask in `secr`). Results are shown in Table 8.1 along with the previous
 7923 models without a mask. We see AIC values are smaller for the model without the mask.
 7924 It is probably acceptable to compare these different fits (with and without habitat mask)
 7925 by AIC because we recognize the mask as having the effect of modifying the random
 7926 effects distribution (i.e., of the activity centers, *s*) and the results should be sensitive to
 7927 choice of the distribution for *s*. That said, we tend to prefer the mask model because it
 7928 makes sense to exclude the areas of open water from the state-space of *s*. For females the
 7929 model-averaged density is 3.88 individuals per 1000 km^2 and for males the model-averaged
 7930 density estimate is 4.46 individuals per 1000 km^2 as we see here:

7931 > `model.average (model0b,model1b,model2b,model3b,model4b,realnames="D")`

```

7932
7933      estimate   SE.estimate      lcl      ucl
7934 session=1 3.876615e-05 1.189102e-05 2.153795e-05 6.977518e-05
7935 session=2 4.459658e-05 1.323696e-05 2.523280e-05 7.882022e-05

```

7936 This is quite a bit higher than that based on the rectangular state-space (i.e., not
 7937 specifying a habitat mask). This is not surprising given that **the state-space is part**
 7938 **of the model** and the specific state-space modification we made here, which reduces the
 7939 area from the rectangular state-space, should be extremely important from a biological
 7940 standpoint (i.e., wolverines are not actively using open ocean).

8.2 BAYESIAN MODEL SELECTION

7941 Model selection is somewhat less straightforward as a Bayesian, and there is no canned
 7942 all-purpose method like AIC. As such we recommend a pragmatic approach, in general,
 7943 for all problems, based on a number of basic considerations:

- 7944 (1) For a small number of fixed effects we think it is reasonable to adopt a conventional
 7945 “hypothesis testing” approach – i.e., if the posterior for a parameter overlaps zero
 7946 substantially, then it is probably reasonable to discard that effect from the model.
- 7947 (2) Calculation of posterior model probabilities: In some cases we can implement methods
 7948 which allow calculation of posterior model probabilities. One such idea is the indicator
 7949 variable selection method from Kuo and Mallick (1998). For this, we introduce a latent
 7950 variable $w \sim \text{Bern}(.5)$ and expand the model to include the variable w as follows:

$$\text{logit}(p_{ijk}) = \alpha_0 + w * \alpha_1 * C_{ijk}.$$

7951 The importance of the covariate C is then measured by the posterior probability that
 7952 $w = 1$.

7953 (3) The Deviance Information Criterion (DIC): Bayesian model selection is now routinely
 7954 carried out using DIC ((Spiegelhalter et al., 2002)), although its effectiveness in hier-
 7955 archical models depends very much on the manner in which it is constructed (Millar,
 7956 2009). We recommend using it if it leads to sensible results, but we think it should be
 7957 calibrated to the extent possible for specific classes of models. This has not yet been
 7958 done in the literature for SCR models, to our knowledge.

7959 (4) Logical argument: For something like sex specificity of certain parameters, it seems
 7960 to make sense to leave an extra parameter in the model no matter what because, bio-
 7961 logically, we might expect a difference (e.g., home range size). In some cases failure to
 7962 apply logical argument leads to meaningless tests of gratuitous hypotheses (Johnson,
 7963 1999).

7964 In all modeling activities, as in life itself, the use of logical argument should not be under-
 7965 utilized.

7966 8.2.1 Model selection by DIC

7967 The availability of AIC makes the use of likelihood methods convenient for problems where
 7968 likelihood estimation is achievable. For Bayesian analysis, DIC seemed like a general-
 7969 purpose equivalent, at least for a brief period of time after its invention. However, there

7970 seem to be many variations of DIC, and a consistent version is not always reported across
 7971 computing platforms. Even statisticians don't have general agreement on practical issues
 7972 related to the use of DIC (Millar, 2009). Despite this, it is still widely reported. We think
 7973 DIC is probably reasonable for certain classes of models that contain only fixed effects,
 7974 or for which the latent variable structure is the same across models so that only the fixed
 7975 effects are varied (this covers many SCR model selection problems). However, it would be
 7976 useful to see some calibration of DIC for some standardized model selection problems.

7977 Model deviance is defined as negative twice the log-likelihood; i.e., for a given model
 7978 with parameters θ : $\text{Dev}(\theta) = -2 * \log L(\theta|\mathbf{y})$. The DIC is defined as the posterior mean
 7979 of the deviance, $\overline{\text{Dev}}(\theta)$, plus a measure of model complexity, p_D :

$$\text{DIC} = \overline{\text{Dev}}(\theta) + p_D$$

7980 The standard definition of p_D is

$$p_D = \overline{\text{Dev}}(\theta) - \text{Dev}(\bar{\theta})$$

7981 where the 2nd term is the deviance evaluated at the posterior mean of the model parameter(s), $\bar{\theta}$. The p_D here is interpreted as the effective number of parameters in the model.
 7982 Gelman et al. (2004) suggest a different version of p_D based on one-half the posterior
 7983 variance of the deviance:

$$p_V = \text{Var}(\text{Dev}(\theta)|\mathbf{y})/2.$$

7984 This is what is produced from **WinBUGS** and **JAGS** if they are run from **R2WinBUGS** or
 7985 **R2jags**, respectively. It is less easy to get DIC summaries from **rjags**, so we used **R2jags**
 7986 in our analyses below.

7988 8.2.2 DIC analysis of the wolverine data

7989 We repeated the analysis of the wolverine models with sex specificity, but this time doing
 7990 a Bayesian analysis paralleling the likelihood analysis we did above in **secr**, using the
 7991 logit/log parameterization of the model parameters. To do so in **BUGS**, we used dummy
 7992 variables. Thus, we can express models allowing for sex specificity using a dummy variable
 7993 **Sex** and new parameters (α_{sex} , β_{sex}) which represent the effect of **Sex** at level 1:

$$\text{logit}(p_{0,i}) = \alpha_0 + \alpha_{sex} \mathbf{Sex}_i$$

7994 and

$$\log(\sigma_i) = \log(\sigma_0) + \beta_{sex} \mathbf{Sex}_i.$$

7995 In these expressions, the sex variable \mathbf{Sex}_i is a binary variable where $\mathbf{Sex}_i = 0$ corresponds to female, and $\mathbf{Sex}_i = 1$ corresponds to male.

7996 Unlike the multi-session model in **secr**, we carry out the analysis of the sex-specific
 7997 model here by putting all of the data into a single data set, and explicitly accounting for
 7998 the covariate 'sex' in the model by assigning it a Bernoulli prior distribution with ψ_{sex}
 8000 being the proportion of males in the population. In this case, we produce "Model 0" above,
 8001 the model with no sex effect on density, by setting the population proportion of males at
 8002 one-half: $\psi_{sex} = 0.5$ (see also Sec. 7.2.4). As usual, handling of missing values of the
 8003 sex variable is done seamlessly which might be a practical advantage of Bayesian analysis

8004 in situations where sex is difficult to record in the field which may lead to individuals of
 8005 unknown sex (i.e., missing values).

8006 The **BUGS** model specification for the most complex model, Model 4, is shown in
 8007 Panel 8.1. This model has sex-specific intercept, scale parameter, σ , and density. We
 8008 provide an **R** script named `wolvSCR0ms` in the `scrbook` package which will fit each model.
 8009 The function uses **JAGS** by default for the fitting, using the `R2jags` package. The kernel
 8010 of this function is the model specification in Panel 8.1, which gets modified depending on
 8011 the model we wish to fit using a command line option `model`. For example, `model = 1`
 8012 fits the model with constant parameter values for males and females, but sex-specific
 8013 population sizes (`model = 0` constrains the male probability parameter, ψ_{sex} , to be 0.5).
 8014 The **R** function fits each of the 5 models using a binary indicator variable to turn ‘on’ or
 8015 ‘off’ each effect. Here is how we obtain the MCMC output for each of the 5 models:

```
8016 > wolv0 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=0)
8017 > wolv1 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=1)
8018 > wolv2 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=2)
8019 > wolv3 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=3)
8020 > wolv4 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=4)
```

8021 We fitted the 5 models to the wolverine data and summarize the DIC computation
 8022 results in Table 8.2. The model rank has model 0, model 2, model 1, model 4, model 3.
 8023 Interestingly, this is the same order as the models based on AIC_c which we found above
 8024 (see Table 8.1). The posterior mean and SD of model parameters under the 5 models are
 8025 given in Table 8.3.

Table 8.2. DIC results for the 5 models of sex specificity fitted to the wolverine camera trapping data, using the function `wolvSCR0ms`. Results are based on 3 chains of length 61000 yielding 180000 posterior samples.

Model	Meandev	p_D	DIC	Rank
Model 0	441.01	77.09	518.10	1
Model 1	441.78	77.504	519.28	3
Model 2	440.12	78.440	518.56	2
Model 3	443.31	79.478	522.79	5
Model 4	441.24	80.078	521.32	4

8026 8.2.3 Bayesian model averaging with indicator variables

8027 A convenient way to deal with model selection and averaging problems in Bayesian analysis
 8028 by MCMC is to use the method of model indicator variables (Kuo and Mallick,
 8029 1998). Using this approach, we expand the model to include a set of prescribed models
 8030 as specific reductions of a larger model. This has been demonstrated in some specific
 8031 capture-recapture models in Royle and Dorazio (2008, Sec. 3.4.3), and Royle (2009b) and
 8032 in the context of SCR by Tobler et al. (2012). A useful aspect of this method is that
 8033 model-averaged parameters are produced by default. We emphasize the need to be careful
 8034 of reporting model-averaged parameters that don’t have a common interpretation in

```

alpha.sex ~ dunif(-3,3)          ## Prior distributions
beta.sex ~ dunif(-3,3)
sigma0 ~ dunif(0,50)
alpha0 ~ dnorm(0,.1)
psi ~ dunif(0,1)                 ## Data augmentation parameter
psi.sex ~ dunif(0,1)              ## Probability of 'male'

for(i in 1:M){                   ## DA loop
  wsex[i] ~ dbern(psi.sex)       ## Latent sex state (male = 1)
  z[i] ~ dbern(psi)              ## DA variables, activity centers, etc..
  s[i,1] ~ dunif(Xl,Xu)
  s[i,2] ~ dunif(Yl,Yu)
  logit(p0[i]) <- alpha0 + alpha.sex*wsex[i]
  log(sigma.vec[i]) <- log(sigma0) + beta.sex*wsex[i]
  alpha1[i] <- 1/(2*sigma.vec[i]*sigma.vec[i])
  for(j in 1:ntraps){
    mu[i,j] <- z[i]*p[i,j]
    y[i,j] ~ dbin(mu[i,j],K[j])
    dd[i,j] <- pow(s[i,1] - traplocs[j,1],2) + pow(s[i,2] - traplocs[j,2],2)
    p[i,j] <- p0[i]*exp( - alpha1[i]*dd[i,j] )
  }
}

```

Panel 8.1: Part of the **BUGS** specification for a complete sex specificity of model parameters. This is a simplified version of the model contained in the **wolvSCR0ms** script, because it does not contain the on/off switches for creating the various sub-models.

Table 8.3. Posterior summaries of model parameters for models with varying sex specificity of model parameters. Model 0 = no sex specificity, model 4 = fully sex-specific (see text). Models are based on the Gaussian encounter probability model, each with 21000 iterations, 1000 burn-in, 3 chains for a total of 60000 posterior samples.

Parameter	model 0		model 1		model 2		model 3		model 4	
	Mean	SD								
N	60.02	11.91	60.24	11.93	59.37	11.97	59.67	11.97	58.77	11.75
D	5.79	1.15	5.81	1.15	5.72	1.15	5.75	1.15	5.66	1.13
α_0	-2.81	0.18	-2.82	0.17	-2.44	0.25	-2.82	0.18	-2.43	0.25
α_{sex}	0.00	1.73	0.00	1.73	-0.75	0.34	0.00	1.73	-0.79	0.36
σ_0	0.64	0.06	0.64	0.05	0.66	0.06	0.65	0.08	0.63	0.09
β_{sex}	0.00	1.73	-0.01	1.73	0.01	1.74	-0.01	0.17	0.10	0.18
ψ_{sex}	0.50	0.29	0.52	0.10	0.56	0.10	0.52	0.11	0.54	0.11
ψ	0.30	0.07	0.30	0.07	0.30	0.07	0.30	0.07	0.30	0.07
deviance	441.01	12.42	441.78	12.45	440.12	12.53	443.31	12.61	441.24	12.66
	$pD = 77.1$		$pD = 77.5$		$pD = 78.4$		$pD = 79.5$		$pD = 80.1$	
	$DIC = 518.1$		$DIC = 519.3$		$DIC = 518.6$		$DIC = 522.8$		$DIC = 521.3$	

the different models because they are meaningless (averaging apples and oranges....). For example, if a regression parameter is in a specific model then the posterior is informed by the data and a specific MCMC draw is from the appropriate posterior distribution. On the other hand, if the regression parameter is not in the model then the MCMC draw is obtained directly from the prior distribution, and so we need to think carefully about whether it makes sense to report an average of such a thing (in the vast majority of cases the answer is no). But some parameters like N or density, D , do have a consistent interpretation and we support producing model-averaged results of those parameters.

To implement the Kuo and Mallick approach, we expand the model to include the latent indicator variables, say w_m , for variable m in the model, such that

$$w_m = \begin{cases} 1 & \text{linear predictor includes covariate } m \\ 0 & \text{linear predictor does not include covariate } m \end{cases}$$

We assume that the indicator variables w_m are mutually independent with

$$w_m \sim \text{Bernoulli}(0.5)$$

for each variable $m = 1, 2, \dots$, in the model. For example, with 2 variables, the expanded model has the linear predictor:

$$\text{logit}(p_{ijk}) = \alpha_0 + \alpha_1 w_1 C_{1,i} + \alpha_2 w_2 C_{2,ijk}$$

where, let's suppose, $C_{1,i}$ is an individual covariate such as sex, and $C_{2,ijk}$ is a behavioral response covariate which is individual-, trap-, and occasion-specific. We can assume a parallel model specification on the parameter σ which is liable to vary by individual level covariates such as sex:

$$\log(\sigma_i) = \beta_0 + \beta_1 w_3 C_{1,i}.$$

Using this indicator variable formulation of the model selection problem we can characterize unique models by the sequence of w variables. In this case, each unique sequence (w_1, w_2, w_3) represents a model, and we can tabulate the posterior frequencies of each model by post-processing the MCMC histories of (w_1, w_2, w_3) , as we demonstrate shortly. This method then evaluates all possible combinations of covariates or 2^m models.

Conceptually, analysis of this expanded model within the data augmentation framework does not pose any additional difficulty. One broader, technical consideration is that posterior model probabilities are well known to be sensitive to priors on parameters (Aitkin, 1991; Link and Barker, 2006). See also Royle and Dorazio (2008, Sec. 3.4.3) and Link and Barker (2010, Sec. 7.2.5). What might normally be viewed as vague or non-informative priors, are not usually innocuous or uninformative when evaluating posterior model probabilities. The use of AIC seems to avoid this problem largely by imposing a specific and perhaps undesirable prior that is a function of the sample size (Kadane and Lazar, 2004). One solution is to compute posterior model probabilities under a model in which the prior for parameters is fixed at the posterior distribution under the full model (Aitkin, 1991). At a minimum, one should evaluate the sensitivity of posterior model probabilities to different prior specifications.

Analysis of the wolverine data

The **R** script `wolvSCROms` in the package `scrbook` provides the model indicator variable implementation for the fully sex-specific SCR model. It is run by setting `model=5` in the function call. We note again that it is not very useful to report most parameter estimates from this model because their marginal posterior is a mixture from the prior (when a value of the indicator variable of 0 is sampled) and draws informed by the data (i.e., from the posterior, when a 1 is drawn for the indicator variable w). On the other hand, the parameters N and density D should be reported and they represent marginal posteriors over all models in the model set. In effect, model averaging is done as part of the MCMC sampling. The variable ‘mod’ contains the two binary indicator variables (w above) which pre-multiply the ‘sex’ term in each of the p_0 and σ 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 ψ_{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:

```
8086 > mod <- wolv5$BUGSoutput$sims.list$mod
8087 > mod <- paste(mod[,1],mod[,2],mod[,3],sep="")
8088 >
8089 > table(mod)
8090 mod
8091   000   001   010   011   100   101   110   111
8092 17181 4935 1057 296 25211 8337 2275 708
8093
8094 > round( table(mod)/length(mod) , 3)
8095 mod
8096   000   001   010   011   100   101   110   111
8097 0.286 0.082 0.018 0.005 0.420 0.139 0.038 0.012
```

8098 This results in a comparison of all 8 possible models (based on $m = 3$ covariates) instead
 8099 of just the 5 models we originally proposed. We see that the best model is that labeled
 8100 100 which, according to our construction above, has `mod[1]=1, mod[2]=0` and `mod[3]=0`.
 8101 This is the model having sex-specific baseline encounter probability p_0 , and $\psi_{sex} = 0.5$.
 8102 This model has posterior model probability 0.420. The model with no sex specificity at
 8103 all (the model with label 000) has posterior probability 0.286 and the remaining posterior
 8104 mass is distributed over the other six models. We could arrive at a qualitatively similar
 8105 conclusion using a more ad hoc approach based on looking at the posterior mass for each
 8106 parameter under the full model (model 4; see Table 8.3, in part). Considering the sex-
 8107 specific intercept, it appears to be very important as its posterior mass is mostly away
 8108 from 0. On the other hand, the coefficient on log-sigma is concentrated around 0, and
 8109 the estimated ψ_{sex} (probability that an individual is a male) is 0.54 with a large posterior
 8110 standard deviation. We might therefore be inclined to discard the sex effect on $\log(\sigma)$
 8111 based on classical thinking-like-a-hypothesis-testing-person and settle for the model with
 8112 a sex-specific intercept in the encounter probability model. This is consistent with our
 8113 indicator variable approach which found that model (1,0,0) has posterior probability of
 8114 0.420. Looking at the posteriors for each parameter to thin the model down is consistent
 8115 with these results. We can obtain model-averaged estimates from the indicator variable
 8116 approach, which produces direct model-averaged estimates of N and D :

```
8117   mu.vect sd.vect  2.5%   25%   50%   75% 97.5% Rhat n.eff
8118 D     5.695   1.133  3.759  4.916  5.591  6.362  8.193 1.002 3600
8119 N    59.077  11.758 39.000 51.000 58.000 66.000 85.000 1.002 3600
```

8120 We obtain a model-averaged estimate (posterior mean) for density of $D = 5.695$ which
 8121 is hardly any different from our model specific estimates (Table 8.3) and, in particular,
 8122 from model 2 which has only a sex-specific intercept.

8123 8.2.4 Choosing among detection functions

8124 Another approach to implementing model indicator variables is to introduce a categorical
 8125 “model identity” variable which is itself a parameter of the model. Using this approach,
 8126 then each distinct model is associated with a unique set of covariates or other set of model
 8127 features. This is convenient especially when we cannot specify the linear predictor as
 8128 some general model that reduces to various alternative sub-models simply by switching
 8129 binary variables on or off. In the context of SCR models, choosing among different en-
 8130 counter probability models would be an example. For this case we do something like this
 8131 `mod ~ dcat(probs[])` where `probs` is a vector with elements $1/(\#models)$, and the en-
 8132 counter probability matrix is filled in depending on the value of `mod`. In particular, instead
 8133 of a 2-dimensional array `p[i,j]`, we build `p[i,j,m]` for each of $m = 1, 2, \dots, M$ models.
 8134 An example with 3 distinct models is:

```
8135   mod ~ dcat(probs[])
8136   ##
8137   ## Using a double loop construction fill-in p[,] for each model:
8138   ##
8139   p[i,j,1] <- p0[1]*exp( - alpha1[1]*dist2[i,j] )
```

```

8140 p[i,j,2] <- 1-exp(-p0[2]*exp( - alpha1[2]*dist2[i,j] ) )
8141 logit(p[i,j,3]) <- p0[3] - alpha1[3]*dist2[i,j]
8142
8143 mu[i,j] <- z[i]*p[i,j,mod]
8144 y[i,j] ~ dbin(mu[i,j],K[j])

```

8145 As before the posterior probabilities can be highly sensitive to priors on the different
8146 model parameters and sometimes mixing is really poor and, in general, we've experienced
8147 mixed success trying to carry out model selection using this construction. We do provide
8148 a template **R/JAGS** script (`wolvSCR0ms2`) in the `scrbook` package which has an example
8149 of choosing among 3 different encounter probability models: The Gaussian encounter
8150 probability, Gaussian hazard, and logistic model with the square of distance (defined
8151 in Sec. 7.1). The key things to note are that there are 3 intercepts and 3 different
8152 '`alpha1`' parameters (the coefficient on distance). The parameters should not be regarded
8153 as equivalent across the models, so it is important to have them separately defined (and
8154 estimated) for each model. In our analysis we used a vague normal prior (precision = 0.1)
8155 for the intercept parameter (either log or logit-scale of baseline encounter probability p_0)
8156 and a `Uniform(0,5)` prior for one-half the inverse of the coefficient on distance-squared. In
8157 the **BUGS** model specification the priors look like this:

```

8158 for(i in 1:3){
8159   alpha0[i] ~ dnorm(0,.1)
8160   sigma[i] ~ dunif(0,5)
8161   alpha1[i] <- 1/(2*sigma[i]*sigma[i])
8162 }

```

8163 Then, we create a probability of encounter for each individual, trap *and* model so that
8164 the holder object "p" in the model description is a 3-dimensional array (sometimes this
8165 would have to be a 4 or 5-d array in more complex models with time effects, etc..), so that
8166 construction of the encounter probability models look like this:

```

8167 p[i,j,1] <- p0[1]*exp( - alpha1[1]*dist2[i,j] )
8168 p[i,j,2] <- 1-exp(-p0[2]*exp( - alpha1[2]*dist2[i,j] ) )
8169 logit(p[i,j,3]) <- p0[3] - alpha1[3]*dist2[i,j]

```

8170 where

```

8171 logit(p0[1]) <- alpha0[1]
8172 log(p0[2]) <- alpha0[2]
8173 p0[3] <- alpha0[3]

```

8174 You can experiment with the `wolvSCR0ms2` script to investigate the importance of different
8175 models of encounter probability and whether they have an affect on the inferences.

8.3 EVALUATING GOODNESS-OF-FIT

8176 In practical settings, we estimate parameters of a desirable model, or maybe fit a bunch
8177 of models and report estimates from all of them or a model-averaged summary of density.

8178 An important question is: Is our model worth anything? In other words, does the model
8179 appear to be an adequate description of our data? Formal assessment of model adequacy or
8180 goodness-of-fit is a challenging problem and there are no all-purpose algorithms for doing
8181 this in either frequentist or Bayesian paradigms. Moreover, there are some philosophical
8182 challenges to evaluating model fit, such as, if we do model averaging then should all of
8183 the models have to fit? Or should the averaged model have to fit? What if none of the
8184 models fit? We don't know the answers to these questions and we won't try to answer
8185 them. Instead, we will provide what guidance we can on taking the first steps to evaluating
8186 fit, of a single model, as if it were a cherished family heirloom of great importance. We
8187 suggest that if you have a model that you really like, a single model, then it is a sensible
8188 thing to check that the model is a good fit to your data. If it is not, we do not imagine
8189 that the model is useless but just that some thought should be put into why the model
8190 doesn't fit so that, perhaps, some remediation might happen as future data are collected.
8191 After all, you may have spent 2, 3 or many more years of your life collecting that data set,
8192 perhaps thousands of hours, and therefore it seems a reasonable proposition to expect to
8193 do some estimation and analysis of the model regardless of model fit. You can still learn
8194 something from a model that does not pass some technical litmus test of model fit.

8195 Conceptually, we can think of evaluation of model fit as follows: if we simulate data
8196 under the model in question, do the simulated realizations resemble the data set that we
8197 actually have? For either Bayesian or classical inference, the basic strategy to assessing
8198 model fit is to come up with a fit statistic that depends on the parameters and the data
8199 set, which we denote by $T(\mathbf{y}, \theta)$, and then we compute this for the observed data set, and
8200 compare its value to that computed for perfect data sets simulated under the correct model.
8201 In the case of classical inference, we will often rely on the standard practice of parametric
8202 bootstrapping (Dixon, 2002), where we simulate data sets conditional on the MLE $\hat{\theta}$ and
8203 compare realizations with what we've observed. The R package **unmarked** (Fiske and
8204 Chandler, 2011) contains generic bootstrapping methods for certain hierarchical models,
8205 including distance sampling (e.g., see Sillett et al., 2012, for an application). In simple
8206 cases, using classical inference methods, it is sometimes possible to identify a test statistic
8207 of theoretical merit, perhaps with a known asymptotic distribution. For examples from
8208 capture-recapture see Burnham et al. (1987), Lebreton et al. (1992), and Chapt. 5 of
8209 Cooch and White (2006). For Bayesian analysis we use the Bayesian p-value method
8210 (Gelman et al., 1996) (we introduced the Bayesian p-value in sec. 3.9.1). Using this
8211 approach, data sets are simulated based on a posterior sample of the model parameters
8212 θ and some fit statistic for the simulated data sets, usually based on the discrepancy of
8213 the observed data from its expected values, is compared to that for the actual data. In
8214 most cases, whether Bayesian or frequentist, the main idea for assessing model fit is the
8215 same: We compare data sets from the model we're interested in with the data set we have
8216 in hand. If they appear to be consistent with one another, then our faith in the model
8217 increases, at least to some extent, and we say "the model fits."

8218 To date, we are unaware of any goodness-of-fit applications based on likelihood analysis
8219 of SCR models. For Bayesian analysis of SCR models, there has not been a definitive or
8220 general proposal for a fit statistic or even a class of fit statistics, although a few specialized
8221 implementations of Bayesian p-values have been provided (Royle, 2009b; Gardner et al.,
8222 2010a; Royle et al., 2011a; Gopalaswamy et al., 2012a,b; Russell et al., 2012). While
8223 we universally adopt the Bayesian p-value approach, and suggest some fit statistics in

the following text, we caution that there is no general expectation to support how well they should do. As such, one might consider doing some kind of custom evaluation or calibration when using such methods, if the power of the test (ability to reject under specific departures from the model) is of paramount interest. We note that this uncertain power or performance of the Bayesian p-value is not a weakness of the Bayesian approach because the same issue applies in using bootstrap approaches applied to classical analysis of models, if we were to devise such methods.

8.4 THE TWO COMPONENTS OF MODEL FIT

For most SCR models, there are at least two distinct components of model fit, and we propose to evaluate these two distinct components individually. First, we can ask, are the data consistent with the *observation* model, conditional on the underlying point process? We can evaluate this based on the encounter frequencies of individuals *conditional* on (posterior samples of) the underlying point process $\mathbf{s}_1, \dots, \mathbf{s}_N$. We discuss some potential fit statistics for addressing this in the next section. Second, we can evaluate whether the data appear consistent with the *state* process model (i.e., the “uniformity” assumption of the point process). For the simple model of independence and uniformity, this is similar to the assumption of *complete spatial randomness* (CSR) which we consider in Sec. 8.4.1 below. Actually, this is not strictly the assumption of CSR because of the binomial assumption on N under data augmentation, so we instead use the term *spatial randomness*.

8.4.1 Testing uniformity or spatial randomness

Historically, especially in ecology, there has been an extraordinary amount of interest in whether a realization of a point process indicates “complete spatial randomness,” i.e., that the points are distributed uniformly and independently in space. Two good references for such things are Cressie (1991, Ch. 8) and Illian et al. (2008)¹. In the context of animal capture-recapture studies, the spatial randomness hypothesis is manifestly false, purely on biological grounds. Typically individuals will be clustered, or more regular (for territorial species), than expected under spatial randomness and heterogeneous habitat will generate the appearance of clustering even if individuals are distributed independently of one another. While we recommend modeling spatial structure explicitly when possible (Chapts. 11, 12, 13), the uniformity assumption may be an adequate description of data sets in some situations. Further, we find that it is generally flexible enough to reflect non-uniform patterns in the data, because we do observe some direct information about some of the point locations.

The basic technical framework for evaluating the spatial randomness hypothesis is based on counts of activity centers in cells or bins. For that we use any standard goodness-of-fit test statistic, based on gridding (i.e., binning) the state-space of the point process into $g = 1, 2, \dots, G$ cells or bins, and we tabulate $N_g \equiv N(\mathbf{x}_g)$ the number of activity centers in bin g , centered at coordinate \mathbf{x}_g . Specifically, let $B(\mathbf{x})$ indicate a bin centered at coordinate

¹We 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, accessed January 24, 2013.

8261 \mathbf{x} , then² $N(\mathbf{x}) = \sum_{i=1}^N I(\mathbf{s}_i \in B(\mathbf{x}))$ is the population size of bin $B(\mathbf{x})$. In Sec. 5.11.1,
 8262 we used the summaries $N(\mathbf{x})$ for producing density maps from MCMC output. Here, we
 8263 use them for constructing a fit statistic. We have used the Freeman-Tukey statistic of this
 8264 form:

$$T(\mathbf{N}, \theta) = \sum_g (\sqrt{N_g} - \sqrt{\mathbb{E}(N_g)})^2$$

8265 where $\mathbb{E}(N_g)$ is estimated by the mean bin count. An alternative conventional assessment
 8266 of fit is based on the following statistic: Conditional on N , the total number of activity
 8267 centers in the state-space \mathcal{S} , the bin counts N_g should have a binomial distribution. It will
 8268 usually suffice to approximate the binomial cell counts by Poisson cell counts, in which
 8269 case we can use the classical “index-of-dispersion” test (Illian et al., 2008, p. 87), based
 8270 on the variance-to-mean ratio:

$$ID = (G - 1) * s^2 / \bar{N}$$

8271 where s^2 is the sample variance of the bin counts and \bar{N} is the sample mean. When the
 8272 point process realization is *observed*, as in classical point pattern modeling (but not in
 8273 SCR), this statistic has approximately a Chi-square distribution on $(G - 1)$ degrees-of-
 8274 freedom under the spatial randomness hypothesis. If $s^2 / \bar{N} > 1$, clustering is suggested
 8275 whereas, $s^2 / \bar{N} < 1$ suggests the point process is too regular.

8276 Whatever statistic we choose as our basis for assessing spatial randomness, *the im-*
 8277 *portant technical issue is that we don’t observe the point process and so the standard*
 8278 *statistics for evaluating spatial randomness cannot be computed directly. However, using*
 8279 *Bayesian analysis, we do have a posterior sample of the underlying point process and*
 8280 *so we suggest computing the posterior distribution of any statistic in a Bayesian p-value*
 8281 *framework. For a given posterior draw of all model parameters, N is known, based on the*
 8282 *value of the data augmentation variables z_i , and so we can obtain a posterior sample of*
 8283 *$N(\mathbf{x})$ by taking all of the output for MCMC iterations $m = 1, 2, \dots$, and doing this:*

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

8284 Thus, $N(\mathbf{x})^{(1)}, N(\mathbf{x})^{(2)}, \dots$, is the Markov chain for the derived parameter $N(\mathbf{x})$.

8285 In addition to computing the bin counts for each iteration of the MCMC algorithm,
 8286 at the same time we generate a realization of the activity centers \mathbf{s}_i under the spatial
 8287 randomness model, and we obtain bin counts for these “new” data, $\tilde{N}(\mathbf{x})$. For each of
 8288 the posterior samples – that of the real data, and that of the posterior simulated data, we
 8289 compute the fit-statistic. The fit statistic based on the actual data is:

$$T(\mathbf{N}, \theta) = \sum_x (\sqrt{N(\mathbf{x})} - \sqrt{\tilde{N}(\mathbf{x})})^2$$

8290 whereas the fit statistic based on a simulated realization of points under the spatial ran-
 8291 domness hypothesis is:

$$T(\tilde{\mathbf{N}}, \theta) = \sum_x (\sqrt{\tilde{N}(\mathbf{x})} - \sqrt{\tilde{N}(\mathbf{x})})^2$$

² $I(arg)$ is the indicator function which evaluates to 1 if *arg* is true, otherwise 0

8292 And we compute the Bayesian p-value by tallying up the proportion of times that $T(\tilde{\mathbf{N}}, \theta)$
 8293 is larger than $T(\mathbf{N}, \theta)$, as an estimate of: $p = \Pr(T(\tilde{\mathbf{N}}, \theta) > T(\mathbf{N}, \theta))$. The **R** function
 8294 **SCRgof** in our package **scrbook** will do this, given the output from **JAGS** (see below).

8295 Sensitivity to bin size

8296 Evaluating fit based on bin counts in point process models are sensitive to the number of
 8297 bins (Illian et al., 2008, p. 87-88). This is related to the classical problem of fit testing
 8298 for binary regression because in a point process model, as the number of grid cells gets
 8299 small, the grid cell counts go to 0 or 1 and standard fit statistics (e.g., based on deviance
 8300 or Pearson residuals) are known not to be very useful. There is some good discussion of
 8301 this in McCullagh and Nelder (1989, Sec. 4.4.5). What it boils down to is, using the
 8302 example of the Pearson residual statistic considered by McCullagh and Nelder (1989), the
 8303 fit statistic is exactly a deterministic function of the sample size only, which clearly should
 8304 not be regarded as useful for model fit. This is why, in order to do a check of model fit
 8305 when you have a binary response, one must always aggregate the data in some fashion. In
 8306 the context of testing spatial randomness, computing the test statistic we described above
 8307 has us chop up the region \mathcal{S} into bins, and tally up N_g , the frequency of activity centers
 8308 in each bin g . Suppose that we choose the bin size to be extremely small such that $\mathbb{E}(N_g)$
 8309 tends to N/G (N being the number of activity centers). Further, N_g tends to a binary
 8310 outcome. Therefore the fit statistic has N components that have value $N_g = 1$, and it has
 8311 $G - N$ components that have value $N_g = 0$. Therefore, the fit statistic resembles:

$$T(\mathbf{N}, \theta) = \sum_{g \ni N_g=1}^N (1 - \sqrt{N/G})^2 + \sum_{g \ni N_g=0}^{G-N} (N/G)^2 = N(1 + (G - N)/G)$$

8312 (here \ni means “such that”). If G is huge relative to N , then we see that this tends to
 8313 about $2 * N$, which does not provide any meaningful assessment of model fit. So if you
 8314 look at this in the limit in which the bin counts become binary, the fit statistic loses all
 8315 its variability to the specific model used and is just a deterministic function of N . As a
 8316 practical matter, it probably makes sense to restrict the number of bins to *fewer* than the
 8317 number of observed individuals in the sample size. In typical SCR applications this will
 8318 therefore result, usually, in very large (and few) bins, and presumably not much power.

8319 There are some extensions that help resolve the issue of sensitivity to bin size. We can
 8320 construct fit statistics based not just on quadrat counts but also the neighboring quadrat
 8321 counts – this is the Greig-Smith method (Greig-Smith, 1964). In addition, there are a
 8322 myriad of “distance methods” for evaluating point process models, and we believe that
 8323 many of these can (and will) be adapted to SCR models. Again the main feature is that
 8324 the point process on which inference is focused is completely latent in SCR models – so
 8325 this makes the fit assessment slightly different than in classical point processes. That said,
 8326 the methods should be adaptable, e.g., in a Bayesian p-value kind of way.

8327 Sensitivity to state-space extent

8328 An issue that we have not investigated is that any model assessment that applies to a *latent*
 8329 point process is probably sensitive to the size of the state-space. As the size of the state-
 8330 space increases then the cell counts (far away from the data) *are* independent binomial
 8331 counts with constant density, and so we can overwhelm the fit statistic with extraneous
 8332 “data” simulated from the posterior, which is equal to the prior as we move away from the

8333 data, and therefore uninformed by the observed data that live in the vicinity of the trap
 8334 array. Therefore we recommend computing these goodness-of-fit statistics in the vicinity
 8335 of the trap array only. Perhaps, as an ad hoc rule-of-thumb, less than the average trap
 8336 spacing from the rectangle enclosing the trap array. For example, if the average trap
 8337 spacing is, say, 10 km, then the bins used to obtain the observed and predicted activity
 8338 centers should not extend any further from the traps than 5 km. This should be a matter
 8339 of future research.

8340 8.4.2 Assessing fit of the observation model

8341 In evaluating the spatial randomness hypothesis, we could draw on well-established ideas
 8342 from point process modeling. On the other hand, it is less clear how to approach goodness-
 8343 of-fit evaluation of the observation model. For most SCR problems, we have a 3-dimensional
 8344 data array of *binary* observations, y_{ijk} for individual i , trap j and sample occasion k . As
 8345 discussed in the previous section, we need to construct fit statistics based on observed and
 8346 expected frequencies that are aggregated in some fashion. In practice, the data will be
 8347 too sparse to have much power, unless the data are highly aggregated. We recommend
 8348 focusing on summary statistics that represent aggregated versions of y_{ijk} over 1 or 2 of
 8349 the dimensions. We describe 3 such fit statistics below. We recognize that, depending on
 8350 the model, some information about model fit will be lost by summarizing the data in this
 8351 way. For example if there is a behavioral response and we aggregate over time to focus
 8352 on the individual and trap level summaries then some information about lack of fit due
 8353 to temporal structure in the data is lost.

8354 **Fit statistic 1: individual x trap frequencies** We summarize the data by individual
 8355 and trap-specific counts y_{ijk} aggregated over all sample occasions. Using standard
 8356 “dot notation” to represent summed quantities, we express that as: $y_{ij\cdot} = \sum_{k=1}^K y_{ijk}$.
 8357 Conditional on \mathbf{s}_i , the expected value under any encounter model is:

$$\mathbb{E}(y_{ij\cdot}) = p_{ij} K$$

8358 (or K_j if the traps are operational for variable periods). If there is time-varying structure
 8359 to the model, then expected values would have to be computed according to $\mathbb{E}(y_{ij\cdot}) =$
 8360 $\sum_k p_{ijk}$. Then we can define a fit statistic from the Freeman-Tukey residuals according
 8361 to:

$$T_1(\mathbf{y}, \theta) = \sum_i \sum_j (\sqrt{y_{ij\cdot}} - \sqrt{\mathbb{E}(y_{ij\cdot})})^2$$

8362 where we use θ here to represent the collection of all parameters in the model. This is
 8363 conditional on \mathbf{s} as well as on the data augmentation variables \mathbf{z} . We compute this statistic
 8364 for *each* iteration of the MCMC algorithm for the observed data set and also for a new
 8365 data set simulated from the posterior distribution, say $\tilde{\mathbf{y}}$.

8366 We could also use a similar fit statistic derived from summarizing over traps to obtain
 8367 an $n_{\text{ind}} \times K$ matrix of count statistics. We imagine that either summary of the data will
 8368 probably be too disaggregated (have mostly values of 0) in most practical settings to have
 8369 much power.

8370 **Fit statistic 2: Individual encounter frequencies.** SCR models represent a
 8371 type of model for heterogeneous encounter probability, like model M_h , but with an ex-
 8372 plicit factor (space) that explains part of the heterogeneity. For model M_h , the individual

8373 encounter frequencies are the sufficient statistic for model parameters, and so it makes in-
 8374 tuitive sense to provide some kind of omnibus fit assessment of the core heuristic that SCR
 8375 model is adequately explaining the heterogeneity using a model M_h -like statistic based
 8376 on individual encounter frequencies. So, we build a fit statistic based on the individual
 8377 total encounters (Russell et al., 2012), $y_{i..} = \sum_j \sum_k y_{ijk}$. In addition, the expected value
 8378 is a similar summary over traps and occasions: $\mathbb{E}(y_{i..}) = \sum_j \sum_k p_{ijk}$. Then, we define
 8379 statistic T_2 according to:

$$T_2(\mathbf{y}, \theta) = \sum_i (\sqrt{y_{i..}} - \sqrt{\mathbb{E}(y_{i..})})^2$$

8380 We imagine this test statistic should provide an omnibus test of extra-binomial variation
 8381 and should therefore capture some effect of variable exposure to encounter of individuals,
 8382 although we have not carried out any evaluations of power under specific alternatives.
 8383 Obviously, in using this statistic, we lose information on departures from the model that
 8384 might only be trap- or time-specific.

8385 **Fit Statistic 3: Trap frequencies.** We construct an analogous statistic based
 8386 on aggregating over individuals and replicates to form trap encounter frequencies: $y_{.j} =$
 8387 $\sum_i \sum_k y_{ijk}$ (Gopalaswamy et al., 2012b) and the expected value is a similar summary
 8388 over individuals and occasions: $\mathbb{E}(y_{.j}) = \sum_i \sum_k p_{ijk}$. Then statistic T_3 is:

$$T_3(\mathbf{y}, \theta) = \sum_j (\sqrt{y_{.j}} - \sqrt{\mathbb{E}(y_{.j})})^2$$

8389 This seems like a sensible fit statistic because we can think of SCR models as spatial
 8390 models for counts (Chandler and Royle, In press). Therefore, we should seek models that
 8391 provide good predictions of the observable spatial data, which are the trap totals. In this
 8392 context, it might even make sense to pursue cross-validation based methods for model
 8393 selection. Cross-validation is a standard method of evaluating models such as in kriging
 8394 or spline smoothing, so we could as well develop such ideas based on the trap-specific
 8395 frequencies.

8396 8.4.3 Does the SCR model fit the wolverine data?

8397 We use the ideas described in the previous section to evaluate goodness-of-fit of the SCR
 8398 model to the wolverine camera trapping data.

8399 We consider first whether the simple model of spatial randomness of the activity
 8400 centers is adequate. We think that the encounter model shouldn't have a large effect
 8401 on whether the spatial randomness assumption is adequate or not, so we fit "Model 0"
 8402 (in which parameters are *not* sex-specific) using an **R** script provided in the function
 8403 **wolvSCR0gof** which will default to fitting the model in **JAGS**. This is the same script as
 8404 **wolvSCR0ms** except that it saves the MCMC output for the activity centers **s** and the data
 8405 augmentation variables **z**, which are required in order to compute the Bayesian p-value
 8406 test of spatial randomness.

8407 The MCMC output is processed with the **R** function **SCRgof** which computes the test
 8408 of spatial randomness based on bin counts, using the Bayesian p-value calculation. The
 8409 function **SCRgof** requires a few things as inputs: (1) the output from a **BUGS** run (in
 8410 particular, the activity center coordinates and the data augmentation variables); (2) the

8411 number of bins to create for computing spatial frequencies of activity centers; (3) the trap
 8412 locations and, (4) the buffer around the trap array to use in computing the bin counts.
 8413 This buffer could be that used in defining the state-space for the model fitting, but we
 8414 think it should be relatively tighter to the trap array than the state-space used in model-
 8415 fitting. For the wolverine analysis, where we're using 10-km grid cells (1 unit = 10 km)
 8416 and a 20 km buffer for model fitting, we'll use a state-space buffer of 0.4 units (4 km) for
 8417 computing the fit statistic. The **R** code to fit the model and obtain the goodness-of-fit
 8418 result is as follows:

```
8419 > wolv1 <- wolvSCR0gof(nb=1000,ni=6000,buffer=2,M=200,model=0)
8420
8421 > bugsout <- wolv1$BUGSoutput$sims.list
8422
8423 > traplocs <- wolverine$wtraps[,2:3]
8424 > traplocs[,1] <- traplocs[,1] - min(traplocs[,1])
8425 > traplocs[,2] <- traplocs[,2] - min(traplocs[,2])
8426 > traplocs <- traplocs/10000
8427
8428 > set.seed(2013) # set seed so Bayesian p-value is the same each time
8429
8430 > SCRgof(bugsout,5,5,traplocs=traplocs,buffer=.4)
8431
8432 Cluster index observed: 1.099822
8433 Cluster index simulated: 1.000453
8434 P-value index of dispersion: 0.408
8435 P-value2 freeman-tukey: 0.6842667
```

8436 The output produced by **SCRgof** is the index of dispersion based on the ratio of the variance to the mean (see above), which is computed as the posterior mean index of dispersion for the latent point process, and also the average value for simulated data. If this value is > 1 then clustering is suggested, which we see a (very) minor amount of evidence for here. Two Bayesian p-values are produced: the first is based on the cluster index, and the 2nd is based on the Freeman-Tukey statistic calculated as described in Sec. 8.4.1. Because our p-values aren't close to 0 or 1, we judge that the model of spatial randomness provides an adequate fit to the data. You can verify that a similar result is obtained if we use the model with fully sex-specific parameters (Model 4).

8445 Next, we did a Bayesian p-value analysis of the observation component of the model,
 8446 using the 3 fit statistics described in Sec. 8.4.2. These statistics can be calculated as
 8447 part of the **BUGS** model specification or by post-processing the MCMC output returned
 8448 from a **BUGS** run. The **R** script **wolvSCR0gof** contains the relevant calculations. For
 8449 example, to compute fit statistic 1, we have to add some commands to the **BUGS** model
 8450 specification such as this (note: this is only a fraction of the model specification):

```
8451 .....
8452 for(j in 1:ntraps){
8453   mu[i,j] <- w[i]*p[i,j]
8454
8455   y[i,j] ~ dbin(mu[i,j],K[j])
```

```

8456   ynew[i,j] ~ dbin(mu[i,j],K[j])
8457
8458   err[i,j] <- pow(pow(y[i,j],.5) - pow(K[j]*mu[i,j],.5),2)
8459   errnew[i,j] <- pow(pow(ynew[i,j],.5) - pow(K[j]*mu[i,j],.5),2)
8460 }
8461
8462 Tlobs <- sum(err[,])
8463 Tnew <- sum(errnew[,])
8464 .....

```

8465 Similar calculations are carried out to obtain the posterior samples of test statistics 2
8466 (individual totals) and 3 (trap totals). For the wolverine data, the Bayesian p-value
8467 calculations produce:

```

8468 > mean(wolv1$BUGSoutput$sims.list$T1new>wolv1$BUGSoutput$sims.list$T1obs)
8469 [1] 0
8470
8471 > mean(wolv1$BUGSoutput$sims.list$T2new>wolv1$BUGSoutput$sims.list$T2obs)
8472 [1] 0.17
8473
8474 > mean(wolv1$BUGSoutput$sims.list$T3new>wolv1$BUGSoutput$sims.list$T3obs)
8475 [1] 0.02066667

```

8476 Based on statistic T_2 , we might conclude that the model is adequate for explaining
8477 individual heterogeneity although the other two statistics suggest a general lack of fit of
8478 the observation model. A similar result is obtained using the fully sex-specific model. We
8479 note that one individual was captured 8 times in one trap, which is pretty extreme under
8480 a model which assumes independent Bernoulli trials. We summarize that the trap-counts
8481 simply are not well-explained by this model.

8482 In attempt to resolve this problem, we extended the model to include a local (trap-
8483 specific) behavioral response (following Royle et al. (2011b)) which can be fitted using
8484 the sample **R** script **wolvSCRMb**. To fit a model using **WinBUGS**, and then compute the
8485 Bayesian p-values we do this:

```

8486 > wolv.Mb <- wolvSCRMb(nb=1000,ni=6000,buffer=2,M=200)
8487
8488 > mean(wolv.Mb$sims.list$T1new>wolv.Mb$sims.list$T1obs)
8489 [1] 0.9666667
8490
8491 > mean(wolv.Mb$sims.list$T2new>wolv.Mb$sims.list$T2obs)
8492 [1] 0.3644667
8493
8494 > mean(wolv.Mb$sims.list$T3new>wolv.Mb$sims.list$T3obs)
8495 [1] 0.4990667

```

8496 Given that this model seems to fit better, we might prefer reporting estimates under
8497 this model, which we do in Table 8.4. (the behavioral response parameter is labeled α_2
8498 in the table). Estimated density is about 1 individual higher per 1000 km² compared

8499 with the various models that lack a behavioral response. It might be useful to try these
 8500 fit assessment exercises using the habitat mask as described in Sec. 5.10. That takes
 8501 an extremely long time to run in **BUGS** though, especially for the behavioral response
 8502 model.

Table 8.4. Posterior summary statistics for local (trap-specific) behavioral response model M_b fitted to the wolverine camera trapping data using **WinBUGS**. The parameter α_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
σ	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
α_1	0.69	0.19	0.37	0.67	1.10	1.00	730
α_2	2.50	0.27	1.99	2.50	3.04	1.00	700
ψ	0.36	0.10	0.20	0.35	0.58	1.00	2600
T_1^{obs}	54.71	6.12	43.69	54.39	67.47	1.00	3900
T_1^{new}	64.73	7.62	50.93	64.39	80.96	1.00	3900
T_2^{obs}	13.93	4.07	7.25	13.53	23.04	1.00	5700
T_2^{new}	12.65	3.35	6.93	12.36	20.07	1.00	2000
T_3^{obs}	12.80	1.74	9.80	12.64	16.61	1.00	2400
T_3^{new}	12.94	3.05	7.77	12.67	19.58	1.00	15000

8.5 QUANTIFYING LACK-OF-FIT AND REMEDIATION

8503 Molinari-Jobin et al. (2013) used a strategy for assessing model fit in dynamic occupancy
 8504 models (Royle and Kéry, 2007) similar to that which we suggested above. They con-
 8505 structed a fit statistic based on aggregating the data over replicate samples (k), to obtain
 8506 the total detections per site i and year j . They used a Bayesian p-value analysis based on
 8507 a Chi-squared test statistic (also see Kéry and Schaub, 2012, Chapt. 12). Their analysis
 8508 suggested a model that didn't fit, and, so they computed the "lack-of-fit ratio" (see Kéry
 8509 and Schaub, 2012, Sec. 12.3) – the ratio of the fit statistic computed for the actual data to
 8510 that of the replicate data sets. They interpret this analogous to the over-dispersion coeffi-
 8511 cient in generalized linear models (McCullagh and Nelder, 1989), usually called the c-hat
 8512 statistic in capture-recapture literature (see Cooch and White, 2006, Chapt. 5). Molinari-
 8513 Jobin et al. (2013) reported the lack-of-fit ratio for their model to be 1.14 which suggests
 8514 a minor lack-of-fit, compared to perfect data having a value of 1, because the posterior
 8515 standard deviations will be too small by a factor of $\sqrt{1.14} = 1.07$. In classical capture-
 8516 recapture applications of goodness-of-fit assessment, inference for non-fitting models is
 8517 dealt with by inflating the resulting SEs (of the non-fitting model), by the square-root of
 8518 c-hat. We believe that these ideas related to quantifying lack-of-fit and understanding its
 8519 effect could also be applied to SCR models, although we have not yet explored this.

8.6 SUMMARY AND OUTLOOK

8520 In this chapter, we offered some general strategies for model selection and model checking,
8521 or assessment of model fit. We think the strategies we outlined for model selection are fairly
8522 standard and can be effectively applied to many SCR modeling problems. Some technical
8523 issues of Bayesian analysis need to be addressed (in general) before Bayesian methods
8524 are more generally useful and accessible. For one thing, Bayesian model selection based
8525 on the indicator variable approach of Kuo and Mallick (1998) can be tediously slow even
8526 for small data sets, and so improved computation will improve our ability to do Bayesian
8527 model selection in practical situations. Also, and most importantly, sensitivity to prior
8528 distributions is an important issue. Further research and practice might identify preferred
8529 prior configurations for SCR that provide a good calibration in relevant model selection
8530 problems. Finally, we believe that cross-validation should prove to be a useful method
8531 in model assessment and selection, as SCR models are a form of spatial model of counts,
8532 and so it is natural to pick models that predict the observable spatial counts (i.e., at trap
8533 locations) well.

8534 For Bayesian model assessment, or goodness-of-fit checking, we suggested a framework
8535 based on independent testing of the spatial model of independence and uniformity, and
8536 testing fit of the observation model conditional on the underlying point process. These
8537 ideas are based on mostly *ad hoc* attempts in a number of published applications (Royle
8538 et al., 2009a, 2011a; Gopalaswamy et al., 2012b; Russell et al., 2012, e.g.). While we think
8539 this general strategy should be fruitful, we know of no studies on the power to detect
8540 various model departures, and so the ideas should be viewed as experimental. We have
8541 not discussed assessment of model fit for SCR models using likelihood methods, although
8542 we imagine that standard bootstrapping ideas should be effective, perhaps based on the
8543 fit statistics (or similar ones) we suggested here for computing Bayesian p-values.

8544 Clearly there is much research to be done on assessment of model fit in SCR models.
8545 For testing the spatial randomness hypothesis, we used a classical approach based on
8546 count frequencies, in which point locations are put into spatial bins. Other approaches
8547 from spatial point process modeling should be pursued including nearest-neighbor methods
8548 or distance-based methods. In addition, studies to evaluate the power to detect relevant
8549 departures from the standard assumptions, and the robustness of inferences about N or
8550 density, need to be conducted. If the spatial randomness model appears inadequate, it
8551 is possible to fit models that allow for a non-uniform distribution of points (see Chapt.
8552 11) and even point process models that allow for interactions among points (Reich et al.,
8553 2012). On the other hand, we expect that most of these Bayesian p-value tests will have
8554 low power in typical data sets consisting of a few to a few dozen individuals. As such,
8555 failure to detect a lack of fit may not be that meaningful. But, on the other hand, it
8556 may not make a difference in terms of density estimates either. We think inference about
8557 density should be relatively insensitive to departures from spatial randomness, because
8558 we get to observe direct information on some component of the population, component
8559 of density is *observed*. For those activity centers, the assumed model of the point process
8560 should exert little influence on the placement of the activity centers. Conversely, as is
8561 the case with classical closed population models (Otis et al., 1978; Dorazio and Royle,
8562 2003; Link, 2003), inferences may be somewhat more sensitive to bad-fitting models for
8563 the observation process.

9

8564

8565

ALTERNATIVE OBSERVATION MODELS

8566

8567 In previous chapters we considered various models of *encounter probability*, both in terms
8568 of parametric functions of distance and also a myriad of covariate models (Chapt. 7 and
8569 elsewhere). However, we have so far only considered a specific probability model for the
8570 observations (we'll call this the "observation model") – the Bernoulli encounter process
8571 model which, in **secr**, is the *proximity detector* model. This assumes that individual and
8572 trap-specific encounters are independent Bernoulli trials.

8573 In this chapter, we focus on developing additional observation models. The observa-
8574 tion model could be thought of as being determined by the type of device – or the type of
8575 "detector" using the terminology of **secr** (Efford, 2011a). We consider models that apply
8576 when observations are not binary and, in some cases, that do not require independence of
8577 the observations. We present models when the data are encounter *frequencies*, based on the
8578 Poisson distribution, and observation models based on the multinomial distribution. For
8579 example, if sampling devices can detect an individual some arbitrary number of times dur-
8580 ing an interval, then it is natural to consider observation models for encounter frequencies,
8581 such as the Poisson model. Another type of encounter device is the "multi-catch" device
8582 (Efford et al., 2009a) which is a physical device that can capture and hold an arbitrary
8583 number of individuals. A typical example is a mist-net for birds (Borchers and Efford,
8584 2008). It is natural to regard observations from these kinds of studies as independent
8585 multinomial observations. A related type of device that produces *dependent* multinomial
8586 observations are the so-called *single-catch* traps (Efford, 2004; Efford et al., 2009a). The
8587 canonical example are small-mammal live traps which catch and hold a single individual.
8588 Competition among individuals for traps induces a complex dependence structure among
8589 individual encounters. To date, no formal inference framework has been devised for this
8590 method although it stands to reason that the independent multinomial model should be
8591 a good approximation in some situations (Efford et al., 2009a). We analyze a number of
8592 examples of these different observation models using **JAGS** and also the **R** package **secr**
8593 (Efford, 2011a).

9.1 POISSON OBSERVATION MODEL

8594 The models we analyze in Chapt. 5 assumed binary observations – i.e., standard encounter
 8595 history data – so that individuals are captured at most one time in a trap on any given
 8596 sample occasion. This makes sense for many types of DNA sampling (e.g., based on hair
 8597 snares) because distinct visits to sampled locations or devices cannot be differentiated.
 8598 However, for some encounter devices, or methods, the potential number of encounters is
 8599 *not* fixed, and so it is possible to encounter an individual some arbitrary number of times
 8600 during any particular sampling episode. That is, we might observe encounter frequencies
 8601 $y_{ijk} > 1$ for individual i , trap j and sampling interval k . As an example, if a camera
 8602 device is functioning properly it may be programmed to take photos every few seconds if
 8603 triggered. For a second example, suppose we are searching a quadrat or length of trail
 8604 for scat, we may find multiple samples from the same individual. Therefore, we seek
 8605 observation models that accommodate such encounter frequency data. In general, any
 8606 discrete probability mass function could be used for this purpose, including the standard
 8607 models for count data used throughout ecology, the Poisson and negative binomial. Here
 8608 we focus on using the Poisson model only although other count frequency models are
 8609 possible for SCR models (Efford et al., 2009b).

8610 Let y_{ijk} be the frequency of encounter for individual i , in trap j , during occasion k ,
 8611 then assume:

$$y_{ijk} \sim \text{Poisson}(\lambda_{ij})$$

8612 where the expected encounter frequency λ_{ij} depends on both individual and trap. As we
 8613 did in the binary model of Chapt. 5, we now seek to model the expected value of the
 8614 observation (which was p_{ij} in Chapt 5) as a function of the individual activity center \mathbf{s}_i .
 8615 We propose

$$\lambda_{ij} = \lambda_0 k(\mathbf{x}_j, \mathbf{s}_i)$$

8616 Where $k(\mathbf{x}, \mathbf{s})$ is any positive valued function, such as the negative exponential or the
 8617 bivariate Gaussian kernel, and λ_0 is the baseline encounter rate – the expected number
 8618 of encounters if a trap is placed precisely at an individuals home range center (note: in
 8619 `secr` the notation for this is g_0). Then, $\lambda_0 k(\mathbf{x}_j, \mathbf{s}_i)$ is the expected encounter rate in trap
 8620 \mathbf{x}_j for an individual having activity center \mathbf{s}_i . Note that

$$\log(\lambda_{ij}) = \log(\lambda_0) + \log(k(\mathbf{x}_j, \mathbf{s}_i)).$$

8621 Equating $\alpha_0 \equiv \log(\lambda_0)$, and, if $k(\mathbf{x}, \mathbf{s}) \equiv \exp(-d(\mathbf{x}, \mathbf{s})^2/(2\sigma^2))$ (i.e., the Gaussian model),
 8622 then:

$$\log(\lambda_{ij}) = \alpha_0 - \alpha_1 d(\mathbf{x}_j, \mathbf{s}_i)^2 \quad (9.1.1)$$

8623 where $\alpha_1 = 1/(2\sigma^2)$, which is the same linear predictor as we have seen for the Bernoulli
 8624 model in Chapt. 5. This Poisson SCR model is therefore a type of Poisson generalized
 8625 linear mixed model (GLMM).

8626 We can accommodate covariates at the level of individual-, trap- or sample occasion
 8627 by including them on the baseline encounter rate parameter λ_0 . For example, if C_j is
 8628 some covariate that depends on trap only, then we express the relationship between λ_0
 8629 and C_j as:

$$\log(\lambda_{0,ijk}) = \alpha_0 + \alpha_2 C_j$$

8630 and therefore covariates on the logarithm of baseline encounter probability appear also as
 8631 linear effects on λ_{ij} . In general, covariates might also affect the coefficient on the distance

term (α_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 λ then $x + y$ is Poisson with mean 2λ). 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 λ_0 serve the same role as affecting the base encounter rate. Since the observation model is the same, probabilistically speaking, for all values of K , evidently we need only $K = 1$ "survey" from which to estimate model parameters (Efford et al., 2009b). We know this intuitively, as sampling by multiple traps serves as replication in SCR models. This has great practical relevance to the conduct of capture-recapture studies and the use of SCR models. For example, if individuality is obtained by genetic information from scat sampling, one should only have to carry out a single spatial sampling of the study area. However, one must be certain that sufficient spatial recaptures will be obtained so that effective estimation is possible.

9.1.1 Poisson model of space usage

It is natural to interpret the Poisson encounter model as a model of space usage resulting from movement of individuals about their home range (Sec. 5.4). Imagine we have perfect samplers in every pixel of the landscape so that whenever an individual moves from one pixel to another, we can record it. Let m_{ij} be the number of times individual i was recorded in pixel j (i.e., it selected or used pixel j). Then, we might think of the Poisson model for the observed *use* frequencies:

$$m_{ij} \sim \text{Poisson}(\lambda_0 k(\mathbf{x}_j, \mathbf{s}_i))$$

where λ_0 is related to the baseline movement rate of the animal (how often it moves). This model of space usage gives rise to the standard resource selection function (RSF) models (see Chapt. 13). But now suppose our samplers are not perfect but, rather, record only a fraction of the resulting visits. A sensible model is

$$y_{ij}|m_{ij} \sim \text{Binomial}(m_{ij}, p).$$

The marginal distribution of y_{ij} is:

$$y_{ij} \sim \text{Poisson}(p_0 k(\mathbf{x}_j, \mathbf{s}_i)).$$

where p_0 is a composite of the movement rate and conditional detection probability p . Therefore, we see that encounters accumulate in proportion to the frequency of outcomes of an individual using space (or "selecting resources").

We introduced an interpretation of SCR models in terms of movement and space usage in Sec. 5.4, and it is one of the main underlying concepts of SCR models that is not present in ordinary capture-recapture models. As we noted there, the underlying model of space usage is only as complex as the encounter probability model which has been, so far in this book, only symmetric and stationary (does not vary in space). We generalize this model of space usage substantially in Chapt. 13.

9.1.2 Poisson relationship to the Bernoulli model

8668 There is a sense in which the Poisson and Bernoulli models can be viewed as consistent with
 8669 one another. Note that under the Poisson model, the relationship between the expected
 8670 count and the probability of counting “at least 1”, is given by
 8671

$$\Pr(y > 0) = 1 - \exp(-\lambda) \quad (9.1.2)$$

8672 where $\mathbb{E}(y) = \lambda$. Therefore, if we equate the event “encountered” with the event that the
 8673 individual was captured at least 1 time under the Poisson model, i.e., $y > 0$, then it would
 8674 be natural to set $p_{ij} = \Pr(y > 0)$ according to Eq. 9.1.2. That is, we can use Eq. 9.1.2
 8675 as the model for encounter probability for binary observations. This is the “hazard rate”
 8676 model in distance sampling.

8677 In fact, as λ gets small, the Poisson model is a close approximation to the Bernoulli
 8678 model in the sense that outcomes concentrate on $\{0, 1\}$, i.e., $\Pr(y \in \{0, 1\}) \rightarrow 1$ as $\lambda \rightarrow 0$.
 8679 Indeed, under the Poisson model, $\Pr(y > 0) \rightarrow \lambda$ for small values of λ . This phenomenon
 8680 is shown in Fig. 9.1 where the left panel shows a plot of $\lambda_{ij} = \lambda_0 k(\mathbf{x}_j, \mathbf{s}_i)$ vs. distance and
 8681 superimposed on that is a plot of $p_{ij} = 1 - \exp(-\lambda_{ij})$ vs. distance, for values $\lambda_0 = 0.1$
 8682 and $\sigma = 1$, and the right panel shows a plot of $\Pr(y > 0)$ vs. $\mathbb{E}(y)$. We see that the two
 8683 quantities are practically indistinguishable. This is convenient in some cases because the
 8684 Poisson model might be more tractable to fit (or even vice versa). For an example, see the
 8685 models described in Chapt. 18, and we also consider another case in Sec. 20.1.5 below.
 8686 To evaluate the closeness of the approximation, you can use the following R commands
 8687 which we used to produce Fig. 9.1:

```
8688 > x <- seq(0.001, 5, , 200)
8689 > lam0 <- .1
8690 > sigma <- 1
8691 > lam <- lam0*exp(-x**/(2*sigma*sigma))
8692
8693 > par(mfrow=c(1,2))
8694 > p1 <- 1-exp(-lam)
8695 > plot(x, lam, ylab="E[y] or Pr(y>0)", xlab="distance", type="l", lwd=2)
8696 > lines(x,p1,lwd=2,col="red")
8697 > plot(lam, p1, xlab="E[y]", ylab="Pr(y>0)", type="l", lwd=2)
8698 > abline(0,1,col="red")
```

8699 To summarize, if y is Poisson then, as λ gets small,

$$\begin{aligned} \Pr(y > 0) &\approx \mathbb{E}(y) \\ 1 - \exp(-\lambda_0 k(\mathbf{x}, \mathbf{s})) &\approx \lambda_0 k(\mathbf{x}, \mathbf{s}) \end{aligned} \quad (9.1.3)$$

8700 What all of this suggests is that if we have very few observations > 1 in our SCR data
 8701 set, then we won’t lose much information by using the Bernoulli model. On the other
 8702 hand, the Poisson model may have some advantages in terms of analytic or numerical
 8703 tractability in some cases. Further, this approximation explains the close correspondence
 8704 we have found between these two versions of the Gaussian encounter probability model
 8705 (Sec. 5.4). Namely, the Gaussian hazard model and the Gaussian encounter probability
 8706 model are close approximations because $1 - \exp(-\lambda) \approx \lambda$ if λ is small.

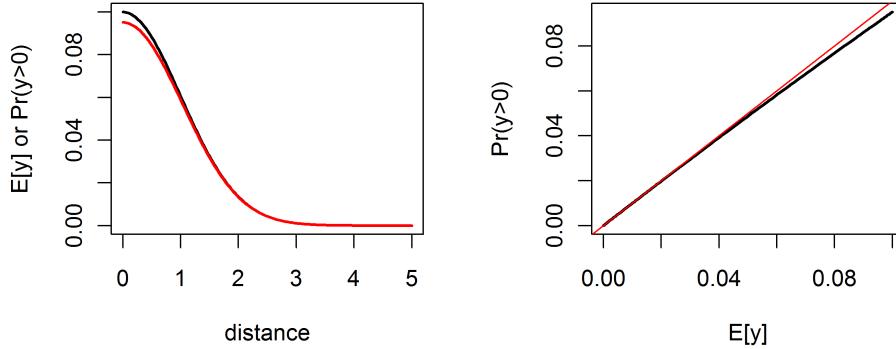


Figure 9.1. Poisson approximation to the binomial. As the Poisson mean approaches 0, then $\Pr(y > 0)$ under the Poisson model approaches λ and therefore $y \sim \text{Poisson}(\lambda)$ is well-approximated by a Bernoulli model with parameter λ .

Even in such cases where the Poisson and Bernoulli models are not quite equivalent, we might choose to truncate individual encounter frequencies to binary observations anyhow (transforming counts to 0/1 is called “quantizing”). We might do this intentionally in some cases, such as when the distinct encounter events are highly dependent as often happens in camera trap studies when the same individual moves back-and-forth in front of a camera during a short period of time. But sometimes, truncation is a feature of the sampling. For example, in the case of bear hair snares, the number of encounters might be well approximated by a Poisson distribution but we cannot determine unique visits and so only get to observe the binary event “ $y > 0$ ”. In this case, we might choose to model the encounter probability for the binary encounter using Eq. 9.1.4. This is equivalent to the complementary log-log link model, or the “Gaussian hazard” as we called it in Chapt. 5:

$$\text{cloglog}(p_{ij}) = \log(\lambda_0) + \log(k(\mathbf{x}, \mathbf{s}))$$

where $\text{cloglog}(u) = \log(-\log(1 - u))$.

9.1.3 A cautionary note on modeling encounter frequencies

Other models for counts might be appropriate. For example, ecologists are especially fond of negative binomial models for count data (Ver Hoef and Boveng, 2007; White and Bennetts, 1996; Kéry et al., 2005) but other models for excess-Poisson variation are possible. For example, we might add a normally distributed random effect to the linear predictor (Coull and Agresti, 1999).

As a general rule we favor the Bernoulli observation model even if our sampling scheme

8727 produces encounter frequencies. The main reason is that, with frequency data, we are
 8728 forced to confront a model choice problem (i.e., Poisson, negative binomial, log-normal
 8729 mixture) that is wholly unrelated to the fundamental space usage process that underlies
 8730 the genesis of many types of SCR data. Repeated encounters over short time intervals are
 8731 not likely to be the result of independent encounter events. E.g., an individual moving back
 8732 and forth in front of a camera yields a cluster of observations that is not informative about
 8733 the underlying spatial structure of the population. Similarly in scat surveys dogs are used
 8734 to locate scats which are processed in the lab for individuality (Kohn et al., 1999; MacKay
 8735 et al., 2008; Thompson et al., 2012). The process of local scat deposition is not strictly
 8736 the outcome of movement or space usage but rather the outcome of complex behavioral
 8737 considerations as well as dependence in detection of scat by dogs. For example, dogs find
 8738 (or smell) one scat and then are more likely to find one or more nearby ones, if present, or
 8739 they get into a den or latrine area and find many scats. The additional assumption required
 8740 to model variation in observed frequencies (i.e., conditional on location) provides relatively
 8741 no information about space usage and density, and we feel that the model selection issue
 8742 should therefore be avoided.

8743 To elaborate on this, we suppose that an individual with activity center \mathbf{s} visits
 8744 a particular pixel \mathbf{x} with some probability $p(\mathbf{x}, \mathbf{s})$, and then, once there, deposits a
 8745 number of scat, or visits a camera some number of times with frequency $y(\mathbf{x}, \mathbf{s}) \geq 0$.
 8746 We describe the outcome of this movement/usage process with a two-level hierarchical
 8747 model of the form: $[y|w][w|p(\mathbf{x}, \mathbf{s})]$ where $w(\mathbf{x}, \mathbf{s})$ is a binary variable that indicates
 8748 whether the individual with activity center \mathbf{s} used pixel \mathbf{x} during some interval, and let
 8749 $w(\mathbf{x}, \mathbf{s}) \sim \text{Bernoulli}(p(\mathbf{x}, \mathbf{s}))$. If we suppose encounter frequency y is independent of \mathbf{x} and
 8750 \mathbf{s} conditional on the use variable w , then we see that the model for y (amount of use) does
 8751 not depend on \mathbf{s} .

8752 9.1.4 Analysis of the Poisson SCR model in BUGS

8753 We consider the simplest possible model here in which we have no covariates that vary
 8754 over sample occasions $k = 1, 2, \dots, K$ so that we work with the aggregated individual-
 8755 and trap-specific encounters:

$$y_{ij} = (\sum_{k=1}^K y_{ijk}) = \text{Poisson}(K\lambda_0 k(\mathbf{x}_j, \mathbf{s}_i))$$

8756 and we consider the bivariate normal form of $k(\mathbf{x}, \mathbf{s})$:

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

8757 so that

$$\log(\lambda_{ij}) = \alpha_0 - \alpha_1 d(\mathbf{x}_j, \mathbf{s}_i)^2$$

8758 where $\alpha_0 = \log(\lambda_0)$ and $\alpha_1 = 1/(2\sigma^2)$.

8759 As usual, we approach Bayesian analysis of these models using data augmentation
 8760 (Sec. 4.2). Under data augmentation, we introduce a collection of all-zero encounter
 8761 histories to bring the total size of the data set up to M , and a corresponding set of data
 8762 augmentation variables $z_i \sim \text{Bern}(\psi)$. Then the observation model is specified conditional
 8763 on z according to:

$$y_{ij} \sim \text{Poisson}(z_i K \lambda_{ij})$$

which evaluates to a point mass at $y = 0$ if $z = 0$. In other words, the observation model under data augmentation is a zero-inflated Poisson model which is easily analyzed by Bayesian methods, e.g., in one of the **BUGS** dialects or, alternatively, using likelihood methods, which we neglect here although the same principles as in Chapt. 6 apply.

9.1.5 Simulating data and fitting the model

Simulating a sample SCR data set under the Poisson model requires only a couple minor modifications to the procedure we used in Chapt. 5 (see the function `simSCR0`). In particular, we modify the block of code which defines the model to be that of $E(y)$ and not $\Pr(y = 1)$, and we change the random variable generator from `rbinom` to `rpois`:

```
8773 ##  
8774 ## S =activity centers and traplocs defined as in simSCR0()  
8775 ##  
8776 ## Compute distance between activity centers and traps:  
8777 > D <- e2dist(S,traplocs)  
8778  
8779 ## Define parameter values:  
8780 > alpha0 <- -2.5  
8781 > sigma <- 0.5  
8782 > alpha1 <- 1/(2*sigma*sigma)  
8783  
8784 ## Encounter probability model:  
8785 > muy <- exp(alpha0)*exp(-alpha1*D*D)  
8786  
8787 ## Now generate the encounters of every individual in every trap  
8788 > Y <-matrix(NA,nrow=N,ncol=ntraps)  
8789 > for(i in 1:nrow(Y)){  
8790   Y[i,] <- rpois(ntraps,K*muy[i,])  
8791 }
```

We modified our simulation code from Chapt. 5 to simulate Poisson encounter frequencies for each trap and then we analyze an ideal data set using **BUGS**. This Poisson simulator function `simPoissonSCR` is available in the `scrbook` package (it can produce 3-d encounter history data too, although we don't do that here). Here is an example of simulating a data set and harvesting the required data objects, and doing the data augmentation:

```
8798 ## Simulate data and extract data elemements  
8799 ##  
8800 > data <- simPoissonSCR(discard0=TRUE,rnd=2013)  
8801 > y <- data$Y  
8802 > nind <- nrow(y)  
8803 > X <- data$traplocs  
8804 > K <- data$K  
8805 > J <- nrow(X)
```

```

8806 > xlim <- data$xlim
8807 > ylim <- data$ylim
8808
8809 ## Data augmentation
8810 > M <- 200
8811 > y <- rbind(y,matrix(0,nrow=M-nind,ncol=ncol(y)))
8812 > z <- c(rep(1,nind),rep(0,M-nind))

```

8813 The process for fitting the model in **WinBUGS** or **JAGS** is identical to what we've
 8814 done previously in Chapt. 5. In particular, we set up some starting values, package
 8815 the data and inits, identify the parameters to be monitored, and then send everything
 8816 off to our MCMC engine. Here it all is for fitting the Poisson observation model (these
 8817 commands are shown in the help file for `simPoissonSCR`):

```

8818 ## Starting values for activity centers
8819 ##
8820 > sst <- X[sample(1:J,M,replace=TRUE),]
8821 > for(i in 1:nind){
8822   if(sum(y[i,])==0) next
8823   sst[i,1] <- mean( X[y[i,>0,1] ) )
8824   sst[i,2] <- mean( X[y[i,>0,2] ) )
8825 }
8826 ## Dithered a little bit from trap locations
8827 > sst <- sst + runif(nrow(sst)*2,0,1)/8
8828 > data <- list (y=y,X=X,K=K,M=M,J=J,xlim=xlim,ylim=ylim)
8829 > inits <- function(){
8830   list (alpha0=rnorm(1,-2,.4),alpha1=runif(1,1,2),s=sst,z=z,psi=.5)
8831 }
8832 > parameters <- c("alpha0","alpha1","N","D")

```

8833 Next, we write the **BUGS** model to an external file:

```

8834 > cat("
8835 model{
8836   alpha0 ~ dnorm(0,.1)
8837   alpha1 ~ dnorm(0,.1)
8838   psi ~ dunif(0,1)
8839
8840   for(i in 1:M){
8841     z[i] ~ dbern(psi)
8842     s[i,1] ~ dunif(xlim[1],xlim[2])
8843     s[i,2] ~ dunif(ylim[1],ylim[2])
8844     for(j in 1:J){
8845       d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
8846       y[i,j] ~ dpois(lam[i,j])
8847       lam[i,j] <- z[i]*K*exp(alpha0)*exp(- alpha1*d[i,j]*d[i,j])
8848     }
8849   }

```

```

8850   N <- sum(z[])
8851   D <- N/64
8852 }
8853 ",file = "SCR-Poisson.txt")

```

8854 To fit the model we execute **bugs** in the usual way:

```

8855 > library(R2WinBUGS)
8856 > out1 <- bugs (data, inits, parameters, "SCR-Poisson.txt", n.thin=1,
8857           n.chains=3,n.burnin=1000,n.iter=2000,working.dir=getwd(),
8858           debug=TRUE)

```

8859 Or, using **JAGS** via **rjags** we would do something like this:

```

8860 > library(rjags)
8861 > jm <- jags.model("SCR-Poisson.txt", data=data, inits=inits,
8862           n.chains=3, n.adapt=1000)
8863 > out2 <- coda.samples(jm, parameters, n.iter=1000, thin=1)

```

8864 Summarizing the output from the **WinBUGS** run produces the following:

```

8865 > print(out1,digits=2)
8866 Inference for Bugs model at "SCR-Poisson.txt", fit using WinBUGS,
8867 3 chains, each with 2000 iterations (first 1000 discarded)
8868 n.sims = 3000 iterations saved
8869      mean    sd  2.5%   25%   50%   75% 97.5% Rhat n.eff
8870 alpha0  -2.57  0.19 -2.95 -2.69 -2.57 -2.44 -2.19 1.00 2600
8871 alpha1   2.34  0.36  1.69  2.08  2.32  2.57  3.12 1.00 3000
8872 N       114.13 15.25 87.97 103.00 113.00 124.00 147.00 1.01 370
8873 D       1.78  0.24  1.37  1.61  1.77  1.94  2.30 1.01 370
8874 deviance 329.95 21.92 290.00 314.20 329.50 344.40 375.80 1.00 1700
8875 ...
8876 [..some output deleted..]
8877 ...

```

8878 9.1.6 Analysis of the wolverine study data

8879 We reanalyzed the data from the wolverine camera trapping study that were first introduced in Sec. 5.9. We modified the **R** script from the function **wolvSCR0** to fit the Poisson model (see the help file for **wolvSCR0pois**). Executing this function produces the results shown in Table 9.1. The results are almost indistinguishable from the Bernoulli model fitted previously, where we had a posterior mean for N of 59.84 and σ was 0.64. You can edit the script **wolvSCR0pois** to obtain more posterior samples, or modify the model in some way.

Table 9.1. Results of fitting the SCR model with Poisson encounter frequencies to the wolverine camera trapping data. Posterior summaries were obtained using **WinBUGS** with 3 chains, each with 6000 iterations, discarding the first 1000 as burn-in, to yield a total of 15000 posterior samples.

Parameter	Mean	SD	2.5%	50%	97.5%	Rhat	n.eff
N	60.12	11.91	40.00	59.00	87.00	1	630
D	5.80	1.15	3.86	5.69	8.39	1	630
$\log(p_0)$	-2.89	0.17	-3.22	-2.89	-2.57	1	5000
λ_0	0.06	0.01	0.04	0.06	0.08	1	5000
σ	0.64	0.06	0.54	0.64	0.76	1	730
ψ	0.30	0.07	0.19	0.30	0.45	1	650

9.1.7 Count detector models in the secr package

The R package **secr** will fit Poisson or negative binomial encounter frequency models. The formatting of data and structure of the analysis proceeds in a similar fashion to the Bernoulli model described in Sec. 6.5, except that we specify the `detector='count'` option when the traps object is created. The set-up proceeds as follows:

```

8891 > library(secr)
8892 > library(scrbook)
8893 > data(wolverine)
8894
8895 > traps <- as.matrix(wolverine$wtraps)
8896 > dimnames(traps) <- list(NULL,c("trapID","x","y",paste("day",1:165,sep="")))
8897 > traps1 <- as.data.frame(traps[,1:3])
8898 > trapfile1 <- read.traps(data=traps1,detector="count")

```

You can proceed with analysis of these data and compare/contrast with the Bayesian analysis given above, or the results of the Bernoulli model fitted in Chapt. 6.

9.2 INDEPENDENT MULTINOMIAL OBSERVATIONS

Several types of encounter devices yield multinomial observations in which an individual can be caught in a single trap during a particular encounter occasion, but traps might catch any number of individuals. Mist netting is the canonical example of such a “multi-catch” device (Efford et al., 2009a). Also some kinds of bird or mammal cage-traps hold multiple animals, as do pit-fall traps which are commonly used for many species of herptiles. Another type of sample method that might be viewed (in some cases) as a multi-catch device are area-searches of, for example, reptiles where we think of a small polygon as the “trap” – we could get multiple individuals (turtles, lizards) in the same plot but not, in the same sample occasion, at different plots. The key features of this independent multinomial or multi-catch model are: (1) capture of an individual in a trap is *not* independent of its capture in other traps, because initial capture precludes capture in any other trap and (2) individuals behave independently of one another, so whether a trap captures some individual doesn’t have an affect on whether it captures another. A

8914 type of model in which the 2nd assumption is violated are the “single catch” trap systems
 8915 which we address in Sec. 20.1.5 below.

8916 In this case we assume the observation \mathbf{y}_{ik} for individual i during sample occasion k is
 8917 a multinomial observation which consists of a sequence of 0’s and a single 1 indicating the
 8918 trap of capture, or “not captured”. For the “not captured” event we define an additional
 8919 outcome, by convention element $J + 1$ of the vector. As an example, if we capture an
 8920 individual in trap 2 during some occasion of a study involving $J = 6$ traps. Then, the
 8921 multinomial observation has length $J+1 = 7$, and the observation is $\mathbf{y}_i = (0, 1, 0, 0, 0, 0, 0)$.
 8922 An individual not captured at all would have the observation vector $(0, 0, 0, 0, 0, 0, 1)$. If
 8923 we sample for 5 occasions in all and the individual is also caught in trap 4 during occasion
 8924 3, but otherwise uncaptured, then the 5 encounter observations for that individual are as
 8925 follows:

8926 occassion	-----trap ----- "not captured"						
	1	2	3	4	5	6	7

8929 1	0	1	0	0	0	0	0
8930 2	0	0	0	0	0	0	1
8931 3	0	0	0	1	0	0	0
8932 4	0	0	0	0	0	0	1
8933 5	0	0	0	0	0	0	1

8934 Statistically we regard the *rows* of this data matrix as *independent* multinomial trials.

8935 Analogous to our previous Bernoulli and Poisson models, we seek to construct the
 8936 multinomial cell probabilities for each individual, as a function of *where* that individual
 8937 lives, through its center of activity \mathbf{s} . Thus we suppose that

$$\mathbf{y}_{ik} | \mathbf{s}_i \sim \text{Multinomial}(1, \boldsymbol{\pi}(\mathbf{s}_i)) \quad (9.2.1)$$

8938 where $\boldsymbol{\pi}(\mathbf{s}_i)$ is a vector of length $J + 1$, where $\pi_{i,J+1}$, the last cell, corresponds to the
 8939 probability of the event “not captured”. Now we have to construct these cell probabilities
 8940 in some meaningful way that depends on each individual’s \mathbf{s} . We use the standard
 8941 multinomial logit with distance as a covariate:

$$\pi_{ij} = \frac{\exp(\alpha_0 - \alpha_1 d_{ij})}{1 + \sum_j \exp(\alpha_0 - \alpha_1 d_{ij})}$$

8942 for $j = 1, 2, \dots, J$ and, for $J + 1$, i.e., “not captured”,

$$\pi_{i,(J+1)} = \frac{\exp(0)}{1 + \sum_j \exp(\alpha_0 - \alpha_1 d_{ij})}$$

8943 or, more commonly, we use d_{ij}^2 to correspond to our Gaussian kernel model for encounter
 8944 probability. Whatever function of distance we use in the construction of multinomial prob-
 8945 abilities will have a direct correspondence to the standard encounter probability models
 8946 we used in the Bernoulli or Poisson models as well (see Sec. 5.4).

8947 It is convenient to express these multinomial models short-hand as follows, e.g., for
 8948 the Gaussian encounter probability model:

$$\text{mlogit}(\pi_{ij}) = \alpha_0 - \alpha_1 d_{ij}^2$$

8949 In this way we can refer to models with covariates in a more concise way. For example, a
 8950 model with a trap-specific covariate, say C_j , is:

$$\text{mlogit}(\pi_{ij}) = \alpha_0 - \alpha_1 d_{ij}^2 + \alpha_2 C_j$$

8951 or we could include occasion-specific covariates too, such as behavioral response.

8952 A statistically equivalent distribution to the multinomial is the *categorical* distribution.

8953 If \mathbf{y} is a multinomial trial with probabilities $\boldsymbol{\pi}$ than the *position* of the non-zero element of
 8954 \mathbf{y} is a categorical random variable with probabilities $\boldsymbol{\pi}$. We express this for SCR models
 8955 as

$$\mathbf{y}|\mathbf{s} \sim \text{Categorical}(\boldsymbol{\pi}(\mathbf{s}))$$

8956 In the SCR context, the categorical version of the multinomial trial corresponds to the
 8957 *trap of capture*. Using our example above with 6 traps then we could as well say y_{ik} is a
 8958 categorical random variable with possible outcomes (1, 2, 3, 4, 5, 6, 7) where outcome $y = 7$
 8959 corresponds to “not captured.” Obviously, how this is organized or labeled is completely
 8960 irrelevant, although it is convenient to use the integers 1 to $(J + 1)$ where $J + 1$ is the
 8961 event not captured. Therefore, for our illustration in the previous table, $y_{i1} = 2$, $y_{i2} = 7$,
 8962 $y_{i3} = 4$ and so on.

8963 For simulating and fitting data in the **BUGS** engines we will typically use the cat-
 8964 egorical representation of the model because it is somewhat more convenient. We have
 8965 found that fitting multinomial models in **WinBUGS** is less efficient than **JAGS** (Royle
 8966 and Converse, in review), which we use in the subsequent examples involving multinomial
 8967 observation models.

8968 9.2.1 Multinomial resource selection models

8969 The multinomial probabilities in Eq. 9.2.2 look similar to the multinomial resource selec-
 8970 tion function (RSF) model for telemetry data (Manly et al., 2002; Lele and Keim, 2006).
 8971 This suggests how we might model landscape or habitat covariates using such methods
 8972 – i.e., by including them as explicit covariates in a larger multinomial model for “use” –
 8973 which, if we take the product of use with encounter, produces a model for the observable
 8974 encounter data. This leads naturally to the development of models that integrate RSF
 8975 data from telemetry studies with SCR data (Royle et al., 2012a), which is the topic of
 8976 Chapt. 13.

8977 9.2.2 Simulating data and analysis using JAGS

8978 We’re going to show the nugget of a simulation function which is used in the function
 8979 **simMnSCR** found in the **R** package **scrbook**. The first lines of the following **R** code make
 8980 use of some things that you need to define, but we omit them here (e.g., **xlim**, **ylim** are
 8981 the boundaries of the state-space, **N** is the population size, etc.):

```
8982 ##
8983 ## Simulate random activity centers:
8984 ##      (first define N, xlim, ylim, etc..)
8985 ##
8986 > S <- cbind(runif(N,xlim[1],xlim[2]),runif(N,ylim[1],ylim[2]))
```

```

8987
8988 ## Distance from each individual to each trap
8989 > D <- e2dist(S,traplocs)
8990
8991 ## Set parameter values
8992 > sigma <- 0.5
8993 > alpha0 <- -1
8994 > alpha1 <- -1/(2*sigma*sigma)
8995
8996 ## make an empty data matrix and fill it up with data
8997 > Ycat <- matrix(NA,nrow=N,ncol=K)
8998 > for(i in 1:N){
8999   for(k in 1:K){
9000     lp <- alpha0 + alpha1*D[i,]*D[i,]
9001     cp <- exp(c(lp,0))
9002     cp <- cp/sum(cp)
9003     Ycat[i,k] <- sample(1:(ntraps+1),1,prob=cp)
9004   }
9005 }
```

9006 We save the data in the matrix `Ycat` to clarify that it is the categorical observation
 9007 representing “trap of capture”. The matrix `Ycat` here has the maximal dimension N
 9008 and so, to do an analysis that mimics a real situation, we would have to discard the
 9009 uncaptured individuals. The function `simMnSCR` in the package `scrbook` will also simulate
 9010 data that includes a behavioral response which will be the typical situation in small-
 9011 mammal trapping problems (see Converse and Royle, 2012, for details).

9012 Here we use our function `simMnSCR` to simulate a data set with $K = 7$ occasions. We’ll
 9013 run the model using `JAGS` which we have found is much more effective for this class of
 9014 models. We get the data set-up for analysis by augmenting the size of the data set to
 9015 $M = 200$. In addition we choose starting values for s and the data augmentation variables
 9016 z . For starting values of s we cheat a little bit here and use the true values for the observed
 9017 individuals and then augment the $M \times 2$ matrix \mathbf{S} with $M - n$ randomly selected activity
 9018 centers. Our function `spiderplot` returns the mean observed location of individuals for
 9019 use as starting values for the `nind` encountered individuals. The parameters input to
 9020 `simMnSCR` are the intercept α_0 , $\sigma = \sqrt{1/(2\alpha_1)}$ for the Gaussian encounter probability
 9021 model, and α_2 is the behavioral response parameter. The data simulation and set-up
 9022 proceeds as follows:

```

9023 > set.seed(2013)
9024 > parms <- list(N=100,alpha0= -.40, sigma=0.5, alpha2= 0)
9025 > data <- simMnSCR(parms, K=7, ssbuff=2)
9026 > nind <- nrow(data$Ycat)
9027
9028 > M <- 200
9029 > Ycat <- rbind(data$Ycat,matrix(nrow(data$X)+1,nrow=(M-nind),ncol=data$K))
9030 > Sst <- rbind(data$S,cbind(runif(M-nind,data$xlim[1],data$xlim[2]),
9031                           runif(M-nind,data$ylim[1],data$ylim[2])))
9032
```

```
9032 > zst <- c(rep(1,160),rep(0,40))
```

9033 The model specification is not much more complicated than the binomial or Poisson
 9034 models given previously. The main consideration is that we define the cell probabilities for
 9035 each trap $j = 1, 2, \dots, J$ and then define the last cell probability, $J+1$, for “not captured”,
 9036 to be the complement of the sum of the others. The code is shown in Panel 9.1. In the
 9037 last lines of code here we specify N and density, D , as derived parameters.

9038 To fit the model, we need to package everything up (inits, parameters, data) and send
 9039 it off to **JAGS** to build an MCMC simulator for us (these commands are executed in
 9040 the help file for `simMnSCR`). In addition to the usual data objects, we also pass the limits
 9041 of the assumed rectangular state-space (`ylim`, `xlim`, both 1×2 vectors) and the scale of
 9042 the standardized units, called `trap.space` here because we typically will define the trap
 9043 coordinates to be an integer grid. If the trap spacing is 10 m and we want units of density
 9044 computed in terms of individuals per meter-squared, then we input `trap.space=10`. The
 9045 analysis is carried out as follows:

```
9046 > inits <- function(){ list (z=zst,sigma=rnorm(1,.5,1) ,S=Sst) }  

9047  

9048 # Parameters to monitor  

9049 > parameters <- c("psi","alpha0","alpha1","sigma","N","D")  

9050  

9051 # Bundle the data. Note this reuses "data"  

9052 > data <- list (X=data$X,K=data$K, trap.space=1,Ycat=Ycat,M=M,  

9053   ntraps=nrow(data$X),ylim=data$ylim,xlim=data$xlim)  

9054  

9055 > library(R2jags)  

9056 > out <- jags (data, inits, parameters, "model.txt", n.thin=1,  

9057   n.chains=3, n.burnin=1000, n.iter=2000)
```

9058 The posterior summaries are provided in the following **R** output (recall that $N = 100$,
 9059 $\alpha_0 = -0.40$, and $\sigma = 0.5$):

```
9060 > out  

9061 Inference for Bugs model at "model.txt", fit using jags,  

9062   3 chains, each with 2000 iterations (first 1000 discarded)  

9063   n.sims = 3000 iterations saved  

9064      mu.vect sd.vect    2.5%     25%     50%     75%   97.5% Rhat n.eff  

9065 D        1.873   0.189   1.531   1.750   1.859   2.000   2.250 1.006 1300  

9066 N       119.867  12.107  98.000 112.000 119.000 128.000 144.000 1.006 1300  

9067 alpha0   -0.435   0.151  -0.738  -0.535  -0.439  -0.331  -0.146 1.004  580  

9068 alpha1    2.195   0.286   1.658   2.004   2.180   2.372   2.785 1.003 2400  

9069 psi      0.599   0.069   0.465   0.552   0.599   0.645   0.739 1.006 1400  

9070 sigma    0.480   0.032   0.424   0.459   0.479   0.500   0.549 1.003 2400  

9071 deviance 892.164 21.988 850.922 877.417 891.561 906.246 937.728 1.003  950  

9072  

9073 [... output deleted ....]
```

```

model{
psi ~ dunif(0,1)
alpha0 ~ dnorm(0,10)
sigma ~ dunif(0,10)
alpha1 <- 1/(2*sigma*sigma)

for(i in 1:M){
  z[i] ~ dbern(psi)
  S[i,1] ~ dunif(xlim[1],xlim[2])
  S[i,2] ~ dunif(ylim[1],ylim[2])
  for(j in 1:ntraps){
    #distance from capture to the center of the home range
    d[i,j] <- pow(pow(S[i,1]-X[j,1],2) + pow(S[i,2]-X[j,2],2),1)
  }
  for(k in 1:K){
    for(j in 1:ntraps){
      lp[i,k,j] <- exp(alpha0 - alpha1*d[i,j])*z[i]
      cp[i,k,j] <- lp[i,k,j]/(1+sum(lp[i,k,]))
    }
    cp[i,k,ntraps+1] <- 1-sum(cp[i,k,1:ntraps]) # last cell = not captured
    Ycat[i,k] ~ dcat(cp[i,k,])
  }
}
N <- sum(z[1:M])
A <- ((xlim[2]-xlim[1])*trap.space)*((ylim[2]-ylim[1])*trap.space)
D <- N/A
}

```

Panel 9.1: **BUGS** model specification for the independent multinomial observation model. For data simulation and model fitting see the help file `?simMnSCR` in the **R** package `scrbook`.

9074 9.2.3 Multinomial relationship to the Poisson

9075 The multinomial is related to the Poisson encounter rate model by a conditioning argument.
 9076 Let y_{ij} be the number of encounters for individual i in trap j . If $y_{ij} \sim \text{Poisson}(\lambda_{ij})$,
 9077 then, conditional on the *total* number of captures (i.e., across all traps), $y_i = \sum_j y_{ij}$, the
 9078 trap encounter frequencies are multinomial with probabilities

$$\pi_{ij} = \frac{\lambda_{ij}}{\sum_j \lambda_{ij}}$$

9079 for $j = 1, 2, \dots, J$. Or equivalently the *trap of capture* is categorical with probabilities π_{ij}
 9080 as given above. Under the Gaussian kernel model, these probabilities are:

$$\pi_{ij} = \frac{\exp(-\alpha_1 d(\mathbf{x}_i, \mathbf{s}_i)^2)}{\sum_j \exp(-\alpha_1 d(\mathbf{x}_i, \mathbf{s}_j)^2)} \quad (9.2.2)$$

9081 where, we note, the intercept α_0 has canceled from both the numerator and denominator.
 9082 This makes sense because, here, these probabilities describe the trap-specific capture prob-
 9083 abilities *conditional on capture*. Therefore, the model is not completely specified, absent
 9084 a model for the “overall” probability of encounter or the expected frequency of captures,
 9085 say ϕ_i . Depending on how we specify a model for this quantity ϕ_i , we can reconcile it
 9086 directly with the Poisson model. Let y_i be the total number of encounters for individual
 9087 i and suppose y_i has a Poisson distribution with mean ϕ_i . Then, marginalizing Eq. 9.2.1
 9088 over the Poisson distribution for y_i produces the original set of *iid* Poisson frequencies
 9089 with probabilities:

$$\lambda_{ij} = \phi_i \pi_{ij}$$

9090 for $j = 1, 2, \dots, J$. In particular, if we suppose that $\phi_i = \sum_j \exp(\alpha_0 - \alpha_1 d(\mathbf{x}, \mathbf{s})^2)$ then
 9091 the marginal distribution of y_{ij} is Poisson with mean $\exp(\alpha_0 - \alpha_1 d(\mathbf{x}, \mathbf{s})^2)$, equivalent to
 9092 Eq. 9.1.1.

9093 In summary, the Poisson and multinomial models are equivalent in how they model
 9094 the distribution of captures among traps. It stands to reason that, if the encounter
 9095 rate of individuals is low, we could use the Poisson and multinomial models interchange-
 9096 ably. In fact, based on our discussion in Sec. 9.1.2 above we could use any of the bino-
 9097 mial/Poisson/multinomial models with little ill-effect when encounter rate is low.

9098 9.2.4 Avian mist-netting example

9099 We analyze data from a mist-netting study of ovenbirds, conducted at the Patuxent
 9100 Wildlife Research Center, Laurel MD, by D.K. Dawson and M.G. Efford. The data from
 9101 this study are available in the **secr** package, and have been analyzed previously by Efford
 9102 et al. (2004), see also Borchers and Efford (2008). Forty-four mist nets spaced 30 m apart
 9103 on the perimeter of a 600-m x 100-m rectangle were operated on 9 or 10 non-consecutive
 9104 days in late May and June for 5 years from 2005-2009. The ovenbird data can be loaded
 9105 as follows:

```
9106 > library(secr)
9107 > data(ovenbird)
```

9108 The data set consists of adult ovenbirds caught during sampling in each of 5 years, 2005-
9109 2009. (one ovenbird was killed in 2009, indicated by a negative net number in the encounter
9110 data file). As with most mist-netting studies, nets are checked multiple times during a
9111 day (e.g., every hour during a morning session). However, for this data set, the within-day
9112 recaptures are not included so each bird has at most a single capture per day. Therefore
9113 the multinomial model (detector type ‘multi’ in **secr**) is appropriate. Although several
9114 individuals were captured in more than one year, this information is not used in the models
9115 presently offered in **secr**, but we do make use of it in the development of open models in
9116 Chapt. 16.

9117 **Multiple sample sessions**

9118 Up to this point we have only dealt with a basic closed population sampling situation
9119 consisting of repeated sample occasions on a single population of individuals using a single
9120 array of traps. In practice, many studies produce repeated samples over longer periods
9121 of time over which demographic closure isn’t valid, or at different locations where the
9122 populations are completely distinct. We adopt the **secr** terminology of *session* for such
9123 replication by groups of time or space, and the models are *multi-session* models, although
9124 we think of such models as being relevant to any stratified population (see Chapt. 14).
9125 We introduced **secr**’s multi-session models in Sec. 6.5.4. In the case of the ovenbird data,
9126 sampling was carried out in multiple years, with a number of sample occasions within
9127 each year (9 or 10), a type of data structure commonly referred to as “the robust design”
9128 (Pollock, 1982). In this context, it stands to reason that there is recruitment and mortality
9129 happening across years. In Chapt. 16 we model these processes explicitly but, here, we
9130 provide an analysis of the data that does not require explicit models for recruitment and
9131 survival, regarding the yearly populations as independent strata, and fitting a multi-session
9132 model.

9133 When the sessions represent explicit time periods, the multi-session model of **secr** can
9134 be thought of as a type of open population model. In particular, a special case of open
9135 models arises when we assume N_t (time-specific population sizes) are independent from
9136 one time period or session to the next – this can be thought of as a “random temporary
9137 emigration” model of the Kendall et al. (1997) variety, and this is the multi-session model
9138 implemented in **secr**. In particular, by assuming that N_t is Poisson with mean Λ_t , one can
9139 model variation in abundance among sessions based on the Poisson-integrated likelihood
9140 in which parameters of Λ_t appear directly in the likelihood as we noted in Sec. 6.5.4.
9141 We provide an analysis (below) of the ovenbird data here using the multi-session models
9142 in **secr**. We formalize the multi-session model approach from a Bayesian perspective
9143 using data augmentation in Chapt. 14 (Converse and Royle, 2012; Royle and Converse,
9144 in review).

9145 A 3rd way to develop models for stratified or grouped populations, not based on
9146 multi-session models, but that is convenient in **BUGS**, is to regard the data from each
9147 session as an independent data set with its own N_t parameter, and do T distinct data
9148 augmentations. Because each N_t is regarded as a free parameter, independent of the
9149 other parameters, we’ll call this the nonparametric multi-session model to distinguish it
9150 from the multi-session model which assumes the N_t are related to one another by having
9151 been generated from a common Poisson distribution. We can analyze this model in the
9152 normal context of data augmentation by augmenting each year separately in the same
9153 **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 ¹

We demonstrate these 3 approaches to analyzing grouped/stratified data using the ovenbird data: (1) In the following section, we provide the nonparametric multi-session model with unconstrained N_t ; (2) we demonstrate the Poisson model-based multi-session models from **secr** both here (following section) and in Chapt. 14 from a Bayesian standpoint; (3) later, in Chapt. 16, we provide a fully dynamic “spatial Jolly-Seber” model and apply it to the ovenbird data.

Analysis in JAGS

The ovenbird data are provided as a multi-session **capthist** object **ovenCH** which, by regarding years as independent strata, or sessions, allows for the fitting of the multi-session model. For doing a Bayesian analysis in one of the **BUGS** engines (we use **JAGS** here) there are a number of ways to structure the data and describe the model. We can analyze either a 2-d data set with all years (data augmented) “stacked” into a data set of dimension $(5 * M) \times 10$ (5 years, M = size of the augmented data set, K = 10 replicate sample occasions). Or, we could produce a 3-d array $(M \times J \times K)$. We adopted the former approach, analyzing the data as a 2-d array and creating an additional categorical variable for “year” to indicate which stratum (year) each record goes with.

Data on individual sex is included with **secr**, but we provide an analysis of a single model for all adults, constant σ 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 σ 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:

```
9189 ovenbird.model.1    fitted secr model -- null
9190 ovenbird.model.1b   fitted secr model -- g0 net shyness
9191 ovenbird.model.1T   fitted secr model -- g0 time trend within years
9192 ovenbird.model.h2   fitted secr model -- g0 finite mixture
9193 ovenbird.model.D    fitted secr model -- trend in density across years
```

¹We do not know of **secr** documentation that states this (or contradicts it). We think this is what is being done, based partially on conversations or emails with M.G. Efford, D.L. Borchers, the various publications on **secr**, and our own thinking about it.

```

model{
  alpha0 ~ dnorm(0,.1)
  sigma ~ dunif(0,200)
  alpha1 <- 1/(2*sigma*sigma)

  A <- ((xlim[2]-xlim[1]))*((ylim[2]-ylim[1]))
  for(t in 1:5){
    N[t] <- inprod(z[1:bigM],yrdummy[,t])
    D[t] <- (N[t]/A)*10000 # Put in units of per ha
    psi[t] ~ dunif(0,1)
  }

  for(i in 1:bigM){ # bigM = total size of jointly augmented data set
    z[i] ~ dbern(psi[year[i]])
    S[i,1] ~ dunif(xlim[1],xlim[2])
    S[i,2] ~ dunif(ylim[1],ylim[2])

    for(j in 1:ntraps){ # X = trap locations, S = activity centers
      d2[i,j] <- pow(pow(S[i,1]-X[j,1],2) + pow(S[i,2]-X[j,2],2),1)
    }
    for(k in 1:K){
      Ycat[i,k] ~ dcat(cp[i,k,])
      for(j in 1:ntraps){
        lp[i,k,j] <- exp(alpha0 - alpha1*d2[i,j])*z[i]*(1-dead[i,k])
        cp[i,k,j] <- lp[i,k,j]/(1+sum(lp[i,k,1:ntraps]))
      }
      cp[i,k,ntraps+1] <- 1-sum(cp[i,k,1:ntraps]) # Last cell = not captured
    }
  }
}

```

Panel 9.2: **BUGS** model specification for the non-parametric multi-session model in which each N_t is independent of the other. The implied prior (by data augmentation) is that $N_t \sim \text{Uniform}(0, 100)$. To fit this model to the ovenbird data, see `?SCRovenbird` in the **R** package `scrbook`.

Table 9.2. Posterior summary statistics for the ovenbird mist-netting data based on the independent multinomial (“multi-catch”) encounter process model. Parameters ψ , N and D are indexed by year. MCMC was done using jags with 3 chains, each with 11000 iterations, discarding the first 1000, for a total of 30000 posterior samples.

Parameter	Mean	SD	2.5%	50%	97.5%	Rhat	n.eff
$D[1]$	0.983	0.211	0.636	0.966	1.455	1.002	1900
$D[2]$	1.023	0.209	0.673	1.003	1.492	1.001	7100
$D[3]$	1.208	0.238	0.807	1.186	1.749	1.004	740
$D[4]$	0.896	0.195	0.575	0.880	1.333	1.002	3000
$D[5]$	0.753	0.177	0.465	0.734	1.149	1.001	4000
α_0	-3.479	0.160	-3.797	-3.477	-3.171	1.005	490
α_1	0.000	0.000	0.000	0.000	0.000	1.003	1100
σ	76.214	6.125	65.569	75.758	89.360	1.003	1100
$N[1]$	80.423	17.283	52.000	79.000	119.000	1.002	1900
$N[2]$	83.685	17.077	55.000	82.000	122.000	1.001	7100
$N[3]$	98.822	19.483	66.000	97.000	143.000	1.004	740
$N[4]$	73.288	15.962	47.000	72.000	109.000	1.002	3000
$N[5]$	61.589	14.468	38.000	60.000	94.000	1.001	4000
$\psi[1]$	0.403	0.092	0.246	0.395	0.606	1.002	1600
$\psi[2]$	0.419	0.091	0.260	0.412	0.620	1.001	6400
$\psi[3]$	0.494	0.102	0.315	0.486	0.723	1.004	760
$\psi[4]$	0.368	0.086	0.221	0.361	0.555	1.002	3200
$\psi[5]$	0.310	0.079	0.178	0.302	0.485	1.002	3500

9194 The model fit objects provided in `secr` are based on the use of the habitat mask.
 9195 To make the analyses consistent with our previous analysis in **JAGS**, we refit all of the
 9196 models here without the habitat mask. The re-analysis proceeds as follows, changing the
 9197 “trend in density across years” model to allow for year-specific density:

```

9198 ## Fit constant-density model
9199 > ovenbird.model.1 <- secr.fit(ovenCH)
9200 ## Fit net avoidance model
9201 > ovenbird.model.1b <- secr.fit(ovenCH, model = list(g0 ~ b))
9202 ## Fit model with time trend in detection
9203 > ovenbird.model.1T <- secr.fit(ovenCH, model = list(g0 ~ T))
9204 ## Fit model with 2-class mixture for g0
9205 > ovenbird.model.h2 <- secr.fit(ovenCH, model = list(g0 ~ h2))
9206 ## Fit a model with session (year)-specific Density
9207 > ovenbird.model.DT <- secr.fit(ovenCH, model = list(D ~ session))

```

9208 All of these can be fitted easily in **JAGS** but the model we fitted previously is roughly
 9209 equivalent to the last model, `ovenbird.model.DT`, because we allowed for year-specific
 9210 population sizes (and hence density). So, we’ll compare our results from **JAGS** to that
 9211 model. The `secr` output is extensive and so we do not reproduce it completely here. By

9212 default, it summarizes the trap information for each year, encounter information, and then
 9213 output for each year. Here is an abbreviated version for `ovenbird.model.DT`:

```

9214 > print(ovenbird.model.DT,digits=2)
9215
9216 secr.fit( capthist = ovenCH, model = list(D ~ session), buffer = 300 )
9217 secr 2.3.1, 14:46:52 23 Jan 2013
9218
9219 $`2005`
9220 Object class      traps
9221 Detector type    multi
9222 Detector number   44
9223 Average spacing   30.27273 m
9224 x-range           -50 49 m
9225 y-range           -285 285 m
9226
9227 [... deleted ...]
9228
9229          2005 2006 2007 2008 2009
9230 Occasions     9   10   10   10   10
9231 Detections    35   42   52   30   33
9232 Animals       20   22   26   19   16
9233 Detectors     44   44   44   44   44
9234
9235 Model          : D~session g0~1 sigma~1
9236 Fixed (real)   : none
9237 Detection fn   : halfnormal
9238 Distribution    : poisson
9239 N parameters    : 7
9240 Log likelihood  : -1119.845
9241 AIC            : 2253.689
9242 AICc           : 2254.868
9243
9244 [... deleted ...]
```

9245 To do model selection we use the handy helper-function `AIC` as follows (output edited
 9246 to fit on the page):

```

9247 AIC (ovenbird.model.1, ovenbird.model.1b, ovenbird.model.1T,
9248          ovenbird.model.h2, ovenbird.model.DT)
9249
9250          model detectfn npar logLik     AIC     AICc     dAICc
9251 ovenbird.model.1T [edited output]  4 -1111.850 2231.700 2232.109  0.000
9252 ovenbird.model.1b        ....      4 -1117.615 2243.229 2243.637 11.528
9253 ovenbird.model.h2        ....      3 -1121.164 2248.327 2248.570 16.461
9254 ovenbird.model.1         ....      5 -1119.762 2249.524 2250.143 18.034
9255 ovenbird.model.DT        ....      7 -1119.845 2253.689 2254.868 22.759
```

9256 We see that our DT model is way down at the bottom of the list. Instead, the model with
 9257 a time-trend (within-season) in detection probability is preferred, followed by a behavioral
 9258 response. We encourage you to adapt the **JAGS** model specification for such models which
 9259 is easily done (see Chapt. 7 for many examples). We provide the summary results for the
 9260 model having $D \sim \text{session}$ as follows:

```

9261 > print(ovenbird.model.DT,digits=2)
9262
9263 secr.fit( capthist = ovenCH, model = list(D ~ session), buffer = 300 )
9264 secr 2.3.1, 14:46:52 23 Jan 2013
9265
9266 [...deleted....]
9267
9268 Fitted (real) parameters evaluated at base levels of covariates
9269
9270   session = 2005
9271     link estimate SE.estimate    lcl    ucl
9272 D      log      0.920      0.228  0.571  1.484
9273 g0     logit     0.028      0.004  0.021  0.037
9274 sigma   log     78.566      6.379 67.025 92.095
9275
9276   session = 2006
9277     link estimate SE.estimate    lcl    ucl
9278 D      log      0.963      0.238  0.598  1.553
9279 g0     logit     0.028      0.004  0.021  0.037
9280 sigma   log     78.566      6.379 67.025 92.095
9281
9282   session = 2007
9283     link estimate SE.estimate    lcl    ucl
9284 D      log      1.139      0.282  0.706  1.836
9285 g0     logit     0.028      0.004  0.021  0.037
9286 sigma   log     78.566      6.379 67.025 92.095
9287
9288   session = 2008
9289     link estimate SE.estimate    lcl    ucl
9290 D      log      0.832      0.206  0.516  1.341
9291 g0     logit     0.028      0.004  0.021  0.037
9292 sigma   log     78.566      6.379 67.025 92.095
9293
9294   session = 2009
9295     link estimate SE.estimate    lcl    ucl
9296 D      log      0.701      0.173  0.435  1.130
9297 g0     logit     0.028      0.004  0.021  0.037
9298 sigma   log     78.566      6.379 67.025 92.095

```

9299 The point estimates (MLEs) of density are uniformly lower than the Bayesian estimates
 9300 (posterior means) shown in Table 9.2. We expect some difference in this direction due

9301 to small-sample skew of the posterior. In addition, there may be slight differences due
 9302 to the fact that **secr** multi-session model assumes that the N_t have a Poisson prior, but
 9303 the implementation in **JAGS** using data augmentation is based on a binomial prior. The
 9304 estimated σ is very similar between the **JAGS** analysis and **secr**.

9.3 SINGLE-CATCH TRAPS

9305 The classical animal trapping experiment is based on a physical trap which captures a
 9306 single animal and holds that individual until subsequent molestation by a biologist. This
 9307 type of observation model – the “single-catch” trap – was the original situation considered
 9308 in the context of spatial capture-recapture by Efford (2004). Nowadays, capture-recapture
 9309 data are more often obtained by other methods (DNA from hair snares, or scat sampling,
 9310 camera traps etc...) but nevertheless the single-catch traps are still widely used in small
 9311 mammal studies (Converse et al., 2006b; Converse and Royle, 2012) and other situations.

9312 The single-catch model is basically a multinomial model but one in which the number
 9313 of available traps is reduced as each individual is captured. As such, the constraints on the
 9314 joint likelihood for the sample of n encounter histories are very complicated. As a result,
 9315 at the time of this writing, there has not been a formal development of either likelihood or
 9316 Bayesian analysis of this model and applications of SCR models to single-catch systems
 9317 have used the independent multinomial model as an approximation (see below).

9318 Nevertheless, we can make some progress to describing the basic observation model
 9319 formally. In particular, if we imagine that all of the individuals captured queued up at
 9320 the beginning of the capture session to draw a number indicating their order of capture,
 9321 then there is a nice conditional structure resulting from a “removal process” operating on
 9322 the traps. The first individual captured has the multinomial observation model:

$$\mathbf{y}_1 \sim \text{Multinomial}(\boldsymbol{\pi}_1)$$

9323 whereas the 2nd individual captured also has a multinomial encounter probability model
 9324 but with the trap which captured the first individual removed. We might express this as:

$$\mathbf{y}_2 \sim \text{Multinomial}(\boldsymbol{\pi}_2)$$

9325 where

$$\pi_{2j} = \frac{(1 - y_{1j}) * \exp(\alpha_0 - \alpha_1 d_{ij}^2)}{\sum_j (1 - y_{1j}) * \exp(\alpha_0 - \alpha_1 d_{ij}^2)}$$

9326 and so on for $i = 3, 4, \dots, n$. In a certain way, this model is a type of local behavioral
 9327 response model but where the response is to other individuals being captured. Evidently,
 9328 the **order of capture** is relevant to the construction of these multinomial cell probabilities.
 9329 More generally, the *time* of capture of an individual in any trapping interval will
 9330 affect the encounter probability of subsequently captured individuals, but we think that
 9331 order of capture might lead to a practical approximation to the single-catch process (this
 9332 is how we simulate the data in our function **simScSCR**). In the simulation of single catch
 9333 data, we randomly ordered the population of individuals for each sample occasion, and
 9334 then cycled through them, turning off each trap if an individual was captured in it.

9335 **9.3.1 Inference for single-catch systems**

9336 For the single-catch model, we argued that the observations have a multinomial type of
 9337 observation model, but the multinomial observations have a unique conditional dependence
 9338 structure among them owing to the “removal” of traps as they fill-up with individuals.
 9339 Thus, competition for single-catch traps renders the independence assumptions for the
 9340 independent multinomial model invalid. However, as Efford et al. (2009a) noted, we
 9341 expect “bias to be small when trap saturation (the proportion of traps occupied) is low.
 9342 Trap saturation will be higher when population density is high...” relative to trap density,
 9343 or when net encounter probability is high. Efford et al. (2009a) did a limited simulation
 9344 study and found essentially no effective bias and concluded that estimators of density
 9345 from the misspecified independent multinomial model are robust to the mild dependence
 9346 induced when trap saturation is low. Naturally then, we expect that the Poisson model
 9347 could also be an effective approximation under the same set of circumstances.

9348 In the **R** package **scrbook** we provide a function for simulating data from a single-catch
 9349 system (function **simScSCR**) and fitting the misspecified model (**example(simScSCR)**) in
 9350 **JAGS** so that you can evaluate the effectiveness of this misspecified model for situations
 9351 that interest you.

9352 **9.3.2 Analysis of Efford's possum trapping data**

9353 We provide an analysis here of data from a study of brushtail possums in New Zealand.
 9354 The data are available with the **R** package **secr** (Efford et al., 2009a); see the help file
 9355 **?possum** after loading the **secr** package. Originally the data were analyzed by Efford et al.
 9356 (2005), and a detailed description of the data set is available in the help file, from which
 9357 we summarize:

9358 *Brushtail possums (*Trichosurus vulpecula*) are an unwanted invasive species in New
 9359 Zealand. Although most abundant in forests, where they occasionally exceed densities
 9360 of 15/ha, possums live wherever there are palatable food plants and shelter.*

9361 To load the possum data, execute the following commands:

9362 > library(secr)
 9363 > data(possum)

9364 The study area encompasses approximately 300 ha, and 180 live traps were organized in 5
 9365 distinct grids, shown in Fig. 9.2. Each square arrangement of traps consisted of 36 traps
 9366 with a spacing of 20 m. Thus the squares are 180 m on a side. Individuals were captured,
 9367 tagged, and released over 5 days during April, 2002. A noteworthy aspect of this study is
 9368 that it involves replicated grids selected in some fashion from within a prescribed region.
 9369 From an analysis standpoint, we could adopt the use of the multi-session models which we
 9370 used previously to analyze the ovenbird data. This would be useful if we had covariates
 9371 at the trapping grid level that we wanted to model. Alternatively, we could pool the data
 9372 from all of the grids and analyze them jointly as if they were based on a single trapping
 9373 grid (with 180 traps) which is clearly a reasonable view in this case. In doing this sort
 9374 of pooling, there is an implicit assumption that N_t (t indexing trapping grid in this case)
 9375 is Poisson distributed, with constant mean (Royle, 2004a; Royle et al., 2012b) which we
 9376 also address in Chapt. 14.

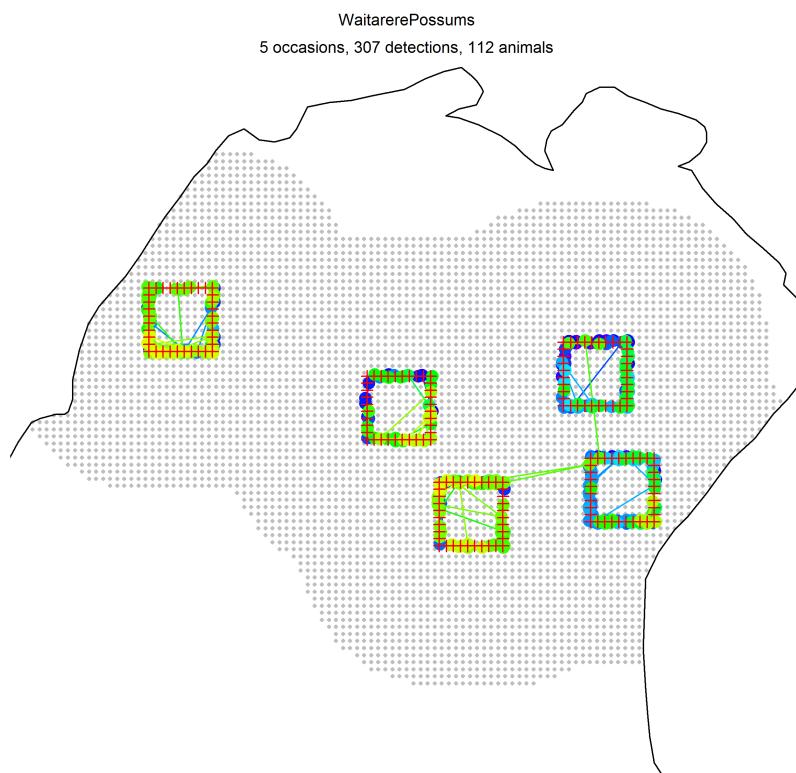


Figure 9.2. Trapping grids used in possum study from Efford et al. (2005), data are contained in the R package `secr` (Efford, 2011a), refer to the help file `?possum` for additional details of this study.

9377 The data file **possumCH** contains 112 encounter histories, and we analyze those here
 9378 although the last 8 of those are recaptures treated as new individuals². The encounter
 9379 process is not strictly a single-catch multinomial process because, as noted in the **possum**
 9380 help file “One female possum was twice captured at two sites on one day, having entered
 9381 a second trap after being released; one record in each pair was selected arbitrarily and
 9382 discarded.” which is a similar situation to what might happen in bird mist net studies, as
 9383 a bird might fly into a net upon release from another. By discarding the two extra-capture
 9384 events, we can satisfactorily view these data as single-catch data, for which **secr** uses the
 9385 independent multinomial likelihood (M. Efford, pers. comm.). If multiple, same-session
 9386 captures were common, then it might be worth developing a model for n_{ik} = the number
 9387 of captures of individual i during sample occasion k , in order to make use of all captures.

9388 For our Bayesian analysis here, we used a rectangular state-space which doesn’t ac-
 9389 count for any geographic boundaries of the survey region, but we note that a habitat mask
 9390 is included in **secr** and it could be used in a Bayesian analysis. Whether or not we use the
 9391 mask is probably immaterial as long as we understand the predictions of N or D over the
 9392 water don’t mean anything biological and we probably wouldn’t report such predictions.
 9393 The **JAGS** model specification is based on that of the ovenbird analysis given previously,
 9394 and so we don’t reproduce the model here. The **R/JAGS** script is called **SCRpossum**,
 9395 which is in the **scrbook** package. The results are summarized in Table 9.3.

Table 9.3. Results of fitting the independent multinomial observation model to the possum trapping data. Strictly speaking, the trapping device is a “single-catch” trap, and the model represents an intentional misspecification. Density is reported in individuals per ha (D_{ha}). Posterior summaries were obtained using **JAGS** with 3 chains, each with 2000 iterations, discarding the first 1000 as burn-in, to yield a total of 3000 posterior samples.

Parameter	Mean	SD	2.5%	50%	97.5%	Rhat	n.eff
N	235.407	17.435	204.000	235.000	270.000	1.009	340
D_{ha}	1.549	0.115	1.343	1.547	1.777	1.009	340
α_0	-0.935	0.167	-1.270	-0.934	-0.605	1.007	870
α_1	0.000	0.000	0.000	0.000	0.000	1.001	2800
σ	52.020	2.675	47.067	51.933	57.585	1.001	2800
ψ	0.783	0.062	0.666	0.782	0.903	1.008	340

9396 The estimated density (posterior mean) is about 1.53 possums/ha. To obtain the **secr**
 9397 results for the equivalent null model, we execute the following command

```
9398 > secr.fit( capthist = possumCH, trace = F )
9399 which produces (edited) summary output:
9400 [... some output deleted ...]
9401
9402 Fitted (real) parameters evaluated at base levels of covariates
9403   link estimate SE.estimate      lcl      ucl
9404 D      log    1.6988930  0.17352645  1.3913904  2.0743547
```

²M. Efford, personal communication

```

9405 g0    logit  0.1968542  0.02256272  0.1563319  0.2448321
9406 sigma  log 51.4689114  2.59981905  46.6204139  56.8216500
9407
9408 [... some output deleted ...]

```

9409 As we've discussed previously, there are many reasons for why there might be differences
9410 between Bayesian and likelihood estimates. But even among likelihood estimates – any
9411 time you run a model there is some numerical integration going on which requires some
9412 specific choices of how to do the integration (see Chapt. 6). For now we just observe that
9413 the estimated density is certainly in the ballpark (compared to those in Table. 9.3), and
9414 so too is the estimated σ .

9.4 ACOUSTIC SAMPLING

9415 The last decade has seen an explosion of technology that benefits the study of animal
9416 populations. This includes DNA sampling methods that allow for identification from
9417 hair or scat, camera trapping and identification software that allow efficient sampling
9418 of many mammals, and the resulting statistical technology that helps us to make sense
9419 of such data (Borchers and Efford, 2008; Royle and Young, 2008; Efford et al., 2009b;
9420 Gopalaswamy et al., 2012b; Sollmann et al., 2013; Chandler and Royle, In press). One
9421 other extremely promising technology area is that of acoustic sampling using microphones
9422 or recording devices. That is, instead of having cameras record encounters, or humans pick
9423 up scat, we can establish an array of (usually) electronic recording devices which, instead of
9424 establishing a visual identity of individuals, record a vocal expression of each individual. In
9425 this context, Efford et al. (2009b) referred to audio recorders as “signal strength proximity
9426 detectors” to distinguish them from other types of proximity detections, including camera
9427 traps, which are *visual* proximity detector. Using audio records, the spatial pattern of the
9428 *signal strength* at the different audio recorders or microphones can be used for inference
9429 about density (Dawson and Efford, 2009; Efford et al., 2009b) in the same way as the
9430 spatial pattern of detections is used in the types of SCR models we have discussed so far.
9431 The basic technical formulation of these models comes from Efford et al. (2009b), and it
9432 was applied to field study of birds by Dawson and Efford (2009). In that study, recording
9433 devices were organized in groups of 4 (in a square pattern), with an array of 5×15 such
9434 clusters of 4, separated by 100 m (300 total recorder locations). This data set, called
9435 **signalCH**, is provided with the **secr** package along with some sample analyses and help
9436 files. See Efford and Dawson (2010), a version of the document **secr-sound.pdf** (that
9437 also comes with the **secr** package) which you can access directly from the main help file
9438 (**?secr**).

9439 Our development here mostly follows Efford et al. (2009b), but we change some nota-
9440 tion to be consistent with our previous material. Let $S(\mathbf{x}, \mathbf{u})$ be the strength of a signal
9441 emanating from signal location \mathbf{u} , as recorded by a device at location \mathbf{x} . Just as ordinary
9442 SCR models represent a model of *encounter frequency* as a function of distance, in acoustic
9443 models, the acoustic SCR model is a model of sound attenuation as a function of distance.
9444 In particular, the acoustic models assumes that S (or a suitable transformation) declines
9445 with distance d from the origin of the sound, to the recording device. In the context of
9446 spatial sampling of animals, the origin is the actual location of some individual animal,

9447 and the recording device is something we nailed to a tree, or mounted on a post. For ex-
 9448 ample, a model of sound attenuation used by Dawson and Efford (2009) is the following:

$$9449 \quad S(\mathbf{x}, \mathbf{u}) = \alpha_0 + \alpha_1 d(\mathbf{x}, \mathbf{u}) + \epsilon \quad (9.4.1)$$

9450 where $\epsilon \sim \text{Normal}(0, \sigma_s^2)$. In many standard situations, S will be measured in decibels,
 9451 which can be any value on the real line. In the conduct of acoustic sampling and the
 9452 development of custom models for your own situation, it would probably be helpful to know
 9453 something about sound dynamics and signal processing. In this model, the parameters
 9454 α_0 , α_1 and σ_s^2 are to be estimated. We abbreviate the set of parameters by $\boldsymbol{\theta}$ for short.

9455 The basic structure of an acoustic SCR study is not really much different from ordinary
 9456 SCR studies. Just as ordinary SCR models require that individuals be encountered at > 1
 9457 trap, these acoustic models require that individuals be heard at > 1 recorder. Therefore,
 9458 the acoustic signals (calls or vocalizations) must be reconcilable and, in fact, reconciled
 9459 successfully by the investigator. In practice, this would require associating signals that
 9460 occur at the same instant with the same individual (or making a decision one way or the
 9461 other). Further, if individuals are actively moving during the sample period (that recorders
 9462 are functioning) then individuals might be double-counted, thereby biasing estimates of
 9463 density. In general, the models produce an estimate of density of sources, and how that is
 9464 interpreted depends on whether individuals are stationary or mobile, and other things. In
 9465 particular, if multiple survey occasions are used (e.g., on different days), then modeling
 9466 movement of individuals would be essential in order to interpret estimates of density
 9467 meaningfully. Models that allow some movement should be possible (see Sec. 9.4.3 below,
 9468 and Chaps. 15 and 16).

9469 9.4.1 The signal strength model

9470 We assert that an individual is detected if S exceeds a threshold, c . The reason for intro-
 9471 ducing this threshold c is that sound recorders will always record some background sound,
 9472 and so effective use of the acoustic SCR models requires specification of the threshold of
 9473 measured signal below which the record is censored (non-detection occurs) because the
 9474 recorded sound is assumed to be background noise. So we assert that an individual is
 9475 detected if $S > c$ which occurs with probability $\Pr(S > c)$, the encounter probability. To
 9476 expand on and formalize this, let S_{ij} be the observed value of S for animal i at detector
 9477 j . The encounter probability is $\Pr(S_{ij} > c)$ which is $\Pr(S_{ij} > c) = 1 - \Pr(S_{ij} < c)$, so
 9478 that, if we standardize the variate we have

$$1 - \Pr\left(\frac{(S_{ij} - \mathbb{E}(S))}{\sigma_s} < \frac{(c - \mathbb{E}(S))}{\sigma_s}\right)$$

9479 This probability calculation requires evaluation of the CDF of a standard normal variate
 9480 say, $\eta = (S_{ij} - \mathbb{E}(S))/\sigma_s$, being less than $\gamma(\boldsymbol{\theta}) = (c - \mathbb{E}(S))/\sigma_s$, which is a function of all
 9481 the parameters α_0 , α_1 , σ_s^2 and also the individual location \mathbf{u} and trap location \mathbf{x} . We'll
 9482 identify it by $\gamma(\boldsymbol{\theta}, \mathbf{x}, \mathbf{u})$ when we need to be explicit about those things. We can compute
 9483 $\Pr(S_{ij} > c) = 1 - \Pr(\eta < \gamma(\boldsymbol{\theta}, \mathbf{x}, \mathbf{u}))$ easily using any software package including **R** which
 9484 has a standard function, **pnorm**, for computing the normal cdf. To be more precise, we'll
 9485 use the **Phi()** to represent the normal cdf. Therefore, an individual is encountered whenever
 9486 $S_{ij} > c$ which happens with probability $\Pr(S_{ij} > c) = 1 - \Phi(\gamma(\boldsymbol{\theta}, \mathbf{x}, \mathbf{u}))$.

Naturally this quantity should depend on *where* an individual is located at the time of recording – what we call it’s instantaneous location, say \mathbf{u} , to distinguish it from it’s home-range center \mathbf{s} (but we outline a model below that contains both \mathbf{u} and \mathbf{s}), and also the trap \mathbf{x} , so we index the quantity γ by those two quantities, in addition to the parameters α_0 , α_1 and σ_s . The probability of detection is therefore

$$p_{ij} = p(\alpha_0, \alpha_1, \sigma | \mathbf{x}_j, \mathbf{u}_i) = 1 - \Phi(\gamma(\cdot))$$

where \mathbf{u}_i is the instantaneous location of individual i and \mathbf{x}_j is the location of trap j . We’ll suppose here that the random variables \mathbf{u}_i have state-space \mathcal{U} ³.

How do we interpret this probability? Well, two things have to happen for an individual to be encountered by a trap: (1) it has to vocalize; (2) the microphone has to record a signal $> c$. These two things together are a product of biological and environmental factors which could include time of day, wind direction and speed, or maybe rain, humidity and other things. The bottom line is a lot of factors are balled up in whether or not the microphone records a sound greater than the threshold.

The observations from an acoustic survey are the signal strength measurements, and the likelihood of the observed signal strength from individual i at detection device j can be specified by noting that the likelihood is the normal pdf for the observed signal if the signal strength is $> c$ and, otherwise, the contribution to the likelihood is $\Phi(\gamma(\cdot))$ (see Eq. 8 of Efford et al. (2009b)):

$$\Pr(S_{ij} | \mathbf{u}_i) = \Phi(\gamma(\cdot))^{1 - I(S_{ij} > c)} \text{Normal}(S_{ij}; \alpha_0, \alpha_1, \sigma_s, \mathbf{x}_j, \mathbf{u}_i)^{I(S_{ij} > c)}$$

We can use this as the basis for constructing the binomial-form of the likelihood as we did in Chapt. 6, which involves the number of individuals not encountered, n_0 . The probability that an individual is *not* captured is equal to the probability that its signal strength doesn’t exceed c at any microphone. The probability of not being captured at a microphone \mathbf{x}_j is:

$$1 - p_{\mathbf{u},j} = \Phi(\gamma(\cdot))$$

and therefore the probability of not being captured at any microphone is:

$$\Pr(\text{all } S_{\mathbf{u},j} < c | \mathbf{u}) = \prod_{j=1}^J (1 - p_{\mathbf{u},j}) = \prod_{j=1}^J \Phi(\gamma(\cdot, \mathbf{x}_j, \mathbf{u}))$$

and therefore the marginal probability of not being captured is

$$\pi_0 = [\text{all } S_{\mathbf{u},j} < c | \boldsymbol{\alpha}] = \int_{\mathcal{U}} \left\{ \prod_{j=1}^J \Phi(\gamma(\boldsymbol{\theta}, \mathbf{x}_j, \mathbf{u})) \right\} d\mathbf{u}$$

which can be used to construct the binomial form of the likelihood as we did in Chapt. 6 (see Eq. 6.2.1).

³We 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

9514 **9.4.2 Implementation in secr**

9515 Fitting acoustic encounter models in **secr** is no more difficult than other SCR models.
 9516 There is a handy manual (**secr-sound.pdf**) with examples (Efford and Dawson, 2010)
 9517 which comes with the **secr** package. The basic process is that **make.capthist** will make a
 9518 **capthist** object from a 3-dimensional encounter array – which is a binary array indicating
 9519 whether each individual was detected or not at each recorder/microphone. In the case
 9520 of signal strength data, **secr** handles the case where # occasions = 1, i.e., the recorders
 9521 obtained data for a single sample occasion, but this is not a general requirement of the
 9522 model for signal strength data (see next section). The “signal” attribute of the **capthist**
 9523 object contains the signal strength in decibels. The best way to include the signal attribute
 9524 is to use **make.capthist** in the usual way, providing it with the encounter data and
 9525 trap data and, in addition, the variable “*c*utval” (which is *c* in our notation above) and
 9526 then provide the signal strength data as an extra column of the **capthist** object. See
 9527 **?make.capthist** for details.

9528 **9.4.3 Implementation in BUGS**

9529 We don’t know of any Bayesian applications of acoustic SCR models, although we imagine
 9530 that implementation of such models in the **BUGS** engines should be achievable. It seems
 9531 easy enough to write down a general hierarchical model that would accommodate sampling
 9532 on repeated occasions. Let \mathbf{s}_i be the home range center, and let \mathbf{u}_{ik} the instantaneous
 9533 location of individual i during sample occasion k (see Chapt. 15 for similar models). The
 9534 model for \mathbf{u}_{ik} can be specified conditional on \mathbf{s}_i . For example, we could assume that \mathbf{u}_{ik}
 9535 are bivariate normal draws with mean \mathbf{s}_i and some variance σ_u^2 . Then, conditional on \mathbf{u}_{ik}
 9536 an individual produces a signal according to the signal attenuation model (Eq. 9.4.1), or
 9537 perhaps some other model. Then we generate the binary encounter data by truncating the
 9538 observed signal at c . This general model then is an example of an SCR model in which
 9539 parameters of a movement model are identifiable (see Sec. 2.6) because there is direct
 9540 information about movement outcomes from the sampling method, unlike other types of
 9541 encounter methods (e.g., camera traps) for which animal locations are restricted to a set of
 9542 fixed, pre-determined points where traps are located. Other types of SCR methods allow
 9543 for movement information too, including some of the search-encounter models (Chapt.
 9544 15).

9545 Instead of developing a Bayesian version of this model here, we leave it to the reader
 9546 to explore simulating data and devising a Bayesian implementation of the acoustic model
 9547 in one of the **BUGS** engines. Note that for a single occasion, you can simulate the data
 9548 using the two stage model (having both \mathbf{s} and \mathbf{u}) or you can simulate \mathbf{u} uniformly without
 9549 dealing with \mathbf{s} in the model. The kernel of the **BUGS** model specification should resemble
 9550 the following snippet:

```
9551 model {
  9552   # Ignoring loops and data augmentation
  9553   u[i,1] ~ dunif(xlim[1], xlim[2])
  9554   u[i,2] ~ dunif(ylim[1], ylim[2])
  9555   mu[i,j] <- alpha0 + alpha1*d[i,j]
  9556   ####
```

```

9557  ##### JAGS has this T() truncation feature
9558  S[i,j] ~ dnorm(mu[i,j], 1/sigma^2)T(c,Inf)
9559  #####
9560  gamma[i,j] <- (c - mu[i,j])/sigma
9561  p[i,j] <- 1 - pnorm(gamma[i,j], 0, 1) # JAGS has pnorm() function
9562  y[i,j] ~ dbern(p[i,j])
9563 }
```

9564 9.4.4 Other types of acoustic data

9565 Efford and Dawson (2010) noted that various other types of acoustic data might arise
 9566 for which SCR-like models would be useful⁴. For example, we could measure the *time of*
 9567 *arrival* of a vocal queue of some sort at multiple recorders to estimate the number and
 9568 origin of N queues. Another example is that where we measure *direction* to a queue from
 9569 multiple devices and do, effectively, a type of statistical triangulation to the multiple but
 9570 unknown number of sources. This has direct relevance to types of double or multiple-
 9571 observer sampling that people do in field studies of birds. Normally 2 observers stand
 9572 in close proximity and record birds, reconciling their detections after data collection.
 9573 An SCR-based formulation of the double-observer method has two observers (or more)
 9574 standing some distance apart, e.g., 50 or 100 meters, and marking individual birds on a
 9575 map (or at least a direction) and a time of detection. The SCR/double-observer method
 9576 could be applied to such data.

9.5 SUMMARY AND OUTLOOK

9577 In this chapter we extended SCR models to accommodate alternative models for the
 9578 observation process, including Poisson and multinomial models. Along with the binomial
 9579 model described in Chapt. 5, this sequence of models will accommodate a substantial
 9580 majority of contemporary spatial capture-recapture problems, including the 4 main types
 9581 of encounter data: binary encounters, multinomial trials from “multi-catch” and “single-
 9582 catch” (Efford, 2004, 2011a; Royle and Gardner, 2011) trap systems, and Poisson encounter
 9583 frequency data from devices that can record multiple encounters of the same individual
 9584 at a device. We summarize the standard observation models and the corresponding **secr**
 9585 terminology in Table 9.4. What we refer to as search-encounter (or area-search) models
 9586 (see Chapt. 15) are distinct from most of the other classes in that the observation location
 9587 can also be random (in contrast to traps, where the location is fixed by design). This
 9588 auxiliary data is informative about an intermediate process related to movement (Royle
 9589 and Young, 2008).

9590 There is a need for other types of encounter models that arise in practice. We identify
 9591 a few of them here, although we neglect a detailed development of them at the present
 9592 time or, in some cases, put that off until later chapters: (1) Removal systems – Sometimes
 9593 traps kill individuals and SCR models can handle that. This can be viewed as a kind of
 9594 open model, with mortality only, and we handle such models (in part) in Chapt. 16; (2)
 9595 There are models for which only specific summary statistics are observable (Chandler and

⁴Some of the following is also related to material presented by D.L. Borchers at the ISEC 2012 conference in Norway.

Table 9.4. Different observation models, where we discuss them in this book, and what the corresponding `secr` terminology is

observation model	Where in this book?	<code>secr</code> name
Bernoulli	Chapt. 5	<code>proximity</code>
Poisson	Sec. 9.1	<code>count</code>
Multinomial (ind)	Sec. 9.2	<code>multi-catch</code>
Multinomial (dep)	Sec. 20.1.5	<code>single-catch</code>
Acoustic	Sec. 9.4	<code>signal</code>
Search-encounter	Chapt. 15	<code>polygon</code> (in part)

9596 Royle, In press; Sollmann et al., 2013) which we cover in Chaps. 18 - 19; (3) We can have
 9597 multiple observation methods working together as in Gopalaswamy et al. (2012b).

9598 There remains much research to be done to formalize models for certain observation
 9599 systems. For example, while we think one will usually be able to analyze single-catch
 9600 systems using the multi-catch model, or even the Bernoulli model if encounter probability
 9601 is sufficiently low, a formalization of the single-catch model would be a useful development
 9602 and, we believe, it should be achievable using one or another of the **BUGS** engines. In
 9603 addition, classical “trapping webs” (Anderson et al., 1983; Wilson and Anderson, 1985a;
 9604 Jett and Nichols, 1987; Parmenter and MacMahon, 1989; Link and Barker, 1994) have
 9605 been around for quite some time and it seems like they are amenable to formulation as
 9606 a type of SCR model although we have not pursued that development simply because
 9607 trapping webs are rarely used in practice.

9608
9609

10

9610

SAMPLING DESIGN

9611 Statistical design is recognized as an important component of animal population studies
9612 (Morrison et al., 2008; Williams et al., 2002). Many biologists have probably been in a
9613 situation where some problem with their data could be traced back to a flaw in study
9614 design. Commonly, design is thought of in terms of number of samples to take, when to
9615 sample, methods of capture, desired sample size (of individuals), power of tests, and related
9616 considerations. In the context of spatial sampling problems, where populations of mobile
9617 animals are sampled by an array of traps or devices, there are a number of critical design
9618 elements. Two of the most important ones are the spacing and configuration of traps
9619 (or sampling devices) within the array. For traditional capture-recapture, conceptual and
9620 heuristic design considerations have been addressed by a number of authors (e.g., Nichols
9621 and Karanth, 2002, Chapt. 11), but little formal analysis focused on spatial design of
9622 arrays has been carried out. Bondrup-Nielsen (1983) investigated the effect of trapping
9623 grid size (relative to animal home range area) on capture-recapture density estimates
9624 using a simulation study and some authors have addressed trap spacing and configuration
9625 by sensitivity “re-analysis” (deleting traps and reanalyzing; Wegge et al., 2004; Tobler
9626 et al., 2008). The scarcity of simulation-based studies looking at study design issues is
9627 surprising, as it seems natural to evaluate prescribed designs by Monte Carlo simulation in
9628 terms of their accuracy and precision. In the past few years, however, a growing number of
9629 simulation studies addressing questions of study design in the context of spatial capture-
9630 recapture have come out (e.g., Marques et al. (2011); Sollmann et al. (2012); Efford and
9631 Fewster (2012); Efford (2011b)), the results of which we will discuss throughout this
9632 chapter.

9633 In this chapter we recommend a general framework for evaluating design choices for
9634 SCR studies using Monte Carlo simulation of specific design scenarios based on trade-offs
9635 between available effort, funding, logistics and other practical considerations – what we
9636 call *scenario analysis*. Many study design related issues can be addressed with preliminary
9637 field studies that will give you an idea of how much data you can expect to collect with a
9638 unit of effort (a camera trap day or a point count survey, for example). But it is also always
9639 useful to perform scenario analysis based on simulation before conducting the actual field

9640 survey not only to evaluate the design in terms of its ability to generate useful estimates,
9641 but also so that you have an expectation of what the data will look like as they are being
9642 collected. This gives you the ability to recognize some pathologies and possibly intervene
9643 to resolve issues before they render a whole study worthless. Suppose you design a study
9644 to place 40 camera traps based on your expectations of parameter values you obtained
9645 from a careful review of the literature, and simulation studies suggest that you should
9646 get 3-5 captures of individuals per night of sampling. In the field you find that you're
9647 realizing 0 or 1 captures per night and therefore you have the ability to sit down and
9648 immediately question your initial assumptions and possibly take some remedial action in
9649 order to salvage your project, your PhD thesis and, hopefully, your career. Simulation
9650 evaluation of design *a priori* is therefore a critical element of any field study.

9651 While we recommend scenario analysis as a general tool to understand your *expected*
9652 *data* before carrying out a spatial capture-recapture study, it is possible to develop some
9653 heuristics and even analytic results related to the broader problem of model-based spatial
9654 design (Müller, 2007) using an explicit objective function based on the inference objective.
9655 We outline an approach in this chapter where we identify a variance criterion, namely, the
9656 variance of an estimator of N for the prescribed state-space. We show that this depends
9657 on the configuration of trap locations, and we provide a framework for optimizing the
9658 variance criterion over the design space (the collection of all possible designs of a given
9659 size). While there is much work to be done on developing this idea, we believe that it
9660 provides a general solution to any type of design problem where the space of candidate
9661 trap locations is well-defined.

10.1 GENERAL CONSIDERATIONS

9662 Many biologists have experience with the design of natural resource surveys from a classical
9663 perspective (Thompson, 2002; Cochran, 2007), a key feature of which involves sampling
9664 space. That is, we identify a sample frame comprised of spatial units and we sample
9665 those units randomly (or by some other method, such as generalized random tessellation
9666 stratified (GRTS) sampling (Stevens Jr and Olsen, 2004)) and measure some attribute.
9667 The resulting inference applies to the attribute of the sample frame. There are some
9668 distinct aspects of the design of SCR studies which many people struggle with in their
9669 attempts to reconcile SCR design with classical survey design problems. We discuss some
9670 of these here.

9671 10.1.1 Model-based not design-based

9672 Inference in classical finite-population sampling is usually justified by “design-based” ar-
9673 guments. This means that properties of estimators (bias, variance) are evaluated over
9674 realizations of the *sample*. The sample is random, but the attribute being observed is not,
9675 for the specific sample that is chosen. For example, imagine we have a landscape gridded
9676 off into 900 1 km \times 1 km grid cells, from which we draw a sample of 100 to measure an
9677 attribute such as “percent developed” which we aim to use in a habitat model. In the
9678 classical design-based view, the attribute (percent developed) is a static quantity for each
9679 of the 900 grid cells and theory tells us that, by taking a random sample, we can expect to
9680 obtain estimators (e.g., of the mean of all 900 grid cells) with good statistical properties,

9681 where the expectation is with respect to the sample of 100 grid cells. For example, if we
9682 repeatedly draw samples of size 100 then, over many such samples, the expected value of
9683 the estimator may be unbiased. Classical design-based sampling does not tell us anything
9684 about the specific 100 sample units that we obtained in our sample. However, in the SCR
9685 modeling framework, properties of our estimators are distinctly model-based. We evaluate
9686 estimators (usually) or care only about a *fixed* sample of spatial locations, averaged over
9687 realizations of the underlying process and data we might generate. Although sometimes
9688 we might condition on the data for purposes of inference (if we have our Bayesian hat on),
9689 the probability model for the data is fundamental to inference, and the spatial sample of
9690 trap locations is always fixed.

9691 **10.1.2 Sampling space or sampling individuals?**

9692 A fundamental question in any sampling problem is what is the sample frame – or the
9693 population we are hoping to extrapolate too. In the context of capture-recapture studies,
9694 it is tempting to think of the sample frame as being spatial (the space within “the study
9695 area”, tiled into quadrats perhaps). Clearly SCR models involve a type of spatial sampling
9696 – we have to identify spatial locations for traps, or arrays of traps. However, unlike
9697 conventional natural resource sampling the attribute we measure is *not* directly relevant
9698 to the *sample location*, such as where we place a trap and, therefore, it may not be
9699 sensible to think of the sample frame as being comprised of spatial units. On the other
9700 hand, capture-recapture studies clearly obtain a sample of *individuals* and SCR models are
9701 models of *individual* encounter and space use. Therefore, it is more natural to think of the
9702 sample frame as a list of N individuals, determined by the definition of the state-space,
9703 or a subset of the state-space, i.e., the study-area, but the number N is unknown. The
9704 purpose of the SCR study is to draw a sample of these N individuals and learn about an
9705 individual attribute – namely, where that individual lives. *That* is the sampling context of
9706 SCR models. SCR models link the observed data (encounter histories) to this individual
9707 attribute via a model (with parameters) which we need to “fit”. Once we fit that model,
9708 we usually use it to make a prediction or estimate of the attribute for individuals that did
9709 not appear in the sample.

9710 Spatial sampling in SCR studies is important, but only as a device for accumulating
9711 individuals in the sample from which we can learn about their inclusion probability. That
9712 is, we’re not interested in any sample unit attribute directly but, rather, we use spatial
9713 units as a means for sampling individuals and obtaining individual level encounter histo-
9714 ries. It makes sense in this context that we should want to choose a set of spatial sample
9715 units that provides an adequate sample size of individuals, perhaps as many as possible.
9716 The key technical consideration as it relates to spatial sampling and SCR is that arbitrary
9717 selection of sample units has a side-effect that it induces unequal probabilities of inclusion
9718 into the sample (i.e., an individual exposed to more traps is more likely to be included
9719 into the sample than an individual exposed to few traps) and so we must also learn about
9720 these unequal probabilities of sample inclusion as we obtain our sample.

9721 The fact that SCR sampling induces unequal probabilities of sampling is consistent
9722 with the classical sampling idea of Horvitz-Thompson estimation which has motivated
9723 capture-recapture models similar to SCR (Huggins, 1989; Alho, 1990). In the Horvitz-
9724 Thompson framework, the sample inclusion probabilities are usually fixed and known.

9725 However, in all real animal sampling problems they are unknown because we never know
9726 precisely where each individual lives and therefore cannot characterize its encounter prob-
9727 ability. Therefore, we have to estimate the sample inclusion probabilities using a model.
9728 SCR models achieve this effect formally, using a fully model-based approach based on a
9729 model that accounts for the organization of individual activity centers and trap locations.
9730 This notion of Horvitz-Thompson estimation suggests that perhaps we should consider
9731 designing SCR studies based on the Horvitz-Thompson variance estimator as a design
9732 criterion. We discuss this a little bit later in this chapter.

9733 **10.1.3 Focal population vs. state-space**

9734 In SCR models we make a distinction between the focal population – the population of
9735 individuals we care about – and those of the state-space, which we are required to prescribe
9736 in order to fit SCR models. These are not the same thing. The geographic scope of the
9737 population of inference is the region within which animals live that you care about in your
9738 study – let’s call this “the study area”. This is often prescribed for political reasons or
9739 legal reasons (e.g. a National Park). To initiate a study, or perhaps motivating the study,
9740 you have to draw a line on a map to delineate a study area, although often it is difficult
9741 to draw this line, and where you draw it is not so much a statistical/SCR issue. On the
9742 other hand, you need to prescribe the state-space to define and fit an SCR model. This
9743 is the region that contains individuals that you *might* capture. This is different from the
9744 study area in most cases. To design a study, you need a well-defined study area, but the
9745 state-space will also be relevant to efficient distribution of traps, and other considerations.

9746 It is helpful to think about this distinction operationally. We define our study area *a priori*.
9747 As a conceptual device, we might think of this as the area that, given an infinite
9748 amount of resources, we might wall off so that we can study a real closed population.
9749 This “study area” should exist independent of any model or estimator of some population
9750 quantity, i.e., the subject-matter context should determine what the study area is. Given
9751 a well-defined study area, we use some method to arrange data collecting devices within
9752 this study area. The method of arrangement can be completely arbitrary but, naturally,
9753 we want to choose arrangements of traps that are better in terms of obtaining statistical
9754 information from the data we wind up collecting.

9755 Lets face it – it’s quite a nuisance that animals move around and this makes the idea of
9756 a spatial study area kind of meaningless in terms of management in most cases. Wherever
9757 you draw a line on a map, there will be animals who live mostly beyond that line that will
9758 sometimes be subjected to observation in your study. One of the benefits of SCR models
9759 is they formalize the exposure and contribution of these individuals to your study. That
9760 is a good thing. Thus, you can probably be a bit sloppy or practical in your definition of
9761 “the study area” and not worry too much.

9762 With these general concepts of spatial sampling and the sampling of individuals in
9763 mind, we can now turn our attention to more specific aspects of study design in SCR
9764 surveys, namely the spatial arrangement of detectors. We discuss some general concepts,
9765 and then focus on a couple of specific case studies that apply to the Bernoulli observation
9766 model or passive detection devices. The general concepts are surely relevant to other SCR
9767 models, and we suspect that the specific case studies are relevant as well.

10.2 STUDY DESIGN FOR (SPATIAL) CAPTURE-RECAPTURE

9768 The importance of adequate trap spacing and overall configuration of the trapping array
9769 has long been discussed in the capture-recapture literature. A heuristic based on recog-
9770 nizing the importance of typical home range sizes (Dice, 1938, 1941) and thus being able
9771 to obtain information about home range size from the trap array is that traps should
9772 be spaced such that the array of available traps exposes as many individuals as possible
9773 but, at the same time, individuals should be captureable in multiple traps. Thus, good
9774 designs should generate a high sample size n (i.e., the number of individuals captured)
9775 and a large number of spatial recaptures. These two considerations form a trade-off in
9776 building designs. On one hand, having a lot of traps very close together should produce
9777 the most spatial recaptures but produce very few unique individuals captured (assuming
9778 that studies are limited in the total number of sampling devices they can deploy). On the
9779 other hand, spreading the traps out as much as possible, in a nearly systematic or regular
9780 design, should yield the most unique individuals, but probably few spatial recaptures. We
9781 will formalize this trade-off later, when we consider formal model-based design of SCR
9782 studies.

9783 Traditional CR models require that all individuals in the study area have a probability
9784 > 0 of being captured, which means that the trap array must not contain “holes” large
9785 enough to contain an animal’s entire home range (Otis et al., 1978). The reason why
9786 “holes” cause a problem in non-spatial models is that they induce heterogeneity in capture
9787 probability. If an animal’s home range lies in or partially in a hole, then it will have a
9788 different probability of being captured than an individual whose home ranges is peppered
9789 with traps. As a consequence, trap spacing is recommended to be on the same order
9790 as the radius of a typical home range (e.g., Dillon and Kelly (2007)). For example,
9791 imagine a camera trap study implemented in South America with the objective to survey
9792 populations of both jaguars (*Panthera onca*) and the much smaller ocelots (*Leopardus*
9793 *pardalis*). Ocelots also have much smaller home ranges and therefore should require closer
9794 trap spacing than the large wide-ranging jaguars. The “no holes” assumption entails
9795 some strong restrictions with respect to study design. Although we need not cover an
9796 area systematically with traps, there has to be some consistent coverage of the entire
9797 area of interest. Often, this is achieved by dividing the study area into grid cells, the
9798 size of which approximates an average home range (or possibly the smallest home range
9799 recorded for the study species in the study area or a similar area; e.g. Wallace et al.
9800 (2003)), and then place (at least) one trap within each cell. In many field situations,
9801 especially when dealing with large mammals and accordingly large study areas, achieving
9802 this consistent coverage can be extremely challenging or even impossible. Depending on
9803 local environmental conditions, parts of the study area can be virtually inaccessible to
9804 humans, because of dense vegetation cover, or unsuitable for setting up detectors, because
9805 of flooding. Even when accessible, setting up traps in difficult habitat conditions can
9806 consume disproportional amounts of time, manpower and other resources. Moreover, even
9807 when the trap spacing does not result in holes, the problem of spatial heterogeneity in
9808 capture probability will still exist because individuals with home ranges near the borders
9809 of the trap array will have a different probability of being captured than individuals that
9810 spend all their time within the trap array.

9811 Where approaches such as MMDM (mean maximum distance moved) are used in
9812 combination with traditional CR models to obtain density estimates (see Chapt. 4), trap

spacing also has a major effect on movement estimates, since it determines the resolution of the information on individual movement (Parmenter et al., 2003; Wilson and Anderson, 1985a). If trap spacing is too wide, there is little or no information on animal movement because most animals will only be captured at one trap (Dillon and Kelly, 2007). In addition, only a trapping grid that is large relative to individual movement can capture the full extent of such movements, and researchers have suggested that the grid size should be at least four times that of individual home ranges to avoid positive bias in estimates of density (Bondrup-Nielsen, 1983). This recommendation originated in small mammal trapping, and it should be relatively easy to follow when dealing with species covering home ranges < 1 ha. However, translated to large mammal research, this can entail having to cover several thousands of square kilometers – a logistical and financial challenge that few projects could realistically tackle.

Though closely related, the requirements in terms of spatial study design for SCR models differ distinctly from those for traditional CR. For one, holes in the study area are of no concern in SCR studies. As a practical matter, some animals within the study area might have vanishingly small probability of being included in the sample, i.e., $p \approx 0$. The nice thing about SCR models is that N is explicitly tied to the state-space, and not the traps which expose animals to encounter. Within an SCR model, extending inference from the sample to individuals that live in these holes represents an extrapolation (prediction of the model outside the range of the data), but one that the model is capable of producing because we have explicit declarations, in the model, that it applies to any area within the state-space (the state-space is a part of the model!), even to areas where we can't capture individuals because we happened to not put a trap near them. Conversely, ordinary capture-recapture models only apply to individuals that have encounter probability that is consistent with the model being considered. Presumably, the existence of a hole in the trap array would introduce individuals with $p = 0$, which is not accommodated in those models. This alone allows for completely new and much more flexible study designs in SCR studies, as compared to traditional CR, such as linear designs, “hollow grids” (detectors trace the outline of a square), or small clusters of grid spread out over larger landscapes (Efford et al., 2005, 2009a; Efford and Fewster, 2012).

Whereas traditional CR studies are concerned with the number of individuals and recaptures and with satisfying the model assumption of all individuals having some probability of being captured, in spatial capture-recapture we are looking at an additional level of information: We need spatially dispersed captures and recaptures. It is not enough to recapture an individual – we need to recapture at least some individuals at several traps. Therefore, in general, design of SCR studies boils down to obtaining three bits of information: total unique individuals captured, total number of recaptures informative about baseline encounter rate, and spatial recaptures, informative about σ .

Most SCR design choices wind up trading these three things against each other to achieve some optimal (or good) mix. So, for example, if we sample a very small number of sites a huge number of times then we can get a lot of recaptures but only very few spatial ones, and few unique individuals etc. This need for spatial recaptures may appear as an additional constraint on study design, but actually, SCR studies are much less restricted than traditional CR studies, because of the way animal movement is incorporated into the model: σ is estimated as a specified function of the ancillary spatial information collected in the survey and the capture frequencies at those locations. This function is able to

make a prediction across distances even when these are latent, including distances larger than the extent of the trap array. When there is enough data across at least some range of distances, the model will do well at making predictions at unobserved distances. The key here is that there needs to be ‘enough data across some range of distances’, which induces some constraint on how large our overall trap array must be to provide this range of distances (e.g., Marques et al., 2011; Efford, 2011b). We will review the flexibility of SCR models in terms of trap spacing and trapping grid size in the following section.

10.3 TRAP SPACING AND ARRAY SIZE RELATIVE TO ANIMAL MOVEMENT

Using a simulation study, Sollmann et al. (2012) investigated how trap spacing and array size relative to animal movement influence SCR parameter estimates and we will summarize their study here. They simulated encounter histories on an 8×8 trap array with regular spacing of 2 units, using a binomial observation model with Gaussian hazard encounter model, across a range of values for the scale parameter σ^* . We refer to the scale parameter as σ^* here, because Sollmann et al. (2012) use a slightly different parametrization of SCR models, in which σ^* corresponds to $\sigma \times \sqrt{2}$.

In Sec. 5.4 we pointed out that under the Gaussian (or half-normal) detection model σ can be converted into an estimate of the 95% home range or “use area” around s_i . Based on this transformation, values for σ^* were chosen so that there was a scenario where the trap array was smaller than a single individual’s home range, i.e. trap spacing was small relative to individual movements ($\sigma^* = 5$), a scenario where spaces between traps were large enough to contain entire home ranges ($\sigma^* = 0.5$), and two intermediate scenarios and where sigma was smaller ($\sigma^* = 1$ unit) and larger ($\sigma^* = 2.5$ units) than the trap spacing, respectively. N was 100, the baseline trap encounter rate λ_0 was 0.5 (on the cloglog scale) for all four scenarios and trap encounters were generated over 4 occasions. Table 10.1 shows the results as the average over 100 simulations.

All model parameters were estimated with relatively low bias (< 10%) and high to moderate precision (relative root mean squared error, RRMSE < 25%) for all scenarios of σ^* , except $\sigma^* = 0.5$ units, under which model parameters were mostly not estimable (therefore excluded from Table 10.1). Data for the latter case mostly differed from the other scenarios in that fewer animals were captured and very few of the captured animals were recorded at more than 1 trap (Table 10.2). For $\sigma^* = 0.5$, abundance (N) was not estimable in 88% of the simulations, and when estimable, was underestimated by approximately 50%. This shows that a wide trap spacing that is considerably too large relative to animal movement may be problematic in SCR studies.

Estimates (posterior means) of N were least biased and most precise under the $\sigma^* = 2.5$ scenario, and in general, all parameters were estimated best under the $\sigma^* = 2.5$ or the $\sigma^* = 5$ scenario. All estimates had the highest relative bias and the lowest precision under the $\sigma^* = 1$ scenario. These results clearly demonstrate that SCR models can successfully handle a range of trap spacing to animal movement ratios, and even when using a trapping array smaller than an average home range: at $\sigma^* = 5$, the home range of an individual was approximately 235 units², while the trapping grid only covered 196 units². Still, the model performed very well.

An important consideration in this simulation study is that all but the $\sigma^* = 0.5$ units

Table 10.1. Mean, relative root mean squared error (RRMSE) of the mean, mode, 2.5% and 97.5% quantiles, relative bias of mean (RB) and 95% Bayesian credible interval (BCI) coverage for spatial capture-recapture parameters across 100 simulations for four simulation scenarios, define by the input value of movement parameter σ^* . N = number of individuals in the state space; λ_0 = baseline trap encounter rate.

Scenario	Mean	rrmse	Mode	2.5%	97.5%	RB	BCI
$\sigma^* = 1 (\sigma = 0.71)$							
N	108.497	0.172	104.099	78.977	143.406	0.085	96
λ_0	0.518	0.248	0.477	0.303	0.752	0.035	94
σ^*	1.008	0.093	0.990	0.857	1.195	0.008	94
$\sigma^* = 2.5 (\sigma = 1.77)$							
N	100.267	0.105	98.456	82.086	121.878	0.003	97
λ_0	0.507	0.118	0.500	0.409	0.623	0.014	92
σ^*	2.501	0.046	2.491	2.267	2.690	< 0.001	92
$\sigma^* = 5 (\sigma = 3.54)$							
N	102.859	0.137	100.756	77.399	130.020	0.029	88
λ_0	0.505	0.075	0.501	0.435	0.580	0.011	93
σ^*	5.023	0.039	5.001	4.687	5.431	0.005	97

Table 10.2. Summary statistics of 100 simulated data sets for four simulation scenarios, defined by the input value of movement parameter σ^* . Individual detection histories were simulated on an 8×8 trap array with regular trap spacing of 2 units.

Scenario	Inds. captured	Total captures	Inds. recaptured	Inds. captured at > 1 trap
$\sigma^* = 0.5$	18.29 (3.84)	25.38 (5.86)	5.52 (2.03)	0.72 (0.95)
$\sigma^* = 1.0$	37.70 (13.44)	69.35 (26.05)	19.48 (7.68)	11.87 (5.43)
$\sigma^* = 2.5$	44.19 (4.67)	231.78 (33.98)	36.60 (4.76)	35.21 (4.73)
$\sigma^* = 5.0$	40.51 (5.15)	427.77 (79.09)	33.09 (4.63)	32.60 (4.76)

scenarios provided reasonably large amounts of data, including 20+ individuals being captured on the trapping grid. When dealing with real-life animals that are often territorial and may have lower trap encounter rates, a very small grid compared to an individual's home range may result in the capture of few to no individuals. In that case, the sparse data will limit the ability of the model to estimate parameters (Marques et al., 2011), which is true of most models.

To further explore the effects of trap spacing and movement on bias and precision of estimates of N , we expanded the simulation study of Sollmann et al. (2012): we considered a regular 7×7 grid, with trap spacing ranging from $0.5 \times \sigma$ to $4 \times \sigma$, with a state-space that had variable size so that the buffer around the traps was constant in units of σ . For each trap spacing scenario we simulated and analyzed 500 data sets and calculated the RRMSE and relative bias for the estimates of N . Figure 10.1 shows the results of this set of simulations. We see that there is clearly an optimal trap spacing, especially in terms of precision, which is highest at a trap spacing of $1.5 - 2.5 \times \sigma$. Efford (2012) reported similar results and highlighted the trade-off between the number of individuals captured and the number of spatial recaptures – intuitively, the former goes up with an increase in

trap spacing, whereas the latter goes down. In summary, in small trap spacing scenarios, the small sample size leads to imprecise estimates, whereas in large trap spacing scenarios, lack of spatial recaptures leads to imprecise and biased estimates.

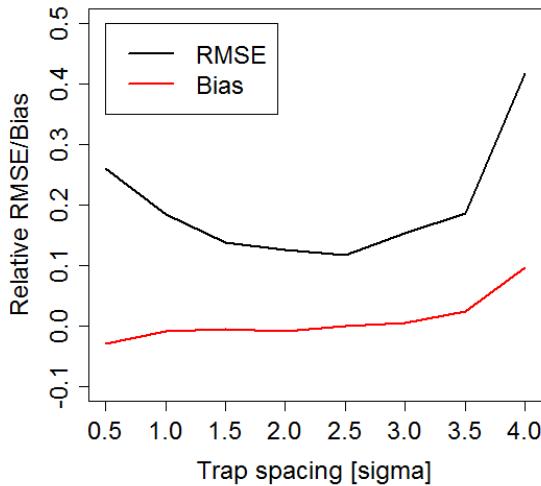


Figure 10.1. Relative bias and RRMSE of estimates of N from an SCR model for a range of trap spacing scenarios.

10.3.1 Black bears from Pictured Rocks National Lakeshore

To see how trap array size influences parameter estimates from spatial capture-recapture models in the real world, Sollmann et al. (2012) also looked at a black bear data set from Pictured Rocks National Lakeshore, Michigan, collected using 123 hair snares distributed over an area of 440 km^2 along the shore of Lake Superior in May-July 2005 (Belant et al., 2005). The SCR model for the bear data included sex-specific encounter rate parameters, and an occasion-specific baseline encounter rate. This was motivated by a) the lower average number of detections for male bears, b) the decreasing number of detections over time in the raw data, and c) the fact that male black bears are known to move over larger areas than females (e.g., Gardner et al., 2010b; Koehler and Pierce, 2003).

To address the impact of a smaller trap array on the parameter estimates, models fitted to the full data set were compared to models fitted to data subsets. The first subset retained only those 50% of the traps closest to the grid center. In the second, only the southern 20% of the traps were retained 10.3.

Reducing the area of the trap array by 50% created a grid polygon of 144 km^2 , which was smaller than an estimated male black bear home range and only 50% larger than a

Table 10.3. Posterior summaries of SCR model parameters for black bears, modified from Sollmann et al. (2012).

	Mean (SE)	Mode	2.5%	97.5%
Full data set				
D	10.556 (1.076)	10.448	8.594	12.792
σ^* (males)	7.451 (0.496)	7.323	6.579	8.495
σ^* (females)	2.935 (0.143)	2.939	2.671	3.226
50% of traps				
D	12.648 (1.838)	12.205	9.307	16.713
σ^* (males)	5.354 (0.511)	5.248	4.472	6.473
σ^* (females)	3.318 (0.277)	3.262	2.841	3.910
20% of traps				
D	6.752 (1.611)	5.953	4.000	10.218
σ^* (males)	9.881 (3.572)	7.566	5.121	18.447
σ^* (females)	2.686 (0.391)	2.657	2.121	3.404

9936 female black bear home range – approximately 260 km² and 100 km², respectively, when
 9937 converting estimates of σ^* to home range size. Table 10.3 shows that this did not greatly
 9938 influence model results, compared to the full data set.

9939 Removing 80% of the traps and thereby reducing the area of the trap array to 64
 9940 km² – well below the average black bear home range – had a great effect on sample size
 9941 (only 25 of the original 83 individuals sampled) and parameter estimates. Particularly,
 9942 male black bear movement was overestimated and imprecise. The combination of the low
 9943 baseline trap encounter rate of males and the considerable reduction in sample size led to
 9944 a low level of information on male movement: 5 of the 12 males were captured at one trap
 9945 only. Although they moved over smaller areas, owing to their higher trap encounter rate,
 9946 females were, on average, captured at more traps (3.4 traps per individual compared to 2.6
 9947 for males) so that their movement estimate remained relatively accurate. Overestimated
 9948 male movements and female trap encounter rates resulted in an underestimate of density
 9949 of almost 40%. This effect is contrary to what we would expect to see in non-spatial
 9950 CR models, where a trapping grid that is small relative to animal movement leads to
 9951 underestimated movement (MMDM) and overestimated density (Bondrup-Nielsen, 1983;
 9952 Dillon and Kelly, 2007; Maffei and Noss, 2008). While this example again demonstrates
 9953 the ability of SCR models to deal with a range of trapping grid sizes, it also clearly shows
 9954 that your study design needs to consider the amount of data you can expect to collect.
 9955 As an alternative to simulation studies, Efford et al. (2009b) provide a mathematical
 9956 procedure to determine the expected number of individuals captured and recaptures for a
 9957 given detector array and set of model parameters.

10.4 SAMPLING OVER LARGE AREAS

9958 Trap spacing is an essential aspect of design of SCR studies. However, it is only the most
 9959 important aspect if one can uniformly cover a study area with traps. In many practical
 9960 situations, where the study area is large relative to effort that can be expended, one has
 9961 to consider other strategies which deviate from a strict focus on trap spacing. There are
 9962 two general strategies that have been suggested for sampling large areas which we think

9963 are useful in practice, either by themselves or combined: Sampling based on *clusters* of
9964 traps and sampling based on *rotating* groups of traps over the landscape.

9965 Karanth and Nichols (2002) describe 3 strategies for moving traps to achieve coverage
9966 of a larger study area, geared toward traditional capture-recapture analysis. Suppose that
9967 sampling the entire area of interest requires sampling G sites, then the 3 strategies are:

- 9968 (1) For every day/sampling occasion, randomly choose x out of your G sites, where x is
9969 the number of trapping devices you have at hand. Obviously, this requires that it be
9970 relatively easy to move traps around.
- 9971 (2) Move blocks of traps that are close to each other in space daily. For example, if you
9972 divide your total study area into 4 blocks, sample block 1 for a day, then move traps
9973 to block 2 for a day, and so forth, and repeat until each block has been sampled for a
9974 sufficient amount of time.
- 9975 (3) If moving blocks of traps daily is too challenging logistically, then you can sample
9976 each block for a certain number of days/occasions before moving cameras to the next
9977 block. In this fashion, you only need to move traps to each block once.

9978 In traditional CR we collapse data across traps and assume all individuals in the study
9979 area have some probability > 0 of being detected. For our data that means that, under
9980 scenario (2) the first occasion is defined as the time it takes to sample all 4 blocks once, the
9981 second occasion consists of the second round of sampling all blocks, etc. Under scenario
9982 (3), we have to combine data from day 1 in each of the blocks to form occasion 1, data
9983 from day 2 in each of the blocks forms occasion 2, and so on. Especially scenario 3 makes
9984 modeling time-dependent detection difficult, since occasion 1 does no longer refer to an
9985 actual day or continuous time interval. We do not have that problem in SCR, where
9986 accounting for sampling effort at each trap is straight forward, as we first demonstrated
9987 for the wolverine example in Sec. 5.9. Because we are dealing with detection at the trap
9988 level, even for design (3) in a spatial framework, we can still look at variation in detection
9989 over time. As such, we don't think that one of the above designs is superior for SCR
9990 models than the other, but rather, all of them may produce adequate SCR data, as long
9991 as overall sample size requirements are met.

9992 Efford and Fewster (2012) looked at the performance of different spatial study designs
9993 for abundance estimation from traditional and spatial capture-recapture models, including
9994 a clustered design, where groups of detectors are spaced throughout the larger region of
9995 interest. They found that in a spatial framework this design performed well, although
9996 there were indications of a slight positive bias in estimates of N . Such a clustered design
9997 enables researchers to increase area coverage without having to increase the number of
9998 traps. Efford and Fewster (2012) note that distribution of clusters has to be spatially
9999 representative – for example, systematic with a random origin. The issue of spatially
10000 representative designs is not limited to SCR and an extensive treatment of the topic can
10001 be found in the distance sampling literature (Buckland et al., 2001). Further, the authors
10002 stress that, if distances among clusters are large and individuals are unlikely to show up
10003 in several clusters, then the method relies on spatial recaptures *within* clusters, meaning
10004 that spacing of detectors within clusters has to be appropriate to the movements of the
10005 species under study. A clustered type of design is also suggested by Efford et al. (2009b)
10006 for acoustic detectors (see Chapt. 9.4) with small groups of such detectors (e.g., 2×2)
10007 being distributed in a probabilistic fashion across the region of interest.

10008 In practice, employing both of these strategies – clustering and rotating traps – might
 10009 be necessary or advantageous. Sun (in prep) used a simulation study to investigate differ-
 10010 ent trap arrangements (Fig. 10.2) for a black bear study based on hair snares distributed
 10011 over a 2625-km² study area. She simulated detection data of bears for 3 trap arrangements
 10012 including a regular (uniform) coverage of traps, clusters of 4 traps each with a gap between
 10013 clusters, and a design in which the clusters were moved mid-way through the study to fill
 10014 the gap (a sequential or “rotating” design). She found that the precision and accuracy of
 10015 estimates of N generally decreased when changing from a uniform to a clustered to a ro-
 10016 tating design, although the loss of efficiency was relatively small when using the clustered
 10017 design. The result seems to support that cluster designs can be effective with relatively
 10018 little loss of efficiency.

10019 Further research on optimal detector configurations, especially for large scale studies, is
 10020 called for (Efford and Fewster, 2012). More generally, work on formalizing and generalizing
 10021 these ideas of spatial study design is needed. We believe the model-based spatial design
 10022 approach that we introduce below is one possible way to do that.

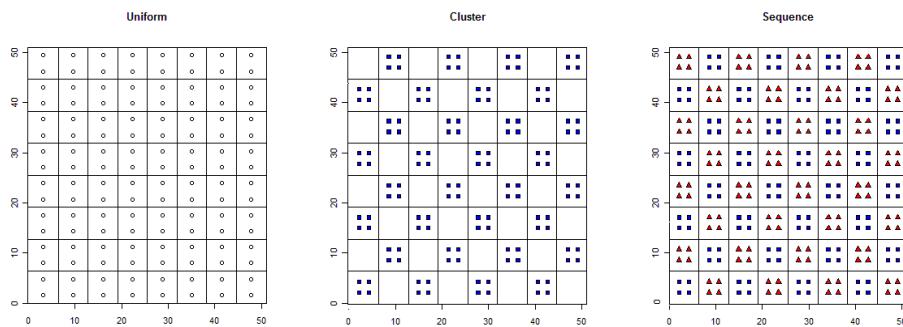


Figure 10.2. Three designs evaluated by Sun (in prep). The left panel shows uniform coverage of the area with traps (hair snares) equally spaced and static for the duration of the period. The central panel shows clusters of 4 traps in close proximity, with larger gaps between clusters. The right panel shows a design in which all grid cells are sampled by the cluster of 4 traps, but in a sequential (in time) manner.

10.5 MODEL-BASED SPATIAL DESIGN

10023 A point we have stressed in previous chapters is that SCR models are basically glorified
 10024 versions of generalized linear models (GLMs) with a random effect that represents a latent
 10025 spatial attribute of individuals, the activity center or home range center. This formulation
 10026 makes analysis of the models readily accessible in freely available software and also allows
 10027 us to adapt and use concepts from this broad class of models to solve problems in spatial
 10028 capture recapture. In particular, we can exploit well-established model-based design con-
 10029 cepts (Kiefer, 1959; Box and Draper, 1959, 1987; Fedorov, 1972; Sacks et al., 1989; Hardin

and Sloane, 1993; Fedorov and Hackl, 1997) to develop a framework for designing spatial trapping arrays for capture-recapture studies. Müller (2007) provides a recent monograph level treatment of the subject that is very accessible.

In the following sections, we adapt these classical methods for constructing optimal designs to obtain the configuration of traps (or sampling devices) in some region (the design space, \mathcal{X}), that minimizes some appropriate objective function based on the variance of model parameters, α , or N , for a prescribed state-space. We show that this criterion – based on the variance of an estimator of N – represents a formal compromise between minimizing the variance of the MLEs of the detection model parameters and obtaining a high expected probability of capture. Intuitively, if our only objective was to minimize the variance of parameter estimates than all of our traps should be in one or a small number of clusters where we can recapture a small number of individuals many times each. Conversely, if our objective was only to maximize the expected probability of encounter then the array should be highly uniform so as to maximize the number of individuals being exposed to capture.

10.5.1 Statement of the design problem

Let \mathcal{X} , the *design space*, denote some region within which sampling could occur and let $\mathbf{X} = \mathbf{x}_1, \dots, \mathbf{x}_J$ denote the *design*, the set of sample locations (e.g., of camera traps), normally we just call these “traps.” The design space \mathcal{X} must be prescribed (a priori). Operationally, we could equate \mathcal{X} to the study area itself (which is of management interest) but, in practical cases, there will generally be parts of the study area that we cannot sample. Those areas need to be excluded from \mathcal{X} . While \mathcal{X} may be continuous, in practice it will be sufficient to represent \mathcal{X} by a discrete collection of points which is what we do here. This is especially convenient when the geometry of \mathcal{X} is complicated and irregular, which would be the case in most practical applications. The technical problem addressed subsequently is how do we choose the locations \mathbf{X} in a manner that produces the “optimal” (lowest variance) for estimating population size or density, or some other quantity of interest.

As usual, we regard the population of N individual “activity centers” as the outcome of a point process distributed uniformly over the state-space \mathcal{S} . The relevance and importance of \mathcal{S} has been established repeatedly in this book, as it defines a population of individuals (i.e., activity centers) and, in practice, it is not usually the same as \mathcal{X} due to the fact that animals move freely over the landscape and the location of traps is typically restricted by policies, ownership, logistics and other considerations. The objective we pursue here is: Given (1) \mathcal{X} , (2) a number of design points, J ; (3) the state-space \mathcal{S} , (4) an SCR model, and (5) a design criterion $Q(\mathbf{X})$, we want to choose *which* J design points we should select in order to obtain the *optimal* design under the chosen model, where the optimality is with respect to $Q(\mathbf{X})$.

What types of functions make reasonable objective functions, $Q(\mathbf{X})$? We will describe some possible choices for $Q(\mathbf{X})$ below, but it makes sense that they should relate to the variance of estimators of one or more parameters of the SCR model.

We motivate the basic ideas of model-based design with a simple model that proves to be an effective caricature of the SCR model that we'll use shortly. Suppose \mathbf{s} is the activity center of a single individual, and \mathbf{s} is known with certainty. Then, for an array

10074 of traps \mathbf{X} we measure a response variable, lets say the strength of an acoustic signal,
 10075 that has a normal distribution. So we have this response variable that has a normal linear
 10076 model of the form:

$$\mathbf{y} = \mathbf{M}(\mathbf{X}, \mathbf{s})'\boldsymbol{\alpha} + \text{error}$$

10077 In our notation here, $\mathbf{M}(\mathbf{X}, \mathbf{s})$ is some design matrix where, in the context of SCR models,
 10078 it has 2 columns (for the basic model): A column of 1's, and then a column of distance
 10079 from each trap \mathbf{x}_j to the activity center \mathbf{s} . The design matrix is therefore, for a single
 10080 individual, a matrix of dimension $J \times 2$.

10081 The inference objective here is to estimate the parameters $\boldsymbol{\alpha}$. The variance-covariance
 10082 matrix of $\hat{\boldsymbol{\alpha}}$ is, suppressing the dependence on \mathbf{X} for notational convenience,

$$\text{Var}(\boldsymbol{\alpha}, \mathbf{X}) = (\mathbf{M}(\mathbf{s})'\mathbf{M}(\mathbf{s}))^{-1}$$

10083 Note that the design points \mathbf{x}_j appear explicitly (in the 2nd column of \mathbf{M}). In considering
 10084 design for estimation in such models it is natural to choose design points, corresponding to
 10085 values of \mathbf{x} , such that the variance of $\hat{\boldsymbol{\alpha}}$ is minimized. Of course, $\boldsymbol{\alpha}$ is a vector, and so the
 10086 “variance” is a matrix (at least 2×2) so we have to work with suitable scalar summaries
 10087 of that matrix, such as the trace (sum of the diagonals) or a function of the determinant,
 10088 etc..

10089 For a population of N individuals, if we know *all* N values of \mathbf{s} , the design matrix
 10090 \mathbf{M} has the same basic structure but with N versions stacked-up on top of one another,
 10091 producing a larger $N * J \times 2$ design matrix. The 2nd column of that matrix contains the
 10092 information about trap locations (the 1st column is still just a column of 1s). Therefore,
 10093 we could easily find the design \mathbf{X} that optimizes some function of the variance-covariance
 10094 matrix of the model parameters.

10095 All of this is fine and good if we happen to know the activity centers for each individual.
 10096 However, this is not a realistic formulation. When \mathbf{s} is unknown, it might make sense to
 10097 consider minimizing the expected (spatially averaged) variance:

$$E_{\mathbf{s}} \{ \text{Var}(\boldsymbol{\alpha}, \mathbf{X}) \} = \sum_{\mathbf{s} \in \mathcal{S}} (\mathbf{M}'(\mathbf{s})\mathbf{M}(\mathbf{s}))^{-1}.$$

10098 However, this is not the expected variance based on sampling a population of N indi-
 10099 viduals, just for a single individual having unknown \mathbf{s} . Because of the matrix inverse in
 10100 this expression, it is not sufficient to use a variance criterion that weighs this variance
 10101 by N . As an alternative, we can maximize the expected *information*, the inverse of the
 10102 variance-covariance matrix, which is probably more appealing from an analytic point of
 10103 view. The information matrix for the data based on a single individual, with known \mathbf{s} , is:
 10104 $\mathcal{I}(\boldsymbol{\alpha}, \mathbf{X}) = (\mathbf{M}'(\mathbf{s})\mathbf{M}(\mathbf{s}))$. For a population of N individuals, let \mathbf{M}_i be the design matrix
 10105 for the individual with activity center \mathbf{s}_i . Then, the total information for all N individuals
 10106 is:

$$\mathcal{I}(\boldsymbol{\alpha}, \mathbf{X}) = \sum_{i=1}^N (\mathbf{M}'_i(\mathbf{s}_i)\mathbf{M}_i(\mathbf{s}_i))$$

10107 The information matrix depends on the design \mathbf{X} through the N individual matrices
 10108 $\mathbf{M}_1, \dots, \mathbf{M}_N$. Now, because we don't know \mathbf{s}_i we can compute the integrated information,

10109 over all possible values of \mathbf{s}_i , and for each \mathbf{s}_i , which is an N -fold summation:

$$E_{\mathbf{s}_1, \dots, \mathbf{s}_N} \mathcal{I}(\boldsymbol{\alpha}, \mathbf{X}) = \sum_{i=1}^N \sum_{s \in \mathcal{S}} (\mathbf{M}'_i(\mathbf{s}_i) \mathbf{M}_i(\mathbf{s}_i))$$

10110 which is just N copies of the integrated (spatially averaged) information:

$$E_{\mathbf{s}_1, \dots, \mathbf{s}_N} \mathcal{I}(\boldsymbol{\alpha}, \mathbf{X}) = N \sum_{s \in \mathcal{S}} (\mathbf{M}'_i(\mathbf{s}_i) \mathbf{M}_i(\mathbf{s}_i)).$$

10111 It therefore seems sensible to base design of SCR studies on some criterion that is
 10112 a function of this expected information matrix. E.g., find the design that maximizes
 10113 the diagonals, or the determinant, or minimizes the trace of the *inverse* (the variance-
 10114 covariance matrix based on N individuals). This can be done for any number of design
 10115 points $\mathbf{x}_1, \dots, \mathbf{x}_J$ using standard exchange algorithms (see Müller, 2007, Chapt. 3) and
 10116 we discuss this below in Sec. 10.5.5. However, our SCR models are not normal linear
 10117 models but, rather, more like Poisson or binomial GLMs. We see in the next section that
 10118 we can adapt these ideas for such models.

10119 10.5.2 Model-based Design for SCR

10120 Following our development of the normal linear model above, suppose for the moment
 10121 that we know \mathbf{s} for a single individual. In this case, its vector of counts of encounter in
 10122 each trap \mathbf{y} are either binomial or Poisson counts, and the linear predictor has this form:

$$g(\mathbb{E}(\mathbf{y})) = \alpha_0 + \alpha_1 \|\mathbf{x} - \mathbf{s}\|^2. \quad (10.5.1)$$

10123 for the Gaussian encounter probability model, or any other model could be used. In vector
 10124 form, we write this as:

$$g(\mathbb{E}(\mathbf{y})) = \mathbf{M}' \boldsymbol{\alpha}$$

10125 where \mathbf{M} is the $J \times 2$ design matrix where the 2nd column contains the squared pairwise
 10126 distances between each individual i and trap j , and thus it depends on both \mathbf{X} and \mathbf{s} .

10127 The asymptotic formula for $\text{Var}(\boldsymbol{\alpha})$ can be cooked up for any type of GLM. As an
 10128 example (we use this below), for the Poisson GLM, the asymptotic variance-covariance
 10129 matrix of $\hat{\boldsymbol{\alpha}}$, considering a single individual having location \mathbf{s} , is¹

$$\text{Var}(\hat{\boldsymbol{\alpha}} | \mathbf{X}, \mathbf{s}) = (\mathbf{M}(\mathbf{s})' \mathbf{D}(\boldsymbol{\alpha}, \mathbf{s}) \mathbf{M}(\mathbf{s}))^{-1}. \quad (10.5.2)$$

10130 This is a function of the design \mathbf{X} as well as \mathbf{s} both of which are balled-up in the re-
 10131 gression design matrix \mathbf{M} , and the matrix \mathbf{D} which is a diagonal matrix having elements
 10132 $\text{Var}(y_j | \mathbf{s}) = \exp(\mathbf{m}' \boldsymbol{\alpha})$ for y_j = the frequency of encounter in trap j and where \mathbf{m}' is the
 10133 j^{th} row of $\mathbf{M}(\mathbf{s})$. We can compute the expected information under the Poisson model with
 10134 known N using this modified formulation. These ideas are meant to motivate technical
 10135 concepts related to model-based design, where we know N , and therefore have a convenient
 10136 variance or information expression to work with. However, in all real capture-recapture
 10137 applications we won't know N , and so we need to address that issue, which we do in the
 10138 next section.

¹ This is basic GLM theory that derives from the fact that the Poisson is a member of the natural exponential family of distributions, e.g., see McCullagh and Nelder (1989) or Agresti (2002).

10139 10.5.3 An Optimal Design Criterion for SCR

10140 There are a number of appealing directions to pursue for deriving a variance-based criterion
 10141 upon which to devise designs for capture recapture studies. For one, we could formulate
 10142 the objective function based on the variance of the Huggins-Alho estimator (Sec. 4.5) of
 10143 N . We find that these expressions depend on individual sample inclusion probabilities
 10144 (if s is close to traps, the individual has a high probability of being encountered and
 10145 *vice versa*), and hence the specific trap locations, and parameters of the model. These
 10146 variance expressions provide a natural design criterion. On the other hand, we find that
 10147 the calculus is a bit tedious at the present time. As an alternative, we devise a variance
 10148 criterion based on the conditional estimator of N having the form

$$\tilde{N} = \frac{n}{\hat{\bar{p}}}$$

10149 where $\hat{\bar{p}}$ is the MLE of the marginal probability that an individual appears in the sample
 10150 of n unique individuals, and it depends on the MLE of the parameters of the encounter
 10151 probability model, $\hat{\alpha}$. We elaborate on the precise form of $\hat{\bar{p}}$ and the variance of its MLE
 10152 below. The variance of this estimator is:

$$\text{Var}\left(\frac{n}{\hat{\bar{p}}}\right) = n^2 * \text{Var}\left(\frac{1}{\hat{\bar{p}}}\right)$$

10153 An important thing to note is that this estimator, and its variance, are *conditional* on the
 10154 sample size of individuals, n . We never set out, in capture-recapture, to obtain a sample
 10155 of n individuals (n is always a stochastic outcome) and so we need to “uncondition” on n .
 10156 Fortunately, this is a simple proposition using standard rules of expectation and variance,
 10157 which produce the following expression:

$$\text{Var}(\tilde{N}(\alpha)) = N\bar{p}\{(1 - \bar{p}) + N\bar{p}\} \left(\frac{\text{Var}(\hat{\bar{p}})}{\bar{p}^4} \right) \quad (10.5.3)$$

10158 The key thing to note about this as a criterion: (1) It depends on \bar{p} , the marginal proba-
 10159 bility of encounter. Clearly the variance decreases as \bar{p} increases. In general, the form of \bar{p}
 10160 depends on the SCR model being used. We will provide an example below. Obviously, \bar{p}
 10161 will depend on the parameter values α . (2) The criterion depends on $\text{Var}(\hat{\bar{p}})$. So, designs
 10162 that estimate \bar{p} well should be preferred. This will also depend on the parameters α and
 10163 *also* the variance of the MLE, $\hat{\alpha}$. Based on these considerations, we suggest a number
 10164 of appealing criteria for constructing spatial designs for capture-recapture studies. For
 10165 convenience we label them Q_1 - Q_4 :

- 10166 (1) $Q_1 = \text{Trace}(\mathbf{V}_\alpha)$ where \mathbf{V}_α is the variance-covariance matrix of the MLE of α . De-
 10167 signs which minimize this criterion are those which are good for estimating the param-
 10168 eters of the encounter probability model.
- 10169 (2) Q_2 is the variance expression in Eq. 10.5.3. Using this criterion, we should prefer
 10170 designs that minimize the variance for estimating N .
- 10171 (3) $Q_3 = 1 - \bar{p}$. Designs which minimize this criterion are those which maximize the
 10172 average capture probability. These should maximize n .
- 10173 (4) $Q_4 = \text{Var}(\hat{\bar{p}})$. We should prefer designs which provide good estimates of \bar{p} .

10174 To make use of any of these criteria in a particular design problem, we need to decide on
 10175 values of N , and the model parameters for computing \bar{p} , and then think about optimizing
 10176 the criterion over all possible designs (see below).

10177 10.5.4 Too much math for a Sunday afternoon

10178 Here we discuss calculation \bar{p} and variance expressions required to compute the design
 10179 criteria above.

10180 Characterizing \bar{p}

10181 In SCR models, an individual with activity center \mathbf{s}_i is captured if it is captured in *any*
 10182 trap and therefore, under the Bernoulli (passive detector) observation model,

$$\bar{p}(\mathbf{s}_i, \mathbf{X}) = 1 - \prod_{j=1}^J (1 - p_{ij}(\mathbf{x}_j, \mathbf{s}_i))$$

10183 where p_{ij} here is the Gaussian (or other) encounter probability model that depends on
 10184 distance between traps and activity centers, say d_{ij} for the distance between individual
 10185 activity center \mathbf{s}_i and trap \mathbf{x}_j . Under the Poisson observation model, with a Gaussian
 10186 hazard model:

$$\bar{p}(\mathbf{s}_i, \mathbf{X}) = 1 - \exp(-\lambda_0 \sum_j \exp(\alpha_1 * d(\mathbf{x}_j, \mathbf{s}_i)^2))$$

10187 where here we emphasized that this is conditional on \mathbf{s}_i and also the design – the trap
 10188 locations \mathbf{x}_j . The *marginal* probability of encounter, averaging over all possible locations
 10189 of \mathbf{s} is:

$$\bar{p}(\mathbf{X}) = 1 - \int_{\mathbf{s}} \bar{p}(\mathbf{s}_i, \mathbf{X}) d\mathbf{s}. \quad (10.5.4)$$

10190 It is important to note that this can be calculated directly *given* the design \mathbf{X} , and
 10191 parameters of the model. This is handy because we see that it is used in the variance
 10192 formulae given subsequently and therefore it is used directly in evaluating any of the
 10193 criteria described above.

10194 Characterizing $\text{Var}(\hat{p})$

10195 Developing an expression for $\text{Var}(\hat{p})$ depends on the observation model. We work here with
 10196 the Poisson observation model, and we do that because the technical argument that follows
 10197 is somewhat easier for that case compared to the Bernoulli model for passive detection
 10198 devices (but see Huggins (1989) and Alho (1990) for additional context). We first express
 10199 the integral in Eq. 10.5.4 as a summation over a fine mesh of points so that:

$$\bar{p}(\mathbf{X}) = \sum_{\mathbf{s}} 1 - \bar{p}(\mathbf{s}_i, \mathbf{X})$$

10200 which under the Poisson observation model is, in a simplified notation:

$$\bar{p}(\mathbf{X}) = \sum_{\mathbf{s}} \left\{ 1 - \exp\left(-\sum_j \exp(\alpha_0 + \alpha_1 d(\mathbf{x}_j, \mathbf{s})^2)\right) \right\}$$

10201 The MLE of $\bar{p}(\mathbf{X})$ has us plug in the MLE of the parameters of the model, in this case
 10202 $\hat{\lambda}_0 = \exp(\hat{\alpha}_0)$ and $\hat{\alpha}_1$. To compute the variance of the MLE of \bar{p} , we note that the variance
 10203 operator can move inside the summation over \mathbf{s} , and the subtraction from 1 doesn't count
 10204 anything, so we have

$$\text{Var}(\hat{p}(\mathbf{X})) = \sum_{\mathbf{s}} \text{Var} \left(\exp \left(- \sum_j \exp(\hat{\alpha}_0 + \hat{\alpha}_1 d(\mathbf{x}_j, \mathbf{s})^2) \right) \right)$$

10205 A few applications of the delta approximation and some arm-waving yields the following
 10206 expression for the variance of \hat{p} :

$$\text{Var}(\hat{p}(\mathbf{X})) = \sum_{\mathbf{s}} \left(\exp(-\hat{\lambda}_{\mathbf{s}}) \right) \left(\sum_j \hat{\lambda}(\mathbf{x}_j, \mathbf{s})^2 (\text{Var}(\hat{\alpha}_0) + d(\mathbf{x}_j, \mathbf{s})^4 \text{Var}(\hat{\alpha}_1)) \right)$$

10207 where $\lambda(\mathbf{x}, \mathbf{s}) = \exp(\alpha_0 + \alpha_1 d(\mathbf{x}, \mathbf{s})^2)$ and $\lambda_{\mathbf{s}} = \sum_{j=1}^J \lambda(\mathbf{x}_j, \mathbf{s})$.

10208 Characterizing $\text{Var}(\hat{\alpha})$

10209 The big picture is this: For a given design \mathbf{X} , we can compute $\text{Var}(\hat{p}(\mathbf{X}))$ – this is just a
 10210 calculation involving sums over all points in the state-space and design points – provided
 10211 we know the variance of the estimator of $\boldsymbol{\alpha}$, $\text{Var}(\hat{\boldsymbol{\alpha}})$ and the parameters of the model.
 10212 However, it is not so easy to write down the analytic form of this matrix. Some calculus
 10213 would have to be done on the conditional likelihood (e.g., from Borchers and Efford (2008))
 10214 to figure out the asymptotic form of this matrix. For our purposes, we think it might
 10215 suffice² to approximate the matrix, using the analogous result from a Poisson or binomial
 10216 GLM assuming that N is known, since we have convenient formulas for those (see Eq.
 10217 10.5.2).

10218 The approximate variance given by Eq. 10.5.2 is conditional on the collection of
 10219 activity centers, $\mathbf{s}_1, \dots, \mathbf{s}_N$. To resolve this, we take the approach outlined previously to
 10220 compute the *expected* information obtained from a particular realization of N individuals,
 10221 and invert that result. In particular, the total information for all N individuals is

$$\mathcal{I}(N) = \mathbf{M}'_1 \mathbf{D}_1 \mathbf{M}_1 + \dots + \mathbf{M}'_N \mathbf{D}_N \mathbf{M}_N$$

10222 We can compute the expected information over *all* elements of the state-space, which is
 10223 just N times the average information of a single individual:

$$\mathbb{E}(\mathcal{I}(N)) = N \sum_{\mathbf{s}} \mathbf{M}(\mathbf{s})' \mathbf{D}(\mathbf{s}) \mathbf{M}(\mathbf{s}).$$

10224 Putting it all together

10225 For a single design, \mathbf{X} , we need to compute this expected information quantity, invert it
 10226 to get the variance of $\hat{\boldsymbol{\alpha}}$, and then either use that variance matrix in the calculation of
 10227 criterion Q_1 , or else evaluate some other quantities along the way to computing the other
 10228 criteria. We can compute \bar{p} (which is Q_3) for a given design without doing any variance
 10229 calculations. If we use $\text{Var}(\hat{\boldsymbol{\alpha}})$ along with \bar{p} , we can compute $\text{Var}(\hat{p})$, which is Q_4 . We can
 10230 combine all of these things together and compute $\text{Var}(\bar{N})$ for a given \mathbf{X} . This gives us Q_2 .

²Warning: But we don't know. No warranty is implied.

10231 10.5.5 Optimization of the criterion

10232 To compute spatial designs that optimize a given criterion, we need to come up with a
 10233 ballpark guess of the model parameters so that the criterion can be evaluated for any
 10234 design. i.e., what are the values of α and N we expect in our study? If we do that,
 10235 and specify the state-space \mathcal{S} then, we can, in theory, optimize the variance criterion
 10236 over all possible configurations of J traps. In formulating the optimization problem note
 10237 that we have J sample locations corresponding to rows of \mathbf{X} . The problem is therefore a
 10238 $2J$ dimensional optimization problem which, for J small, could be solved using standard
 10239 numerical optimization algorithms as exist in almost every statistical computation envi-
 10240 ronment. However, J will almost always be large enough so as to preclude effective use of
 10241 such algorithms. This is a common problem in experimental design, and spatial sampling
 10242 in general, for which sequential exchange or swapping algorithms have been fairly widely
 10243 adopted (e.g., Wynn, 1970; Fedorov, 1972; Mitchell, 1974; Meyer and Nachtsheim, 1995;
 10244 Nychka et al., 1997; Royle and Nychka, 1998). The basic idea is to pose the problem as a
 10245 sequence of 1-dimensional optimization problems in which the objective function is opti-
 10246 mized over 1 or several coordinates at a time. In the present case, we consider swapping
 10247 out \mathbf{x}_j for some point in \mathcal{X} that is nearby \mathbf{x}_j (e.g., a 1st order neighbor). Beginning with
 10248 an initial design, chosen randomly or by some other method, the objective function is eval-
 10249 uated for all possible swaps (at most 4 in the case of 1st order neighbors) and whichever
 10250 point yields the biggest improvement is swapped for the current value. The algorithm is
 10251 iterated over all J design points and this continues until convergence is achieved. Such
 10252 algorithms may yield local optima and optimization for a number of random initial designs
 10253 can yield incremental improvements. We implemented such a swapping algorithm in **R**,
 10254 and it is available as a function in the **scrbook** package with the function **SCRdesign**. The
 10255 algorithm operates on a discrete representation of \mathcal{S} (an arbitrary matrix of coordinates).
 10256 For each point in the design, \mathbf{X} , only the nearest neighbors (the number is specified by
 10257 the user) are considered for swapping into the design during each iteration. For example,
 10258 to compute **ndesigns** = 10 putative optimal designs (each based on a random start) of
 10259 size J = 11, we execute the function as follows:

10260 > des<-SCRdesign(S,X,ntraps=11,ndesigns=10,nn=15,sigma=1)

10261 Where the state-space S , the candidate set X are provided as matrices, **nn** is the number of
 10262 nearest neighbors to inspect for each design point change, and **sigma** is the scale parameter
 10263 of, in this case, a Gaussian hazard model. See the help file **SCRdesign** for examples and
 10264 analysis of the output.

10265 While swapping algorithms are convenient to implement, and efficient at reducing
 10266 the criterion in very high dimensional problems, they do not always yield the global
 10267 optimum. In practice, as in the examples below, it is advisable to apply the algorithm to
 10268 a large number of random starting designs. Our experience is that essentially meaningless
 10269 improvements are realized after searching through a few dozen random starts.

10270 The design criteria we developed above bear a striking resemblance to design criteria
 10271 used to construct so-called space-filling designs (Nychka et al., 1997). Such criteria are
 10272 based on inter-point distances, and space-filling designs seek to optimize some function
 10273 of distance alone, instead of a variance-based objective function. The benefit of this
 10274 approach is that one doesn't have to specify the model to produce a design, and space-
 10275 filling designs have been shown to provide reasonable approximations to designs based on

10276 variance criteria under flexible statistical models (Nychka et al., 1997). This similarity
 10277 suggests that perhaps certain distance-based design criteria might be suitable for SCR
 10278 models. A version of a swapping algorithm used to optimize a space-filling criterion is
 10279 implemented in the **R** package **fields**.

10280 **10.5.6 Illustration**

10281 Because the algorithm operates on a discrete version of \mathcal{S} , it is trivial to apply to situations
 10282 in which the state-space is arbitrary in extent and geometry. However, we consider a
 10283 simplified situation here in order to illustrate the calculation of optimal designs and how
 10284 they look for an idealized situation.

10285 Consider designing a camera trapping study for a square state-space on $[9, 21] \times [9, 21]$
 10286 and with \mathcal{X} being the smaller square $[10, 20] \times [10, 20]$. For this illustration we assumed
 10287 $\alpha_0 = \log(\lambda_0) = -1.7$ and $\sigma = \sqrt{2}$ so that $\alpha_1 = 1/(2\sigma^2) = 1/4$.

10288 Designs of size 11 and 21 were computed using 10 random starting designs. We found
 10289 the optimal design using each of the 4 criteria we described above. To refresh your memory
 10290 Q_1 is the trace of the variance-covariance matrix of $\hat{\alpha}$, Q_2 is the variance of \tilde{N} , Q_3 is $1 - \bar{p}$
 10291 (so the design that minimizes this criterion obtains the highest possible \bar{p}), and Q_4 is the
 10292 variance of \hat{p} . The putative optimal designs (henceforth “best”) are shown in Fig. 10.3.
 10293 There are a few points of some interest here.

10294 The designs are not completely regular but obviously have a systematic look to them.
 10295 For the $J = 11$ designs, the Q_1 design is slightly more compact, with an average closest
 10296 neighbor distance of 2.59 units vs 3.03 units for the Q_3 design. The two designs are qual-
 10297 itatively similar, providing roughly uniform coverage of the design space \mathcal{X} . Conversely,
 10298 the other two criteria produce designs that are highly clustered. Criterion Q_2 which is
 10299 optimal for \tilde{N} , produces 2 clusters of traps, 7 traps in one and 4 traps in the other. Finally,
 10300 designs which are optimal for the criterion Q_4 , the variance of estimating \bar{p} , produce a
 10301 single cluster of traps that is roughly centrally located in the design space. This makes
 10302 sense, because the very dense cluster of traps provides a large number of recaptures near
 10303 the origin $d = 0$, which, intuitively, provides the most information about estimating pa-
 10304 rameter of the encounter probability model. For the $J = 21$ designs, we have an average
 10305 closest neighbor distance of 1.87 for Q_1 and 2.19 for Q_3 but, qualitatively, the structure
 10306 of the designs is similar to $J = 11$. The best design for estimating N (the criterion Q_2)
 10307 produces 2 clusters, but just with more traps. While these illustrations make sense to us,
 10308 we’re not entirely convinced of the implication that 2 clusters of traps should be optimal
 10309 with $J = 21$ total traps. However, it is clear what is going on here is that the tight clusters
 10310 are providing good information about estimating \bar{p} and, by spreading the two clusters out,
 10311 the expected sample size, n , is maximized.

10312 While the designs for Q_1 and Q_3 are not exactly uniform, they are very regular looking
 10313 which we should expect given the regularity of both \mathcal{S} and \mathcal{X} in this case. One thing to
 10314 note is that the trap spacing varies depending on J even though σ is the same, so optimal
 10315 trap spacing should not be viewed as a static thing depending only on the model. Because
 10316 the designs are not exactly regular, the average closest neighbor distance is not exactly
 10317 the same as the trap spacing of a regular grid design.

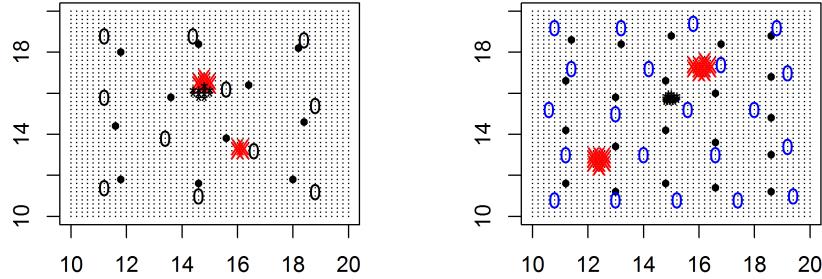


Figure 10.3. Best designs for each of 4 design criteria, produced using the exchange algorithm with 15 nearest-neighbors and based on 10 random starting designs. The left panel shows the best $J = 11$ point designs, and the right panel shows the best $J = 21$ point designs. The solid black dots correspond to the best design for Q_1 , 0 marks the design for Q_3 , "X" for Q_2 and the tightly clustered "*" corresponds to Q_4 .

10.5.7 Density covariate models

Many capture-recapture studies will involve one or more landscape or habitat covariates that are thought to affect density, with the idea of using the methods described in Chapt. 11 for modeling and inference. We imagine that it should be possible to extend the model-based framework described previously to accommodate uncertainty due to having to estimate β , and this could be included as a feature of the design criterion.

In this case, we can think of the captures in a trap being Poisson random variables with mean $\mu(\mathbf{x}, \mathbf{s}) * D(\mathbf{s})$ and we think the same arguments as given above can be used to devise design criteria and optimize them. However, in this case we might not only care about estimating N but also (or instead) inference about the parameters β . Thus, we might choose designs that are good for N or perhaps only good for estimating β or perhaps both. Intuitively, we think these two design objectives conflict with one another to some extent. Model-based approaches should favor areas of higher density, but the design points need to realize variation in the landscape covariates too.

10.6 TEMPORAL ASPECTS OF STUDY DESIGN

The spatial configuration of traps is one of the most important aspects of sampling design for capture-recapture studies. Indeed, as we discussed in the previous section, design under SCR models can be thought of as being analogous to classical model-based spatial design, and the concepts and methods from that field can be brought to bear on the design

of capture-recapture studies. However, there are other aspects of sampling design that should be considered in capture-recapture studies, including the frequency or length of temporal samples. We discuss some of these issues here, although without a detailed or formal analysis.

10340 **10.6.1 Total sampling duration and population closure**

10341 All the models we have discussed so far are *closed population* SCR models, i.e., models
10342 that assume that the population remains constant during our survey. Traditionally, two
10343 different levels of closure have been considered in the capture-recapture literature – de-
10344 mographic and geographic closure (see also Chapt. 4). Demographic closure refers to the
10345 absence of births and deaths, while geographic closure refers to the absence of immigration
10346 and emigration during a study. In traditional capture-recapture, the geographic closure
10347 assumption prohibited (in theory, not in field praxis, of course) any movement off the trap-
10348 ping grid. Kendall (1999) explored a range of scenarios of closure violation, focusing on
10349 different kinds of movement in and out of the study area, and found that several of these
10350 scenarios caused biased abundance estimates from traditional capture-recapture models.

10351 As discussed in Chapt. 5, one main objective of SCR models is to relax the geographic
10352 closure assumption – the model explicitly allows for movements of individuals about their
10353 activity centers, which may have them off the trapping grid for parts of the time, even if
10354 the activity center itself is on the grid. SCR models do, however, assume no permanent
10355 emigration or immigration from the state-space. The interpretation of demographic closure
10356 remains the same in SCR models as it is in traditional CR models.

10357 We have not explored effects of closure violation on SCR abundance and density esti-
10358 mates. Conceptually, we expect estimates to be biased high when births or immigrations
10359 happen during our study. For one, the total number of individuals at the study site during
10360 the course of the study would be higher than at any particular point in time and corre-
10361 spond to a *cumulative* number of individuals in our study area. Further, because some
10362 individuals are not available for detection for the entire study (they only become available
10363 when they are recruited) we would expect detection to be underestimated, potentially
10364 leading to further positive bias in estimates of abundance. Death or emigrations during a
10365 study do not inflate the number of individuals actually on the study area, but as animals
10366 die and become unavailable for detection, we can again imagine a negative bias in baseline
10367 detection and, consequently, some positive bias in N .

10368 To avoid such bias in population estimates, closed population models should typically
10369 be applied to short surveys, where short is relative to the life history of the species under
10370 study. For example, for small mammals, that might mean a few days, whereas for large,
10371 long-lived species with a slow population turnover, several weeks or even a couple of
10372 months can still be considered short. In practice, we have no means of ever guaranteeing
10373 a closed population – even if we sample animals for a day, one of the individuals we record
10374 may be eaten by a predator later that day, or a dispersing individual may arrive just as
10375 we turn our backs. On the other hand, we are faced with the need to collect sufficient
10376 data, which, especially for elusive species, pushes us to sample over longer rather than
10377 shorter time periods. If we do not have enough sampling devices to cover the entire area
10378 of interest at once, rotating study designs (see sec. 10.4) can require even longer sampling
10379 to accumulate sufficient captures and recaptures. So clearly, in temporal study design we

have to strive for a compromise between collecting enough data while still approximating a closed population. For some species we may be able to avoid seasons where violation of demographic closure is particularly likely – for example migration seasons in migratory birds, or specific breeding seasons (or collective suicide season in lemmings). But for many species such biological seasons might be less clear cut. For example, in warm climates tigers and other large cats can breed year round (Nowak, 1999). As a consequence, guidelines as to what time frame adequately approximates a closed population are generally vague and arm-wavy. Unfortunately, we do not have much more to offer on the subject of how to decide on the length of a study, other than to urge you to think about the biology of your study species *before the study* and choose a time window that seems appropriate for that purpose.

10.6.2 Diagnosing and dealing with lack of closure

Once a field study has been conducted, you may wonder whether the collected data contain any evidence that the closure assumption has been met or violated. Relatively few tests for population closure in traditional capture-recapture have been developed, mostly due to the fact that behavioral variation in detection is indistinguishable from violation of demographic closure (Otis et al., 1978; White et al., 1982). Otis et al. (1978) developed a test for population closure that can handle heterogeneity in detection probability, but does not perform well in the presence of time or behavioral variation in p . Stanley and Burnham (1999) developed a closure test for model M_t (time variation in detection), which works well when there is permanent emigration and a large number of individuals migrate. Both tests are implemented in the program **CloseTest** Stanley and Richards (2013).

There are no specific population closure tests for SCR models, for the same reasons that violation of other model assumptions cannot necessarily be distinguished from a lack of population closure. If you are worried that closure might have been violated in your study, one approach of dealing with this problem is to fit an open population model. You can subdivide your study into several periods and fit a spatial version of Pollock's robust design capture-recapture model, which can estimate population size/density for each of these periods (in this context also called primary periods) using models of demographic closure. Alternatively, we may consider fully dynamic models which contain explicit parameters of survival and recruitment (Chapt. 16). These models can be quite time consuming, and if you wanted a faster check you could alternatively fit a spatial Cormack-Jolly-Seber model that only estimates survival. The magnitude of the survival estimates gives you some partial information about population closure in your study – if survival is close to 1 at least there is little evidence of losses of individuals, either through permanent emigration or death. These and other open population models are presented in detail in Chapt. 16. Finally, if your data are too sparse to fit a full-blown open population model, you can subdivide your study into $t = 1, 2, \dots, T$ primary periods and estimate abundance separately for each period's worth of data, possibly sharing the detection parameters across periods, if you can safely assume they remain constant. You can do that by either letting N_t be independent from each other, or by specifying an underlying distribution for all N_t in a multi-session framework as described in Chapt. 14.

10.7 SUMMARY AND OUTLOOK

10422 Design of capture-recapture studies in the context of *spatial* models is an important prob-
10423 lem, but solutions to this problem are mostly *ad hoc* or incomplete at the present time.
10424 As a general rule, we always recommend scenario analysis by Monte Carlo simulation
10425 (Efford and Fewster, 2012; Sollmann et al., 2012; Sun, in prep). This takes a lot of time
10426 but it guarantees forward progress, or at least not doing the dumbest from among several
10427 dumb things. We discussed some examples from the literature that assess trap spacing
10428 and evaluate trap clustering and rotating coverage strategies for sampling large areas. The
10429 nice thing about simulation studies is that we can simulate data for any complex situation
10430 we desire, even if we can't fit the model effectively. Thus, we can always characterize
10431 worst-case situations under pathological model misspecifications.

10432 When designing a spatial capture-recapture study for a single species, trap spacing and
10433 the size of the array can (and should) be tailored to the spatial behavior of that species to
10434 ensure adequate data collection. However, some trapping devices like camera traps may
10435 collect data on more than one species and researchers may want to analyze these data,
10436 too. Independent of the trapping device used, study design will in most cases face a limit
10437 in terms of the number of traps available or logically manageable. As a consequence,
10438 researchers need to find the best compromise between trap spacing and the overall grid
10439 area.

10440 Particularly for large mammal research, SCR models have much more realistic require-
10441 ments in terms of area coverage than non-spatial CR models. In the latter, density esti-
10442 mates can be largely inflated with small trapping grids relative to individual movement
10443 (Maffei and Noss, 2008) – covering at least 4 times the average home range is recom-
10444 mended. Further, we need consistent coverage of the entire study area, as all individuals
10445 in the population of interest must have some probability > 0 to be captured.

10446 In contrast, SCR models work well in study areas on the scale of an individual's home
10447 range (as long as sufficient data is collected) (Sollmann et al., 2012; ?; Marques et al.,
10448 2011), and they provide unbiased estimates for sampling designs that do not expose all
10449 individuals in the sampled population to detectors, i.e., that have "holes" (Efford and
10450 Fewster, 2012). These results, however, should not encourage researchers to design non-
10451 invasive trap arrays based on minimum area requirements and with a minimum number
10452 of detectors. Study design should still strive to expose as many individuals as possible
10453 to sampling and obtain adequate data on individual movement. Large amounts of data,
10454 both individuals and recaptures, do not only improve precision of parameter estimates
10455 (Sollmann et al., 2012; Efford et al., 2004), they also allow including potentially important
10456 covariates (such as gender or time effects in the black bear example – see also Chapt. 7)
10457 into SCR models to obtain density estimates that reflect the actual state of the studied
10458 population.

10459 Beyond the traditional grid-based sampling design, the flexibility of SCR models allows
10460 for different spatial detector arrangements such as linear (see ? for an example), or
10461 dispersed clusters of detectors. How well these different designs perform, comparatively,
10462 remains to be explored.

10463 The other general strategy for constructing spatial designs is a formal model-based
10464 strategy in which we seek the configuration of design points (trap locations) $\mathbf{x}_1, \dots, \mathbf{x}_J$
10465 which are optimal for some formal information-based objective function. This is a standard
10466 approach in classical sampling and experimentation, yet it has not gained widespread use

in ecology. In our view, model-based design under SCR models has great potential due to its coherent formulation and flexibility. On this topic, we have just barely scratched the surface here, showing how to formulate a criterion that is a function of the design, and then optimizing the criterion over all possible designs. Our cursory analysis of model-based design in a single situation did reveal an important aspect of design that has not been discussed in the literature. That is, the optimal spacing of traps in an array depends on the *density* of traps in the state-space. In our analysis, the spacing of 11 and 21 trap optimal designs was quite different. Therefore, this should be considered in practical SCR design exercises.

Conceptually, the information in SCR studies comes in two parts: Recaptures of individuals at different traps (spatial recaptures) and the total sample size of individuals. Maximizing both of these things as objectives induces an explicit trade-off in the construction of capture-recapture designs. We need designs that are good for estimating \bar{p} and also designs that obtain a high sample size of n . Designs that are extremely good only for one or the other will produce bad SCR designs – estimators of density with low precision – or designs in which N is not estimable due to a lack of spatial recaptures. One possible exception is when telemetry data are available (or other auxiliary data). In Chapt. 13 we discuss SCR models that integrate auxiliary information on resource selection obtained by telemetry. Telemetry data are directly informative about the coefficient of the distance term (σ or α_1) and, in fact, can be estimated from telemetry data alone. It stands to reason that, when telemetry data are available, this should affect considerations related to trap spacing. Conceivably even, one might be able to build SCR designs that don't yield any formal spatial recaptures because all of the information about σ is provided by the telemetry data. We have done limited evaluations of the trap spacing problem in the presence of telemetry data, and the results suggest that, while efficient designs have a larger trap spacing than without telemetry data, the realization of some spatial recaptures is important even when telemetry data are available. With the **R** code we provide in Chapt. 13, you should be able to carry out your own custom evaluation of these types of design problems.

10496

Part III

10497

10498

Advanced SCR Models

10499
10500

11

10501
10502

MODELING SPATIAL VARIATION IN DENSITY

10503 Underlying every SCR model is a spatial point process that describes the number and
10504 distribution of animal activity centers. Spatial point processes are characterized by two
10505 key elements: a spatial domain, or state-space \mathcal{S} , and an intensity function which returns
10506 the expected density of points at any location in \mathcal{S} . If the intensity is constant throughout
10507 \mathcal{S} , the point process is said to be homogeneous. Thus far we have focused our attention
10508 on homogeneous point processes whose realized values are the locations of the N activity
10509 centers. When a Poisson prior is placed on N , the model is known as a homogeneous
10510 Poisson point process, which is the classic model of “complete spatial randomness.” A
10511 similar model, that we often use in conjunction with data augmentation and MCMC,
10512 places a binomial prior on N . This is also a model of spatial randomness, and in this
10513 chapter we will compare and contrast the two.

10514 The spatial randomness assumption is often viewed as restrictive because ecological
10515 processes such as habitat selection can result in non-uniform distributions of organisms.
10516 We have argued, however, that this assumption is less restrictive than may be recognized
10517 because a homogeneous point process actually allows for infinite possible “point patterns”,
10518 or realized configurations of activity centers. Furthermore, given enough data, the uniform
10519 prior will have very little influence on the estimated locations of activity centers.
10520 Nonetheless, a homogeneous point process does not allow one to model population density
10521 using covariates, which is an important objective in much ecological research. For
10522 example, even when assuming a homogeneous point process for the activity centers, an
10523 estimated density surface may strongly suggest that individuals are more abundant in one
10524 habitat than another; however, such results do not provide the basis for formally testing
10525 hypotheses about spatial variation in density, and they could not be used to make pre-
10526 dictions about habitat-specific abundance in other regions. A more direct approach is to
10527 replace the homogeneous model with an inhomogeneous model in which the point process
10528 intensity is allowed to vary spatially.

10529 In this chapter, we cover methods for fitting inhomogeneous Poisson and binomial
10530 point process models so that density can be modeled as a function of covariates in much

10531 the same way as is done in generalized linear models. The covariates we consider differ
 10532 from those covered in previous chapters, which were typically attributes of the animal
 10533 (e.g. sex or age) or the trap (e.g. baited or not) and were used to model movement
 10534 or encounter rate. In contrast, here we wish to model covariates that are defined at all
 10535 points in \mathcal{S} , and so we will refer to them as state-space covariates or density covariates.
 10536 These may include continuous covariates such as elevation, or categorical covariates such
 10537 as habitat type. Typically, these state-space covariates are formatted as raster images
 10538 with a prescribed resolution and extent.

10539 One thing to keep in mind when modeling density is that the SCR definition of density
 10540 is different than what is perhaps a more common definition of density in ecology. In SCR
 10541 models, density is defined as the number (or expected number) of *activity centers* in
 10542 some region, whereas in other ecological studies, density is often defined as the number
 10543 of *individuals* in some region at some instant in time. The latter definition is closer to
 10544 the quantity being estimated in distance sampling studies. So which definition is better?
 10545 Does it make more sense to contemplate activity centers or individuals at an instant in
 10546 time? From our perspective, either definition may suffice for a given objective, but we
 10547 note that there exists a formal relationship between the two since an activity center is
 10548 the *average* of an individual's locations during some time period. As such, an activity
 10549 center may be a better descriptor of an individual's preferences than is a location during
 10550 a single instant in time. Moreover, with SCR models we can model both the distribution
 10551 of activity centers (as we will do in this chapter) as well as the distribution of individuals
 10552 during specific instances in time, as is demonstrated in Chapt. 15.

10553 Inhomogeneous Poisson point process models were discussed in the original formulation
 10554 of SCR models by Efford (2004) and were described in more detail by Borchers and Efford
 10555 (2008). We will show that an inhomogeneous point process with a binomial prior on N is
 10556 quite similar to the Poisson model, but is more easily implemented in MCMC algorithms.
 10557 To do so, we will define the data augmentation parameter ψ in terms of the point process
 10558 intensity function, and we will replace the uniform prior on the activity centers with a
 10559 prior that is also derived from the intensity function. Development of this prior, which
 10560 does not have a standard form, is a central component of this chapter. First we begin
 10561 with a review of homogeneous point process models.

11.1 HOMOGENEOUS POINT PROCESS REVISITED

10562 The homogeneous Poisson point process is *the* model of complete spatial randomness and
 10563 is often used in ecology as a null model to test for departures from randomness (Cressie,
 10564 1991; Diggle, 2003; Illian et al., 2008). The Poisson model asserts that the number of points
 10565 in \mathcal{S} is Poisson distributed: $N \sim \text{Poisson}(\mu|\mathcal{S}|)$ where $\mu > 0$ is the intensity parameter
 10566 and $|\mathcal{S}|$ is the area of the state-space. The intensity parameter μ is the density of points,
 10567 and thus multiplying the intensity by the area of some region yields the expected number
 10568 of points in that region. As with all homogeneous point process models, the N points are
 10569 distributed uniformly, which implies that they do not interact with each other in any way
 10570 – for example, they neither attract nor repel one another.

10571 Unlike the Poisson point process, the binomial point process assumes that N is fixed,
 10572 not random. The distinction is illustrated by this simple R code that generates realizations
 10573 from Poisson and binomial point processes in the unit square ($\mathcal{S} = [0, 1] \times [0, 1]$):

```

10574 > Area <- 1                                # Area of unit square
10575 > muP <- 4                                # intensity
10576 > nP <- rpois(1, muP*Area)               # number of points: random
10577 > PPP <- cbind(runif(nP), runif(nP))    # Poisson point pattern
10578 > nB <- 4                                # number of points: fixed
10579 > muB <- nB/Area                          # intensity
10580 > BPP <- cbind(runif(nB), runif(nB))    # binomial point pattern

```

10581 Both of these models are homogeneous because the intensity parameter is constant ($\mu = 4$
 10582 in both cases) and the locations of N the points are mutually independent and uniformly
 10583 distributed. with each other. The key distinction is that N is random in the former and
 10584 fixed in the latter.

10585 Another difference between the Poisson and binomial models is that if the state-space is
 10586 divided into K disjunct regions, the number of points in each region $n(B_k) : k = 1, \dots, K$;
 10587 are independent and identically distributed (i.i.d.) under the Poisson model but not under
 10588 the binomial model. In the Poisson case, the counts are $n(B_k) \sim \text{Poisson}(\mu|B_k|)$, where
 10589 $|B_k|$ is the area of the region B_k . For the binomial model, $n(B_k) \sim \text{Binomial}(N, \pi(B_k))$
 10590 where $\pi(B_k)$ is the proportion of the state-space in B_k ; however, these counts are not i.i.d.
 10591 because the number of points in one region is informative about the number of points in
 10592 another region. For example, if $N = 10$ and if there are 7 points outside the region B_1 ,
 10593 then we can say with certainty that $B_1 = 10 - 7 = 3$.

10594 Fig. 11.1 is meant to further illustrate the characteristics of the binomial model. The
 10595 left panel shows a point pattern realized from a homogeneous binomial point process with
 10596 $N = 50$. The right panel shows the same realization, except that the state-space has
 10597 been discretized into 25 equally-sized disjunct regions, or pixels, and the counts in each
 10598 pixel are shown. Since the pixels are the same size, we have that $\pi(B_k) = 1/25$, and
 10599 the expected number of point in each pixel is $\mathbb{E}(n(B_k)) = N\pi(B_k) = 50/25 = 2$, which
 10600 happens to be the empirical mean in this instance. However, as previously stated, these
 10601 counts are not independent realizations from a binomial distribution since $\sum_k n(B_k) =$
 10602 N . Rather, the model for the entire vector is multinomial: $\{n(B_1), n(B_2), \dots, n(B_k)\} \sim$
 10603 $\text{Multinomial}(N, \{p(B_1), p(B_2), \dots, p(B_K)\})$ (Illian et al., 2008). If you need a refresher on
 10604 the multinomial distribution, refer to Sec. 2.2.3, and consider the following **R** code, which
 10605 generates counts similar to those seen in Fig. 11.1:

```

10606 > n.Bk <- rmultinom(1, size=50, prob=rep(1/25, 25))
10607 > matrix(n.Bk, 5, 5)
10608      [,1] [,2] [,3] [,4] [,5]
10609 [1,]    2    2    2    2    1
10610 [2,]    2    4    0    5    0
10611 [3,]    0    3    2    4    1
10612 [4,]    1    2    1    4    1
10613 [5,]    4    2    4    1    0

```

10614 The dependence among counts has virtually no practical consequence when the number
 10615 of pixels is large. For example, if there are 100 pixels, the number of points in one
 10616 pixels carries very little information about the expected number of points in another
 10617 pixel. However, if there are only 2 pixels, then clearly the number of points in one pixel
 10618 allows one to determine how many points will occur in the remaining pixel.

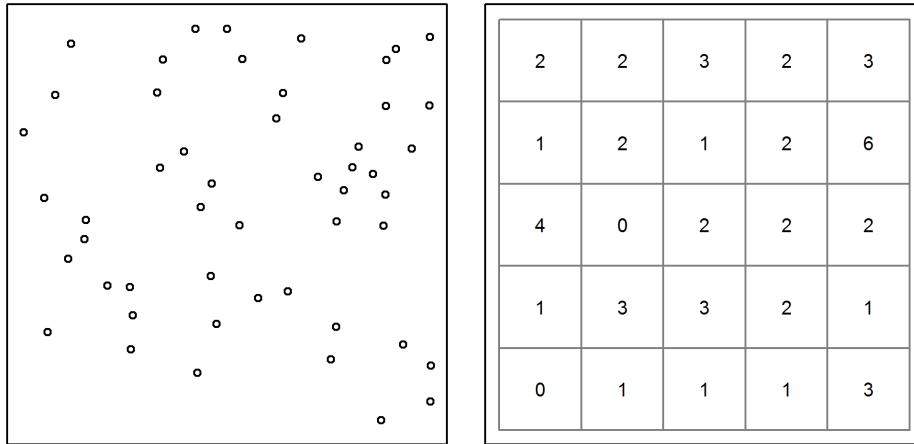


Figure 11.1. Homogeneous binomial point process with $N=50$ points represented in continuous and discrete space.

10619 The discrete representation of space shown in Fig. 11.1 is not only helpful for under-
 10620 standing the properties of a point process, it is also of practical importance when fitting
 10621 SCR models because spatial covariates are almost always represented as rasters, i.e. grids
 10622 with predetermined extent and resolution. In such cases, the definition of the prior for the
 10623 point locations can be changed from the probability that a point occurs at some location
 10624 in space to the probability that it occurs in some pixel of the raster. As we will explain
 10625 in Sec. 11.4.2, this typically involves changing the prior from a uniform distribution to a
 10626 multinomial or categorical distribution.

10627 Having sketched out the basic characteristics of homogeneous Poisson and binomial
 10628 point process models, we will now review their relevance to SCR models before moving
 10629 on to the inhomogeneous models. In a SCR model with a homogeneous point process,
 10630 the intensity parameter μ is interpreted as population density, and N is interpreted as
 10631 population size (i.e. the number of activity centers in \mathcal{S}). These interpretations are true
 10632 regardless of whether we consider the Poisson model or the binomial model, but since N
 10633 is always unknown, one might wonder why we are discussing the binomial model at all.

10634 In our work, we typically adopt the binomial model simply because it is easy to
 10635 implement using MCMC and data augmentation. And while N is truly unknown, we
 10636 use an upper bound, M , which is fixed. Thus, the standard point process we use in
 10637 Bayesian analyses can be regarded in two ways. First, it is a binomial point process with
 10638 M points. Second, in terms of N , it is a thinned binomial point process, where ψ is
 10639 the thinning parameter. With this in mind, the only real difference between the Poisson
 10640 and binomial models, as implemented in SCR contexts, is that in the former, we have
 10641 $N \sim \text{Poisson}(\mu|\mathcal{S}|)$, and in the latter we have $N \sim \text{Binomial}(M, \psi)$. In other words, we
 10642 just have a different prior on N , and when using MCMC, the binomial prior is much more

convenient because it fixes the size of the parameter space and makes it easy to extend the model in each of the ways discussed in this book. It is also worth remembering that the Poisson distribution is the limit of the binomial distribution when M is very large and ψ is very small (Chapt. 2), and thus the two models are much more similar than may appear.

You might have noticed that the intensity parameter μ was not shown for the binomial prior $N \sim \text{Binomial}(M, \psi)$. Instead, we see the data augmentation parameter ψ , which has been used throughout this book, but without much mention of the point process intensity. What then is the relationship between ψ and μ ? As first discussed in Chapt. 5, under data augmentation, the expected value of N is $\mathbb{E}[N] = M\psi$. But, from this chapter, we also know that the expected value of N can be written in terms of μ as $\mathbb{E}[N] = \mu|\mathcal{S}|$. Therefore, $\psi = \mu|\mathcal{S}|/M$ and hence we can directly estimate μ rather than ψ if we want, as will be demonstrated in the next section where the objective is to model μ as a function of spatially-referenced covariates. First, consider the following **R** code, which illustrates some the concepts we just covered:

```

10643 > Area <- 1                      # Area of state-space
10644 > M <- 100                     # Data augmentation size
10645 > mu <- 10                      # Intensity (points per area)
10646 > psi <- (mu*Area)/M          # Data augmentation parameter (thinning rate)
10647 > N <- rbinom(1, M, psi)       # Realized value of N under binomial prior
10648 > cbind(runif(N), runif(N))    # Point pattern from thinned binomial model
10649 [,1]      [,2]
10650 [1,] 0.52779588 0.84306878
10651 [2,] 0.11529168 0.80635046
10652 [3,] 0.06777632 0.66072116
10653 [4,] 0.18694649 0.56761245
10654 [5,] 0.30176929 0.03159091
10655 [6,] 0.84352724 0.89691452
10656 [7,] 0.52766808 0.08871199
10657 [8,] 0.73007529 0.63184825
10658 [9,] 0.01119023 0.69807029

```

11.2 INHOMOGENEOUS POINT PROCESSES

The principal difference between homogeneous and inhomogeneous point processes is that the intensity parameter μ is allowed to vary spatially in the inhomogeneous model. Thus, rather than μ being a fixed constant, it is now a function defined at each point $\mathbf{x} \in \mathcal{S}$. A vast number of options exist for modeling spatial variation in the intensity of a point process (Cox, 1955; Stoyan and Penttinen, 2000; Illian et al., 2008), but here we focus on modeling μ as a function of spatially-referenced covariates and a vector of regression coefficients β ; a function we will denote $\mu(\mathbf{x}, \beta)$. To be clear, $\mu(\mathbf{x}, \beta)$, is a function that returns the expected density of activity centers at location \mathbf{x} , given the covariate values at \mathbf{x} ¹. Since the intensity must be positive, and because the natural logarithm is the

¹The use of \mathbf{x} to denote any point in the state-space could cause confusion because we use \mathbf{x}_j as the location of a trap, but it is standard notation, and the distinction should be evident by the context.

10683 canonical link function of the Poisson generalized linear model (McCullagh and Nelder,
 10684 1989), it is natural to consider the following model:

$$\log(\mu(\mathbf{x}, \boldsymbol{\beta})) = \beta_0 + \sum_{v=1}^V \beta_v C_v(\mathbf{x}) \quad (11.2.1)$$

10685 which says that there are V covariates and β_v is the regression coefficient for covariate
 10686 $C_v(\mathbf{x})$. This covariate, $C_v(\mathbf{x})$, could be any variable defined at all points in the state-
 10687 space, such as habitat type or elevation. Eq. 11.2.1 should look familiar because it is
 10688 the standard linear predictor used in Poisson regression. As with other GLMs, one could
 10689 consider alternative link functions.

10690 Recall from the previous section that for a homogeneous point process, the expected
 10691 number of points in the state-space was simply the intensity parameter multiplied by area:
 10692 $\mathbb{E}(N) = \mu|\mathcal{S}|$. But now that we are regarding the intensity as a function, rather than a
 10693 scalar, this equation is not very useful. So what is $\mathbb{E}(N)$ for an inhomogeneous point
 10694 process? Contemplating a discrete state-space is useful for figuring this out. Imagine
 10695 that the state-space is represented as a raster with many tiny pixels. In this case, we
 10696 will associate \mathbf{x} with pixel ID, i.e. \mathbf{x} just references some pixel with V covariates values
 10697 associated with it. The expected number of individuals in this pixel, say $\mathbb{E}(n(\mathbf{x}))$, can
 10698 intuitively be found by evaluating the intensity function (Eq. 11.2.1) and multiplying it
 10699 by the area of the pixel. In other words, we compute the expected number of individuals
 10700 in a pixel by multiplying the expected value of density for that pixel by the area of the
 10701 pixel. If we do this for each pixel in the state-space, then summing up these values gives
 10702 us what we are after, the expected value of N . Specifically, $\mathbb{E}(N) = \sum_{\mathbf{x} \in \mathcal{S}} \mathbb{E}(n(\mathbf{x}))$. As
 10703 the area of the pixels approaches zero, such that we move from discrete space back to
 10704 continuous space, the summation must be replaced with an integration of the form:

$$\mathbb{E}(N) = \int_{\mathcal{S}} \mu(\mathbf{x}, \boldsymbol{\beta}) d\mathbf{x}. \quad (11.2.2)$$

10705 Together, Eqs. 11.2.1 and 11.2.2 describe a model for spatial variation in density as well
 10706 as population size. The key task in fitting such inhomogeneous point process models is to
 10707 estimate the $\boldsymbol{\beta}$ parameters.

10708 We have now described an approach for modeling the point process intensity, yet in
 10709 order to define the likelihood or to develop an MCMC algorithm for the inhomogeneous
 10710 model, we need to specify the prior distribution for the activity centers. Recall that under
 10711 the homogeneous point process, the prior was $\mathbf{s}_i \sim \text{Uniform}(\mathcal{S})$, for $i = 1, \dots, N$, or
 10712 equivalently:

$$[\mathbf{s}_i] = 1/|\mathcal{S}| \quad (11.2.3)$$

10713 where, as before, $|\mathcal{S}|$ is the area of the state-space. This simply indicates that an activity
 10714 center is just as likely to occur at any location as another. However, if animals exhibit
 10715 habitat selection or simply occur in one region more often than another, it would be
 10716 preferable to replace this prior with one describing the spatial variation in density. Clearly
 10717 this prior should be determined in some way by the spatially-varying intensity function
 10718 $\mu(\mathbf{x}, \boldsymbol{\beta})$. Since the integral of a probability density function (pdf) must be unity, we can
 10719 convert $\mu(\mathbf{x}, \boldsymbol{\beta})$ into a pdf by dividing it by a normalizing constant. In this case, the

normalizing constant is found by integrating $\mu(\mathbf{x}, \boldsymbol{\beta})$ over the entire state-space. The probability density function of the new prior is therefore:

$$[\mathbf{s}_i | \boldsymbol{\beta}] = \frac{\mu(\mathbf{s}_i, \boldsymbol{\beta})}{\int_S \mu(\mathbf{x}, \boldsymbol{\beta}) d\mathbf{x}} \quad (11.2.4)$$

Substituting the uniform prior with this new distribution allows us to fit inhomogeneous binomial point process models to spatial capture-recapture data.

As a practical matter, note that the integral in the denominator of Eq. 11.2.4 is evaluated over space, and since we always regard space as two-dimensional (the state-space is planar), this is a two-dimensional integral that can be approximated using the methods discussed in Chapt. 9, which include Monte Carlo integration and Gaussian quadrature. Alternatively, if our state-space covariates are in raster format, i.e. they are in discrete space, the integral can be replaced with a summation over all the pixels in the raster,

$$[\mathbf{s}_i | \boldsymbol{\beta}] = \frac{\mu(\mathbf{s}_i, \boldsymbol{\beta})}{\sum_{\mathbf{x} \in S} \mu(\mathbf{x}, \boldsymbol{\beta})} \quad (11.2.5)$$

where \mathbf{s} is now defined as “pixel ID” rather than a point in space.

Although the discrete space approach is standard practice, it is technically unjustified because covariate values must be known for all points in space, and a raster is simply a set of spatially-referenced covariate values at an evenly-spaced subset of points (the pixel centers). This same problem is present anytime that we have a sample of the spatial covariates, rather than a function defining their value for all points in space. In such cases, it may be necessary to interpolate the values of the covariates for points in space where they were not measured. One option would be to use a Kriging interpolator, as demonstrated by Rathbun (1996). Another option is to sample the spatial covariates using probabilistic sampling methods, which allow for design-based estimators of their values for the entire study area (Rathbun et al., 2007). Either option could be implemented within maximum likelihood or MCMC estimation methods; however, we do not demonstrate them here because it seems likely that they will be inconsequential in most cases where the raster data are of high resolution, such that the loss of information is negligible when going from continuous space to discrete space. Furthermore, the validity of this assertion, and the level of resolution required to adequately approximate continuous space can often be assessed by checking the consistency of the parameter estimates among varying levels of resolution, as was demonstrated in Chapt. 5.

We now have all the tools needed to fit inhomogeneous point process models. Likelihood-based inference for inhomogeneous Poisson point process models was described by Borchers and Efford (2008) and reviewed in Chapt. 6. Another example is demonstrated in the next section, but first we focus on the binomial model that we favor when conducting Bayesian inference. In the previous section we noted that the data augmentation parameter ψ can be expressed in terms of the intensity parameter μ . The same is true for inhomogeneous models. Specifically, rather than $\mathbb{E}(N) = \psi M$ as before, we use the expected value of N shown in Eq. 11.2.2 which results in

$$\psi = \frac{\int_S \mu(\mathbf{x}, \boldsymbol{\beta}) d\mathbf{x}}{M} \quad (11.2.6)$$

Note that the data augmentation limit M must be high enough so that it is greater than the numerator – i.e., the expected value of N must be less than M .

10759 In the next sections we walk through a few examples, building up from the simplest
 10760 case where we actually observe the activity centers as though they were data. In the second
 10761 example, we fit the inhomogeneous model to simulated data in which density is a function
 10762 of a single continuous covariate. The next example shows an analysis in discrete space
 10763 using both **secr** (Efford, 2011a) and **JAGS** (Plummer, 2003), and in the final example,
 10764 we model the intensity of activity centers for a real dataset collected on jaguars (*Panthera*
 10765 *onca*) in Argentina.

11.3 OBSERVED POINT PROCESSES

10766 In SCR models, the points (activity centers) are not directly observed, but in other contexts they are. Examples include the locations of disease outbreaks, the locations of trees
 10767 in a forest, or the locations of radio-tracked animals. In such cases, it is straightforward
 10768 to fit inhomogeneous point process models and estimate the parameters β from
 10769 Eq. 11.2.1, as we will do in the following example.
 10770

Suppose we knew the locations of N animal activity centers, perhaps as the result of an extensive telemetry study. If we assume N is Poisson distributed and the points are mutually independent of one another, we can fit the inhomogeneous Poisson point process model. The likelihood of this model has two components: $[\{\mathbf{s}_1, \dots, \mathbf{s}_N\}|N]$ and $[N]$. The pdf of the first part is given by Eq. 11.2.4, and with the Poisson assumption we have:

$$\begin{aligned}\mathcal{L}(\beta|\{\mathbf{s}_1, \dots, \mathbf{s}_N\}) &= [\{\mathbf{s}_1, \dots, \mathbf{s}_N\}|N][N] \\ &= \left\{ \prod_{i=1}^N \frac{\mu(\mathbf{s}_i, \beta)}{\int_S \mu(\mathbf{x}, \beta) d\mathbf{x}} \right\} \frac{e^{-\int_S \mu(\mathbf{x}, \beta) d\mathbf{x}} \int_S \mu(\mathbf{x}, \beta) d\mathbf{x}^N}{N!}.\end{aligned}$$

10771 This can be simplified by noting that the denominator in the first component of the model
 10772 cancels with the corresponding piece in the numerator of the second component. And,
 10773 since N is observed and thus does not depend on the parameters, $N!$ can be omitted as
 10774 well. After log-transforming the remaining pieces, we have the log-likelihood often seen in
 10775 textbooks, such as Diggle (2003, pg. 104):

$$\ell(\beta|\{\mathbf{s}_i\}) = \sum_{i=1}^N \log(\mu(\mathbf{s}_i, \beta)) - \int_S \mu(\mathbf{x}, \beta) d\mathbf{x}.$$

10776 Having arrived at the likelihood we could choose a prior distribution for β and obtain the
 10777 posterior distribution using Bayesian methods, or we can find the maximum likelihood
 10778 estimates (MLEs) using standard numerical methods as is demonstrated below.

10779 First, we simulate some data under the model $\mu(\mathbf{x}, \beta) = \beta_0 + \beta_1 \text{ELEV}(\mathbf{x})$, where
 10780 $\text{ELEV}(\mathbf{x})$ is a spatial covariate, say elevation, and $\beta_0 = -6$ and $\beta_1 = 1$. It is worth
 10781 emphasizing that a spatial covariate must be defined at any location in the state-space,
 10782 as is true of the following covariate `elev.fn`:

```
10783 > elev.fn <- function(x) {           # spatial covariate
  10784 +   x <- matrix(x, ncol=2)          # Force x to be a matrix
  10785 +   (x[,1] + x[,2] - 100) / 40.8 # Returns (standardized) "elevation"
  10786 + }
```

```

10787 > # intensity function
10788 > mu <- function(x, beta0, beta1) exp(beta0 + beta1*elev.fn(x=x))
10789 > beta0 <- -6 # intercept of intensity function
10790 > beta1 <- 1 # effect of elevation on intensity
10791 > # Next line computes integral
10792 > EN <- cuhre(2, 1, mu, beta0=beta0, beta1=beta1,
10793 +           lower=c(0,0), upper=c(100,100))$value

```

10794 The function `elev.fn` returns the value of elevation at any location x . The standardization bit is not necessary, but helps with the model fitting below. The next lines of the
 10795 code define the intensity function $\mu(x, \beta)$ in terms of elevation and the regression coefficients.
 10796 The last line uses the `cuhre` function in the `R2Cuba` package (Hahn et al., 2010) to
 10797 compute the expected value of N in a $[0, 100] \times [0, 100]$ square state-space, which is the
 10798 two-dimensional integral of Eq. 11.2.4. This integral could also be computed using a fine
 10799 grid of points as we have done in previous chapters, but it is useful to gain familiarity
 10800 with more efficient integration functions in **R**.

10801 The **R** code above demonstrates how to obtain the expected value of N given a spatial
 10802 covariate and the coefficients defining the intensity function. Now we need to generate a
 10803 realized value of N and distribute the N points in proportion to the intensity function.
 10804 This is not as simple as it was to simulate data from a homogeneous point process because
 10805 the points are no longer uniformly distributed within the state-space. Instead one must
 10806 resort to methods such as rejection sampling, which involves simulating data from a stan-
 10807 dard distribution and then accepting or rejecting each point using probabilities defined
 10808 by the distribution of interest. For more information, readers should consult an accessible
 10809 text such as Robert and Casella (2010). In our example, we simulate from a uniform dis-
 10810 tribution and then accept or reject using the (scaled) probability density function $[s_i | \beta]$
 10811 (Eq. 11.2.4). The following **R** commands demonstrate the use of rejection sampling to
 10812 simulate an inhomogeneous point process for the elevation covariate depicted in Fig. 11.2.

```

10814 > set.seed(31025)
10815 > beta0 <- -6 # intercept of intensity function
10816 > beta1 <- 1 # effect of elevation on intensity
10817 > # Next line computes integral, which is expected value of N
10818 > EN <- cuhre(2, 1, mu, beta0=beta0, beta1=beta1,
10819 +           lower=c(0,0), upper=c(100,100))$value
10820 > EN
10821 [1] 39.96634
10822 > N <- rpois(1, EN) # Realized N
10823 > s <- matrix(NA, N, 2) # This matrix will hold the coordinates
10824 > elev.min <- elev.fn(c(0,0))
10825 > elev.max <- elev.fn(c(100, 100))
10826 > Q <- max(c(exp(beta0 + beta1*elev.min),
10827 +               exp(beta0 + beta1*elev.max)))
10828 > counter <- 1
10829 > while(counter <= N) {
10830 +   x.c <- runif(1, 0, 100); y.c <- runif(1, 0, 100)
10831 +   s.cand <- c(x.c,y.c)

```

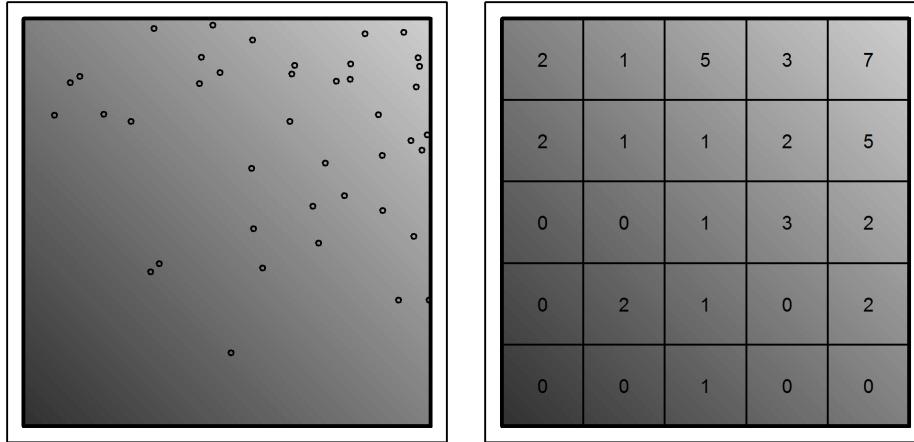


Figure 11.2. An example of a spatial covariate, say elevation, and a realization from an inhomogeneous Poisson point process with $\mu(\mathbf{x}, \boldsymbol{\beta}) = \exp(\beta_0 + \beta_1 \text{ELEV}(\mathbf{x}))$ where $\beta_0 = -6$ and $\beta_1 = 1$.

```

10832 +   pr <- mu(s.cand, beta0, beta1) #/ EN
10833 +   if(runif(1) < pr/Q) {
10834 +     s[counter,] <- s.cand
10835 +     counter <- counter+1
10836 +   }
10837 + }
```

10838 Similar methods are also implemented in the **R** package **spatstat** (Baddeley and Turner, 2005).

10840 The 41 simulated points are shown in Fig 11.2. High elevations are represented by
 10841 light gray and low elevations are darker. The density of points in apparently higher in
 10842 lighter regions suggesting that these simulated animals prefer high elevations. Given these
 10843 points, we will now estimate β_0 and β_1 by minimizing the negative-log-likelihood using
 10844 **R**'s **optim** function.

```

10845 > nll <- function(beta) {
10846 +   beta0 <- beta[1]
10847 +   beta1 <- beta[2]
10848 +   EN <- cuhre(2, 1, mu, beta0=beta0, beta1=beta1,
10849 +                     lower=c(0,0), upper=c(100,100))$value
10850 +   -(sum(beta0 + beta1*elev.fn(s)) - EN)
10851 + }
10852 > starting.values <- c(-10, 0)
10853 > fm <- optim(starting.values, nll, hessian=TRUE)
```

```

10854 > cbind(Est=fm$par, SE=sqrt(diag(solve(fm$hessian)))) # estimates and SEs
10855   Est      SE
10856 [1,] -5.9335547 0.2204693
10857 [2,]  0.9545532 0.1771507

```

Maximizing the Poisson likelihood took a fraction of a second, and we obtained estimates of $\hat{\beta}_0 = -5.93$ and $\hat{\beta}_1 = 0.95$, which are very close to the data-generating values. The 95% confidence interval for $\hat{\beta}_1$ is [0.61, 1.3] and since it does not include zero, the null hypothesis that $\beta_1 = 0$, i.e. that there is no effect of elevation on density, can be rejected. In addition to testing hypotheses, these results can be used to predict population size in new regions or create predicted density surface maps by plugging the parameter estimates into Eqs. 11.2.1 and 11.2.2.

You might wonder if the results would differ if we assumed a binomial rather than a Poisson distribution for N . This can be checked using the following code:

```

10867 > nllBin <- function(beta, M=100) {
10868   +   beta0 <- beta[1]
10869   +   beta1 <- beta[2]
10870   +   EN <- cuhre(2, 1, mu, beta0=beta0, beta1=beta1,
10871   +                     flags=list(verbose=0),
10872   +                     lower=c(0,0), upper=c(100,100))$value
10873   +   N <- nrow(s)
10874   +   psi <- EN/M
10875   +   -(sum(beta0 + beta1*elev.fn(s) - log(EN)) +
10876   +     dbinom(N, M, psi, log=TRUE))
10877   + }
10878 > cbind(Est=fmBin$par, SE=sqrt(diag(solve(fmBin$hessian)))) # est and SE
10879   Est      SE
10880 [1,] -5.9339490 0.1965479
10881 [2,]  0.9545742 0.1771962

```

which indicates that the MLEs are almost identical, and supports the claim that the prior on N has little influence in SCR models. Notice, however, that the standard error for β_0 is smaller under the binomial model than it was under the Poisson model – a difference that will dissipate as M tends toward infinity.

This example demonstrates that if we had the data we wish we had, i.e. if we knew the coordinates of the activity centers, we could easily estimate the parameters governing the underlying point process and make inferences about spatial variation in density and abundance. Unfortunately, in virtually all animal ecology studies, the locations of the N animals, or the N activity centers, cannot be directly observed. Thus we need extra information to estimate the locations of these unobserved points, which in the case of SCR, comes from the locations where each animal is captured.

11.4 FITTING INHOMOGENEOUS POINT PROCESS SCR MODELS

11.4.1 Continuous space

In this example, we will use the same set of points simulated in the previous section to generate spatial capture-recapture data. Specifically, we overlay a grid of 49 traps on the

map shown in Fig. 11.2 and simulate capture histories conditional on the activity centers. Then, we will attempt to estimate the activity center locations as though we did not know where they were, as is the case in real applications. We will also estimate β_0 and β_1 as before and see how the estimates compare when the points are not actually observed. The following **R** code simulates encounter histories under a Poisson observation model (see Chapt. 9), which could be appropriate in camera trapping studies or when using other methods in which animals could be detected multiple times at a trap during a single occasion.

```

10894 > xsp <- seq(20, 80, by=10); len <- length(xsp)
10895 > X <- cbind(rep(xsp, each=len), rep(xsp, times=len)) # traps
10896 > ntraps <- nrow(X); nooccasions <- 5
10897 > y <- array(NA, c(N, ntraps, nooccasions)) # capture data
10898 > sigma <- 5 # scale parameter
10899 > lam0 <- 1 # basal encounter rate
10900 > lam <- matrix(NA, N, ntraps)
10901 > set.seed(5588)
10902 > for(i in 1:N) {
10903 +   for(j in 1:ntraps) {
10904 +     # The object "s" was simulated in previous section
10905 +     distSq <- (s[i,1]-X[j,1])^2 + (s[i,2] - X[j,2])^2
10906 +     lam[i,j] <- exp(-distSq/(2*sigma^2)) * lam0
10907 +     y[i,j,] <- rpois(nooccasions, lam[i,j])
10908 +   }
10909 + }
```

Now that we have a simulated capture-recapture dataset y , we can simulation the posterior distributions of the model parameters using MCMC. A commented Gibbs sampler written in **R** is available in the accompanying **R** package **scrbook** (see **?scrIPP**). This function is not meant to be an all purpose tool for fitting SCR models using MCMC. Instead, it is presented so that interested readers can better understand the computational aspects of the problem and can modify it for their purposes. The function can be used as SO:

```

10927 > fm1 <- scrIPP(y, X, M=150, 10000, xlims=c(0,100), ylims=c(0,100),
10928 +                     space.cov=elev.fn, tune=c(0.4, 0.2, 0.3, 0.3, 7))
10929 > plot(mcmc(fm1$out))
```

which requests 10000 posterior samples and estimates the effect of the spatial covariate, elevation, on density. The argument **space.cov** accepts any spatial covariate that returns a real value for any location in the rectangular state-space defined by the **xlims** and **ylims** arguments. Currently, the function places uniform priors on the parameters σ , λ_0 , β_0 and β_1 , although this could easily be modified. The **tune** argument specifies the tuning parameters used in the Metropolis-within-Gibbs steps of the algorithm. These should be chosen using trial and error to achieve an acceptance rate of between 0.4 and 0.6, roughly. See Chapt. 17 for more details about MCMC.

Results of the analysis are shown in Fig. 11.3 and Table 11.1. Fig. 11.3 displays the trace plots of the Markov chains as well as the posterior distributions for three parameters.

10940 The chains appear to converge rapidly but may need to be run longer to reduce Monte
 10941 Carlo error. Summaries of the posterior distributions are presented in Table 11.1. The
 10942 posterior means for β_0 and β_1 are quite similar to MLEs from the analysis in the previous
 10943 section in which we assumed no observation error. However, we see that the confidence
 10944 intervals are wider. With respect to the other parameters in the model, we see that all of
 10945 the data generating parameter fall within the 95% credible intervals. One thing to note
 10946 is that, although the point estimates for the expected and realized values of N are quite
 10947 similar, the posterior for the realized value of N is more precise. This is to be expected
 10948 because the uncertainty associated with the realized value of N is entirely determined by
 10949 the sampling error. That is, if we could perfectly detect all of the individuals in S , there
 10950 would be no uncertainty about N . In contrast, the variance for expected value of N is
 10951 composed both process error and sampling error. See Chapt. 5 and Efford and Fewster
 10952 (2012) for additional discussion on the difference between realized and expected values of
 10953 abundance.

10954 Fitting continuous space inhomogeneous point process models is somewhat difficult in
 10955 **BUGS** because the “IPP” prior $[s_i|\beta]$, unlike the uniform prior, is not one of the available
 10956 distributions that comes with the software. It is possible to add new distributions in
 10957 **BUGS**, but it is somewhat cumbersome. **secr** allows users to fit continuous space models
 10958 using linear or polynomial functions of the easting and northing coordinates, but it does
 10959 not accept truly continuous covariates that are functions of space. However, these are
 10960 not really important limitations because discrete space versions of the model are straight-
 10961 forward, and virtually all spatial covariates are, or can be, defined as such.

Table 11.1. Summary of posterior distributions from SCR model with inhomogeneous point process.

Parameter	Mean	SD	2.5%	97.5%
$\sigma = 5$	5.232	0.310	4.681	5.858
$\lambda_0 = 1$	0.802	0.119	0.595	1.049
$\beta_0 = -6$	-5.856	0.254	-6.376	-5.393
$\beta_1 = 1$	0.985	0.209	0.575	1.378
$N = 41$	47.615	8.041	35.000	66.000
$E(N) = 39.9$	47.551	10.992	29.837	71.332

10962 11.4.2 Discrete space

10963 To fit inhomogeneous point process models using covariates in discrete space, i.e. in raster
 10964 format, we follow the same steps as outlined in Chapt. 9 – we define s_i as pixel ID,
 10965 and we use the categorical distribution as a prior. This effectively changes the problem
 10966 from estimating the coordinates of an activity center, to estimating the pixel in which an
 10967 activity center is located. As pixel size approaches zero, these two become equivalent. A
 10968 good example is found in (Mollet et al., In review). Here we present an analysis of the
 10969 simulated data shown in the Fig. 11.2. The spatial covariate, let’s call it forest canopy
 10970 height (CANHT), was simulated using using the code shown on the help page `ch11` in
 10971 `scrbook`. The points are the number of activity centers in each pixel, generated from

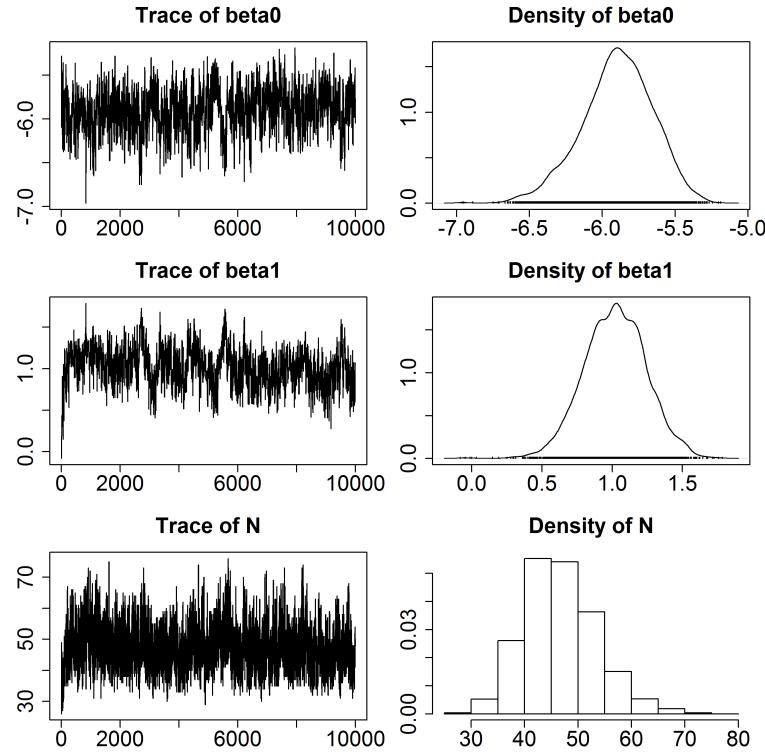


Figure 11.3. Trace plots and posterior distributions from MCMC analysis of SCR model with inhomogeneous point process. Analysis was conducted using the `scrIPP` function in the accompanying **R** package `scrbook`.

10972 a single realization of the inhomogeneous point process model with intensity $\mu(\mathbf{x}, \boldsymbol{\beta}) =$
 10973 $\exp(\beta_0 + \beta_1 \text{CANHT}(\mathbf{x})) \times \text{pixelArea}$, where $\beta_0 = -6$ and $\beta_1 = 1$.

10974 The **BUGS** description of the model is shown in panel 11.1. The vector `probs[]` is
 10975 the prior probability defined by Eq. 11.2.5, which is the probability that an individual's
 10976 activity center is located at pixel \mathbf{x} . `grid` is the matrix of coordinates for each pixel.

10977 This model can also be fit in `secr`, which refers to the raster data as a "habitat mask".
 10978 The habitat mask is essentially a `data.frame` with attributes. The `data.frame` itself has
 10979 2 columns for the coordinates of each of the pixel centers. The attributes of the object
 10980 include information such as the area of the pixels and the spacing between pixel centers.
 10981 If there are covariates, these too are stored as an attribute of the habitat mask, and are
 10982 formatted as a `data.frame` with 1 row per pixel and 1 column per covariate. Once the
 10983 data have been formatted correctly, fitting the model in `secr` is as simple as:

```
10984 > secr1 <- secr.fit(ch, model=D~canht, mask=msk)
```

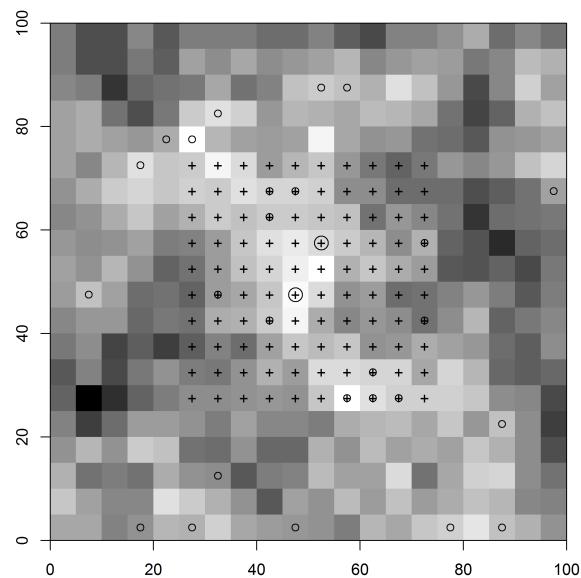


Figure 11.4. Simulated activity centers in discrete space. The spatial covariate, canopy height, is highest in the lighter areas and density increases with canopy height. A single activity center is shown as a small circle, and larger circles represent two activity centers in a pixel. Trap locations are shown as crosses.

Table 11.2. Comparison of **secr** and **JAGS** results. Point estimates from the Bayesian analysis are posterior means. Intervals are lower and upper 95% CIs.

Parameter	Truth	Software	Mean	SD	2.5%	97.5%
λ_0	1.00	JAGS	1.04	0.087	0.88	1.22
	1.00	secr	1.08	0.089	0.92	1.27
σ	10.00	JAGS	10.16	0.373	9.46	10.94
	10.00	secr	9.84	0.350	9.18	10.55
β_1	1.00	JAGS	1.20	0.350	0.50	1.88
	1.00	secr	1.09	0.316	0.47	1.71
N	30.00	JAGS	26.63	2.585	23.00	33.00
	30.00	secr	28.19	3.037	24.49	37.39
$\mathbb{E}(N)$	32.30	JAGS	26.39	5.048	17.25	36.96
	32.30	secr	28.19	6.117	18.52	42.93

where `D~canht` indicates that we want to model density as a function of canopy height, which is defined in the `msk` object. R code to format the data and fit the models using **secr** and **JAGS** is available in `scrbook`, found by issuing the command: `help(ch11secr-jags)`.

Results of fitting the model in **JAGS** and **secr** are shown in Table 11.2 and are similar as expected. The differences that do exist are likely due to the differences in Bayesian and frequentist estimation methods, as discussed in Chapt 3. Either answer may be “more correct” depending upon one’s criteria for correctness!

11.5 ARGENTINA JAGUAR STUDY

Estimating density of large felines has been a priority for many conservation organizations, but few robust methodologies existed before the advent of SCR. Distance sampling is not feasible for such rare and cryptic species, and traditional capture-recapture methods yield estimates that are highly sensitive to the subjective choice of the effective survey area. SCR models provide a powerful alternative because density can be estimated directly and data can be collected using non-invasive methods such as camera traps or hair snares.

In this example, we demonstrate how readily density can be estimated for a globally imperiled species using SCR. Furthermore, we show how inhomogeneous point process models can be used to test important hypotheses regarding the factors affecting density. The data come from an 8-year camera-trapping study designed to assess the impacts of poaching on jaguar density in Argentina, near the borders of Brazil and Paraguay. Additional information about the study is presented in Paviolo et al. (2008) and Paviolo et al. (2009). The expected effect of poaching is a decline in jaguar density due to the direct removal of individuals and the depletion of its main prey species. To conserve jaguars and related species, protected areas have been established and three levels of protection are recognized, as depicted in Fig. 11.5. The dark gray area is the Iguazú National Park that is patrolled regularly by law enforcement officials. The medium gray areas are not protected and rarely patrolled. Finally, the light gray areas are large soybean monocultures, cities and dams which provide no suitable habitat for jaguars

To test for differences in density between the three regions, we modeled the point process intensity parameter as a function of protection status (PROTECT), which we

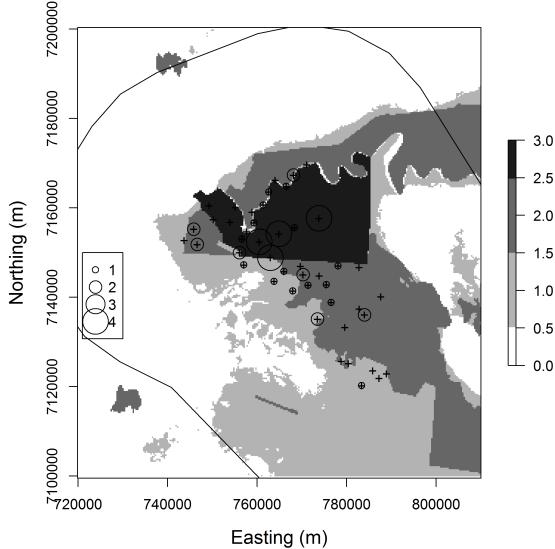


Figure 11.5. Jaguar detections at 46 camera trap stations. The three levels of protection status are no protection (light gray), some protection (gray), and Iguazú National Park (dark gray). Non-habitat (soybean monocultures) is shown in white.

treated as an ordinal variable:

$$\mu(\mathbf{x}, \boldsymbol{\beta}) = \exp(\beta_0 + \beta_1 \text{PROTECT}(\mathbf{x})) \times \text{pixelArea}.$$

We predicted that β_1 would be greater than zero, indicating that jaguar density increases with protection status. In addition to modeling spatial variation in density, we also modeled the scale parameter of the half-normal (or Gaussian) encounter model as sex-specific because male cats typically have larger home ranges than females (Sollmann et al., 2011). Since sex is an individual-specific covariate, and not observed for the individuals that were not captured, a prior distribution is required for the sex of uncaptured individuals. We used a Bernoulli prior with probability 0.5 to describe our uncertainty about sex ratio. Another equivalent option is to augment the data with an equal number of males and females and let the MCMC algorithm determine which of these individuals are actually members of the population.

An additional unique aspect of this study is the highly irregular state-space. Unlike in the examples of simulated data, the geometry of this state-space is not a simple rectangular region. Instead, it is the area south of the Iguazú River, which runs along the northern border of the park shown in dark green in Fig. 11.5, and it excludes the large soybean monocultures. Fitting models in highly convoluted spatial regions raises the question: How does one integrate Eq. 11.2.4 over this irregular space? Earlier we used the function `cuhre`

Table 11.3. Summaries of posterior distributions from the model of jaguar density. σ is the scale parameter of the half-normal detection function. λ_0 is baseline encounter rate, β_1 is the effect of protection status on jaguar density, ρ is the sex-ratio, N is population size. The last three parameters are the density estimates (jaguars/100 km²) for the three levels of protection.

	Mean	SD	2.5%	97.5%
N	35.819	7.9749	23.0000	54.0000
D_{low}	0.906	0.3265	0.3813	1.6682
D_{med}	0.770	0.2841	0.2698	1.4392
D_{high}	1.370	0.3069	0.8315	1.9955
σ_{female}	5501.204	876.8774	4142.2756	7578.5692
σ_{male}	6452.570	915.3623	4970.3215	8505.5219
λ_0	0.006	0.0016	0.0034	0.0098
ψ	0.355	0.0937	0.1998	0.5638
β_0	-4.686	0.2602	-5.2346	-4.2129
β_1	0.174	0.3500	-0.5104	0.8649
Sex Ratio	0.489	0.0550	0.3824	0.6000

in R for the two-dimensional integration, but its `lower` and `upper` arguments essentially assume that the state-space is square. There are methods of transforming the state-space that might allow us to work around this problem, but once again we find that it is most convenient to work in discrete space and sum over all the pixels defining \mathcal{S} .

We fit the model to data from a single year in which 46 camera stations were operational, each consisting of a pair of cameras placed along roads or small trails. Forty-five detections of 16 jaguars (8 males and 8 females) were made over a 95-day sampling period. The mean number of sampling days at each camera station was 48.2. The raw capture data shown in Fig. 11.5 suggest that the highest number of captures was in the national park, but there were also several traps in the park with no captures. Furthermore, few cameras were placed far from the protected areas, making it somewhat difficult to detect differences in density. R code to fit the model is available in `scrbook` on the help page `jaguarDataCh11`. Parameter estimates are shown in Table 11.3.

The results indicate that efforts to protect jaguars by reducing poaching in protected areas are not working as well as hoped for. The posterior probability that $\beta_1 > 0$ was only 0.69, and the posterior mean of realized density was only 51% higher in the national park than in the unprotected area. Fig. 11.6 shows the estimated density surfaces. The first map is the expected density in each of the three values, which was computed by plugging in the posterior mean values of β_0 and β_1 into the log-linear intensity function. The second map is the realized density surface – the conditional-on- N probability distribution of the number of activity centers in each pixel of the rasterized state-space. The expected values would be used if we were interested in making inferences about other areas or time periods, whereas the realized map is the best description of the system during the study period.

We note that there is room for improvement in our analysis, and our results should be considered preliminary. The political boundaries used to demarcate protected areas are not as concrete as we might like. In reality poaching pressure is likely higher near remote park boundaries than in well-guarded park interiors. One option for addressing this would be to use a continuous measure of poaching pressure such as distance from the nearest

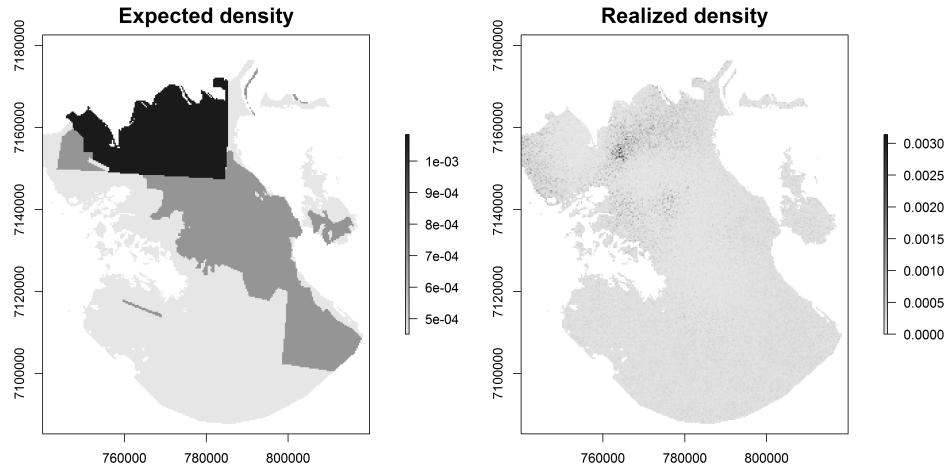


Figure 11.6. Estimated density (activity centers / pixel) surfaces from the analysis of the jaguar data.

town, or some other accessibility metric. It would also be worthwhile to model density separately for each sex because many of the detections outside of the park were of males, and thus it is possible that the sexes use habitat differently (Conde et al., 2010). Other extensions warranting investigation include treating PROTECT as a categorical rather than ordinal variable, and assessing the effects of roads and trails on jaguar movement using the methods described in Chapt. 12. Developing models for these extensions could be readily accomplished by modifying the fitting functions found in the **R** package **scrbook**.

11.6 SUMMARY AND OUTLOOK

One of the distinguishing features of spatial capture-recapture models is that they allow for inference about spatial variation in density without relying on ad hoc approaches for determining the amount of area surveyed. The approach described in this chapter involves modeling the locations of activity centers as outcomes of an inhomogeneous point process with intensity determined by covariates defined at all locations in the state-space. Covariate effects can be evaluated in exactly the same way as is done in generalized linear models, making it easy to interpret the results.

All the examples in this section included a single state-space covariate, but this was for simplicity only. Including multiple covariates poses no additional challenges. Similarly, additional model structure such sex-specific encounter rate parameters or behavioral responses can be accommodated and fit using **secr**, **BUGS**, or by extending the functions in **scrbook**. It is also possible to consider covariates that affect both density and ecological distance as will be described in the next chapter. The ramifications of this are enormous for applied ecological research and conservation efforts because researchers can use capture-recapture data to identify areas where both density and landscape connectiv-

11081 ity are high (Royle et al., 2013). Addressing such questions is simply not possible using
11082 standard, non-spatial capture-recapture methods.

11083 Although we focused on modeling the point process intensity as a function of covariates,
11084 other options for fitting inhomogeneous models exist (Illian et al., 2008). Cox processes are
11085 models in which the point process intensity is a function of spatial random effects. Such
11086 methods are useful for accommodating overdispersion, but it seems unlikely that most SCR
11087 datasets could support such complexity. Gibbs processes are another important class of
11088 models that are distinguished by the interactions of points. Although little work has been
11089 done on such models in the context of SCR studies (Reich et al., 2012), we expect they
11090 will receive more attention because they can be used to model processes such territoriality
11091 (points repel one another) or aggregation (points attract one another). Neyman-Scott
11092 processes are another option for modeling aggregation or clustering, and could be useful
11093 for studying gregarious species.

```

model{
  sigma ~ dunif(0, 20)
  lam0 ~ dunif(0, 5)
  beta0 ~ dunif(-10, 10)
  beta1 ~ dunif(-10, 10)
  for(j in 1:nPix) {
    mu[j] <- exp(beta0 + beta1*CANHT[j])*pixArea
    probs[j] <- mu[j]/EN
  }
  EN <- sum(mu[]) # Expected value of N, E(N)
  psi <- EN/M
  for(i in 1:M) {
    z[i] ~ dbern(psi)
    s[i] ~ dcat(probs[])
    x0g[i] <- grid[s[i],1]
    y0g[i] <- grid[s[i],2]
    for(j in 1:ntraps) {
      dist[i,j] <- sqrt(pow(x0g[i]-traps[j,1],2) + pow(y0g[i]-traps[j,2],2))
      lambda[i,j] <- lam0*exp(-dist[i,j]*dist[i,j]/(2*sigma*sigma)) * z[i]
      y[i,j] ~ dpois(lambda[i,j])
    }
  }
  N <- sum(z[]) # Realized value of N
}

```

Panel 11.1: **BUGS** model specification for the inhomogeneous point process model in discrete space. A nearly equivalent formulation would involve omitting β_0 and modeling the expected number of activity centers as $\mathbb{E}(N) = M\psi$ with $\psi \sim \text{Uniform}(0, 1)$.

11094
11095

12

11096

MODELING LANDSCAPE CONNECTIVITY

11097 Every spatial capture-recapture model that we have considered so far has expressed en-
11098 counter probability as a function of the Euclidean distance between individual activity
11099 centers s and trap locations x . As a practical matter, models based on Euclidean distance
11100 imply circular, symmetric, and stationary home ranges of individuals, and these are not
11101 often biologically realistic. While these simple encounter probability models will often be
11102 sufficient for practical purposes, especially in small data sets, sometimes developing more
11103 complex models of the detection process as it relates to space usage of individuals will
11104 be useful. Animals may not judge distance in terms of Euclidean distance but, rather,
11105 according to the configuration of habitat patches, quality of local habitat, perceived mor-
11106 tality risk, and other considerations. Together, the degree to which these factors facilitate
11107 or impede movement determines landscape connectivity (Tischendorf and Fahrig, 2000),
11108 which is widely recognized to be an important component of population viability (With
11109 and Crist, 1995; Compton et al., 2007). Moreover, because encounter probability and
11110 the distance metric upon which it is based represent outcomes of individual movements
11111 about their home range, ecologists might have explicit hypotheses about how environmen-
11112 tal variables affect the distance metric, and it is therefore desirable to incorporate these
11113 hypotheses directly into SCR models so that they may be formally evaluated statistically.

11114 Although much theory has been developed to predict the effects of decreasing con-
11115nectivity, few empirical studies have been conducted to test these predictions due to the
11116 paucity of formal methods for estimating connectivity parameters (Cushman et al., 2010;
11117 Hanks and Hooten, in press). Instead, ecologists often rely on expert opinion or *ad hoc*
11118 methods of specifying connectivity values, even in important applied settings (Adriaensen
11119 et al., 2003; Beier et al., 2008; Zeller et al., 2012). In addition, no methods are available
11120 for simultaneously estimating population density and connectivity parameters, in spite of
11121 theory predicting interacting effects of density and connectivity on population viability
11122 (Tischendorf et al., 2005; Cushman et al., 2010). In this chapter, following Royle et al.
11123 (2013), we provide a framework for modeling landscape connectivity using SCR models,
11124 by parameterizing models for encounter probability based on “ecological distance”. A
11125 natural candidate framework for modeling ecological distance is the least-cost path which

is used widely in landscape ecology for modeling connectivity, movement and gene flow (Adriaensen et al., 2003; Manel et al., 2003; McRae et al., 2008). In practical applications, variables that influence landscape connectivity, or the effective cost of moving across the landscape, include things like highways (e.g., Epps et al., 2005), elevation (Cushman et al., 2006), ruggedness (Epps et al., 2007), snow cover (Schwartz et al., 2009), distance to escape terrain (Shirk et al., 2010), range limitations (McRae and Beier, 2007), or distance from urban areas, highways, human disturbance or other factors that animals might avoid.

Royle et al. (2013) provided an SCR framework based on least-cost path for modeling landscape connectivity. They parameterized encounter probability based not on Euclidean distance but, rather, on the least-cost path between an individual's activity center and a trap location. This is parameterized in terms of one or more parameters that relate the *resistance* of the landscape to explicit covariates. In this way, SCR models can explicitly accommodate landscape structure and account for connectivity of the landscape. Using this methodological extension of SCR models, it is possible to make formal statistical inferences about movement and connectivity from capture-recapture studies that generate sparse individual encounter history data without subjective prescription of resistance or cost surfaces. While we believe there should be much ecological interest in developing SCR models that account for landscape connectivity, it is also important for obtaining more accurate estimates of density; under simple models of landscape connectivity, incorrectly fitting the basic model SCR0 produces substantial bias in estimates of N and hence density (Royle et al., 2013).

12.1 SHORTCOMINGS OF EUCLIDEAN DISTANCE MODELS

In the standard SCR models, encounter probability is modeled as a function of Euclidean distance. For example, using the binomial observation model (Chapt. 5), let y_{ij} be individual- and trap-specific binomial counts with sample size K and probabilities p_{ij} . The Gaussian model is

$$p_{ij} = p_0 \exp(-d_{ij}^2/(2\sigma^2)) \quad (12.1.1)$$

where $d_{ij} = ||\mathbf{x}_j - \mathbf{s}_i||$ is Euclidean distance. As usual, we will sometimes adopt the log-scale parameterization based on $\log(p_{ij}) = \alpha_0 + \alpha_1 d_{ij}^2$ where $\alpha_0 = \log(p_0)$ and $\alpha_1 = -1/(2\sigma^2)$.

The main problem with the Euclidean distance metric in this encounter probability model is that it is unaffected by habitat or landscape structure, and it implies that the space used by individuals is stationary and symmetric, which may be unreasonable assumptions for some species. By stationary we mean in the formal sense of invariance to translation. That is, the properties of an individual home range centered at some point \mathbf{s} are exactly the same as any other point say \mathbf{s}' . As an example, if the common detection model based on a bivariate normal probability distribution function is used, then the implied space usage by *all* individuals, no matter their location in space or local habitat conditions, is symmetric with circular contours of usage intensity.

In the framework of Royle et al. (2013), SCR models explicitly incorporate information about the landscape so that a unit of distance is variable depending on identified covariates, say $C(\mathbf{x})$. Thus, where an individual lives on the landscape, and the state of the surrounding landscape, will determine the character of its usage of space. In particular, they suggest distance metrics, based on least-cost path, that imply irregular, asymmet-

ric and non-stationary home ranges of individuals. As an example, Fig. 12.1 shows a typical symmetric home range (left panel), and a compressed home range (right panel) resulting from the effect of an environmental variable (center panel) on an animal's movement behavior. We might think of the environmental variable as representing an elevation gradient of a valley and so, for a species that avoids high elevation, space usage will be concentrated in flatter terrain at lower elevations and therefore producing the elliptical home range shape.

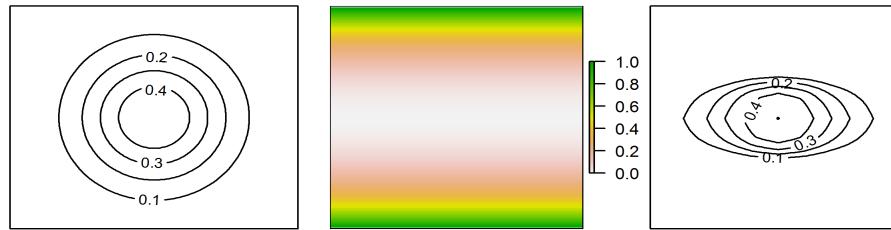


Figure 12.1. A symmetric home range (left), a habitat variable (center) such as representing an elevation gradient, and a non-symmetric home range (right) resulting from the cost imposed on movement by the habitat variable.

12.2 LEAST-COST PATH DISTANCE

We adopt a cost-weighted distance metric here which defines the effective distance between points by accumulating pixel-specific costs determined using a cost function defined by the user. The idea of cost-weighted distance to characterize animal use of landscapes is widely used in landscape ecology for modeling connectivity, movement and gene flow (Beier et al., 2008). For reasons of computational tractability we consider a discrete landscape defined by a raster of some prescribed resolution. The distance between any two points \mathbf{x} and \mathbf{x}' can be represented by a sequence of line segments connecting neighboring pixels, say $\mathbf{l}_1, \mathbf{l}_2, \dots, \mathbf{l}_m$. Then the cost-weighted distance between \mathbf{x} and \mathbf{x}' is

$$d(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{m-1} \text{cost}(\mathbf{l}_i, \mathbf{l}_{i+1}) \|\mathbf{l}_i - \mathbf{l}_{i+1}\| \quad (12.2.1)$$

where $\text{cost}(\mathbf{l}_i, \mathbf{l}_{i+1})$ is the user-defined cost to move from pixel \mathbf{l}_i to neighboring pixel \mathbf{l}_{i+1} in the sequence. Given the cost of each pixel, it is a simple matter to compute the cost-weighted distance between any two pixels, along *any* path, simply by accumulating the incremental costs weighted by distances. In the context of spatial capture-recapture models (and, more generally, landscape connectivity) we are concerned with the *minimum* cost-weighted distance, or the *least-cost path*, between any two points which we will denote by d_{lcp} , which is the sequence $\mathcal{P} = (\mathbf{l}_1, \mathbf{l}_2, \dots, \mathbf{l}_m)$ that minimizes the objective function

11190 defined by Eq. 12.2.1. That is,

$$d_{lcp}(\mathbf{x}, \mathbf{x}') = \min_{\mathcal{P}} \sum_{i=1}^{m-1} \text{cost}(\mathbf{l}_i, \mathbf{l}_{i+1}) \|\mathbf{l}_i - \mathbf{l}_{i+1}\| \quad (12.2.2)$$

11191 The least-cost path distance can be calculated in many geographic information systems
 11192 and other software packages, including the **R** package **gdistance** (van Etten, 2011) which
 11193 we use below.

11194 The key ecological aspect of least-cost path modeling is the development of models
 11195 for pixel-specific cost. A natural approach is to model cost as a function of one or more
 11196 covariates defined on every pixel of the according raster. For example, using a single
 11197 covariate $C(\mathbf{x})$ we define the cost of moving from some pixel \mathbf{x} to neighboring pixel \mathbf{x}' as

$$\log(\text{cost}(\mathbf{x}, \mathbf{x}')) = \alpha_2 \left(\frac{C(\mathbf{x}) + C(\mathbf{x}')}{2} \right) \quad (12.2.3)$$

11198 Thus, if $\alpha_2 = 0$ then substituting $\text{cost}(\mathbf{x}, \mathbf{x}') = \exp(0) = 1$ into Eq. 12.2.2 will produce the
 11199 ordinary Euclidean distance between points. Here we assume the covariate C is positive-
 11200 valued, and we constrain $\alpha_2 \geq 0$ so as to avoid negative costs. While not necessarily
 11201 problematic from a mathematical standpoint, negative costs are unrealistic biologically.

11202 The use of least-cost path models to model landscape connectivity has been around
 11203 for a long time. And, although α_2 is rarely known, conservation biologists design link-
 11204 ages that require this resistance value as input (see Beier et al., 2008, and articles cited
 11205 therein). However, formal inference (e.g., estimation) of parameters is not often done. In-
 11206 stead, in many existing applications of least-cost path analysis, the parameter α_2 is fixed
 11207 by the investigator, or based on expert opinion (Beier et al., 2008), although recently
 11208 researchers have begun to define costs based on resource selection functions¹, animal
 11209 movement (Tracy, 2006; Fortin et al., 2005), or genetic distance data (e.g., Gerlach and
 11210 Musolf (2000); Epps et al. (2007); Schwartz et al. (2009)).

11211 To formalize the use of cost-weighted distance in SCR models, we substitute Eq. 12.2.2
 11212 for Euclidean distance in the expression for encounter probability (Eq. 12.1.1) and maxi-
 11213 mize the resulting likelihood (see below). In doing so, we can directly estimate parameters
 11214 of the least-cost path model, evaluate how landscape covariate influence connectivity, and
 11215 test explicit hypotheses about these things using only individual level encounter history
 11216 data from capture-recapture studies.

11217 12.2.1 Example of Computing Cost-weighted distance

11218 As an example of the cost-weighted distance calculation consider the following landscape
 11219 comprised of 16 pixels with unit spacing identified as follows, along with the pixel-specific
 11220 cost:

	pixel ID	Cost
11221	4 8 12 16	100 1 1 1
11222	3 7 11 15	100 100 1 1

11223 ¹We address the integration of resource selection models based on telemetry data with SCR
 11224 models in Chapt. 13.

```

11224      2 6 10 14          100 100 100 1
11225      1 5   9 13         100 100   1 1

```

11226 We assume the scale is such that the distance between neighboring pixels in any cardinal
 11227 direction is 1 unit, and the distance between neighbors on a diagonal is $\sqrt{2}$ units. We
 11228 assigned low cost of 1 to “good habitat” pixels (or pixels we think of as “highly connected”
 11229 by virtue of being in good habitat) and, conversely, we assign high cost (100) to “bad
 11230 habitat”. This simple cost raster is shown in Fig. 12.2. The **R** commands for creating
 11231 this simple example are as follows (which can be run using the **R** script **SCRed** – see the
 11232 help file for that):

```

11233 > library(raster)
11234 > library(gdistance)
11235 > r <- raster(nrows=4,ncols=4)
11236 > projection(r) <- "+proj=utm +zone=12 +datum=WGS84" # Sets the projection
11237 > extent(r) <- c(.5,4.5,.5,4.5) #sets the extent of the raster
11238 > costs1 <- c(100,100,100,100,1,100,100,100,1,1,100,1,1,1,1,1)
11239 > values(r) <- matrix(costs1,4,4,byrow=FALSE) #assign the costs to the raster
11240 > par(mfrow=c(1,1))
11241 > plot(r)

```

11242 This produces Fig. 12.2.

11243 For this simple case we can easily compute the shortest cost-weighted distance between
 11244 any pixels “by eye”. For example, the shortest cost-weighted distance between pixels 5
 11245 and 9 in this example is 50.5 units: $1 * (100 + 1)/2 = 50.5$, the shortest distance between
 11246 pixels 4 and 8 is also 50.5, while the shortest cost-distance between 4 and 12 is 51.5. What
 11247 is the shortest distance between 7 and 16? Suppose an individual at pixel 7 can move
 11248 diagonal (which has distance $\sqrt{2}$) and pay $\sqrt{2}(100 + 1)/2$, and then move once to the
 11249 right to pay 1 additional unit cost, for a total of 72.4. However, if the individual instead
 11250 moved one unit to the right, to pixel 11, and then diagonally, the total cost is 51.914
 11251 which is the minimum cost-weighted distance in getting from pixel 7 to 16. These two
 11252 ways of moving from 7 to 16 have the same Euclidean distance, but different cost-weighted
 11253 distances according to our cost function.

11254 The least-cost path distances can be computed with just a few **R** commands, and
 11255 these commands can be inserted directly into the likelihood construction for an ordinary
 11256 spatial capture-recapture model. The **R** package **gdistance** calculates least-cost path us-
 11257 ing Dijkstra’s algorithm (Dijkstra, 1959) (from the **igraph** package (Csardi and Nepusz,
 11258 2006)). To compute the least-cost path, or the minimum cost-weighted distances between
 11259 every pixel and every other pixel, we make use of the helper function **transition**, which
 11260 calculates the cost of moving between neighboring pixels. It operates on the inverse-scale
 11261 (“conductance”), and so the **transitionFunction** argument is given as $1/\text{mean}(x)$. The
 11262 function **geoCorrection** modifies this object depending on the projection of the coor-
 11263 dinate system (e.g., it corrects for curvature of the earth’s surface if longitude/latitude
 11264 coordinates are used). The result is fed into the function **costDistance** to compute the
 11265 pair-wise distance matrix. For that, we define the center points of each raster, here these
 11266 are just integers on $[1, 4] \times [1, 4]$. The commands altogether are as follows:

```

11267 > tr1 <- transition(r,transitionFunction=function(x) 1/mean(x),directions=8)

```

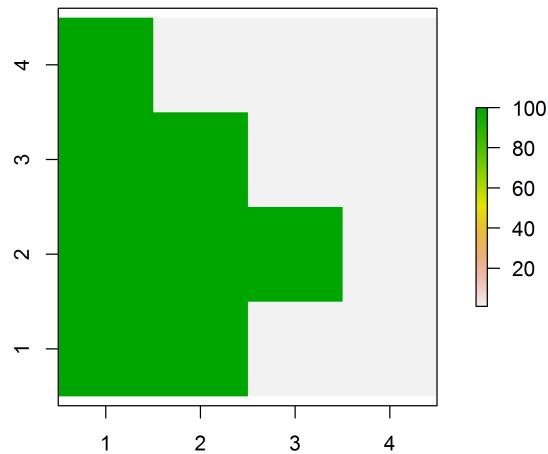


Figure 12.2. A 4×4 raster depicting a binary cost surface, with cost = 1 (white) or 100 (shaded) to represent ease of movement across a pixel.

```

11268 > tr1CorrC <- geoCorrection(tr1,type="c",multpl=FALSE,scl=FALSE)
11269 > pts <- cbind( sort(rep(1:4,4)),rep(4:1,4))
11270 > costs1 <- costDistance(tr1CorrC,pts)
11271 > outD <- as.matrix(costs1)

```

11272 Now we can look at the result and see if it makes sense to us. Here we produce the
 11273 first 5 columns of this distance matrix to illustrate a couple of examples of calculating the
 11274 minimum cost-weighted distance between points:

```

11275 > outD[1:5,1:5]
11276      1       2       3       4       5
11277 1  0.0000 100.0000 200.0000 205.2426 100.0000
11278 2 100.0000   0.0000 100.0000 200.0000 141.4214
11279 3 200.0000 100.0000   0.0000 100.0000 126.1604
11280 4 205.2426 200.0000 100.0000   0.0000 105.2426
11281 5 100.0000 141.4214 126.1604 105.2426   0.0000

```

11282 An interesting case is that between point 1 and 4. Note that simply taking the shortest
 11283 Euclidean distance, weighted by cost, produces a cost-weighted distance of 100×1 to

move from pixel 1 to pixel 2, and similarly from 2 to 3 and 3 to 4, producing a total cost-weighted distance of 300. However, the actual *least-cost path* has cost-weighted distance 205.2426. See if you can figure out the shortest path by inspection.

The key point here is that, once we can compute this distance matrix, we can use it as the distance matrix in computing the encounter probability between activity centers and traps, and we can use our existing MLE technology (Chapt. 6) to fit models that are based on ecological distance.

12.3 SIMULATING SCR DATA USING ECOLOGICAL DISTANCE

Royle et al. (2013) simulated capture-recapture data such that landscape connectivity was governed by a cost function having a single covariate, and they considered two hypothetical covariate landscapes (Fig. 12.3). The landscape here is a 20×20 pixel raster, with extent = $[0.5, 4.5] \times [0.5, 4.5]$. For example, think of each pixel as representing, say, a 1×1 km grid cell with something like “percent developed” or “trail/road density” representing the covariate. For sampling by capture-recapture, imagine that 16 camera traps are established at the integer coordinates $(1, 1), (1, 2), \dots, (4, 4)$. The two covariates were constructed as follows (see `?make.EDcovariates` for the R commands): First is an increasing trend from the NW to the SE (“systematic covariate”), where $C(\mathbf{x})$ is defined as $C(\mathbf{x}) = \text{row}(\mathbf{x}) + \text{col}(\mathbf{x})$ and $\text{row}(\mathbf{x})$ and $\text{col}(\mathbf{x})$ are just the row and column, respectively, of the raster. This might mimic something related to distance from an urban area or a gradient in habitat quality due to land use, or environmental conditions such as temperature or precipitation gradients. In the second case, the covariate was generated using spatially correlated noise to emulate a typical patchy habitat covariate (“patchy covariate”) such as tree or understory density.

For both covariates we use a cost function in which transitions from pixel \mathbf{x} to \mathbf{x}' is given by:

$$\log(\text{cost}(\mathbf{x}, \mathbf{x}')) = \alpha_2 \left(\frac{C(\mathbf{x}) + C(\mathbf{x}')}{2} \right)$$

where $\alpha_2 = 1$ for simulating the observed data. Remember that with $\alpha_2 = 0$ the model reduces to one in which the cost of moving across each pixel is constant, and therefore Euclidean distance is operative. In the left panel of Fig. 12.3, a sample realization of $N = 100$ activity centers is shown. While encounter probability is assumed to be related to landscape connectivity according to the single-variable cost function, individual activity centers are assumed to be uniformly distributed, although we can modify this assumption (See Sec. 12.8 below).

When distance is defined by the cost-weighted distance metric given by Eq. 12.2.2 then individual space-usage varies spatially in response to the landscape covariate(s) used in the distance metric. As a consequence, home range contours are no longer circular, as in SCR models based on Euclidean distance. For example, using one of the covariates we use in our simulation study below (Fig. 12.3, right panel) with a Gaussian encounter probability model but having distance metric defined by Eq. 12.2.2, produces home ranges such as those shown in Fig. 12.4.

To simulate data, we have to load the `scrbook` package and call the function `make.EDcovariates` to generate our raster covariates (see the help file for how that is done). We process the covariate into a least-cost path distance matrix, and then simulate observed encounter

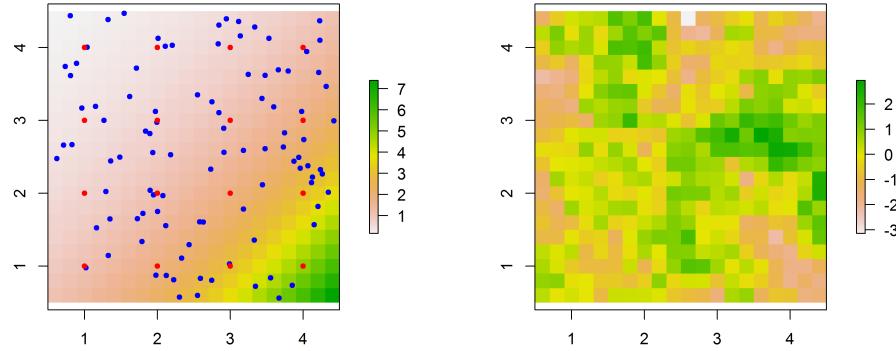


Figure 12.3. Two covariates (defined on a 20×20 grid) used in simulations. Left panel shows a covariate with systematic structure meant to mimic distance from some feature, and the right panel shows a “patchy” covariate. A hypothetical realization of $N = 100$ activity centers (blue dots) is superimposed on the left figure, along with 16 trap locations.

11325 data using standard methods which we have used many times previously in this book.
 11326 The complete set of **R** commands is:

```

11327 ### Grab a covariate
11328 > library(scrbook)
11329 > set.seed(2013)
11330 > out <- make.EDcovariates()
11331 > covariate <- out$covariate.patchy
11332
11333 ### prescribe some settings
11334 > N <- 200
11335 > alpha0 <- -2
11336 > sigma <- .5
11337 > alpha1 <- 1/(2*sigma*sigma)
11338 > alpha2 <-1
11339 > K <- 5
11340 > S <- cbind(runif(N,.5,4.5),runif(N,.5,4.5))
11341
11342 # make up some trap locations
11343 > xg <- seq(1,4,1); yg<-4:1
11344 > traplocs <- cbind( sort(rep(xg,4)),rep(yg,4))
11345 > points(traplocs,pch=20,col="red")
11346 > ntraps <- nrow(traplocs)
```

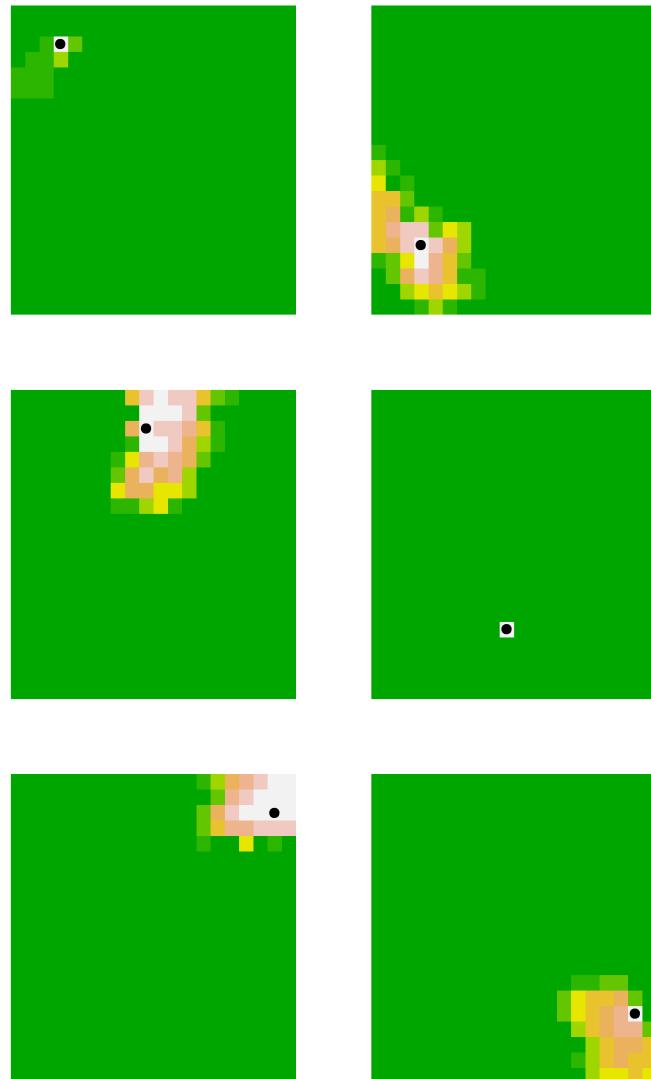


Figure 12.4. Typical home ranges for 6 individuals based on the cost surface shown in the right panel of Fig. 12.3 with $\alpha_2 = 1$. The black dot indicates the home range center and the pixels around each home range center are shaded according to the probability of encounter, if a trap were located in that pixel.

```

11347
11348 ### make a raster and fill it up with the "cost"
11349 > r <- raster(nrows=20,ncols=20)
11350 > projection(r) <- "+proj=utm +zone=12 +datum=WGS84"
11351 > extent(r) <- c(.5,4.5,.5,4.5)
11352 > cost <- exp(alpha2*covariate)
11353
11354 ### compute least-cost path distance
11355 > tr1 <- transition(cost,transitionFunction=function(x) 1/mean(x),directions=8)
11356 > tr1CorrC <- geoCorrection(tr1,type="c",multpl=FALSE,scl=FALSE)
11357 > D <- costDistance(tr1CorrC,S,traplocs)
11358 > probcap <- plogis(alpha0)*exp(-alpha1*D*D)
11359
11360 # now generate the encounters of every individual in every trap
11361 # discard uncaptured individuals
11362 > Y <- matrix(NA,nrow=N,ncol=ntraps)
11363 > for(i in 1:nrow(Y)){
11364 +   Y[i,] <- rbinom(ntraps,K,probcap[i,])
11365 + }
11366 > Y <- Y[apply(Y,1,sum)>0,]

```

12.4 LIKELIHOOD ANALYSIS OF ECOLOGICAL DISTANCE MODELS

11367 Throughout much of this book we rely on Bayesian analysis by MCMC mostly using
 11368 **BUGS**, but sometimes (as in Chapt. 17) developing our own implementations. However,
 11369 occasionally we prefer to use likelihood estimation, such as when we can compare a set
 11370 of models directly by likelihood either to do a direct hypothesis test of a parameter, or
 11371 to tabulate a bunch of AIC values. For the class of models that use least-cost path, we
 11372 also prefer likelihood methods not because they have any conceptual or methodological
 11373 benefit, but simply because they are more computationally efficient to implement (Royle
 11374 et al., 2013).

11375 There are no technical considerations in adapting our formulation of maximum likeli-
 11376 hood estimation (Borchers and Efford, 2008) from Chapt. 6 for the class of models based
 11377 on least-cost path (see the appendix in Royle et al. (2013) for complete details). Likeli-
 11378 hood analysis is really just a straightforward adaptation in which we replace the Euclidean
 11379 distance with least-cost path. Consider the Bernoulli model in which the individual- and
 11380 trap-specific observations have a binomial distribution conditional on the latent variable
 11381 \mathbf{s}_i :

$$y_{ij} | \mathbf{s}_i \sim \text{Binomial}(K, p_{\boldsymbol{\alpha}}(d_{lcp}(\mathbf{x}_j, \mathbf{s}_i; \boldsymbol{\alpha}_2); \boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1)) \quad (12.4.1)$$

11382 where we have indicated the dependence of p on the parameters $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2)$, and also
 11383 d_{lcp} which itself depends on $\boldsymbol{\alpha}_2$, and the latent variable \mathbf{s}_i . We note that the only difference
 11384 between likelihood analysis of this model and the standard Bernoulli model, is the use of
 11385 d_{lcp} here. For the random effect we have $\mathbf{s}_i \sim \text{Uniform}(\mathcal{S})$, we can easily compute the
 11386 integrated (marginal) likelihood of an encounter history. The likelihood is given in the
 11387 **scrbook** package as the function **intlik3ed**. The help file provides an example of its usage
 11388 and for simulating data. To use this function the cost covariate $C(\mathbf{x})$ has to be of class

Table 12.1. Summary output of fitting models based on Euclidean and least-cost path distance to simulated data using the `intlik3ed` function (see `?intlik3ed`). Data were simulated based on the least-cost path model using the “patchy” covariate shown in Fig. 12.3.

Distance metric	-loglik	α_0	α_1	$\log(n_0)$	α_2
True value		-2	2	4.644	1
Euclidean	133.495	-1.885	1.247	3.549	—
Least-cost path (truth)	70.119	-1.780	2.471	4.459	0.046

11389 RasterLayer which requires packages `sp` and `raster` to manipulate.

11390 12.4.1 Example of SCR with least-cost path

11391 Now we use the **R** function `nlm` along with our `intlik3ed` function to obtain the MLEs
 11392 of the model parameters for the data simulated in Sec. 12.3. We’ll do that for both the
 11393 standard Euclidean distance and then for the ecological distance based on the “patchy”
 11394 covariate using the following commands:

```
11395 > frog1<-nlm(intlik3ed,c(alpha0,alpha1,3)),hessian=TRUE,y=Y,K=K,X=traplocs,  

11396   distmet="euclid",covariate=covariate,alpha2=1)  

11397  

11398 > frog2<-nlm(intlik3ed,c(alpha0,alpha1,3,-.3),hessian=TRUE,y=Y,K=K,X=traplocs,  

11399   distmet="ecol",covariate=covariate,alpha2=NA)
```

11400 The summary output for the two model fits is shown in Table 12.1. The model based
 11401 on least-cost path (the data generating model) appears to be much preferred in terms
 11402 of negative log-likelihood. The output parameter order is $\alpha_0, \alpha_1, \log(n_0)$, and $\log(\alpha_2)$
 11403 (remember, we want to keep α_2 positive, so its logarithm is estimated). The data gener-
 11404 ating parameter values were $\alpha_0 = -2$, $\alpha_1 = 2$ and $\log(\alpha_2) = 0$. The simulated sampling
 11405 produced a sample of 96 individuals and so the number of individuals not captured is
 11406 $n_0 = 104$, and $\log(n_0) = 4.64$. We see that the MLEs of the least-cost path model are
 11407 pretty close whereas they are not so close under the misspecified model based on Euclidean
 11408 distance.

12.5 BAYESIAN ANALYSIS

11409 While implementation of these ecological distance SCR models is reasonably straightfor-
 11410 ward, the model cannot be fitted in the **BUGS** engines because least-cost path distance
 11411 cannot be computed. It would be possible to fit the models in **BUGS** if the parameter α_2
 11412 was fixed. In that case, one could compute the least-cost distance matrix ahead of time
 11413 and reference the required elements for a given `s`. Alternatively, it would be possible to
 11414 write a custom MCMC routine using the methods we present in Chapt. 17, although we
 11415 have not yet developed our own MCMC implementation of SCR models with ecological
 11416 distance metrics.

12.6 SIMULATION EVALUATION OF THE MLE

11417 Royle et al. (2013) carried-out a limited simulation study to evaluate the general statistical
 11418 performance of the density estimator under this new model, the effect of mis-specifying
 11419 the model with a normal Euclidean distance metric, and evaluate the general bias and
 11420 precision properties of the MLE using the systematic and patchy landscapes shown in Fig.
 11421 12.3. Their results showed extreme bias in estimates of N when the misspecified Euclidean
 11422 distance is used, and only negligible small-sample bias of 3-5% in the MLE of N using
 11423 the least-cost distance which becomes negligible as the expected sample size increases
 11424 (either due to increasing K , or larger population sizes). The performance of estimating
 11425 the other parameters, including the cost parameter α_2 mirrors the results for estimating
 11426 N . We reproduce a subset of the results from Royle et al. (2013) in Table 12.6 in order
 11427 to highlight some key points.

Table 12.2. Simulation results for estimating population size N for a prescribed state-space with $N = 100$ or $N = 200$ and various levels of replication (K) using the “patchy” landscape shown in Fig. 12.3. For each simulated data set, the SCR model was fitted by maximum likelihood with standard Euclidean distance (“euclid”), or least-cost path (“lcp”), which was the true data-generating model. The summary statistics of the sampling distribution reported are the mean, standard deviation (“SD”) and quantiles (0.025, 0.50, 0.975).

		N=100				
		mean	SD	0.025	0.50	0.975
$K = 3$						
euclid		78.68	18.12	49.40	76.34	125.47
lcp		110.96	28.65	69.55	106.98	181.84
$K = 5$						
euclid		77.85	11.55	59.17	77.44	101.14
lcp		104.44	15.79	78.38	101.47	139.55
$K = 10$						
euclid		78.01	5.26	68.00	77.96	87.81
lcp		100.42	7.56	86.72	100.34	115.47
		N=200				
$K = 3$						
euclid		154.34	33.74	107.00	146.34	221.43
lcp		208.77	49.29	141.68	197.89	325.77
$K = 5$						
euclid		153.39	15.57	129.31	149.54	185.38
lcp		200.91	20.78	164.42	200.47	246.46
$K = 10$						
euclid		156.27	8.51	142.17	156.05	174.55
lcp		198.45	11.44	180.06	198.04	219.52

12.7 DISTANCE IN AN IRREGULAR PATCH

11428 We provide another illustration of how to employ ecological distance calculations in SCR
 11429 models. This example is meant to mimic a situation where we have something like a hard
 11430 habitat boundary such as a habitat corridor or park unit or some other block of relatively
 11431 homogeneous good-quality habitat for some species. This particular system (shown in
 11432 Fig. 12.5) could be habitat surrounded by a suburban wasteland of McDonuts and Beer-
 11433 Marts, much less hospitable habitat for most species. For our purposes, we suppose that
 11434 individuals live within the buffered “f-shaped” region, although we could also imagine the
 11435 negative of the situation in which individuals live outside of the region, so that the polygon
 11436 represents a barrier (a lake) or bad habitat (an urban area) or similar. We describe the
 11437 steps for creating this landscape shortly, so that you can use a similar process to generate
 11438 more relevant landscapes for your own problems.

11439 In this case we’re not going to estimate any parameters of the cost function (though
 11440 you could adapt the analyses of the previous sections to do that) but instead we’re going
 11441 to use ecological distance ideas only to constrain movement within (or to avoid) landscape
 11442 features. Note that, normally, distance “as the crow flies” would not be suitable for
 11443 irregular habitat patches such as that shown in Fig. 12.5.

11444 12.7.1 Basic Geographic Analysis in R

11445 In practical applications our landscape will contain polygons which delineate good or bad
 11446 habitat or other important characteristics of the landscape. These might exist as GIS
 11447 shapefiles or merely as a text file with coordinates defining polygon boundaries. To work
 11448 with polygons in the context of SCR models we need to create a raster, overlay the polygon
 11449 and assign values to each pixel depending on whether pixels are in the polygon or not,
 11450 or how far they are from polygon boundaries. These operations are relatively easy to do
 11451 within a GIS system but we need to be able to do them in **R** in order to compute the
 11452 least-cost paths needed in the likelihood evaluation. Some additional geographic analyses
 11453 have been discussed in Sec. 17.7 where we talked about reading in the shapefile and doing
 11454 SCR analyses with it.

11455 Often we will have GIS shapefiles that define polygons but, here, we create a set of
 11456 polygons by buffering and joining some line segments. In the **R** package **scrbook**, we
 11457 provide a function **make.seg** which allows you to make such line segments given a specific
 11458 trap region. To use **make.seg** we first create a plot region and then call **make.seg** which
 11459 has a single argument being the number of points used to define the line segment. The user
 11460 will click on the visual display until the required number of points has been obtained by
 11461 **make.seg**. In the following set of commands we generate two line segments, **l1** consisting
 11462 of 9 points and **l2** consisting of 5 points, and these reside in a geographic region enclosed
 11463 by $[0, 10] \times [0, 10]$:

```
11464 > library(scrbook)
11465 > library(sp)
11466 > plot(NULL, xlim=c(0,10), ylim=c(0,10))
11467 > l1 <- make.seg(9)
11468 > plot(l1)
11469 > l2 <- make.seg(5)
```

```
11470 > plot(11)
11471 > lines(12)
```

11472 We used this function as above to create a habitat corridor composed of line segments
 11473 of class **SpatialLines** from the **R** package **sp**. The corridor can be loaded from **scrbook**
 11474 by typing the command **data(fakecorridor)**. This data list has 2 line files in it (11 and
 11475 12) and a trap locations file (**traps**). We use some functions from the **R** packages **sp** and
 11476 **rgeos** to join and buffer (by 0.5 units) the two segments. The commands are as follows
 11477 and the result is shown in Fig. 12.5.

```
11478 > data(fakecorridor)
11479 > library(sp)
11480 > library(rgeos)
11481
11482 > buffer <- 0.5
11483 > par(mfrow=c(1,1))
11484 > aa <- gUnion(11,12)
11485 > plot(gBuffer(aa, width=buffer), xlim=c(0,10), ylim=c(0,10))
11486 > pg <- gBuffer(aa, width=buffer)
11487 > pg.coords <- pg@polygons[[1]]@Polygons[[1]]@coords
11488
11489 > xg <- seq(0,10,,40)
11490 > yg <- seq(10,0,,40)
11491
11492 > delta <- mean(diff(xg))
11493 > pts <- cbind(sort(rep(xg,40)),rep(yg,40))
11494 > points(pts, pch=20, cex=.5)
11495
11496 > in pts <- point.in.polygon(pts[,1],pts[,2],pg.coords[,1],pg.coords[,2])
11497 > points(pts[in.pts==1,],pch=20,col="red")
```

11498 In this example, we're not going to estimate parameters of the cost function. Instead,
 11499 the point is to compute ordinary Euclidean distance but restricted by the boundaries of
 11500 the corridor (or patch geometry in general) and thus not distance "as the crow flies." To
 11501 do this, we imagine that animals will tend to severely avoid leaving the buffered habitat
 11502 zone. Therefore, we assign **cost** = 1 if a pixel is within the buffer, and **cost** = 10000 if a
 11503 pixel is outside of a buffer. Therefore the cost to move to a neighboring pixel outside of the
 11504 buffered area is 5000.5 compared to the cost of 1 to move to a neighboring pixel inside the
 11505 buffer. With this cost specification, we can compute the least-cost path distance matrix
 11506 one time and modify our likelihood code to accept the distance matrix as input. We give
 11507 that likelihood in the package **scrbook** as the function **intlik3edv2**. We note also that
 11508 this function accepts a habitat mask in the form of a vector of 0's and 1's that define any
 11509 potential state-space restrictions. i.e., 1 if the pixel is an element of the state-space and 0
 11510 if it is not, and so additional modifications to the geometry of the region could be made.
 11511 However, in the analysis of this simulated data set, we define the state-space to be the
 11512 buffered corridor system. Here we simulate a population of $N = 200$ individuals in the
 11513 corridor system and so we restrict our state-space accordingly for purposes of fitting the

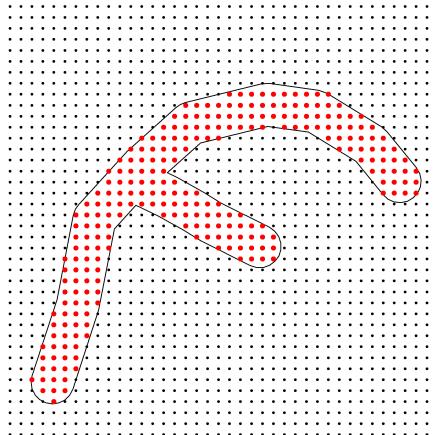


Figure 12.5. A fake wildlife corridor or reserve. The boundary outlines a polygon of suitable habitat surrounded by suburban development.

model. However we encourage you to refit the model without the state-space restriction (for fitting the model only) and then compare the results. The code for doing all of this is in the help file for `intlik3edv2`, which contains the likelihood function and sample **R** script (`?intlik3edv2`).

```

11518  ### Define the cost structure
11519  > cost <- rep(NA,nrow(pts))
11520  > cost[in pts==1]<-1      # low cost to move among pixels but not 0
11521  > cost[in pts!=1]<-10000 # high cost
11522
11523  ### Stuff costs into a raster
11524  > library("raster")
11525  > r <- raster(nrows=40,ncols=40)
11526  > projection(r) <- "+proj=utm +zone=12 +datum=WGS84"
11527  > extent(r) <- c(0-delta/2,10+delta/2,0-delta/2,10+delta/2)
11528  > values(r) <- matrix(cost,40,40,byrow=FALSE)
11529
11530  # check what it looks like
11531  > plot(r)

```

```

11532 > points(pts,pch=20,cex=.4)
11533
11534 # compute ecological distances:
11535 > library("gdistance")
11536 > tr1 <- transition(r,transitionFunction=function(x) 1/mean(x),directions=8)
11537 > tr1CorrC <- geoCorrection(tr1,type="c",multpl=FALSE,scl=FALSE)
11538 > costs1 <- costDistance(tr1CorrC,pts)
11539 > outD <- as.matrix(costs1)

```

11540 In the next block of code we simulate some data and then fit a model to the simulated
 11541 data. Note that the object `traps` is loaded with `data(fakecorridor)` along with the data
 11542 which define the f-shaped patch in Fig. 12.5:

```

11543 > library(scrbook)
11544 > traplocs <- traps$loc
11545 > trap.id <- traps$locid
11546 > ntraps <- nrow(traplocs)
11547
11548 > set.seed(2013)
11549 > N <- 200
11550 > S.possible <- (1:nrow(pts))[in pts==1]
11551 > S.id <- sample(S.possible,N,replace=TRUE)
11552 > S <- pts[S.id,]
11553
11554 > Dtraps <- outD[trap.id,]
11555 > Deuclid <- e2dist(pts[trap.id,],pts)
11556
11557 > alpha0 <- -1.5
11558 > sigma <- 1.5
11559 > alpha1 <- 1/(2*sigma*sigma)
11560 > K <- 10
11561
11562 > probcap <- plogis(alpha0)*exp(-alpha1*D*D)
11563 > Y <- matrix(NA,nrow=N,ncol=ntraps)
11564 > for(i in 1:nrow(Y)){
11565 +   Y[i,] <- rbinom(ntraps,K,probcap[i,])
11566 > }
11567 > Y <- Y[apply(Y,1,sum)>0,]
11568
11569 > frog1 <- nlm(intlik3edv2,c(-2.5,2,log(4)),hessian=TRUE,y=Y,K=K,X=traplocs,
11570 +           S=pts,D=Dtraps,inpoly=in.pts)
11571 > frog2 <- nlm(intlik3edv2,c(-2.5,2,log(4)),hessian=TRUE,y=Y,K=K,X=traplocs,
11572 +           S=pts,D=Deuclid,inpoly=in.pts)

```

11573 These two models fit, with the correctly specified ecological distance, constrained by
 11574 the patch boundaries, and that with the ordinary (misspecified) Euclidean distance are
 11575 summarized in Table 12.3. We find little difference between the two models. In particu-
 11576 lar, 150 individuals were captured and so truth (the number of uncaptured individuals)

Table 12.3. Summary output of fitting models to simulated data in which movement is restricted by the habitat corridor shown in Fig. 12.5. The two models fitted were those based on distance constrained by the corridor boundary (“constrained”) and a misspecified model based on ordinary Euclidean distance which is “as the crow flies”, and cuts through some boundaries. See `?fakecorridor` for the **R** commands to fit these models.

Distance	neg. LL	α_0	α_1	$\log(n_0)$
constrained	-21.892	-1.338	0.332	4.353
Euclidean	-21.128	-1.307	0.382	4.212

is $\log(n_0) = 3.9$. The correct model produces only a slightly more accurate estimate, and it is favored by only 0.7 negative log-likelihood units. Therefore, for this single instance, the results are not too different. This is primarily because the distance between individuals, and traps that they are likely to be captured in, is well-approximated by Euclidean distance.

12.8 ECOLOGICAL DISTANCE AND DENSITY COVARIATES

Habitat characteristics that affect spatial variation in density can also affect home range size and movement behavior. For example, a species that occurs at high density in a forest may be reluctant to venture from a forest patch into an adjacent field. Thus, even if a trap placed in a field is located very close to an animal’s activity center, the probability of capture may be very low. In this case, forest cover is a covariate of both density and encounter probability, and we could model it as such by combining the methods described in this chapter with those described in Chapt. 11.

To demonstrate, we continue with our analysis of the data shown in Fig 11.4.2. Once again, we suppose that density increases with canopy height, but this time, we also allow home range size to decrease as density increases. This commonly-observed phenomenon can be explained by numerous factors such as intra-specific competition (Sillett et al., 2004) or optimal foraging behavior (Tufto et al., 1996; Said and Servanty, 2005).

A question that arises is: Is it possible to estimate the effect of the covariate on density (β_1) and α_2 using standard SCR data? In other words, can we model spatial variation in density and connectivity at the same time, using standard SCR data? Currently, it is not possible to model least-cost distance using **JAGS** or **secr**, so we wrote our own function, **scrDED**, to fit the model using maximum likelihood. An example analysis is provided on the help page for the function in our **R** package **scrbook**. We briefly note here that the function requires the capture history data, the trap locations, and the raster data formatted using the **raster** package (Hijmans and van Etten, 2012). The linear model for the intensity parameter $\mu(s, \beta)$ and the least-cost distance function $lcd(\theta)$ are specified using **R**’s formula interface. A simple function call is

```
11604 > fm <- scrDED(y, traplocs=X, den.formula=~elev, dist.formula=~elev,
11605 + rasters=elev.raster)
```

To assess the possibility of estimating both β and α_2 , we conducted a small simulation study, generating 500 datasets from the model with both parameters set to 1, which

11608 corresponds to the conditions described above. The results indicate that it is possible to
 11609 estimate both parameters (Fig 12.6).

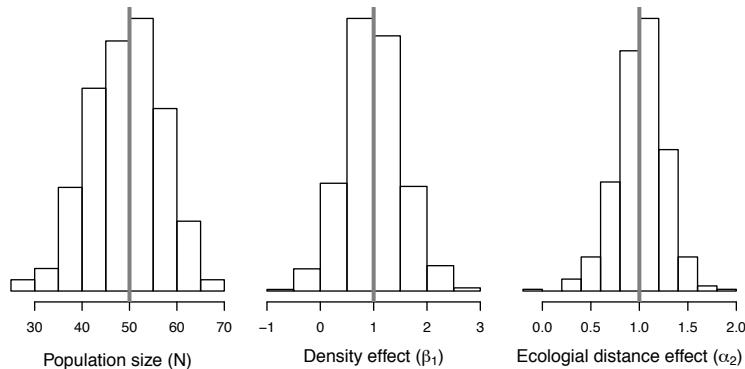


Figure 12.6. Histograms of parameter estimates from 500 simulations under the model in which both density and ecological distance are affected by the same covariate, canopy height. The vertical lines indicate the data-generating value.

12.9 SUMMARY AND OUTLOOK

11610 Almost all published applications of SCR models to date have been based on models for
 11611 the encounter probability that are functions of the Euclidean distance between individual
 11612 activity centers and traps. The obvious limitations of such models are that Euclidean
 11613 distance is unaffected by landscape or habitat structure and implies stationary, isotropic
 11614 and symmetrical home ranges. These are standard criticisms of the basic SCR model
 11615 which we have seen many times in referee reports, or heard in discussions with colleagues.
 11616 However, this should not be seen as criticism that is inherent to the basic conceptual
 11617 formulation of SCR models because, as we have shown here, one can modify the Euclidean
 11618 distance metric to accommodate more realistic formulations of distance that allow for
 11619 inference to be made about landscape connectivity, and model “distance” as a function
 11620 of local habitat characteristics. As such, effective distance between individual home range
 11621 centers and traps varies depending on the local landscape.

11622 How animals use space and therefore how distance to a trap is perceived by individuals
 11623 is not something that can ever be known. We can only ever conjure up models to
 11624 describe this phenomenon and fit those models to limited data on a sample of individuals
 11625 during a limited amount of time. Here we have shown that there is hope to estimate con-
 11626 nectivity parameters that describe how animals use space, from capture-recapture data
 11627 alone, thereby allowing for irregular home range geometry that is influenced by landscape
 11628 structure.

11629 In the presence of functional landscape connectivity, misspecification of the model by
 11630 an ordinary SCR model based on Euclidean distance produces biased estimates of model

parameters (Royle et al., 2013). This is expected because the effect is similar to failing to model heterogeneity, i.e., if we mis-specify “model M_h ” (Otis et al., 1978) with “model M_0 ” (Otis et al., 1978) then we will expect to under-estimate N . So the effect of mis-specifying the ecological distance metric with a standard homogeneous Euclidean distance has the same effect. In our view, this bias is not really the most important reason to consider models of ecological distance. Rather, inference about the structure of ecological distance is fundamental to many problems in applied and theoretical ecology related to modeling landscape connectivity, corridor and reserve design, population viability analysis, gene flow, and other phenomena. Models based on least-cost path distance allow investigators to evaluate landscape factors that influence movement of individuals over the landscape from non-invasively collected capture-recapture data. Therefore SCR models based on ecological distance metrics might aid in understanding aspects of space usage and movement in animal populations and, ultimately, in addressing conservation-related problems such as corridor design.

11645
11646

13

11647
11648
11649

INTEGRATING RESOURCE SELECTION WITH SPATIAL CAPTURE-RECAPTURE MODELS

11650 In Chapt. 5 we briefly discussed the notion of how SCR encounter probability models relate
11651 to models of space usage. When using symmetric and stationary encounter probability
11652 models, SCR models imply that space usage is a decreasing function of distance from an
11653 individual's home range center. This is not a very realistic model in most applications.
11654 In this chapter, we extend SCR models to incorporate models of resource selection, such
11655 as when one or more explicit landscape covariates are available which the investigator
11656 believes might affect how individual animals use space within their home range. This is
11657 what Johnson (1980) called *third-order* selection – a term emphasizing the hierarchical
11658 nature of resource selection.

11659 An appealing feature of SCR models is that they provide a mechanism for modeling
11660 multiple levels of the resource selection hierarchy. For instance, Johnson (1980) de-
11661 fined *second-order* selection as the process determining the location of home ranges on
11662 a landscape, which is exactly the process being modeled using the methods presented in
11663 Chapt. 11. Thus, SCR provides a way of studying the density and distribution of home
11664 range centers, while at the same time allowing for inferences about the use of resources
11665 within home ranges.

11666 Our treatment follows Royle et al. (2012a) who integrated a standard family of resource
11667 selection models based on auxiliary telemetry data into the capture-recapture model for en-
11668 counter probability. They argued that SCR models and resource selection models (Manly
11669 et al., 2002) are based on the same basic underlying model of space usage. The important
11670 distinction between SCR and RSF studies is that, in SCR studies, encounter of individuals
11671 is imperfect (i.e., “ $p < 1$ ”) whereas, with RSF data obtained by telemetry, encounter is
11672 perfect. SCR and telemetry data can therefore be combined in the same likelihood by
11673 formally recognizing this distinction in the model.

11674 There are two important motives for considering a formal integration of RSF models

11675 with capture-recapture. The first is to integrate models of resource use by individuals
11676 with models of population size or density. There is relatively little in the literature on this
11677 topic, although Boyce and McDonald (1999) describe a procedure where (an estimate of)
11678 population size is used to scale resource selection functions to produce a population density
11679 surface. The second reason is because this allows for the integration of auxiliary data from
11680 telemetry studies with capture-recapture data. Telemetry studies are extremely common
11681 in animal ecology for studying movement and resource selection, and capture-recapture
11682 studies frequently involve a simultaneous telemetry component. Telemetry data has been
11683 widely used in conjunction with capture-recapture data using standard non-spatial models.
11684 For example, White and Shenk (2001) and Ivan (2012) suggested using telemetry data to
11685 estimate the probability that an individual is exposed to capture-recapture sampling.
11686 However, their estimator requires that individuals are telemetry-tagged in proportion to
11687 this unknown quantity, which seems impossible to achieve in many studies. In addition,
11688 they do not directly integrate the telemetry data with the capture-recapture model so that
11689 common parameters are jointly estimated. Sollmann et al. (in revision) and Sollmann et al.
11690 (2013) used telemetry data to directly inform the parameter σ from the bivariate normal
11691 SCR model in order to improve estimates of density, although these models do not include
11692 an explicit resource selection component.

11693 Formal integration of capture-recapture with telemetry data for the purposes of mod-
11694 eling resource selection has a number of immediate benefits. For one, telemetry data
11695 provide direct information about σ (Sollmann et al., 2013, in revision). As a result, this
11696 leads to improved estimates of model parameters, and also has design consequences (see
11697 Sec. 10.7). In addition, active resource selection by animals induces a type of heterogene-
11698 ity in encounter probability, which is misspecified by standard SCR encounter probability
11699 models. Animals that use more space due to the configuration of habitat or landscape
11700 features, stand to be exposed to more traps than animals that use less space. As a result,
11701 estimates of population size or density under models that do not account for resource
11702 selection can be biased (Royle et al., 2012a). Finally, because the resource selection model
11703 translates directly to a model for encounter probability for spatial capture-recapture data,
11704 the implication of this is that it allows us to estimate resource selection model parameters
11705 directly from SCR data, i.e., *absent* telemetry data. This fact should broaden the practical
11706 relevance of spatial capture-recapture not just for estimating density, but also for directly
11707 studying movement and resource selection.

13.1 A MODEL OF SPACE USAGE

11708 Assume that the landscape is defined in terms of a discrete raster of one or more covariates,
11709 having the same dimensions and extent. Let $\mathbf{x}_1, \dots, \mathbf{x}_G$ identify the center coordinates
11710 of G pixels that define a landscape, organized in the matrix $\mathbf{X}_{G \times 2}$. Let $C(\mathbf{x})$ denote a
11711 covariate defined for every pixel \mathbf{x} . We suppose that individual members of a population
11712 wander around space in some manner related to the covariate $C(\mathbf{x})$.

11713 As a biological matter, use is the outcome of individuals moving around their home
11714 range (Hooten et al., 2010), i.e., where an individual is at any point in time is the result of
11715 some movement process. However, to understand space usage, it is not necessary to enter-
11716 tain explicit models of movement, just to observe the outcomes, and so we don't elaborate
11717 further on what could be sensible or useful models of movement, but we imagine existing

methods of hierarchical or state-space models are suitable for this purpose (Ovaskainen, 2004; Jonsen et al., 2005; Forester et al., 2007; Ovaskainen et al., 2008; Patterson et al., 2008; Hooten et al., 2010; McClintock et al., 2012). We consider explicit movement models in the context of SCR models later chapters of this book (Chaps. 15 and 16). Here we adopt more of a phenomenological formulation of space usage as follows: If an individual appears in pixel \mathbf{x} at some instant, this is defined as a decision to “use” pixel \mathbf{x} . Thus, over any prescribed time interval, the percentage of time an individual spends in each pixel is theoretically knowable. Or, if we sample some number of points during that interval, say R , then the frequency of use decisions is, conceivably, observable by some omnipotent accounting mechanism (e.g., telemetry that doesn’t malfunction). In this case, let m_{ij} be the *true* use frequency of pixel j by individual i – i.e., the number of times individual i used pixel j . We assume the vector of use frequencies $\mathbf{m}_i = (m_{i1}, \dots, m_{iG})$ has a multinomial distribution:

$$\mathbf{m}_i \sim \text{Multinomial}(R, \boldsymbol{\pi}_i)$$

where $R = \sum_j m_{ij}$ is the total number of “use decisions” made by individual i and

$$\pi_{ij} = \frac{\exp(\alpha_2 C(\mathbf{x}_j))}{\sum_x \exp(\alpha_2 C(\mathbf{x}))}$$

for each $j = 1, 2, \dots, G$ pixels. This is a standard RSF model (Manly et al., 2002) used to model telemetry data. In particular, this is “protocol A” of (Manly et al., 2002) where all available landscape pixels are censused (i.e., known without error), and used pixels are sampled randomly for each individual. The parameter α_2 is the effect of the landscape covariate $C(\mathbf{x})$ on the relative probability of use. Thus, if α_2 is positive, the relative probability of use increases as the covariate increases.

In practice, we don’t get to observe m_{ij} for all individuals but, instead, only for a small subset which we capture and telemeter. For the telemetered individuals, we assume they use resources according to the same RSF model as the population as a whole. To extend this model to make it more realistic, and consistent with the formulation of SCR models, let \mathbf{s} denote the center of an individual’s home range and let $d_{ij} = \|\mathbf{x}_j - \mathbf{s}_i\|$ be the distance from the home range center of individual i , \mathbf{s}_i , to pixel j , \mathbf{x}_j . We modify the space usage model to accommodate that space use will be concentrated around an individual’s home range center:

$$\pi_{ij} = \frac{\exp(-\alpha_1 d_{ij}^2 + \alpha_2 C(\mathbf{x}_j))}{\sum_x \exp(-\alpha_1 d_{ij}^2 + \alpha_2 C(\mathbf{x}))} \quad (13.1.1)$$

The parameters α_1 , α_2 and the activity centers \mathbf{s} can be estimated directly from telemetry data, using standard likelihood methods based on the multinomial likelihood (Johnson et al., 2008). Normally this model is expressed in terms of the scale parameter σ , $\alpha_1 = 1/(2\sigma^2)$, and the multinomial model Eq. 13.1.1 can be understood as a compound model of space usage governed by distance-based “availability” according to a Gaussian kernel, and also “use”, conditional on availability (Johnson et al., 2008; Forester et al., 2009). In other words, the model suggests a kind of distance-based availability in which a pixel is less available to an individual if it is located further away from \mathbf{s}_i .

Eq. 13.1.1 resembles standard SCR encounter probability models that we have used previously, but here the model includes an additional covariate $C(\mathbf{x})$ (see Chapt. 9). In

particular, under this model for space usage or resource selection, if we have no covariates at all, or if $\alpha_2 = 0$, then the probabilities π_{ij} are directly proportional to the SCR model for encounter probability, *if we have a trap in every pixel*. Therefore, setting $\alpha_2 = 0$, the probability of use for pixel j is:

$$p_{ij} \propto \exp(-\alpha_1 d_{ij}^2).$$

Clearly, whatever function of distance we use in the RSF model implies an equivalent model of space usage (Sec. 5.4) as an SCR model for encounter probability. In particular, for whatever model we choose for p_{ij} in an ordinary SCR model, we can modify the distance component in the RSF function in Eq. 13.1.1 to be consistent with that model by setting:

$$\pi_{ij} \propto \exp(\log(p_{ij}) + \alpha_2 C(\mathbf{x}_j))$$

(see Forester et al. (2009)).

One difference between this multinomial observation model for resource use data and those that we have considered in previous chapters is that it includes the normalizing constant $\sum_x \exp(-\alpha_1 d_{ij}^2 + \alpha_2 C(\mathbf{x}_j))$, which ensures that the use distribution is a proper probability density function. In that sense, the model has the same form as the multinomial SCR model described in Chapt. 9 except that, here, the probability density of use locations is distributed over the whole state space \mathcal{S} , not just the subset of locations where we have traps. In a sense, we view telemetry data as a perfect sampling of space, equivalent to having a trap in each pixel, and the number of captures (uses by an individual) is fixed by design.

13.1.1 A simulated example

For a simulated landscape (shown in Fig. 13.1), Royle et al. (2012a) depicted some typical space usage patterns under the model described above, which we reproduce here in Fig. 13.2. The covariate in this case was simulated using a kriging model of correlated random noise with the following R commands:

```
> set.seed(1234)
> gr <- expand.grid(1:40,1:40)
> Dmat<-as.matrix(dist(gr))
> V <- exp(-Dmat/5)
> C <- t(chol(V))%*%rnorm(1600)
```

The resulting covariate vector \mathbf{C} is multivariate normal with mean 0 and variance-covariate matrix \mathbf{V} which, here, has pairwise correlations which decay exponentially with distance. The use densities shown in Fig. 13.2 were simulated with $\alpha_1 = 1/(2\sigma^2)$, with $\sigma = 2$, and the coefficient on $C(\mathbf{x})$ set to $\alpha_2 = 1$. The resulting space usage densities – or “home ranges” – exhibit clear non-stationarity in response to the structure of the underlying covariate, and they are distinctly asymmetrical. We note that if α_2 were set to 0, the 8 home ranges shown here would be proportional to a bivariate normal kernel with $\sigma = 2^1$. The commands for the kriging model, and those to produce Fig. 13.1 are in the package `scrbook` (see `?RSF_example`).

¹This is why we have always referred to the similar-looking model for encounter probability as the Gaussian or bivariate normal model, instead of half-normal.

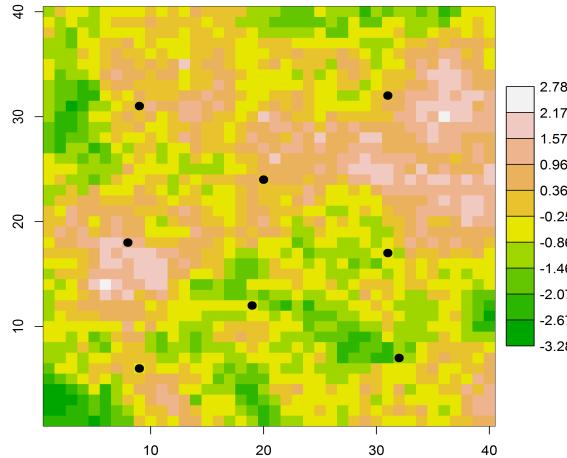


Figure 13.1. A typical habitat covariate reflecting habitat quality or hypothetical utility of the landscape to a species under study. Home range centers for 8 individuals are shown with black dots.

13.1.2 Poisson model of space use

A natural way to motivate the multinomial model of space usage is to assume that individuals make a sequence of resource selection decisions so that the outcomes m_{ij} are *independent* Poisson random variables:

$$m_{ij} \sim \text{Poisson}(\lambda_{ij})$$

where

$$\log(\lambda_{ij}) = a_0 - \alpha_1 d_{ij}^2 + \alpha_2 C(\mathbf{x}_j).$$

In this case, the number of visits to any particular cell is affected by the covariate $C(\mathbf{x})$ but has a baseline rate, $\exp(a_0)$, related to the amount (in an expected value sense) of movement occurring over some time interval. This is an equivalent model to the multinomial model given previously in the sense that, if we condition on the total sample size $R = \sum_j m_{ij}$, then the vector \mathbf{m}_i has a multinomial distribution with probabilities given by Eq. 13.1.1 (see also Chapt. 9).

In practice, we never observe “truth”, i.e., the actual use frequencies m_{ij} . Instead, we observe a sample of the actual use outcomes by an individual. As formulated in Sec. 5.4, we assume a binomial (“random”) sampling model:

$$y_{ij} \sim \text{Binomial}(m_{ij}, p_0).$$

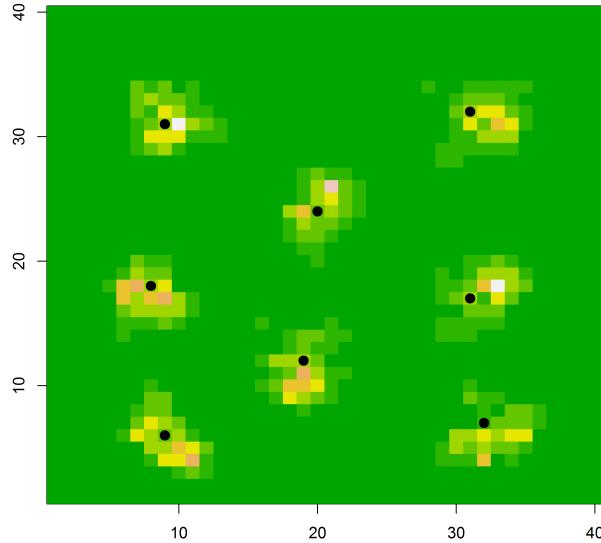


Figure 13.2. Space usage patterns of 8 individuals under a space usage model that contains a single covariate which is shown in Fig. 13.1. The plotted value is the multinomial probability π_{ij} for pixel j under the model in Eq. 13.1.1.

11808 We can think of these counts as arising by thinning the underlying point process (here,
 11809 aggregated into pixels) where p_0 is the thinning rate of the point process. In this case,
 11810 the marginal distribution of the observed counts y_{ij} is also Poisson but with mean

$$\log(\mathbb{E}(y_{ij})) = \log(p_0) + a_0 - \alpha_1 d_{ij}^2 + \alpha_2 C(\mathbf{x}_j).$$

11811 Thus, the space-usage model (RSF) for the thinned counts y_{ij} is the same as the space-
 11812 usage model for the original variables m_{ij} . This is because if we remove m_{ij} from the
 11813 conditional model by summing over its possible values, then the vector \mathbf{y}_i is *also* multi-
 11814 nominal with cell probabilities

$$\pi_{ij} = \frac{\lambda_{ij}}{\sum_j \lambda_{ij}}$$

11815 where any constant (the intercept term a_0 and thinning rate p_0) cancel from the numer-
 11816 ator and denominator. Thus, the underlying multinomial RSF model applies to the true
 11817 unobserved count frequencies \mathbf{m}_i and also those produced from thinning or sampling, \mathbf{y}_i .

13.2 INTEGRATING CAPTURE-RECAPTURE DATA

11818 The key to combining RSF data with SCR data is to note that the Poisson model of space
 11819 usage given above is exactly our Poisson encounter probability model from Chapt. 9, only
 11820 with a spatial covariate $C(\mathbf{x})$, and some arbitrary intercept off-set related to the sampling
 11821 rate by the telemetry device. We've used exactly this model for our SCR data (Chapt. 7),
 11822 but with a different intercept, α_0 , unrelated to the intercept of the Poisson use model for
 11823 telemetry described above but, rather, to the efficiency of the capture-recapture encounter
 11824 device. In other words, we view camera traps (or other devices) located in some pixel \mathbf{x}
 11825 (or multiple pixels) as being equivalent to being able to turn on a type of (less perfect)
 11826 telemetry device only in that pixel. Therefore, data from a camera trapping are Poisson
 11827 random variables for every pixel j where a trap is located:

$$y_{ij} | \mathbf{s}_i \sim \text{Poisson}(\lambda_{ij})$$

11828 with

$$\log(\lambda_{ij}) = \alpha_0 - \alpha_1 d_{ij}^2 + \alpha_2 C(\mathbf{x}_j).$$

11829 The parameters α_1 and α_2 are shared with the multinomial model for the telemetry data.

11830 Alternatively, the SCR study can produce binary encounters depending on the type of
 11831 sampling being done, where $y_{ij} = 1$ if the individual i visited the pixel containing a trap
 11832 and was detected, then we imagine that y_{ij} is related to the latent variable m_{ij} being the
 11833 event $m_{ij} > 0$, which occurs with probability

$$p_{ij} = 1 - \exp(-\lambda_{ij}) \quad (13.2.1)$$

11834 and then the observed encounter frequencies for individual i and trap j , from sampling
 11835 over K occasions are binomial:

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

11836 A key point here is that if resource selection is happening, then it appears as a covariate
 11837 on encounter rate (or encounter probability) in the same way as ordinary covariates which
 11838 we discussed in Chapt. 7.

11839 To construct the likelihood for SCR data when we have direct information on space
 11840 usage from telemetry data, we regard the two samples (SCR and RSF) as independent
 11841 of one another, and we form the likelihood for each set of observations as a function of
 11842 the same underlying parameters. The joint likelihood then is the product of the two
 11843 components.

11844 In particular, let $\mathcal{L}_{scr}(\alpha_0, \alpha_1, \alpha_2, N; \mathbf{y})$ be the likelihood for the SCR data in terms of
 11845 the basic encounter probability parameters and the total (unknown) population size N ,
 11846 and let $\mathcal{L}_{rsf}(\alpha_1, \alpha_2; \mathbf{m})$ be the likelihood for the RSF data based on telemetry which, be-
 11847 cause the sample size of telemetered individuals is fixed, does not depend on N . Assuming
 11848 independence of the two datasets, the joint likelihood is the product of these two pieces:

$$\mathcal{L}_{rsf+scr}(\alpha_0, \alpha_1, \alpha_2, N; \mathbf{y}, \mathbf{m}) = \mathcal{L}_{scr}(\alpha_0, \alpha_1, \alpha_2, N; \mathbf{y}) \times \mathcal{L}_{rsf}(\alpha_1, \alpha_2; \mathbf{m}),$$

11849 where the \mathcal{L}_{scr} is the standard integrated likelihood (Chapt. 6), and the RSF likelihood
 11850 contribution is the multinomial telemetry likelihood having cell probabilities Eq. 13.1.1.
 11851 The R code for maximizing the joint likelihood was given in the supplement to Royle
 11852 et al. (2012a), and we include a version of this in the **scrbook** package, see `?intlik3rsf`,
 11853 which also shows how to simulate data and fit the combined SCR+RSF model.

13.3 SW NEW YORK BLACK BEAR STUDY

11854 Royle et al. (2012a) applied the integrated SCR+RSF model to data from a study of
 11855 black bears (*Ursus americanus*) in a region of approximately 4,600 km² in southwestern
 11856 New York. These data come from a research project by C. Sun (Sun, in prep) at Cornell
 11857 University, and it is a different data set than our Fort Drum bear study data set which
 11858 we've analyzed in previous chapters. The data can be loaded from the **scrbook** package
 11859 with the command **data(nybears)**. We reproduce the findings of Royle et al. (2012a) in
 11860 this section.

11861 The data are based on a noninvasive genetic capture-recapture study using 103 hair
 11862 snares in June and July, 2011. Hair snares were baited and scented and checked weekly
 11863 for hair (Sun, in prep). The study yielded relatively sparse encounter histories of 33 indi-
 11864 viduals with a total of 14 recaptures and 27 individuals captured 1 time only. Telemetry
 11865 data were collected on 3 telemetry-collared individuals, which produced locations for each
 11866 bear approximately once per hour. Telemetry locations were thinned to once per 10 hours
 11867 to produce movement outcomes that might be more independent. This produced 195
 11868 telemetry locations used in the RSF component of the model. Elevation was used as the
 11869 covariate for this model, a standardized version of which is shown in Fig. 13.3 along with
 11870 the number of individuals captured at each hair snare site.

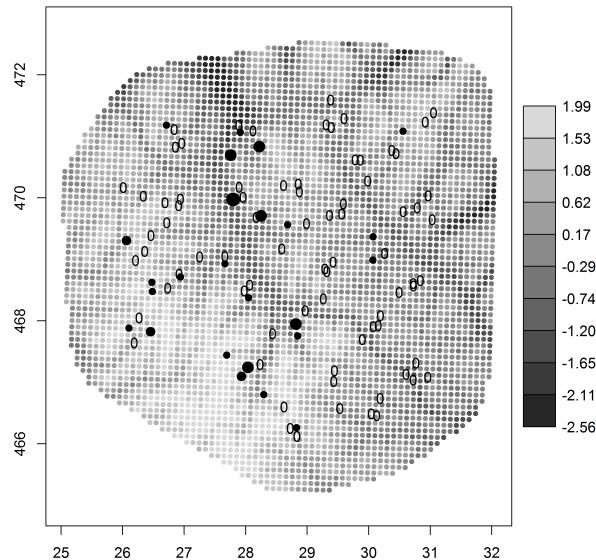


Figure 13.3. Elevation (standardized), hair snare locations are marked by the number of individuals captures at each site. The largest size solid mark corresponds to 4 individuals captured, the smallest to 1 individual. Hair snares that produced no individuals are given by "0".

11871 There are a number of models that could be fitted to these data based on the combination
11872 of SCR and RSF data as well as the elevation covariate. The models fit here are
11873 based on the Gaussian hazard trap encounter/space usage model, including an ordinary
11874 SCR model with no covariates or telemetry data, the SCR model with elevation affecting
11875 either λ_0 or density $D(\mathbf{x})$ (Chapt. 11), and models that use telemetry data. The 6 models
11876 fitted were:

11877 Model 1, SCR: ordinary SCR model
11878 Model 2, SCR+p(C): ordinary SCR model with elevation as a covariate on baseline
11879 encounter probability λ_0 .
11880 Model 3, SCR+D(C): ordinary SCR model with elevation as a covariate on density only.
11881 Model 4, SCR+p(C)+D(C): ordinary SCR model with elevation as a covariate on both
11882 baseline encounter probability and density.
11883 Model 5, SCR+p(C)+RSF: SCR model including data from 3 telemetered individuals.
11884 Model 6, SCR+p(C)+RSF+D(C): SCR model including telemetered individuals and
11885 with elevation as a covariate on density.

11886 Parameter estimates for the six models are given in Table 13.1 (reproduced from Royle
11887 et al. (2012a), see also the help file `?nybears`). It is tempting to want to compare these
11888 different models by AIC but, because models 5 and 6 involve additional data, they cannot
11889 be compared with models 1-4.

11890 By looking at Table 13.1, it is clear based on the negative log likelihood for just Models
11891 1-4, that those containing an elevation effect on density are preferred (Model 3 and 4).
11892 The parameter estimates indicate a positive effect of elevation on density, which seems to
11893 be consistent with the raw capture data shown in Fig. 13.3. Despite this strong effect of
11894 elevation, the estimates of N under each of these models only ranged from 93 – 103 bears
11895 for the 4600 km² state-space, and so estimated density is pretty consistent across models.
11896 If we consider not just density, but space usage (i.e., looking at the parameter α_2), the
11897 effect of elevation is negative. Thus, elevation, appears to affect density and space usage
11898 differently. It was suggested that density operates at the second-order scale of resource
11899 selection and “....is largely related to the spacing of individuals and their associated home
11900 ranges across the landscape. On the other hand, our RSF was defined based on selection of
11901 resources within the home range (third-order).” (Royle et al., 2012a) The positive effect of
11902 density on elevation is consistent with some other studies on black bears (e.g. Frary et al.,
11903 2011), and the negative effect of elevation on space usage can be attributed to seasonal
11904 variation in food availability, usage of corridors, or environmental conditions.

11905 Models 5 and 6 include the additional telemetry data, thus the negative log-likelihoods
11906 are not directly comparable to the first 4 models, but we can still make a few important
11907 observations. First is that the parameter estimates under these two models are consistent
11908 with Model 4 in that elevation had a strong effect on both density and space usage. In
11909 comparing models 5 and 6, the latter model which includes elevation as an effect on density
11910 reduces the negative log-likelihood by 5 units. Additionally, including the telemetry data
11911 reduces the standard errors (SE) of the density and space usage parameters and as we
11912 would expect, the incorporation of telemetry data also reduces the SE for σ . The increased
11913 precision for the estimated population size (N) is negligible with the use of telemetry data
11914 in this case. However, that may be different if more telemetry information were available.
11915 Model 6 (SCR+p(C)+RSF+D(C)), was used to produce maps of density (Fig. 13.4) and

Table 13.1. Summary of model-fitting results for the black bear study. Parameter estimates are for the intercept (α_0), logarithm of σ , the scale parameter of the Gaussian hazard encounter model, β is the coefficient of elevation on density, and the total population size N of the state-space. Standard errors are in parentheses. The SCR data are based on $n = 33$ individuals, and the telemetry data are based on 3 individuals.

model	α_0	$\log(\sigma)$	α_2	N	β	-loglik
SCR(elev)	-2.860 (0.390)	-1.117 (0.139)	0.175 (0.248)	95.8 (22.99)		122.738
SCR	-2.729 (0.345)	-1.122 (0.140)	—	93.9 (22.06)		122.990
SCR+D(elev)	-2.715 (0.353)	-1.133 (0.139)	—	94.2 (21.90)	1.247 (0.408)	118.007
SCR(elev)+D(elev)	-2.484 (0.391)	-1.157 (0.142)	-0.384 (0.276)	103.5 (26.56)	1.571 (0.463)	117.075
SCR(elev)+RSF	-3.068 (0.272)	-0.814 (0.036)	-0.281 (0.118)	81.6 (17.65)		1271.739
SCR(elev)+RSF+D(elev)	-3.070 (0.272)	-0.810 (0.037)	-0.371 (0.124)	89.1 (20.55)	1.273 (0.411)	1266.700

space usage (Fig. 13.5) showing the effect of elevation on both components of the model. The map of space usage shows the relative probability of using a pixel \mathbf{x} relative to one having the mean elevation, given a constant distance to the individual's activity center.

13.4 SIMULATION STUDY

Using the simulated landscape shown in Fig. 13.1, Royle et al. (2012a) presented results of a simulation study considering populations of $N = 100$ and $N = 200$ individuals exposed to encounter by a 7×7 array of trapping devices, with $K = 10$ sampling occasions, using the Gaussian hazard model (Eq. 13.2.1) with

$$\log(\lambda_{ij}) = -2 - \frac{1}{2\sigma^2} d_{ij}^2 + 1 * C(\mathbf{x}_j).$$

where $\sigma = 2$. They looked at the effect of misspecification of the resource selection model with an ordinary model SCR0 (i.e. no habitat covariates affecting the trap encounter model), and the performance of the MLEs, under SCR+telemetry designs having 2, 4, 8, 12, and 16 telemetered individuals (with 20 independent telemetry fixes *per* individual). Three models were fitted: (i) the SCR only model, in which the telemetry data were not used; (ii) the integrated SCR/RSF model which combined all of the data for jointly estimating model parameters; and (iii) the RSF only model which just used the telemetry data alone (and therefore the parameters α_0 and N are not estimable). An abbreviated version of the results from Royle et al. (2012a) is summarized in Table 13.2. We provide an **R** script (see `?RSFsim`) that can be modified for further analysis and exploration.

One thing we see is a pretty dramatic negative bias in estimating N if the model SCR0 is fitted (interestingly, there is much less bias in estimating σ). Overall, though, when either the SCR model with covariate or the joint SCR+RSF model is fitted, the MLEs exhibit little bias for the parameter values simulated here. In terms of RMSE, there is

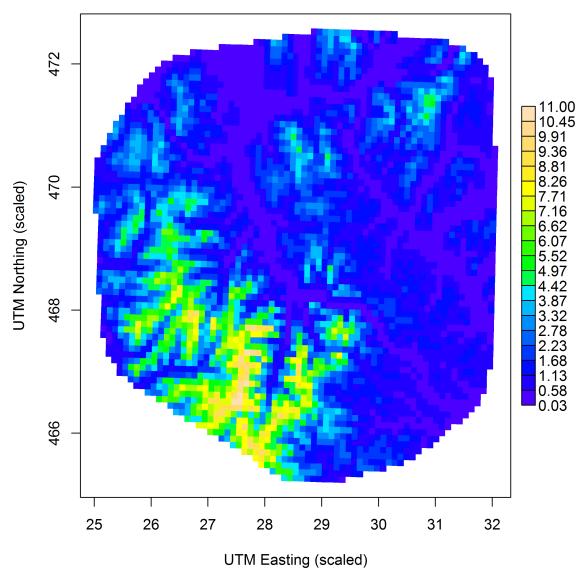


Figure 13.4. Predicted density of black bears (per 100 km²) in southwestern New York study area.

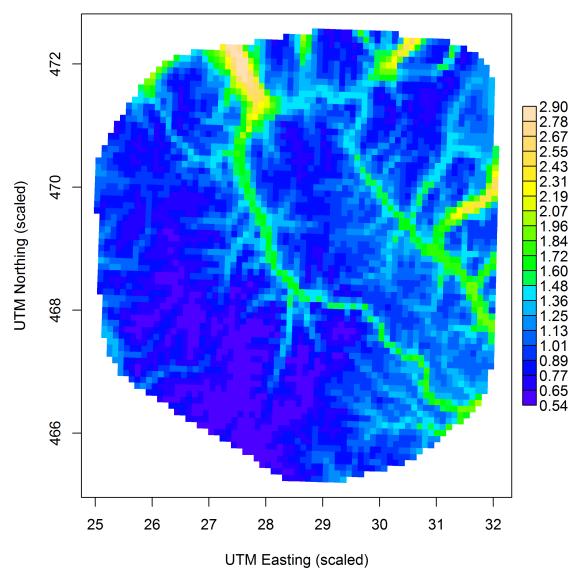


Figure 13.5. Relative probability of use of pixel x compared to a pixel of mean elevation, at a constant distance from the activity center.

only a slight \approx 5-10% reduction in RMSE of the estimator of N when we have at least 2 telemetered individuals. Thus, estimating N benefits only slightly from the addition of telemetry data, which is because information about the intercept, α_0 , comes only from the capture-recapture data. However, there is a large improvement in precision (50-60%) for estimating the scale parameter σ . While this doesn't translate much into improved estimation of N , it suggests that it should be relevant to the design of SCR studies for which trap spacing is one of the main considerations (Chapt. 10). In terms of study design these results also suggest that, perhaps, spatial recaptures are not needed if some telemetry data are available (in Chapt. 19, in the context of mark-resight models, we show a case study of raccoons where additional telemetry data allows estimating model parameters in spite of a very low number of spatial recaptures (Sollmann et al., 2013)). The resource selection parameter α_2 is well-estimated even *without* telemetry data. The fact that parameters of resource selection can be estimated from ordinary capture-recapture data should have considerable practical relevance in the study of animal populations and landscape ecology. For the highest sample size of telemetered individuals ($n = 16$), the RMSE for estimating this parameter only decreases from about 0.09 to 0.07.

Table 13.2. This table summarizes the sampling distribution of the MLE of model parameters for models fitted to data generated under a resource selection model. The models fitted include the misspecified model, which is a basic model SCR0 (with no covariate), the SCR model with the covariate on encounter probability, and the SCR model including the covariate and a sample of telemetered individuals (n is the number of individuals telemetered). Data were simulated with $N = 200$ individuals, $\alpha_2 = 1$ and $\sigma = 2$.

	\hat{N}	RMSE	$\hat{\alpha}_2$	RMSE	$\hat{\sigma}$	RMSE
$n=2$						
SCR+C(x)	199.11	14.28	0.99	0.09	2.00	0.090
SCR+RSF	199.11	13.80	0.99	0.09	2.00	0.079
SCR0	161.48	39.98	—	—	1.84	0.180
$n=4$						
SCR only	199.67	13.87	1.00	0.09	2.00	0.090
SCR/RSF	199.65	13.59	1.00	0.09	2.00	0.072
SCR0	161.32	40.00	—	—	1.83	0.191
$n=8$						
SCR only	199.24	15.49	0.99	0.10	2.01	0.093
SCR/RSF	199.55	14.17	0.99	0.08	2.00	0.063
SCR0	161.46	40.06	—	—	1.84	0.184
$n=12$						
SCR only	200.41	15.16	0.99	0.10	2.00	0.086
SCR/RSF	200.95	13.04	1.00	0.08	2.00	0.051
SCR0	162.40	38.95	—	—	1.84	0.185
$n=16$						
SCR only	199.16	15.62	1.00	0.09	2.00	0.095
SCR/RSF	199.63	13.38	1.00	0.07	2.00	0.052
SCR0	160.93	40.44	—	—	1.84	0.190

13.5 RELEVANCE AND RELAXATION OF ASSUMPTIONS

11953 In constructing the combined likelihood for RSF and SCR data, we assumed the data from
11954 capture-recapture and telemetry studies were independent of one another. This implies
11955 that whether or not an individual enters into one of the data sets has no effect on whether
11956 it enters into the other data set. We cannot foresee situations in which violation of this
11957 assumption should be problematic or invalidate the estimator under the independence
11958 assumption. In some cases it might so happen that some individuals appear in *both* the
11959 RSF and SCR data sets. In this case, ignoring that information should entail only an
11960 incremental decrease in precision because a slight bit of information about an individuals
11961 activity center is disregarded.

11962 Our model pretends that we do not know anything about the telemetered individuals
11963 in terms of their encounter history in traps. In principle it should not be difficult to admit
11964 a formal reconciliation of individuals between the two lists. In that case, we just combine
11965 the two conditional likelihoods before we integrate s from the conditional likelihood. This
11966 would be almost trivial to do if *all* individuals were reconcilable (or none, as in the case
11967 we have covered here). But, in general, we think you will often have an intermediate case,
11968 i.e., either none will be or at most a subset of telemetered guys will be known and there
11969 will be some individuals of unknown mark status. In that case, basically a type of marking
11970 uncertainty or misclassification, is clearly more difficult to deal with (see Chapt. 19 for
11971 some additional context).

11972 We developed the model in a discrete landscape which regarded potential trap locations
11973 and the covariate $C(\mathbf{x})$ as being defined on the same set of points. In practice, trap
11974 locations may be chosen independent of the definition of the raster and this does not pose
11975 any challenge or novelty to the model as it stands. In that case, the covariate(s) need to be
11976 defined at each trap location. The model should be applicable also to covariates that are
11977 naturally continuous (e.g., distance-based covariates) although, in practice, it will usually
11978 be sufficient to work with a discrete representation of such covariates.

11979 The multinomial RSF model for telemetry data assumes independent observations of
11980 resource selection. This would certainly be reasonable if telemetry fixes are made far apart
11981 in time (or thinned). However, as noted by Royle et al. (2012a), the independence as-
11982 sumption is *not* an assumption of spatially independent movement outcomes in geographic
11983 space. Active resource selection should probably lead to the appearance of spatially de-
11984 pendent outcomes, regardless of how far apart in time the telemetry locations are. Even
11985 if resource selection observations are dependent, use of the independence model probably
11986 yields unbiased estimators while under-stating the variance. Development of integrated
11987 SCR+RSF models that accommodate more general models of movement is needed.

13.6 SUMMARY AND OUTLOOK

11988 How animals use space is of fundamental interest to ecologists and is important in the
11989 conservation and management of many species. Investigating space use is normally done
11990 using telemetry and models referred to as resource selection functions (Manly et al., 2002)
11991 but in all of human history, animal resource selection has *never* been studied using capture-
11992 recapture models. Instead, essentially all applications of SCR models have focused on
11993 density estimation. It is intuitive, however, that space usage or resource selection should
11994 affect encounter probability and thus it should be highly relevant to density estimation in

1195 SCR applications, and, vice versa, SCR applications should yield data relevant to resource
1196 selection questions. The development in this chapter shows clearly that these two ideas
1197 can be unified within the SCR methodological framework so that classical notions of
1198 resource selection modeling can be addressed simultaneous to modeling of animal density.
1199 What we find is that if animal resource selection is occurring, this can be modeled as
1200 covariate on encounter probability, with or without the availability of auxiliary telemetry
1201 data. If telemetry data do exist, we can estimate parameters jointly by combining the two
1202 likelihood components – that of the SCR data and that of the telemetry data.

1203 Active resource selection by individuals induces a type of heterogeneous encounter
1204 probability, and this induces (possibly severe) bias in the estimated population size for
1205 a state-space when default symmetric encounter probability models are used. As such,
1206 it is important to account for resource selection when relevant covariates are known to
1207 influence resource selection patterns. Aside from properly modeling this selection-induced
1208 heterogeneity, integration of RSF data from telemetry with SCR models achieves a number
1209 of useful advances: First, it leads to an improvement in our ability to estimate density, and
1210 also an improvement in our ability to estimate parameters of the RSF function. As many
1211 animal population studies have auxiliary telemetry information, the incorporation of such
1212 information into SCR studies has broad applicability to many studies. It seems possible
1213 even to estimate density now, with no spatial recaptures, provided telemetry data are
1214 available. Secondly, the integrated model allows for the estimation of RSF model param-
1215 eters directly from SCR data *alone*. This establishes clearly that SCR models *are* explicit
1216 models of resource selection. In our view, this greatly broadens the utility and importance
1217 of capture-recapture studies beyond their primary historical use of estimating density or
1218 population size. Finally, we note that telemetry information provide direct information
1219 about the home range shape parameter, σ in our analyses above, and its estimation is
1220 greatly improved with even moderate amounts of telemetry data (see also Sollmann et al.
1221 (2013) and Sollmann et al. (in revision). This should have some consequences in terms
1222 of the design of capture-recapture studies (Chapt. 10), especially as it relates to trap
1223 spacing.

1224 Simultaneously conducting telemetry studies with capture-recapture is extremely com-
1225 mon in field studies of animal populations. However, the simultaneous, integrated analysis
1226 of the two sources of data is uncommon. The new class of integrated SCR/RSF models
1227 based on Royle et al. (2012a) allows researchers to model how the landscape and habitat
1228 influence the movement and space use of individuals around their home range, using non-
1229 invasively collected capture-recapture data that can be augmented with telemetry data.
1230 This should improve our ability to understand, and study, aspects of space usage and
1231 it might, ultimately, aid in addressing conservation-related problems such as reserve or
1232 corridor design. This should greatly expand the relevance and utility of spatial capture-
1233 recapture beyond its use for density estimation.

12034
12035

12036
12037

14

STRATIFIED POPULATIONS: MULTI-SESSION AND MULTI-SITE DATA

12038 In this chapter, we describe SCR models for situations when we have multiple distinct
12039 sample groups, strata or “sessions” (the term used in **secr**) each with a population size
12040 parameter N_g , for group g . Such “stratified” populations are commonplace in capture-
12041 recapture studies, especially in the context where the strata represent distinct spatial
12042 regions, yet most SCR applications have been based on models that are distinctly single-
12043 population models. This is done either by analyzing separate data sets one-at-a-time,
12044 producing many, if not dozens, of independent estimates of abundance, or by pooling data
12045 from multiple study areas. A standard example that arises frequently is that in which
12046 multiple habitat patches (often refuges, parks or reserves) are sampled independently
12047 with the goal of estimating the population size of some focal species in each reserve.
12048 If there are parameters that can be shared across sessions or groups, it makes sense to
12049 combine the data together into a single model that permits the sharing of information
12050 about some parameters, but provides individual estimates of abundance for each land
12051 unit. A similar situation is that in which a number of replicate trap arrays are located
12052 within a landscape, sometimes for purposes of evaluating the effects of management actions
12053 or landscape structure on populations. This is a common situation in studies of small
12054 mammals (Converse et al., 2006a,b; Converse and Royle, 2012), or in mist-netting of birds
12055 (DeSante et al., 1995), but there are examples of large-scale monitoring of carnivores and
12056 other species too, e.g., tigers (Jhala et al., 2011).

12057 In previous chapters, we’ve analyzed data for a number of examples that have a natural
12058 stratification or group structure. In Chapt. 9, we analyzed the ovenbird data as an
12059 example of a multi-catch (independent multinomial) model, where we used year as the
12060 stratification variable, and the possum data set (illustrating the single-catch situation) in
12061 which the group structure arose from the use of 5 distinct trap arrays. In Chaps. 7 and 8
12062 we fitted models with sex-specificity of parameters using multi-session models, where the
12063 stratification variable in that case was sex. In this chapter, we focus on Bayesian analysis
12064 of stratified SCR models using data augmentation (Converse and Royle, 2012; Royle et al.,
12065 2012b). The technical modification of data augmentation to deal with such models is that

it is based on a model for the joint distribution of the stratum-specific population sizes, N_g , *conditioned* on their total. This results in a multinomial distribution for all N_g , which we can analyze in some generality using data augmentation. As a practical matter, specification of this multinomial distribution for the N_g parameters *induces* a distribution for an individual covariate, say g_i , which is “group membership”. This is extremely handy to analyze by MCMC in the various **BUGS** engines that you are familiar with by now, and the flexibility of model specification in **BUGS** is why we focus a whole chapter here on Bayesian analysis by data augmentation. However, we have noted previously that the **R** package **secr** fits a class of multi-session models which we have already seen (Sec. 6.5.4), and we used **secr** to analyze several case studies using the multi-session models including the ovenbird (Sec. 9.2.4) and the possum data (Sec. 9.3.2), and models with sex-specific parameters in Chaps. 7 and 8.

In the stratified population models considered here, an individual is assumed to be a member of a single stratum, so that the population sizes N_g for the g strata are independent of one another. However, stratified or multi-session SCR models are also directly relevant when the stratification index is time, either involving distinct periods within a biological season, or even across years. In this case, individuals might belong to multiple of the strata, but, the models discussed in this chapter do not acknowledge that explicitly. Unlike the case in which the strata represent spatial units, with temporally defined strata, we imagine a fully dynamic, or demographically open model for N might be appropriate – one that involves survival and recruitment. We deal with those models specifically in Chapt. 16. However, the stratified models covered here can be thought of as a primitive type of model for open systems in which the population sizes are assumed to be *independent* across temporal strata, and so we might still find them useful in cases where the strata are temporal periods or sessions.

14.1 STRATIFIED DATA STRUCTURE

We suppose that $g = 1, 2, \dots, G$ strata (or groups), having sizes N_g , and state-spaces \mathcal{S}_g , are sampled using some capture-recapture method producing sample sizes of n_g unique individuals and encounters y_{ijk} for individual $i = 1, 2, \dots, \sum_{g=1}^G n_g$. Right now we won’t be concerned with the details of every type of capture-recapture observation model so, for context, and to develop some technical notions, we consider a Bernoulli encounter model in which individual and trap-specific encounter frequencies are binomial counts: $y_{ij} \sim \text{Binomial}(K, p_{ij})$. Let g_i be a covariate (integer-valued, $1, \dots, G$) indicating the group membership of individual i . This covariate is *observed* for the sample of captured individuals but not for individuals that are never captured.

To illustrate the prototypical data structure for stratified SCR data, we suppose that a population comprised of 4 groups is sampled $K = 5$ times. Then, a plausible data set has the following structure:

```
12103   individual (i) : 1 2 3 4 5 6 7 8 9 10
12104   total    encounters (y) : 1 1 3 1 1 2 2 4 1 1
12105       group (g)      : 1 1 1 2 3 3 3 3 4 4
```

This data set indicates three individuals were captured in group 1 (captured 1, 1, and 3 times), a single individual was captured in group 2, four individuals were captured in group 3, and two individuals were captured in group 4.

12109 A key idea discussed shortly is that the assumption of certain models for the collection
 12110 of abundance variables N_g implies a specific model for the group membership variable g_i .
 12111 Then, the data from all groups can be pooled, and analyzed as data from a single popu-
 12112 lation with the appropriate model on g_i , without having to deal with the N_g parameters
 12113 in the model directly. In this way, we can easily build hierarchical models for stratified
 12114 populations, using an *individual* level parameterization of the model. Obviously this is
 12115 important for SCR models as they all possess at least one individual level random effect
 12116 in the form of the activity center \mathbf{s} . In the context of stratified or multi-session type mod-
 12117 els, the “population membership” variable g_i is a *categorical* type of individual covariate
 12118 (Huggins, 1989; Alho, 1990; Royle, 2009b). Before considering SCR models specifically, in
 12119 the next section we talk a little bit about the technical formulation of data augmentation
 12120 for stratified populations in the context of ordinary closed population models.

14.2 MULTINOMIAL ABUNDANCE MODELS

12121 One of the key ideas to Bayesian analysis of stratified population models is that we make
 12122 use of multinomial models for allocating individuals into strata or sessions. We do this
 12123 because it allows us to analyze the models by data augmentation (Converse and Royle,
 12124 2012; Royle and Converse, in review), and it has a natural linkage to the Poisson model,
 12125 which is commonly used throughout ecology to model variation in abundance.

12126 To motivate the technical framework, consider sampling $g = 1, 2, \dots, G$ groups having
 12127 unknown sizes N_g , and we wish to impose model structure on the group-specific population
 12128 size variables using a Poisson distribution:

$$N_g \sim \text{Poisson}(\lambda_g) \quad (14.2.1)$$

12129 with

$$\log(\lambda_g) = \beta_0 + \beta_1 C_g \quad (14.2.2)$$

12130 where C_g is some measured attribute for group g . We could generalize this a bit by
 12131 considering a random effect in Eq. 14.2.2, producing over-dispersed population sizes N_g .
 12132 For the special case of adding log-gamma noise, this results in negative binomial models
 12133 for N_g .

12134 To develop a data augmentation scheme for this group-structured model, let’s think
 12135 about doing data augmentation on each population *individually*, by assuming that

$$N_g \sim \text{Binomial}(M_g, \psi)$$

12136 where $\psi \sim \text{Uniform}(0, 1)$ as usual. A key point is that we allow M_g to be population
 12137 specific but ψ is constant. We could do this multi-population data augmentation by just
 12138 picking each M_g to be some large integer (as we always do by data augmentation; see
 12139 Sec. 9.2.4). However, we want to pick M_g in a way that induces the correct structure
 12140 on N_g . If we want to enforce our Poisson model on N_g from above, we naturally choose
 12141 M_g to be Poisson also, in which case the marginal distribution of N_g is also Poisson,
 12142 but with mean $\psi \exp(\beta_0 + \beta_1 C_g)$. Here, clearly ψ and β_0 are confounded (see below for
 12143 more discussion). Regardless, for multiple groups that we want to model jointly, the key
 12144 point is that we impose the structure that we desire for N_g , on the super-population
 12145 parameters M_g . To implement this model at the individual level we need to get rid of the

12146 M_g parameters (which is the entire motivation of data augmentation in the first place).
 12147 So we condition on the “total super-population” size $M_T = \sum_g M_g$ (in a sense, this is
 12148 the super-super-population!). Then, the vector $\mathbf{M} = (M_1, \dots, M_G)$ has a multinomial
 12149 distribution:

$$\mathbf{M}|M_T \sim \text{Multinomial}(M_T; \boldsymbol{\pi}) \quad (14.2.3)$$

12150 where $\pi_g = \lambda_g / \sum_g \lambda_g$. This is handy because we can implement this model, e.g., in
 12151 **BUGS**, by introducing a variable g_i for each $i = 1, 2, \dots, M_T$ which is the “group mem-
 12152 bership” of each individual in the super-super-population. Then, conditional on g_i , an
 12153 individual is either “real”, or a pseudo-individual, according to the binary data augmen-
 12154 tation variable z_i . As specified in **BUGS** pseudo-code, the model is:

```
12155     psi ~ dunif(0,1)
12156     for(g in 1:G){
12157         pi[g] <- lambda[g]/sum(lambda[])
12158     }
12159     g[i] ~ dcat(pi[1:G])
12160     z[i] ~ dbern(psi)
```

12161 This produces a vector of population size parameters $\mathbf{N} = (N_1, \dots, N_G)$ which are ap-
 12162 proximately, for large M_T , independent Poisson random variables.

12163 When we apply data augmentation to the multinomial joint distribution, the ψ pa-
 12164 rameter takes the place of N_T , the total population size (across all groups or strata).
 12165 In addition, by constructing the model conditional on the total, N_T , we lose information
 12166 about the intercept β_0 ¹ but this is recovered in the data augmentation parameter ψ . Thus,
 12167 one of these parameters has to be fixed. We can set $\beta_0 = 0$ or else we can fix ψ (see Chapt.
 12168 11). The constraint can be specified by noting that, under the binomial data augmenta-
 12169 tion model $\mathbb{E}(N_T) = \psi M_T$ and, under the Poisson model, $\mathbb{E}(N_T) = \sum_g \exp(\beta_0 + \beta_1 C_g)$
 12170 and so we can set

$$\psi = \frac{1}{M_T} \sum_g \exp(\beta_0 + \beta_1 C_g).$$

12171 The linkage of β_0 and ψ was also discussed in Chapt. 11 in the context of building spatial
 12172 models for density. In that case, β_0 was the intercept of the intensity function and one
 12173 could choose to estimate either β_0 or the data augmentation parameter ψ .

12174 14.2.1 Implementation in **BUGS**

12175 The **BUGS** implementation of data augmentation for structured populations is straight-
 12176 forward. For each individual in the super-super-population we introduce a latent variable
 12177 g_i to indicate which *population* the individual belongs to, and we introduce a second
 12178 variable z_i to indicate whether the individual is alive or not. So, the latent structure for
 12179 the M_g variables and the binomial sampling of those super-population sizes is equivalently
 12180 represented by the latent variable pair (g_i, z_i) where g_i is categorical with prior probabili-
 12181 ties π_s and $z_i \sim \text{Bernoulli}(\psi)$. In particular, the multinomial assumption for the latent

¹ A technical argument is that the total N_T is the sufficient statistic for β_0 in the multinomial model and so, by conditioning on the total, β_0 is no longer a free parameter.

variables M_g is formulated in terms of “group membership” for each individual in the super-super-population of size M_T according to:

$$g_i \sim \text{Categorical}(\boldsymbol{\pi})$$

with $\boldsymbol{\pi} = (\pi_1, \dots, \pi_G)$ and $\pi_g = \lambda_g / (\sum_g \lambda_g)$. The binomial sampling is described by the binary variables z_1, \dots, z_{M_T} such that

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

where ψ is constrained as noted in the previous section. The **BUGS** model specification for this individual-level formulation of the model is shown in Panel 14.1 for an ordinary closed population model (model M_0). This actually shows two equivalent formulations. In the left panel we have ψ and β_0 as free parameters. The right panel shows the equivalent model but recognizing the constraint between ψ and β_0 . Running these models using the `multisession.sim` function, you can verify that the two parameters are not uniquely estimable. In particular, using the model (representation 1) in the left-hand side of Panel 14.1, you will see that draws of β_0 appear to be draws from the prior distribution, uninformed by the data, supporting the point we made previously that ψ and β_0 are not uniquely informed by the data.

14.2.2 Groups with no individuals observed

In practical settings, when the groups represent small populations, it will sometimes happen that some groups have no encountered individuals or even that $N_g = 0$ for some groups. This is dealt with implicitly in the development of the model shown in Panel 14.1 in the sense that the *prior* for N_g has the proper dimension (namely, G multinomial cells of non-zero probability) and thus some posterior mass may occur on non-zero values of N_g even if the *data* contain no representatives of group g . You can try this out to verify for yourself.

14.2.3 The group-means model

Under the Poisson model for group abundance N_g , even with a constant mean λ , each stratum or group may have a different realized population size, and this comes at the low price of a single parameter in the model (λ or, equivalently, the data augmentation parameter ψ). Thus, for a single parameter in this group-structure model, we are able to realize variation in the N_g parameters. In a sense, this is a benefit of the group structure in which N_g are regarded as random variables.

To accommodate more flexibility than afforded by the single-parameter Poisson model, there are a couple of choices: (1) We could allow the mean to be group specific such as: $N_g \sim \text{Poisson}(\lambda_g)$ where each λ_g is its own free parameter, independent of each others. This produces a model with G distinct “fixed” parameters, and effectively renders the Poisson assumption irrelevant as it doesn’t induce any “Bayesian shrinkage” (Sauer and Link, 2002) or impose any group structure on the population sizes N_g . It should provide estimates that are effectively the same as analyzing each data set independently, or using the independent binomial prior that we introduced in Chapt. 9, where some information

Implementation 1	Implementation 2
<pre> model { # This will show that psi and b0 # are confounded. p ~ dunif(0,1) beta0 ~ dnorm(0,.1) beta1 ~ dnorm(0,.1) psi ~ dunif(0,1) for(j in 1:G){ log(lam[j]) <- beta0+beta1*C[j] gprobs[j]<-lam[j]/sum(lam[1:G]) } for(i in 1:M){ g[i] ~ dcat(gprobs[]) z[i] ~ dbern(psi) mu[i] <- z[i]*p y[i] ~ dbin(mu[i],K) } N <- sum(z[1:M]) } </pre>	<pre> model { # This version constrains psi with # the intercept parameter p ~ dunif(0,1) beta0 ~ dnorm(0,.1) beta1 ~ dnorm(0,.1) psi <- sum(lam[])/M for(j in 1:G){ log(lam[j]) <- beta0+beta1*C[j] gprobs[j]<-lam[j]/sum(lam[1:G]) } for(i in 1:M){ g[i] ~ dcat(gprobs[]) z[i] ~ dbern(psi) mu[i] <- z[i]*p y[i] ~ dbin(mu[i],K) } N <- sum(z[1:M]) } </pre>

Panel 14.1: BUGS model specification for a capture-recapture model with constant encounter probability and Poisson subpopulation sizes, N_g , with mean depending on a single covariate $C[j]$. Two versions of the model: The first one describes the model in terms of the intercept β_0 and DA parameter ψ , which are confounded. The required constraint is indicated in the specification under Implementation 2.

12219 might be borrowed from the different groups for estimating the encounter probability
 12220 parameters. Under this model, we constraint one of the λ_g parameters to be 0, and N_g
 12221 for that group is taken up by the data augmentation parameter ψ ; (2) Alternatively, we
 12222 could identify specific fixed covariates which might explain variation across groups. Each
 12223 additional covariate adds only 1 additional fixed parameter to the model; (3) A flexible
 12224 formulation that provides something of an intermediate model, between that of a constant
 12225 λ and independent group specific λ_g 's, is that in which we put a prior on λ_g . For example,
 12226 if we assume

$$\lambda_g \sim \text{Gamma}(a, b)$$

12227 this corresponds to imposing a Dirichlet compound-multinomial model on the population
 12228 size vector, or, marginally, a negative binomial model on N_g . See Takemura (1999) for
 12229 some discussion of such models relevant to data augmentation. For this model, we impose
 12230 the constraint $b = 1$ to account for conditioning on the total population size N_T to use
 12231 data augmentation.

14.2.4 Simulating stratified capture-recapture data

It is helpful, as always, to simulate some data in order to understand the model. Suppose we cracked the conservation lotto jackpot and obtained funding to carry out a camera trapping study of some flashy carnivore in 20 forest patches or reserves, using a 5×5 array of traps. Here we will consider an ordinary closed population model, model M_0 , and we suppose there is some forest level covariate, say Dist = disturbance regime, perhaps measured by an index of trail density or something. We imagine a model for patch-level population size such as the following:

$$N_g \sim \text{Poisson}(\lambda_g)$$

$$\log(\lambda_g) = \beta_0 + \beta_1 \text{Dist}_g$$

We simulate some population sizes and encounter data under this model as follows:

```

12241 > set.seed(2013)
12242 > G <- 20                                # G = 20 groups or strata
12243 > beta0 <- 3                             # Abundance model parameters
12244 > beta1 <- .6
12245 > p <- .3                               # Encounter probability
12246 > K <- 5                                # Sample occasions for capture-recapture
12247 > Dist <- rnorm(G)                         # Simulate covariate
12248 > lambda <- exp(beta0+beta1*Dist)        # Simulate population sizes
12249 > N <- rpois(G,lambda=lambda)

12250
12251 > y <- NULL                            # Simulate model M0 data
12252 > for(g in 1:G){
12253 +   if(N[g]>0)
12254 +     y <- c(y, rbinom(N[g],K,p))
12255 + }
12256 > g<- rep(1:G,N)
12257
12258 > ## Now keep the group id and encounter frequency only for
12259 > ## individuals that are captured
12260 > g<-g[y>0]
12261 > y<-y[y>0]
```

That's it! We just simulated a population size model and capture-recapture data for the populations inhabiting $G = 20$ forest patches (the "groups" in this situation). To fit this model, we need to augment the \mathbf{g} and \mathbf{y} data objects, and then we can run the model in **JAGS** or **WinBUGS** using the code given in Panel 14.1. See the help file `?multisession.sim` for doing this analysis with these simulated data.

14.3 OTHER APPROACHES TO MULTI-SESSION MODELS

The multinomial super-population model allows for the joint modeling of a collection of population sizes using data augmentation. However, as we demonstrated in Sec. 9.2.4, we can analyze the models by putting independent binomial priors on each N_g and doing

the data augmentation independently for each population by itself. This is not any more or less difficult than the multinomial formulation but, we imagine, it could be slightly less efficient computationally. In this case we could build in among-group structure by modeling the DA parameter ψ as being variable for each subject, as a function of group-specific variables (see Hendriks et al., 2013, for an example). For example, if C_g is the value of some covariate for group g , then we could have $z_i \sim \text{Bernoulli}(\psi_i)$ with

$$\text{logit}(\psi_i) = \beta_0 + \beta_1 C_{g_i}$$

This implies a binomial model for the stratum population sizes:

$$N_g \sim \text{Binomial}(M, \psi_g).$$

If M is large then the N_g are approximately independent Poisson random variables with means $\psi_g M$.

As we noted in Chapt. 6, the multi-session models in **secr** are based on a Poisson prior for N_g with mean Λ_g , and then among group structure is modeled in the parameter Λ_g . In our view, either model (binomial based on data augmentation, or Poisson) is satisfactory for any application of capture-recapture to stratified populations. The main advantage of the formulation we provided here over that implemented in **secr** is we have quite a bit more flexibility in specifying models of all sorts, either in the population size model for N_g , or for the capture-recapture model. For example, Royle and Converse (in review) fitted a model having random group effects on encounter probability and abundance (i.e., extra-Poisson variation).

14.4 APPLICATION TO SPATIAL CAPTURE-RECAPTURE

Although we developed the implementation of Bayesian models for stratified populations using ordinary closed population models, the underlying ideas are completely general and can be applied equally to spatial capture-recapture models without any novel considerations. We already discussed (Chapt. 4) that SCR models are ordinary closed population models but with an individual covariate which is the activity center s_i , and the observation model has to be defined for each trap. With this in mind, it should be obvious how the **BUGS** specification in Panel 14.1 can be modified to accommodate a group-structured SCR situation. Specifically, we include the prior distribution for s_i and the observation model that relates s_i to the probability of encounter for individual i and trap j , as we've done so many times in previous chapters.

14.4.1 Multinomial (“multi-catch”) observations

We discuss Bayesian analysis of the multi-session model using data augmentation in the context of a multinomial observation model such as for a multi-catch sampling situation². For context, we return to the ovenbird data set, from the **R** package **secr**, which we introduced in Chapt. 9. Another example can be found in Royle and Converse (in review),

²This might be slightly confusing that we are considering multinomial observation models *and* multinomial models for group-specific abundance parameters N_g , but we will take care to be clear about this along the way.

12303 who applied the model to a small mammal trapping problem which involved replicate
 12304 “single-catch” arrays of traps, in a study of the effects of forest management practices on
 12305 small-mammal densities. The ovenbird data is a type of multi-catch observation model
 12306 where the group index variable is “year” and, in our earlier analyses, we analyzed the
 12307 data set using independent binomial priors for N_g within data augmentation in **JAGS**,
 12308 as well as with a Poisson prior in **secr** using the multi-session models. We mirror the
 12309 **secr** analysis here, but using the data augmentation formulation leading to a multinomial
 12310 distribution for N_g we introduced above.

12311 To refresh your memory about the multinomial observation model, let $\mathbf{y}_{ik} = (y_{i1k}, y_{i2k}, \dots, y_{iJk}, y_{i,J+1,k})$
 12312 be the spatial encounter history for individual i , during sample occasion k where the last
 12313 element $y_{i,J+1,k}$ corresponds to “not captured”. For mist nets, an individual can be cap-
 12314 tured in at most one trap. Then, the vector $(y_{i1k}, y_{i2k}, \dots, y_{iJk}, y_{i,J+1,k})$, contains a single
 12315 1 and the remaining values are 0. This $(J + 1) \times 1$ vector \mathbf{y}_{ik} is a multinomial trial:

$$\mathbf{y}_{ik} \sim \text{Multinomial}(n = 1; \boldsymbol{\pi}_{ik})$$

12316 where $\boldsymbol{\pi}_{ik}$ is a $(J + 1) \times 1$ vector where each element represents the probability of being
 12317 encountered in a trap (for elements 1, …, J) or not captured at all (element $J + 1$).

12318 For the multinomial observation model, the encounter probability vector is a func-
 12319 tion of distance between trap locations and individual activity centers, modeled on the
 12320 multinomial logit scale. The Gaussian encounter probability model is:

$$\text{mlogit}(\pi_{ij}) = \eta_{ij} = \alpha_0 - \alpha_1 \text{dist}(\mathbf{x}_j, \mathbf{s}_i)^2 \quad (14.4.1)$$

12321 where $\alpha_1 = 1/(2\sigma^2)$ and σ is the scale parameter of the Gaussian model. Then,

$$\boldsymbol{\pi}_{ij} = \exp(\eta_{ij}) / [1 + \sum_j \exp(\eta_{ij})]$$

12322 for each $j = 1, 2, \dots, J$, and the last cell corresponding to the event “not captured” is:

$$\pi_{i,J+1} = 1 - \sum_{j=1}^J \pi_{ij}$$

12323 There are no novel technical considerations in order to model covariates of any kind.
 12324 For example, in many studies we are concerned with a behavioral response to physical
 12325 capture. This is typical in small-mammal trapping studies, and also in mist-net studies
 12326 of birds where individuals exhibit net avoidance after first capture. For this, let C_{ik} be
 12327 a covariate of previous encounter (i.e., $C_{ik} = 0$ before the occasion of first capture, and
 12328 $C_{ik} = 1$ thereafter), then we include this covariate in our multinomial observation model
 12329 as follows:

$$\text{mlogit}(\pi_{ijk}) = \eta_{ijk} = \alpha_0 - \alpha_1 \text{dist}(\mathbf{x}_j, \mathbf{s}_i)^2 + \alpha_2 C_{ik}$$

12330 We note that, in this case, the multinomial probabilities depend not only on individual
 12331 and trap, but also on sample occasion.

Table 14.1. Posterior summaries for the Bayesian stratified population (“multi-session”) model fitted to the ovenbird data. Results are based on 3 chains, each with 5000 iterations (first 1000 discarded), for a total of 12000 iterations saved.

	Mean	SD	2.5%	50%	97.5%	Rhat
D[1]	0.883	0.191	0.562	0.868	1.308	1.002
D[2]	0.972	0.200	0.624	0.954	1.418	1.001
D[3]	1.146	0.224	0.758	1.125	1.638	1.001
D[4]	0.836	0.183	0.538	0.819	1.247	1.001
D[5]	0.705	0.167	0.428	0.685	1.088	1.001
N[1]	72.208	15.596	46.000	71.000	107.000	1.002
N[2]	79.478	16.367	51.000	78.000	116.000	1.001
N[3]	93.725	18.327	62.000	92.000	134.000	1.001
N[4]	68.399	14.952	44.000	67.000	102.000	1.001
N[5]	57.665	13.659	35.000	56.000	89.000	1.001
alpha0	-3.465	0.159	-3.779	-3.465	-3.155	1.004
alpha1	0.000	0.000	0.000	0.000	0.000	1.009
beta0[1]	4.250	0.244	3.754	4.257	4.710	1.001
beta0[2]	4.349	0.233	3.872	4.356	4.786	1.001
beta0[3]	4.516	0.220	4.059	4.522	4.930	1.001
beta0[4]	4.194	0.248	3.697	4.202	4.664	1.001
beta0[5]	4.013	0.275	3.456	4.022	4.524	1.001
psi	0.371	0.051	0.281	0.367	0.482	1.001
sigma	77.918	6.314	66.963	77.240	91.583	1.009

14.4.2 Reanalysis of the Ovenbird data

Here we use Bayesian analysis by data augmentation to fit a model that approximates the Poisson model with expected value $\mathbb{E}(N_g) = \lambda_g$ where we model effects on the log-mean scale according to:

$$\log(\lambda_g) = \beta_0 + \beta_1 C_g.$$

We considered only two models here: A model with year-specific abundance, and a model with a linear trend in density over time, so $C_g \equiv \text{Year}$. However, using the Kuo and Mallick (1998) indicator variable selection idea (see Chapt. 8), the linear trend term was found to have little or no posterior probability, so we do not reproduce analyses of that here (but see the `ovenbird.ms` function for the **R** script). We show the **BUGS** model specification for the year-specific abundance model in Panel 14.2. Note the construction of the multinomial cell probabilities which distribute individuals among years, based on the year-specific mean λ_t . On the log-scale, each of these parameters has a diffuse normal prior: `beta0[t] ~ dnorm(0, 0.01)`. A few lines of model specification that compute the derived population size parameters and density are not shown, but you can look at the **R** script `ovenbird.ms` in `scrbook` to run this analysis, and produce the posterior summaries shown in Table 14.1.

We previously analyzed these data in Sec. 9.2.4 using `secr` and the “one-at-a-time” data augmentation approach (independent binomial priors for N_t). To reproduce those results from `secr` for the equivalent model we execute this command:

```
> ovenbird.model.DT<-secr.fit(ovenCH,model=list(D~session),buffer=300)
```

```

model {
  alpha0 ~ dnorm(0,.01)                      # Prior distributions
  sigma ~ dunif(0,200)
  alpha1 <- 1/(2*sigma*sigma)
  psi <- sum(lambda[]) / bigM
  for(t in 1:5){
    beta0[t] ~ dnorm(0,0.01)                  # Year-specific abundances
    log(lambda[t]) <- beta0[t]
    pi[t] <- lambda[t]/sum(lambda[])
  }                                            # Calculate multinomial probs
  for(i in 1:bigM){
    z[i] ~ dbern(psi)
    yrid[i] ~ dcat(pi[])
    S[i,1] ~ dunif(xlim[1],xlim[2])          # Activity centers
    S[i,2] ~ dunif(ylim[1],ylim[2])
    for(j in 1:ntraps){
      d2[i,j] <- pow(pow(S[i,1]-X[j,1],2) + pow(S[i,2]-X[j,2],2),1)
    }
    for(k in 1:K){
      Ycat[i,k] ~ dcat(cp[i,k])
      for(j in 1:ntraps){                      # Construct trap enc. probs.
        lp[i,k,j] <- exp(alpha0 - alpha1*d2[i,j])*z[i]*(1-died[i,k])
        cp[i,k,j] <- lp[i,k,j]/(1+sum(lp[i,k,1:ntraps]))
      }
      cp[i,k,ntraps+1] <- 1-sum(cp[i,k,1:ntraps]) # last cell = not captured
    }
  }
}

```

Panel 14.2: BUGS model specification for a stratified (multi-session) SCR model using data augmentation. This shows a multinomial (“multi-catch”) type of observation model, used to analyze the ovenbird data. Some code to tally up the derived population sizes and density parameters is omitted. See ovenbird.ms script

Table 14.2. Estimates for the multi-session model fitted to the ovenbird data using `secr`. The model had a year-specific density parameter, and constant encounter probability parameters.

2005					
	Link	Estimate	SE Estimate	LCL	UCL
D	log	0.920	0.228	0.571	1.484
g0	logit	0.028	0.004	0.021	0.037
sigma	log	78.567	6.379	67.025	92.095
2006					
	Link	Estimate	SE Estimate	LCL	UCL
D	log	0.963	0.238	0.598	1.553
g0	logit	0.028	0.004	0.021	0.037
sigma	log	78.566	6.379	67.025	92.095
2007					
	Link	Estimate	SE Estimate	LCL	UCL
D	log	1.139	0.282	0.706	1.836
g0	logit	0.028	0.004	0.021	0.037
sigma	log	78.566	6.379	67.025	92.095
2008					
	Link	Estimate	SE Estimate	LCL	UCL
D	log	0.832	0.206	0.516	1.341
g0	logit	0.028	0.004	0.021	0.037
sigma	log	78.566	6.379	67.025	92.095
2009					
	Link	Estimate	SE Estimate	LCL	UCL
D	log	0.701	0.173	0.435	1.130
g0	logit	0.028	0.004	0.021	0.037
sigma	log	78.566	6.379	67.025	92.095

12352 Note, small values of `buffer` can produce a warning that it is too small relative to the
 12353 indicated value of σ (which has posterior mass up to near $\sigma = 100$). The `secr` results
 12354 are as follows shown in Table 14.2. There are, as always, slight differences between the
 12355 MLEs shown here and the posterior summaries shown Table 14.1. The absolute difference
 12356 between the MLEs and the Bayesian posterior means was .037, -.011, -.006, -.004 and
 12357 -.004 for years 1 to 5, respectively.

14.5 SPATIAL OR TEMPORAL DEPENDENCE

12358 The models described here, and including the multi-session formulation used in `secr`,
 12359 assume that the population sizes N_g are *independent* (in a limiting sense, under data
 12360 augmentation). As a practical matter, this precludes the sharing of individuals among
 12361 populations (i.e., the same individual cannot be captured in multiple groups) which can
 12362 be violated in a number of situations. First, when the groups represent sampling in distinct
 12363 time periods (seasons, years) but of the same functional population (a standard “robust
 12364 design” situation), it is possible that some individuals remain in the population from one
 12365 time period to the next. In this situation, by disregarding individual identity across groups,
 12366 the models ignore a slight bit of dependence of N_g which may entail some incremental loss
 12367 of efficiency. We imagine this should have little practical effect unless survival probability

12368 is extremely high between the periods. Estimators of parameters obtained by assuming
12369 independence should be conservative in their statement of precision, but they should be
12370 unbiased (or, rather, ignoring the dependence should not affect the bias of the estimator
12371 much if at all).

12372 A second distinct situation is that in which the stratification variable is *spatial*, and
12373 the strata (e.g., trap arrays or other sampling mechanism) are in relatively close spatial
12374 proximity to one another so that individuals can sometimes be encountered by more than
12375 one array (e.g., the possum data, see Fig. 9.2). This case is somewhat easier to deal
12376 with in the analysis because we can build a model in which the state-space is the joint
12377 state-space enclosing all of the trapping arrays, and we preserve individual identity in an
12378 ordinary SCR model, just with a larger array of traps that is the union of the trap arrays
12379 of all sample groups. This may be impractical when the trap arrays are far apart creating
12380 only a slight bit of overlap of populations, because, in that case, the combined state-
12381 space may contain a huge population that one has to deal with in the MCMC (remember
12382 that increasing M increases computation time). (Royle et al., 2011a) had this problem
12383 in an analysis of data from a sample of 1 km quadrats using a search-encounter type
12384 model (discussed in the following chapter). Even in this case the independent N_g model
12385 is probably not too detrimental to inferences that apply to explaining marginal variation
12386 in N_g , such as habitat or landscape effects that are modeled on the expected value of N_g .

14.6 SUMMARY AND OUTLOOK

12387 Capture-recapture data are not always collected as single isolated studies but, instead,
12388 data are often grouped or stratified in some natural way, either because a number of
12389 distinct trap arrays are used, or sampling occurs in several forest patches, or over time.
12390 Often this is motivated by specific objectives, e.g., the trap arrays or units represent
12391 experimental replicates, or sometimes just to derive more valid estimates of density by
12392 obtaining a representative (ideally, random) sample of space within some region. The fact
12393 that data are grouped in such a way raises the obvious technical problem of having to
12394 combine data from multiple arrays, sites or otherwise defined groups in a single unified
12395 model that accommodates explicit sources of variation in density among these groups.
12396 This is naturally accomplished by developing an explicit model for variation in N , e.g., a
12397 Poisson GLM or similar (Converse and Royle, 2012; Royle et al., 2012b).

12398 In this chapter, we outlined an approach to Bayesian analysis of multi-session models
12399 using data augmentation Converse and Royle (2012); Royle and Converse (in review). This
12400 approach gives us one method for building explicit models for N_g and also gives us great
12401 flexibility in specifying the encounter model using standard or novel capture-recapture
12402 modeling considerations. Certain types of multi-session models can be fitted easily in
12403 **secr** (see Chapt. 9) and we suspect that platform will be satisfactory for many problems
12404 you encounter. However, as always, we believe the flexible model-building platform of the
12405 **BUGS** language can be beneficial in many situations.

12406 A common applied context of these multi-session models is when replicate arrays are
12407 used to address explicit hypotheses about the effects of landscape variation or modification
12408 on abundance. For example, in studies of forestry practices and their effects on local fauna,
12409 small mammal grids are used as experimental units, and the “dependent variable” is N
12410 (or density) of small mammals (or some small mammal focal species) for each trap array,

which is not observable. Thus, hierarchical models are needed to directly address the basic hypotheses of such studies. Another distinct context for the application of multi-session models is when the populations are temporally structured (e.g., the ovenbird data), such as when sampling occurs in distinct seasons or years. In these applications, we view multi-session models as a simplified type of open population model, an open model *without* explicit Markovian dynamics. They are analogous to what is usually referred to as models of random temporary emigration (Kendall et al., 1997; Chandler et al., 2011). The models are not incorrect, just simplified, reduced versions of more general Markovian models, and with fewer parameters to estimate. We cover general Markovian models in Chapt. 16.

12420
12421

15

12422
12423

MODELS FOR SEARCH-ENCOUNTER DATA

12424 In this chapter we discuss models for search-encounter data. These models are useful in
12425 situations where the locations of individuals, say \mathbf{u}_{ik} for individuals i and sample occasions
12426 k , are observed directly by searching space (often delineated by a polygon) in some
12427 fashion, rather than restricted to fixed trap locations. In all the cases addressed in this
12428 chapter, both detection probability and parameters related to movement can be estimated
12429 using such models. To formalize this notion a little bit using some of the ideas we've in-
12430 troduced in previous chapter, most of the SCR models we've talked about in the book
12431 involve just two components of a hierarchical model, the observation component, which
12432 we denote by $[y|\mathbf{s}]$ (e.g., Bernoulli, Poisson, or multinomial), and the process component
12433 describing the activity center model $[\mathbf{s}]$, the point process model for the activity centers.
12434 The search-encounter models described here involve an additional component for the loca-
12435 tions conditional on the activity centers. We write this as follows: The observation model
12436 has the form $[y|\mathbf{u}]$, and the process model has two components, a movement model $[\mathbf{u}|\mathbf{s}]$,
12437 which describes the individual encounter locations conditional on \mathbf{s} , and the point process
12438 model $[\mathbf{s}]$. Because we can resolve parameters of the $[\mathbf{u}|\mathbf{s}]$ component, search-encounter
12439 models are slightly more complicated, and also more biologically realistic. Conversely,
12440 when we have an array of fixed trap locations, the movement process is completely con-
12441 founded with the encounter process because the list of potential observation locations is
12442 prescribed, a priori, independent of any underlying movement process.

12443 A few distinct types of situations exist where search-encounter models come in handy.
12444 The prototypical, maybe ideal, situation Royle et al. (2011a) is where we have a single
12445 search path through a region of space from which observations are made (just as in the
12446 typical distance sampling situation, using a transect). As we walk along the search path,
12447 we note the location of each individual that is detected, *and their identity* (this is different
12448 from distance sampling in that sense). Alternatively, we could delineate a search area, and
12449 conduct a systematic search of that region. An example is that of Royle and Young (2008),
12450 which involved a plot search for lizards. They assumed the plot was uniformly searched
12451 which justified an assumption of constant encounter probability, p , for all individuals

12452 within the plot boundaries. The data set was ≥ 1 location observations for each of a
 12453 sample of n individuals. The recent paper by Efford (2011a) discussed likelihood analysis
 12454 of similar models. In the terminology of `secr` such models are referred to as models for
 12455 *polygon detectors*.

15.1 SEARCH-ENCOUNTER DESIGNS

12456 Before we discuss models for search-encounter data, we'll introduce some types of sampling
 12457 situations that produce individual location data by searching space. We imagine there are
 12458 a lot more sampling protocols (and variations) than identified here, but these are some
 12459 of the standard situations that we have encountered over the last few years in developing
 12460 applications of SCR models. For our purposes here we recognize 4 basic sampling designs,
 12461 each of which might have variations due to modification of the basic sampling protocol.

12462 15.1.1 Design 1: Fixed Search Path

12463 A useful class of models arises when we have a fixed search-path or line, or multiple such
 12464 lines, in some region (Fig. 15.1) from which individual detections are made. We assume the
 12465 survey path is laid out *a priori* in some manner that is done independent of the activity
 12466 centers of individuals and the collection of data does not affect the lines. The purpose
 12467 of this assumption, in the models described subsequently, is to allow us to assume that
 12468 the activity centers are uniformly distributed on the prescribed state-space. Alternatively,
 12469 explicit models could be entertained to mitigate a density gradient or covariate effects (see
 12470 Chapt. 11). The situation depicted in Fig. 15.1 shows the search path traversing several
 12471 delineated polygons, although the polygon boundaries may or may not affect the potential
 12472 locations of individuals (see below).

12473 A number of variations of this fixed search path situation are possible, and these
 12474 produce slightly different data structures and corresponding modifications to the model,
 12475 although we do not address all of these from a technical standpoint here:

- 12476 Protocol (1a). We know the search path and record the locations of individuals.
- 12477 Protocol (1b). We record the location of individuals and the location on the search path
 where we first observed the individual.
- 12479 Protocol (1c). We record the closest perpendicular distance. This is a typical distance
 sampling situation, and this is a type of hybrid SCR/distance sampling model.

12481 15.1.2 Design 2: Uniform search intensity

12482 In the uniform search intensity model (or just “uniform search”), we have one or more
 12483 well-defined sample areas (polygons), such as a quadrat or a transect, and we imagine that
 12484 the area is uniformly searched so that encounter probability is constant for all individuals
 12485 within the search area. This type of sampling method is often called “area search” in the
 12486 bird literature (Bibby et al., 1992). Sampling produces locations of individuals within the
 12487 well-defined boundaries of the sample area. The polygon boundaries defining the sample
 12488 unit are important because they tell us that $p = 0$ by design outside of the boundary.

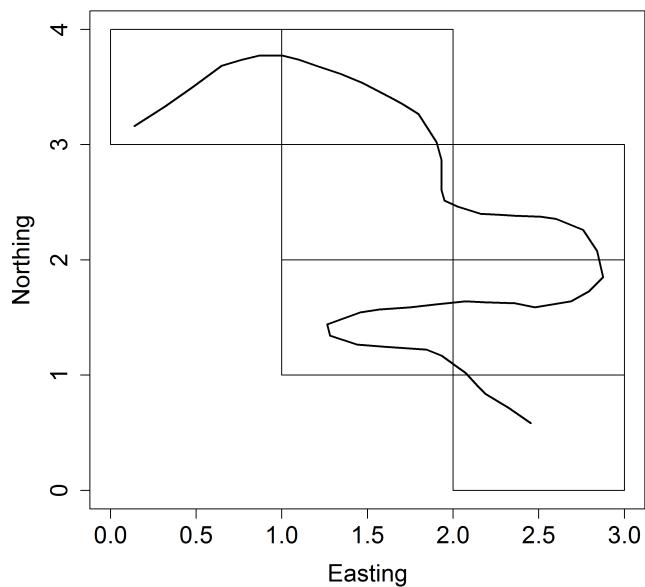


Figure 15.1. A survey line through parts of 7 quadrats in a hypothetical landscape. An observer travels the transect and identifies individuals in the vicinity of the line, recording their identity and location.

12489 Using the example from the Fig. 15.1, but ignoring the survey line through the plot
 12490 (pretend it doesn't exist), we imagine that each of the identified quadrats is uniformly
 12491 searched, which is to say, we assume that each individual within the boundaries of the
 12492 *quadrat* has an equal probability of being detected. In the context of replicate sampling
 12493 occasions (e.g., on consecutive days), individuals may move on or off of the plot, and so
 12494 individuals may have different probabilities of being *available* to encounter, based on the
 12495 closeness of their activity center to the quadrat boundaries. However, given that they're
 12496 available, the uniform search model assumes they have constant encounter probability.

15.2 A MODEL FOR FIXED SEARCH PATH DATA

12497 In contrast to most of the models described in this book (but see Sec. 9.4), we develop
 12498 models for encounter probability that depend explicitly on the instantaneous location \mathbf{u}_{ik} ,
 12499 for individual i at sample occasion k , say $p_{ik} \equiv p(\mathbf{u}_{ik}) = \Pr(y_{ik} = 1 | \mathbf{u}_{ik})$. Note that \mathbf{u} is
 12500 unobserved for the $y = 0$ observations and thus we cannot analyze the conditional-on- \mathbf{u}
 12501 likelihood directly. Instead, we regard \mathbf{u} as random effects and assume a model for them,
 12502 which allows us to handle the problem of missing \mathbf{u}_{ik} values (Sec. 15.4.1). We assume
 12503 that individuals do not move *during* a sampling occasion or, if they do, the individual is
 12504 not added to the data set twice.

12505 To develop encounter probability models for this problem we cannot just use the
 12506 previous models because the "trap" is actually a line or collection of line segments (e.g.,
 12507 Fig. 15.1). Intuitively, $\Pr(y_{ik} = 1 | \mathbf{u}_{ik})$ should increase as \mathbf{u}_{ik} comes "close" to the line
 12508 segments \mathbf{X} . It seems reasonable to express closeness by some distance metric $\|\mathbf{u}_{ik} - \mathbf{X}\|$
 12509 is the distance between locations \mathbf{u}_{ik} and \mathbf{X} , and then assume

$$\text{logit}(p_{ik}) = \alpha_0 + \alpha_1 \|\mathbf{u}_{ik} - \mathbf{X}\|.$$

12510 For the case where \mathbf{X} describes a wandering line, some kind of average distance from \mathbf{u} to
 12511 the line might be reasonable; possible alternatives include the absolute minimum distance
 12512 or the mean over specific segments of the line (within some distance), etc. . . . We could
 12513 also have a model without an explicit distance component, by assuming that individuals
 12514 within a certain distance from the search path are encountered with equal probability. In
 12515 this case, we have only a single parameter α_0 but must also specify the distance limit.

12516 15.2.1 Modeling total hazard to encounter

12517 Because the line \mathbf{X} is not a single point (like a camera trap) we have to somehow describe
 12518 the total encounter probability induced by the line. A natural approach is to model the
 12519 total hazard to capture (Borchers and Efford, 2008), which is standard in survival analysis,
 12520 and also distance sampling (Hayes and Buckland, 1983; Skaug and Schweder, 1999). The
 12521 individual is detected if encountered at any point along \mathbf{X} . Naturally, covariates are
 12522 modeled as affecting the hazard rate and we think of distance to the line as a covariate
 12523 acting on the hazard. Let $h(\mathbf{u}_{ik}, \mathbf{x})$ be the hazard of individual i being encountered by
 12524 sampling at a point \mathbf{x} on occasion t . For example, one possible model assumes, for all
 12525 points $\mathbf{x} \in \mathbf{X}$,

$$\log(h(\mathbf{u}_{ik}, \mathbf{x})) = \alpha_0 + \alpha_1 * \|\mathbf{u}_{ik} - \mathbf{x}\|. \quad (15.2.1)$$

Additional covariates could be included in the hazard function in the same way as for any model of encounter probability that we've discussed previously. The total hazard to encounter anywhere along the survey path, for an individual located at \mathbf{u}_{ik} , say $H(\mathbf{u}_{ik})$, is obtained by integrating over the surveyed line, which we will evaluate numerically by a discrete sum where the hazard is evaluated at the set of points \mathbf{x}_j along the surveyed path:

$$H(\mathbf{u}_{ik}) = \exp(\alpha_0) \left\{ \sum_{j=1}^J \exp(\alpha_1 * ||\mathbf{u}_{ik} - \mathbf{x}_j||) \right\} \quad (15.2.2)$$

where \mathbf{x}_j is the j^{th} row of \mathbf{X} defining the survey path as a collection of line segments which can be arbitrarily dense, but should be regularly spaced. Then the probability of encounter on a given sampling occasion is

$$p_{ik} \equiv p(\mathbf{u}_{ik}) = 1 - \exp(-H(\mathbf{u}_{ik})). \quad (15.2.3)$$

Its possible that the search path could vary by sampling occasion, say \mathbf{X}_k , which can easily be accommodated in the model simply by calculating the total hazard to encounter for each distinct search path.

This is a reasonably intuitive type of encounter probability model in that the probability of encounter is large when an individual's location \mathbf{u}_{ik} is close to the line in the average sense defined by Eq. 15.2.2, and vice versa. Further, consider the case of a single survey point, i.e., $\mathbf{X} \equiv \mathbf{x}$, which we might think of as a camera trap location. In this case note that Eq. (15.2.3) is equivalent to

$$\log(-\log(1 - p_{ik})) = \alpha_0 + \alpha_1 * ||\mathbf{u}_{ik} - \mathbf{x}||$$

which is to say that distance is a covariate on detection that is linear on the complementary log-log scale, which is similar to the "trap-specific" encounter probability of our Bernoulli encounter probability model (see Chapt. 5). The difference is that, here, the relevant distance is between the "trap" (i.e. the survey lines) and the individual's present location, \mathbf{u}_{ik} , which is observable. On the other hand, in the context of camera traps, the distance is that between the trap and a latent variable, \mathbf{s}_i , representing an individual's home range or activity center which is not observed.

A key assumption of this formulation of the model is that encounters at each point along the line, \mathbf{x}_j , are independent of each other point. Then, the event that an individual is encountered *at all* is the complement of the event that it is not encountered *anywhere* along the line (Hayes and Buckland, 1983). In this case, the probability of not being encountered at trap j is: $1 - p(\mathbf{u}_{ik}, \mathbf{x}_j) = \exp(-h(\mathbf{u}_{ik}, \mathbf{x}_j))$ and so the probability that an individual is not encountered at all is $\prod_j \exp(-h(\mathbf{u}_{ik}, \mathbf{x}_j))$. The encounter probability is therefore the complement of this, which is precisely the expression given by Eq. 15.2.3.

Any model for encounter probability can be converted to a hazard model so that encounter probability based on total hazard can be derived. We introduced this model above:

$$\log(h(\mathbf{u}_{ik}, \mathbf{x})) = \alpha_0 + \alpha_1 * ||\mathbf{u}_{ik} - \mathbf{x}||.$$

which is usually called the Gompertz hazard function in survival analysis, and it is most often written as $h(t) = a \exp(b * t)$ in which case $\log(h(t)) = \log(a) + b * t$. In the context

of survival analysis, t is “time” whereas, in SCR models, we model hazard as a function of distance. The Gaussian model has a squared-distance term:

$$\log(h(\mathbf{u}_{ik}, \mathbf{x})) = \alpha_0 + \alpha_1 * \|\mathbf{u}_{ik} - \mathbf{x}\|^2.$$

Borchers and Efford (2008) use this model:

$$h(\mathbf{u}_{ik}, \mathbf{x}) = -\log(1 - \text{expit}(\alpha_0) \exp(\alpha_1 * \|\mathbf{u}_{ik} - \mathbf{x}\|^2))$$

which produces a normal kernel model for *probability of detection* at the point level. i.e., $\Pr(y = 1) = 1 - \exp(-h) = h_0 \exp(\alpha_1 * \|\mathbf{u}_{ik} - \mathbf{x}\|^2)$ where $h_0 = \text{logit}^{-1}(\alpha_0)$. Another model is:

$$\log(h(\mathbf{u}_{ik}, \mathbf{x})) = \alpha_0 + \alpha_1 * \|\mathbf{u}_{ik} - \mathbf{x}\|$$

which is a Weibull hazard function.

15.2.2 Modeling movement outcomes

We have so far described the model for the encounter data in a manner that is conditional on the locations \mathbf{u}_{ik} , some of which are unobserved. Naturally, we should specify a model for these latent variables – i.e., a movement model – so that we could either do a Bayesian analysis by MCMC (Royle and Young, 2008; Royle et al., 2011a) or compute the marginal likelihood (Efford, 2011a). To develop such a model, we adopt what is now customary in SCR models – we assume that individuals are characterized by a latent variable, \mathbf{s}_i , which represents the activity center. This leads to some natural models for the movement outcomes \mathbf{u}_{ik} conditional on the activity center \mathbf{s}_i . Royle and Young (2008) used a bivariate normal model:

$$\mathbf{u}_{ik} | \mathbf{s}_i \sim \text{BVN}(\mathbf{s}_i, \sigma_{move}^2 \mathbf{I}),$$

where \mathbf{I} is the 2×2 identity matrix. We consider alternatives below. This is a primitive model of individual movements about their home range but we believe it will be adequate in many capture-recapture studies which are often limited by sparse data.

We adopt our default assumption for the activity centers \mathbf{s} :

$$\mathbf{s}_i \sim \text{Uniform}(\mathcal{S}); \quad i = 1, 2, \dots, N.$$

The usual considerations apply in specifying the state-space \mathcal{S} – either choose a large rectangle, or prescribe a habitat mask to restrict the potential locations of \mathbf{s} .

15.2.3 Simulation and analysis in JAGS

Here we will simulate a sample data set that goes with the situation described in Fig. 15.1 and then analyze the data in **JAGS**. We begin by defining the state-space containing all of the grid cells in the rectangle $[-1, 4] \times [-1, 5]$, which contains 30 1×1 cells. The survey line in Fig. 15.1 traverses 7 of those 1×1 boxes. We define the total population to be 4 individuals per grid cell (1×1). To set this up in **R**, we do this:

```
12591 > xlim <- c(-1, 4)
12592 > ylim <- c(-1, 5)
12593 > perbox <- 4
12594 > N <- 30*perbox # Total of 30 1x1 quadrats
```

12595 The line in Fig. 15.1 is an irregular mesh of points obtained by an imperfect manual
 12596 point-and-clicking operation, which mimics the way in which GPS points come to us. In
 12597 order to apply our model we need a regular mesh of points. We can obtain a regular
 12598 mesh of points from the irregular mesh by using some functions in the packages `rgeos` and
 12599 `sp`, especially the function `sample.Line`, which produces a set of equally-spaced points
 12600 along a line. The **R** commands are as follows (the complete script is given in the function
 12601 `snakeline`):

```
12602 > library(rgeos)
12603 > library(sp)
12604 > line1 <- source("line1.R")
12605
12606 > line1 <- as.matrix(cbind(line1$value$x,line1$value$y))
12607 > points <- SpatialPoints(line1)
12608
12609 > sLine <- Line(points)
12610 > regpoints <- sample.Line(sLine,250,type="regular") # Key step!
```

12611 Next, we set a random number seed, simulate activity centers and set some model parameters
 12612 required to simulate encounter history data. In the following commands you can see
 12613 where the regular mesh representation of the sample line is extracted from the `regpoints`
 12614 object which we just created:

```
12615 > set.seed(2014)
12616 > sx <- runif(N,xlim[1],xlim[2])
12617 > sy <- runif(N,ylim[1],ylim[2])
12618
12619 > sigma.move <- .35
12620 > sigma <- .4
12621 > alpha0 <- .8
12622 > alpha1 <- 1/(2*(sigma^2))
12623 > X <- regpoints@coords
12624 > J <- nrow(X)
```

12625 Next we're going to simulate data which we do in 2 steps: For each individual in the
 12626 population and for each of K sample occasions, we simulate the location of the individual
 12627 as a bivariate normal random variable with mean \mathbf{s}_i and $\sigma_{move} = 0.35$. Next, we compute
 12628 the encounter probability model using Eq. 15.2.3, with the bivariate normal hazard model,
 12629 and then retain the data objects corresponding to individuals that get captured at least
 12630 once. All of this goes according to the following commands:

```
12631 > K <- 10 ## Sample occasions = 10
12632 > U <- array(NA,dim=c(N,K,2)) ## Array to hold locations
12633 > y <- pmat <- matrix(NA,nrow=N,ncol=K) ## Initialize
12634 > for(i in 1:N){
12635 +   for(k in 1:K){
12636 +     U[i,k,] <- c(rnorm(1,sx[i],sigma.move),rnorm(1,sy[i],sigma.move))
12637 +     dvec <- sqrt( (U[i,k,1] - X[,1])^2 + (U[i,k,2] - X[,2])^2 )
```

```

12638 +      loghaz <- alpha0 - alpha1*dvec*dvec
12639 +      H <- sum(exp(loghaz))
12640 +      pmat[i,k] <- 1-exp(-H)
12641 +      y[i,k] <- rbinom(1,1,pmat[i,k])
12642 >    }
12643 >  }
12644 > Ux <- U[,1]
12645 > Uy <- U[,2]
12646 > Ux[y==0] <- NA
12647 > Uy[y==0] <- NA

```

12648 In the commands shown above, we define matrices, U_x and U_y , that hold the observed
 12649 locations of individuals during each occasion. Note that, if an individual is *not* captured,
 12650 we set the value to `NA`. We pass these partially observed objects to **JAGS** to fit the model.

12651 Finally, we do the data augmentation and we make up some starting values for the
 12652 location coordinates that are missing. For these, we cheat a little bit (for convenience and
 12653 hopefully to improve the efficiency of the MCMC for the simulated data sets) and use the
 12654 actual activity center values. In practice, we might think about using the average of the
 12655 observed locations.

```

12656 > ncap <- apply(y,1,sum)
12657 > y <- y[ncap>0,]
12658 > Ux <- Ux[ncap>0,]
12659 > Uy <- Uy[ncap>0,]

12660
12661 > M <- 200
12662 > nind <- nrow(y)
12663 > y <- rbind(y,matrix(0,nrow=(M-nrow(y)),ncol=ncol(y)))
12664 > Namat <- matrix(NA,nrow=(M-nind),ncol=ncol(y))
12665 > Ux <- rbind(Ux,Namat)
12666 > Uy <- rbind(Uy,Namat)
12667 > S <- cbind(runif(M,xlim[1],xlim[2]),runif(M,ylim[1],ylim[2]))
12668 > for(i in 1:nind){
12669 +   S[i,] <- c( mean(Ux[i,],na.rm=TRUE),mean(Uy[i,],na.rm=TRUE))
12670 > }
12671 > Ux.st <- Ux
12672 > Uy.st <- Uy
12673 > for(i in 1:M){
12674 +   Ux.st[i,!is.na(Ux[i,])]<-NA
12675 +   Uy.st[i,!is.na(Uy[i,])]<-NA
12676 +   Ux.st[i,is.na(Ux[i,])]<-S[i,1]
12677 +   Uy.st[i,is.na(Uy[i,])]<-S[i,2]
12678 + }

```

12679 The **BUGS** model specification is shown in Panel 15.1, although we neglect the stan-
 12680 dard steps showing how to bundle the `data`, `inits`, and farm all of this stuff out to **JAGS**
 12681 (see the help file for `snakeline` for the complete script). Simulating the data as described
 12682 above, and fitting the model in Panel 15.1 produces the results in Table 15.1.

```

model {

  alpha0~dunif(-25,25)          # Priors distributions
  alpha1~dunif(0,25)
  lsigma~dunif(-5,5)
  sigma.move<-exp(lsigma)
  tau<-1/(sigma.move*sigma.move)
  psi~dunif(0,1)

  for(i in 1:M){ # Loop over individuals
    z[i]~dbern(psi)
    s[i,1]~dunif(xlim[1],xlim[2])   # Activity center model
    s[i,2]~dunif(ylim[1],ylim[2])
    for(k in 1:K){                 # Loop over sample occasions
      ux[i,k] ~ dnorm(s[i,1],tau)  # Movement outcome model
      uy[i,k] ~ dnorm(s[i,2],tau)
      for(j in 1:J){ # Loop over each point defining line segments
        d[i,k,j]<- pow(pow(ux[i,k]-X[j,1],2) + pow(uy[i,k]-X[j,2],2),0.5)
        h[i,k,j]<-exp(alpha0-alpha1*d[i,k,j]*d[i,k,j])
      }
      H[i,k]<-sum(h[i,k,1:J])       # Total hazard H
      p[i,k]<- z[i]*(1-exp(-H[i,k]))
      y[i,k] ~ dbern(p[i,k])
    }
  }
  # Population size is a derived quantity
  N<-sum(z[])
}

```

Panel 15.1: **BUGS** model specification for the fixed search path model, based on that from Royle et al. (2011a). See the help file `?snakeline` for the **R** code to simulate data and fit this model.

Table 15.1. Posterior summary statistics for the simulated fixed search path data. These are based on 3 chains, and a total of 9000 posterior samples. The data generating parameter values were $N = 100$, $\sigma_{move} = 0.35$, $\sigma = 0.4$, and $\alpha_0 = 0.8$. The parameter $\alpha_1 = 1/(2\sigma^2)$.

Parameter	Mean	SD	2.5%	50%	97.5%	Rhat
N	117.626	5.675	107.000	117.000	129.000	1.015
α_0	1.305	0.494	0.425	1.280	2.387	1.009
α_1	3.806	0.423	3.050	3.777	4.733	1.008
σ_{move}	0.347	0.008	0.332	0.347	0.364	1.023
σ	0.364	0.020	0.325	0.364	0.405	1.008
ψ	0.587	0.044	0.501	0.588	0.673	1.006

12683 15.2.4 Hard plot boundaries

12684 The previous development assumed that locations of individuals can be observed anywhere
 12685 in the state-space, determined only by the encounter probability model as a function of
 12686 distance from the search path. However, in many situations, we might delineate a plot
 12687 which restricts where individuals might be observed (as in the situation considered by
 12688 Royle and Young (2008)). For such cases we truncate the encounter probability function
 12689 at the plot boundary, according to:

$$p(\mathbf{u}_{ik}) = (1 - \exp(-H(\mathbf{u}_{ik})))I(\mathbf{u}_{ik} \in \mathcal{X}) \quad (15.2.4)$$

12690 where \mathcal{X} is the surveyed polygon and the indicator function $I(\mathbf{u}_{ik} \in \mathcal{X}) = 1$ if $\mathbf{u}_{ik} \in \mathcal{X}$
 12691 and 0 otherwise. That is, the probability of encounter is identically 0 if an individual
 12692 is located *outside* the plot at sample period t . We demonstrated how to do this in the
 12693 **BUGS** language below for a model of uniform search intensity (area-search model).

12694 15.2.5 Analysis of other protocols

12695 In the situation elaborated on above (what we called “Protocol 1a”), the sample path is
 12696 used to locate individuals and, whether or not an individual is encountered, is a function
 12697 of the total hazard to encounter along the whole line. We think there are a number of
 12698 variations of this basic design that might arise in practice. A slight variation (what we
 12699 called “Protocol 1b”) is based on recording location of individuals and also the location
 12700 on the transect where we observed the individual. The probability of encounter is the
 12701 probability of encounter prior to the point on the line where the detection takes place
 12702 (Skaug and Schweder, 1999). This is exactly a distance-sampling observation model, but
 12703 with an additional hierarchical structure that describes the individual locations about their
 12704 activity centers. There are no additional novel considerations in analysis of this situation
 12705 compared to Protocol 1a, and so we have not given it explicit consideration here. Similarly,
 12706 “Protocol 1c” is a slight variation of this – instead of recording the point on the line where
 12707 the individual was first detected, we use, instead, the point on the line that has the shortest
 12708 perpendicular distance. This is a classical distance sampling observation model, and it
 12709 represents an intentional misspecification of the model but it seems that the effect of this
 12710 is relatively minor, or, otherwise, we imagine people wouldn’t do it.

15.3 UNSTRUCTURED SPATIAL SURVEYS

12711 A common situation in practice is that in which sampling produces a survey path, but
 12712 the path was not laid out *a priori* but, rather evolves opportunistically during the course
 12713 of sampling, a situation we'll call an unstructured spatial survey (Thompson et al., 2012;
 12714 Russell et al., 2012). We imagine that the survey path evolves in response to information
 12715 about animal presence, which could be both the number of unique individuals or the
 12716 amount of sign in the local search area. The motivating problem has to do with area
 12717 searches using dog teams, in which the dogs usually wander around hunting scat, and their
 12718 search path is based on how they perceive the environment and what they're smelling.
 12719 This violates the main assumptions that the line is placed *a priori*, independent of density
 12720 and unrelated to detectability.

12721 The analysis framework implemented by Thompson et al. (2012) and Russell et al.
 12722 (2012) is based on a heuristic justification wherein the sampling of space is imagined
 12723 to have been grid-structured, with grid cells that are large enough so that dogs are not
 12724 influenced by scat or sign beyond the specific cell being searched. Then, we assume the dog
 12725 applies a consistent search strategy to each cell so that that resulting cell-level detections
 12726 can be regarded as independent Bernoulli trials with probability p_{ij} depending on the
 12727 distance $\|\mathbf{x}_j - \mathbf{s}_i\|$ between the grid cell with center \mathbf{x}_j , and individual with activity
 12728 center \mathbf{s}_i and the amount of search effort (or length of the search route) within a cell.
 12729 In other words, we use an ordinary SCR type of model but treating the center point of
 12730 each cell as an effective "trap". The deficiency with this approach is that some of the
 12731 "sub-grid" resolution information about movement is lost, so we probably lose precision
 12732 about any parameters of the movement model when the cells are large relative to a typical
 12733 home range size. We discuss a couple of examples below.

12734 15.3.1 Mountain lions in Montana

12735 Russell et al. (2012) analyzed mountain lion (*Puma concolor*) encounter history data to
 12736 assess the status of mountain lions in the Blackfoot Mountains of Montana. The data
 12737 collection was based on opportunistic searching by hunters with dogs, who tree the lion
 12738 (Fig. 15.2). Tissue is extracted with a biopsy dart and analyzed in the lab for individual
 12739 identity. They used $5 \text{ km} \times 5 \text{ km}$ grid cells for binning the encounters, and the length
 12740 of the search path in each grid cell as a covariate of effort (C_j) that each grid cell was
 12741 searched. The model is the Gaussian hazard model with baseline encounter probability
 12742 that depended on sex and effort in each grid cell, on the log scale:

$$\log(\lambda_{0,ij}) = \alpha_0 + \alpha_2 \log(C_j) + \alpha_3 \text{Sex}_i$$

12743 Note for grid cells that were not searched, $C_j = 0$ and, for those, the constraint $\lambda_{0,ij} = 0$
 12744 was imposed so that the probability of encounter was identically 0.

12745 One problem encountered by Russell et al. (2012) in their analysis is the possibility
 12746 of dependence in encounters because of group structure in the data (usually, juveniles in
 12747 association with their mother). In this situation, in addition to dependence of encounter,
 12748 multiple individuals have effectively the same activity center, thus violating a number of
 12749 assumptions related to the ordinary SCR model. To resolve this problem, the authors
 12750 made some assumptions about group association and fitted models where group served as
 12751 the functional individual.



Figure 15.2. Mountain lion. Run! Photo credit: Bob Wiesner.

12752 **15.3.2 Sierra National Forest Fisher Study**

12753 Here we consider a more detailed example and provide the data and **R** script for this
 12754 analysis. The data come from an analysis of individual encounter histories of the fisher
 12755 (*Martes pennanti*) by Thompson et al. (2012). The survey area was divided into 15 ap-
 12756 proximately 1,400 ha hexagons (Fig. 15.3), which is roughly the size of a female fisher's
 12757 home range, and each hexagon was surveyed 3 times by sniffer dog teams searching space
 12758 for scat. The dogs were given considerably latitude to determine their route. Thus, the
 12759 search path is not laid out a priori but rather evolves opportunistically, based on what
 12760 the dog senses at a local scale. The authors divided the region into 1 km grid cells (also
 12761 shown in Fig. 15.3).

12762 We provide the data from this study in the **scrbook** package, and it can be loaded with
 12763 the command **data(fisher)**. The **R** script **SCRfisher** produces the posterior summary
 12764 statistics shown in Table 15.2. One thing is relatively poor mixing of the Markov chains
 12765 here due to sparse data and a fairly long run is probably necessary.

15.4 DESIGN 2: UNIFORM SEARCH INTENSITY

12766 A special case of a search-encounter type of model arises when it is possible to subject
 12767 a quadrat (or quadrats) to a uniform search intensity. This could be interpreted as an
 12768 exhaustive search, or perhaps just a thorough systematic search of the available habitat.
 12769 The example considered by Royle and Young (2008) involved searching a 9 ha plot for
 12770 horned lizards (Fig. 15.4) by a crew of several people. It was felt in that case that complete

Table 15.2. Posterior summary statistics for the fisher study data, based on 30000 posterior samples. Here $\lambda_0 = \exp(\alpha_0)$. This example exhibits relatively poor mixing due to sparse data, and the Rhat statistic should be reduced by obtaining a larger posterior sample.

Parameter	Mean	SD	2.5%	50%	97.5%	Rhat
N	315.889	230.041	12.000	280.000	738.775	1.133
σ	4.745	2.909	0.163	4.650	9.704	1.020
λ_0	0.003	0.033	0.000	0.000	0.016	1.097
α_1	0.188	0.170	0.005	0.138	0.641	1.002
ψ	0.413	0.300	0.016	0.366	0.964	1.131

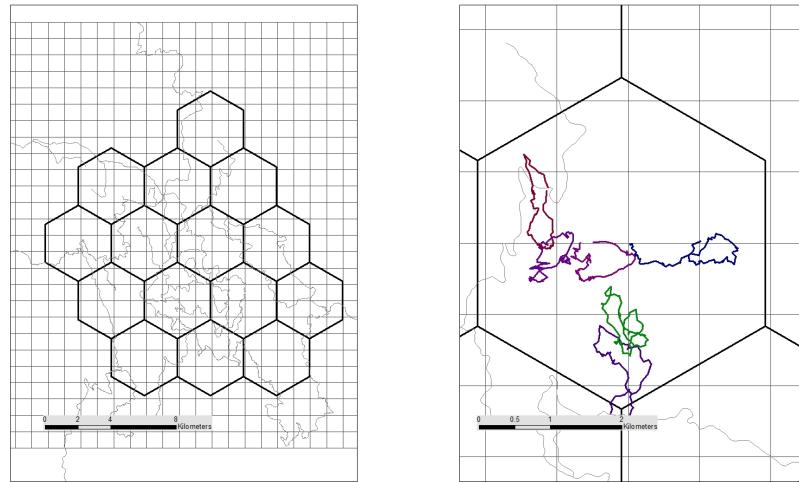


Figure 15.3. Fisher study area showing the gridding system (left panel). The larger hexagons are approximately the size of a typical female home range. The 1 km grid cells define the SCR model grid, where the center point of each one served as a trap. The right panel shows the GPS trackline of the dog team through one of the grid cells. The total length of the trackline was used as a covariate on encounter probability. Credit: Craig Thompson, U.S. Forest Service

12771 and systematic (i.e., uniform) coverage of the plot was achieved. In general, however, we
 12772 think you could have a random sample of the plot and approximate that as a uniform
 12773 coverage – this is kind of a design-based argument justifying the uniform search intensity
 12774 model (we haven't simulated this situation, but it would be worth investigating).



Figure 15.4. A flat-tailed horned lizard showing its typical cryptic appearance in its native environment. Detection of flat-tailed horned lizards is difficult because they do not run when approached. Instead they shuffle under the sand or press down and remain motionless as shown in the picture. The horns are employed only as a last resort if the camouflage fails. *Photo credit: Kevin and April Young*

12775 It is clear that this uniform search intensity model is a special case of the fixed search
 12776 path model in the sense that the probability of encounter of an individual is a constant
 12777 p_0 if the individual is located in the polygon \mathcal{X} during sample occasion k , i.e.,

$$p(\mathbf{u}_{ik}) = p_0 I(\mathbf{u}_{ik} \in \mathcal{X})$$

12778 which resembles Eq. 15.2.4 except replacing the encounter probability function with con-
 12779 stant p_0 .

12780 Subsequently, we give a simple analysis using simulated data and simple movement
 12781 models for \mathbf{u} , including a bivariate normal model and a random walk. For further examples
 12782 and analyses, we refer you to Royle and Dorazio (2008), who reanalyzed the lizard data
 12783 from Royle and Young (2008), and Efford (2011b) and Marques et al. (2011).

12784 **15.4.1 Alternative movement models**

12785 As with the general fixed search path model (“Design 1”), we require a model to describe
 12786 the movement outcomes \mathbf{u}_{ik} . In the analysis of Royle and Young (2008), a simple bivariate

12787 Gaussian movement model was used, in which

$$\mathbf{u}_{ik} | \mathbf{s}_i \sim \text{Normal}(\mathbf{s}_i, \sigma_{move}^2 \mathbf{I}),$$

12788 However, clearly more general versions of the model can be developed. For example, imagine
 12789 a situation where the successive surveys of a bounded sample polygon are relatively
 12790 close together in time so that successive locations of individuals are not well-approximated
 12791 by the Gaussian movement model, which implies independence of locations. Naturally we
 12792 might consider using an auto-regressive or random-walk type of model in which the suc-
 12793 cessive coordinate locations of individual i behave as follows:

$$\begin{aligned} u_{1,i,k} | u_{1,i,k-1} &\sim \text{Normal}(u_{1,i,k-1}, \sigma_{move}^2) \\ u_{2,i,k} | u_{2,i,k-1} &\sim \text{Normal}(u_{2,i,k-1}, \sigma_{move}^2) \end{aligned}$$

12794 here we use the notation u_1 and u_2 for the easting and northing coordinates, respectively.
 12795 (and, for clarity, we are using commas in the sub-scripting here when we have to refer to
 12796 time-lags). In addition, we require that the initial locations have a distribution and, for
 12797 that, we might begin with a simple model such as the uniformity model:

$$\mathbf{u}_{i,1} \sim \text{Uniform}(\mathcal{S})$$

12798 which effectively takes the place of the model for \mathbf{s}_i that we typically use. Under this
 12799 model, individuals don't have an activity center but, rather, they drift through space
 12800 more-or-less randomly based just on their previous location. See Ovaskainen (2004) and
 12801 Ovaskainen et al. (2008) for development and applications of similar movement models
 12802 in the context of capture-recapture data, and also our discussion of a similar model that
 12803 might arise in acoustic surveys (Sec. 9.4). We could allow for dependent movements
 12804 about a central location \mathbf{s}_i using a bivariate auto-regression or similar type of model with
 12805 parameter ρ , e.g.,

$$\mathbf{u}_{i,k} | \mathbf{s}_i \sim \text{BVN}(\rho * (\mathbf{u}_{i,k-1} - \mathbf{s}_i), \sigma_{move}^2 \mathbf{I}).$$

12806 We don't have any direct experience fitting these movement models to real capture-
 12807 recapture data, but we imagine they should prove effective in applications that yield large
 12808 sample sizes of individuals and recaptures.

12809 15.4.2 Simulating and fitting uniform search models

12810 The **R** script `uniform_search`, in the `scrbook` package, we provide a script for simulating
 12811 and fitting search-encounter data using the iid Gaussian model and also the random walk
 12812 model. The **BUGS** model specification is shown in Panel 15.2 for the random walk
 12813 situation. We encourage you to adapt this model and the simulation code for the auto-
 12814 regression movement model. To fit this model to data, we set up the run with **JAGS** using
 12815 the standard commands. We did not specify starting values for the missing coordinate
 12816 locations although we imagine that **JAGS** should perform better if we provide decent
 12817 starting values, e.g., the last observed location or some other reasonable location. We
 12818 imagine that resource selection could be parameterized in this movement model as well,
 12819 perhaps using similar ideas to those described in Chapt. 13.

12820 The following script simulates a population of N individuals and their locations at
 12821 each of 4 times to see if they are in a square [3,13] or not. This simulates a random walk
 12822 thing so we imagine that the sampling occasions are close together in time. The initial
 12823 state is assumed to be uniformly distributed on the state-space which, in this case, is the
 12824 square $[0, 16] \times [0, 16]$. We store the movement outcomes here in a 3-d array U, instead of
 12825 in two separate 2-d arrays (one for each coordinate) as we did above. The R commands
 12826 are as follows:

```

12827 > N <- 100
12828 > nocc <- 4
12829 > Sx <- Sy <- matrix(NA,nrow=N,ncol=nocc)
12830 > sigma.move <- .25
12831
12832 # Simulate initial coordinates on the square:
12833 > Sx[,1] <- runif(N,0,16)
12834 > Sy[,1] <- runif(N,0,16)
12835
12836 > for(t in 2:nyear){
12837 +   Sx[,t] <- rnorm(N,Sx[,t-1],sigma.move)
12838 +   Sy[,t] <- rnorm(N,Sy[,t-1],sigma.move)
12839 + }
12840
12841 # Now we generate encounter histories on a search rectangle
12842 #   with sides [3,13]:
12843 > Y <- matrix(0,nrow=N,ncol=nyear)
12844 > for(i in 1:N){
12845 +   for(t in 1:nyear){
12846 +     # IF individual is in the sample unit we can capture it:
12847 +     if( Sx[i,t] > 3 & Sx[i,t]< 13 & Sy[i,t]>3 & Sy[i,t]<13 )
12848 +       Y[i,t] <- rbinom(1,1,.5)
12849 +   }
12850 + }
12851
12852 # Subset data. If an individual is never captured, cannot have him in our data set
12853 > cap<- apply(Y,1,sum) > 0
12854 > Y <- Y[,cap,]
12855 > Sx <- Sx[,cap,]
12856 > Sy <- Sy[,cap,]
12857
12858 > Sx[Y==0] <- NA
12859 > Sy[Y==0] <- NA
12860
12861 ## Data augmentation:
12862 > M <- 200
12863 > Y <- rbind(Y,matrix(0,nrow=(M-nrow(Y)),ncol=nyear))
12864 > Sx <- rbind(Sx,matrix(NA,nrow=(M-nrow(Sx)),ncol=nyear))
12865 > Sy <- rbind(Sy,matrix(NA,nrow=(M-nrow(Sy)),ncol=nyear))
```

```

12866
12867 # Make 3-d array of coordinates "U"
12868 > U <- array(NA,dim=c(M,nyear,2))
12869 > U[,,1] <- Sx
12870 > U[,,2] <- Sy

```

```

model{
psi ~ dunif(0,1)                                # Prior distributions
tau ~ dgamma(.1,.1)
p0 ~ dunif(0,1)
sigma.move <- sqrt(1/tau)

for (i in 1:M){
  z[i] ~ dbern(psi)
  U[i,1,1] ~ dunif(0,16)                         # Initial location
  U[i,1,2] ~ dunif(0,16)

  for (k in 2:n.occasions){
    U[i,k,1] ~ dnorm(U[i,k-1,1], tau)
    U[i,k,2] ~ dnorm(U[i,k-1,2], tau)
  }
  for(k in 1:n.occasions){
    # Test whether the actual location is in- or outside the
    # survey area. Needs to be done for each grid cell
    inside[i,k] <- step(U[i,k,1]-3) * step(13-U[i,k,1]) *
      step(U[i,k,2]-3) * step(13-U[i,k,2])
    Y[i,k] ~ dbern(mu[i,k])
    mu[i,k] <- p0 * inside[i,k] * z[i]
  }
}
N <- sum(z[])                                     # Population size, derived
}

```

Panel 15.2: **BUGS** model specification for the uniform search intensity model similar to Royle and Young (2008) but with a random walk movement model. help file `?uniform_search` in the **R** package `scrbook`.

15.4.3 Movement and Dispersal in Open Populations

In Chapt. 16 we discuss many aspects of modeling open populations, including some aspects of modeling movement and dispersal and the relevance of SCR models to these

12874 problems. However, given the introduction of the uniform search model above, this is
 12875 clearly relevant to modeling movement and dispersal in open populations. In particular,
 12876 the model described in Panel 15.2 could easily be adapted to an open population by
 12877 conditioning on the first, and introducing a latent “alive state” with survival parameter
 12878 ϕ_t . This would be a spatial version of the standard Cormack-Jolly-Seber model (Chapt.
 12879 16.3)¹.

15.5 PARTIAL INFORMATION DESIGNS

12880 The prototype search-encounter (Design 1) and uniform search (Design 2) cases are ideal
 12881 in the sense that they produce both precise locations of individuals and also a precise
 12882 characterization of the manner in which individuals are encountered by sampling space.
 12883 We have seen a number of studies that, in an ideal world, would have generated data
 12884 consistent with one of these situations but, for some practical reason or other reason,
 12885 partial or no spatial information about the search area or the locations of individuals was
 12886 collected (or retained), and so the models described above could not be used. We imagine
 12887 (indeed, have encountered) at least 3 distinct situations:

- 12888 (a) The search path is not recorded, but locations of individuals are recorded
- 12889 (b) The search path is recorded, but locations of individuals are not.
- 12890 (c) The search path is not recorded, and the locations are not recorded, just raw sum-
 12891 maries for prescribed areas or polygons.

12892 For analysis of these search-encounter designs with partial information, we see a num-
 12893 ber of options of varying levels of formality, depending on the situation (and these are
 12894 largely untested). For (a) You could always assume uniform search intensity, which might
 12895 be reasonable if the plots were randomly searched. Otherwise, its validity would depend
 12896 on the precise manner in which the search activity occurred. For (b) or (c), we could
 12897 adopt the approach we took in the fisher analysis above, and map the locations to the
 12898 center of each plot, thinking of the plot as an effective trap, and using the search path
 12899 length as a covariate. A 4th case with even less information is that in which we don’t
 12900 record individual identity at all. Instead, we just have total count frequencies in each plot.
 12901 This model is precisely the one considered by (Chandler and Royle, In press) and this is
 12902 the focus of Chapt. 18.

15.6 SUMMARY AND OUTLOOK

12903 The generation of spatial encounter history data in ecological studies is widespread. While
 12904 such data have historically been obtained mostly by the use of arrays of fixed traps (catch
 12905 traps, camera traps, etc..), in this chapter we showed that SCR models are equally rel-
 12906 evant to a large class of “search-encounter” problems which are based on organized or
 12907 opportunistic searches of spatial areas. Standard examples include “area searches” in bird
 12908 population studies, use of detector dogs to obtain scat samples, from which DNA can
 12909 be obtained to determine individual identity, or sampling along a fixed search path (or
 12910 transects) by observers noting the locations of detected individuals (this is common in

¹Some work related to this is currently being carried out by our colleagues Torbjørn Ergon and Michael Schaub.

12911 sampling for reptiles and amphibians). The latter situation closely resembles distance
12912 sampling but, with repeated observations of the same individual (on multiple occasions),
12913 it has a distinct capture-recapture element to it. In a sense, the fixed search path models
12914 are hybrid SCR-DS models.

12915 Many models for search-encounter data have three elements in common. They contain:
12916 (1) a model for encounter conditional on locations of individuals; (2) a model that describes
12917 how these observable animal locations are distributed in space about their activity centers;
12918 and (3) a model for the distribution of activity centers. We interpret the 2nd model
12919 component as an explicit movement model, and the existence of this component is distinct
12920 from most of the other models considered in this book. One of the key conceptual points
12921 is that, with these search-encounter types of designs, the locations of observations are *not*
12922 biased by the locations of traps but, rather, locations of individuals can occur anywhere
12923 within search plots or quadrats, or in the vicinity of a transect or search path. Because we
12924 can obtain direct observations of location – outcomes of movement – for individuals, it is
12925 possible to resolve explicit models of movement from search-encounter data. We considered
12926 the simple case of the independent bivariate normal movement model, and also a random
12927 walk type model, which can easily be fitted in the **BUGS** engines. We imagine much
12928 more general movement models can be fitted, although we have had limited opportunities
12929 to pursue this and in most practical capture-recapture studies, we will probably be limited
12930 by sparse data in the complexity of the movement models that could be considered.

12931 Search-encounter sampling is fairly common, although we think that many people don't
12932 realize that it can produce encounter history data that is amenable to the development
12933 of formal models for density, movement and space usage. We believe that these protocols
12934 will become more appealing as methods for formal analysis of the resulting encounter
12935 history data become more widely known. At the same time, search-encounter models will
12936 increase in relevance in future studies of animal populations because so many new methods
12937 of obtaining encounter history data can be based on DNA extracted from animal tissue
12938 or scat, which is easy to obtain by searching space opportunistically. In addition, as the
12939 cost of obtaining individual identity from scat or tissue decreases, its widespread collection
12940 and use in capture-recapture models can only increase.

12941
12942

12943

16

OPEN POPULATION MODELS

16.1 INTRODUCTION

In the previous chapters we focused mostly on closed population models for estimating density and for inference about spatial variation in density and space usage. However, a thorough understanding of population dynamics requires information about both spatial *and* temporal variation in population density and demographic parameters. In this chapter, we discuss modeling the processes governing spatial and temporal population dynamics, namely survival, recruitment, and movement over larger temporal scales (e.g., migration, dispersal, etc...). The ability to estimate these parameters is critical to both basic and applied ecological research (Knape, 2012). For example, testing hypotheses about life history trade-offs requires accurate estimates of both survival and fecundity (Caswell, 1989; Nichols et al., 1994). Inference about density-dependent population regulation, which has fascinated theoretical ecologists for well over a century, is likewise best accomplished by directly studying the factors affecting survival and fecundity, rather than the more common approach of modeling time series data (Nichols et al., 2000b). A mechanistic understanding of population changes, which is essential for basic ecological and conservation related questions, requires useful models of vital rates. Furthermore, if we know how environmental variables affect demographic parameters, we can make predictions about population changes under different future scenarios. We can also assess the sensitivity of parameters such as population growth rate to variation in survival or fecundity. Although matrix population models are often used for these purposes (Caswell, 1989; Sæther and Bakke, 2000), the same objectives can be accomplished by computing posterior predictive distributions of projected population sizes as part of the MCMC algorithm.

The modeling framework we will develop in this chapter is based on a formulation of the classical Cormack-Jolly-Seber (CJS) and Jolly-Seber (JS) type models (Cormack, 1964; Jolly, 1965; Seber, 1965) that are amenable to modeling individual effects, including individual covariates. There is a long history of use of these models in fisheries, wildlife, and ecology studies (Pollock et al., 1990; Lebreton et al., 1992; Pradel, 1996; Williams et al., 2002; Schwarz and Arnason, 2005; Gimenez et al., 2007). Additionally, there have

12972 been many modifications and developments of the CJS and JS models including dealing
 12973 with individuals that do not have a well defined home range but instead are moving
 12974 through the sampled area (transients), dealing with more than one site or state (multi-state
 12975 models, where states maybe geographic units, reproductive stage, etc.), and addressing
 12976 individual movement through spatially implicit models.

12977 For the first time, these models can fully integrate the movement of individuals in the
 12978 vicinity of the trap array with their encounter histories to simultaneously estimate density,
 12979 survival, and recruitment in a spatial model. For many species, such as those that are rare
 12980 or not often observed by researchers, this allows inferences to be made about survival and
 12981 recruitment without having to physically capture individuals. Additionally, another reason
 12982 for extending SCR models to open populations arises purely from a sampling perspective.
 12983 Longer time periods are often needed to sample rare or elusive species to ensure that
 12984 enough captures and recaptures are produced. This extended time frame can quickly lead
 12985 to violations in the assumption of population closure (see also Chapt. 10). For example,
 12986 the European wildcat study that was mentioned in Chapt. 7 (see Kéry et al. (2011) for
 12987 details) was conducted over a year-long period. While the researchers in that study used
 12988 a closed population model, they did model variation in detection as a function of time
 12989 to account for seasonal variation in behavior. Another approach would have been to use
 12990 an open population model to account for possible changes in the population over time
 12991 (however, the spatial capture recapture open models had not been developed at the time
 12992 of the wildcat study, so we'll forgive the authors for not having used this more appropriate
 12993 model).

12994 In this chapter, we present the traditional JS model and the spatial version, demon-
 12995 strating both with an example of mist-netting of ovenbirds, which was also analyzed in
 12996 Chapt. 9. Then we review the traditional CJS, multi-state CJS, and then describe the
 12997 spatial model. In this section, we will use a an example of American shad. Finally, we
 12998 end by discussing some of the new approaches to dynamics of activity centers including
 12999 correlated movement and dispersal.

13000 16.1.1 Brief overview of population dynamics

13001 The most basic formulation of models for population growth stems from an idea originally
 13002 used in accounting, the balance sheet (see ?, Chapt. 3 for a more complete description). To
 13003 gain a mechanistic understanding of population dynamics, it is important to understand
 13004 four fundamental processes that drive population size: births and immigrants (i.e., popu-
 13005 lation “credits”) and deaths and emigrants (i.e., population “debits”). The population at
 13006 time $t + 1$ is a function of these four components:

$$N(t + 1) = N(t) + B(t) + I(t) - D(t) - E(t)$$

13007 where $N(t)$ is the population size at time t , $B(t)$ and $I(t)$ are the credits (additions)
 13008 from births and immigrants at time t , and $D(t)$ and $E(t)$ are the debits (losses) due to
 13009 deaths and emigration. This balance equation model is known as the “BIDE model”. A
 13010 simple population growth model under density independence, assuming no immigration
 13011 or emigration, can be derived as:

$$N(t + 1) = N(t) + N(t)r(t)$$

where $r(t) = b(t) - d(t)$. Here, $b(t)$ and $d(t)$ are the per-capita birth and death rates and thus $r(t)$ is the per-capita growth rate. Models which are based only on the intrinsic population growth rate, ‘ r ’, however, do not retain much information about the underlying drivers of the population dynamics. Density-dependent, age structured, stochastic effects on growth, spatially structured, and competition models (e.g., Lotka-Volterra) all are derivations of the basic BIDE model.

In closed population models, we focus on estimating the population size, N , but in open population models we are interested in the dynamics that arise between years or seasons and thus we focus not only on $N(t)$ but on the processes that drive population changes. By taking the basic parameters in the BIDE model and reconceptualizing them, they can then be related to the commonly used parameters in JS and CJS models, described in more detail throughout this chapter. In the absence of movement, deaths (D) can be estimated in the CJS model and both D and B (births) can be estimated in the JS model. However, in considering movement, it becomes difficult to distinguish births from immigrants and deaths from emigrants because data are usually only collected in one area and when the animal leave that area we cannot determine its fate.

For example, survival ($\phi(t)$) is defined as the probability of an individual surviving from time t to $t + 1$, and often this is called ‘*apparent* survival’ because deaths and emigration cannot be separated. Mortality, the probability of dying from time t to $t + 1$, is $1 - \phi(t)$. Recruitment (γ) is the probability of a new individual entering the population between t to $t + 1$, which includes both those born into the population and immigrants. This inability to distinguish between the different forms of losses and gains does not allow researchers to test specific hypotheses about population dynamics. To address this, Nichols and Pollock (1990) applied the robust design to a two age class situation in order to separate estimates of recruitment into immigration and *in situ* reproduction. While models that focus on the population growth rate tend to lose important information on population dynamics, more recent work has been done to estimate the contributions of survival and recruitment to the per capita growth rate, ‘ r ’, using capture-recapture data and a reverse-time modeling approach (Pradel, 1996; Nichols et al., 2000a). All of these model improvements have provided invaluable information in the study of population dynamics, but none explicitly incorporate animal movement.

16.1.2 Animal movement related to population demography

One issue that arises frequently in traditional open population models is that movement can make it difficult to distinguish survival from emigration. For example, we know that movement of transients and temporary emigration will affect the estimates of survival, causing us to refer to estimates as “apparent survival” (Lebreton et al., 1992). This is because an animal that appears in the population for a short period of time and then leaves is going to appear as though it has died. Due to this problem, there has been a significant amount of work developing models to deal with temporary emigration and transients in both closed and open capture-recapture models (Kendall et al., 1997; Pradel et al., 1997; Hines et al., 2003; Clavel et al., 2008; Gilroy et al., 2012; Chandler et al., 2011). Because movement is modeled directly within the SCR framework, we can better understand the impact of animals moving onto and off of the trap array and hence we can improve our estimates of survival by combining the traditional CJS and JS models with

13056 the SCR model.

13057 While demographic parameters such as survival rates, population growth, etc. are
13058 influenced by density (Fowler, 1981; Murdoch, 1994; Saether et al., 2002), it is also likely
13059 that movement of individuals can influence these parameters. It is generally accepted that
13060 population structure (i.e., age, stage, or size distribution) can affect both population size
13061 and growth over time (Caswell and Werner, 1978). We also know that how animals dis-
13062 tribute themselves in space can directly influence the age or stage structure of a population
13063 – this can be behavioral, habitat related, or some combination of factors. For example,
13064 if habitat is limited, some younger members of the population might have trouble finding
13065 and/or defending a territory. Ultimately, this may lower survival for a certain age class
13066 in the population directly impacting the population structure.

13067 Dispersal can also affect population structure. In population ecology, dispersal can be
13068 related with access to reproduction, population regulation, habitat quality, as well as the
13069 linking of local populations in metapopulation ecology (Clobert et al., 2001; Ovaskainen,
13070 2004; Ovaskainen et al., 2008). It is known that dispersal may be influenced by density-
13071 dependence (Matthysen, 2005); for example, competition may cause individuals to be
13072 more likely to emigrate from an area, or individuals may leave an area in search of a mate
13073 or partner. We discuss modeling dispersal with capture recapture data a bit further in
13074 Sec. 16.4 at the end of the chapter.

16.2 JOLLY-SEBER MODELS

13075 16.2.1 Traditional Jolly-Seber models

13076 The JS model was developed as a way to estimate not only detection and abundance, but
13077 survival and recruitment (new individuals coming into the population) based on capture-
13078 recapture data (Jolly, 1965; Seber, 1965). There are a number of ways that researchers
13079 have formulated the JS model (Cooch and White, 2006) and while all are slightly different,
13080 the resulting estimates of abundance and the driving parameters such as survival and
13081 some form of recruitment should be equivalent. Commonly used formulations are the
13082 Link-Barker (Link and Barker, 2005), Pradel-recruitment (Pradel, 1996), Burnham JS
13083 (?), and the super-population formulation of Schwarz and Arnason (1996). In all of these
13084 models, the parameter of interest is recruitment, or how new individuals arrive into the
13085 population. Therefore one of the main differences between the various models is how new
13086 entrants into the population are parameterized.

13087 Traditionally, sampling for the JS model included only one data collection event per
13088 primary occasion and this allowed for the estimation of survival and recruitment. However,
13089 without repeated visits within a primary occasion, there is not enough data to allow for
13090 variation in detection and this lead to potentially inaccurate estimates of population size.
13091 This lead Pollock (1982) to devise the robust design in order to allow for heterogeneity
13092 in capture probability (by sex, age, social status, etc.) and trap response under the JS
13093 model. We present the robust design approach as it is more flexible and generalizing to
13094 the spatial version of the JS model will be much simpler. The basic idea is that there are
13095 primary occasions (e.g., years, seasons) and we allow the population to be “open” between
13096 the primary occasions. This means that individuals can enter and leave the population
13097 (i.e., births, deaths, immigration, emigration can occur) between the primary occasions
13098 and within a primary occasion, the population is assumed to be closed to these processes.

13099 The standard JS model does not allow for variation in detection probability between in-
 13100 dividuals or within a primary occasion because only one sample is collected per primary
 13101 period. However, when multiple samples are taken within a primary occasion (we call
 13102 these “secondary occasions”), then variation in detection probability can be modeled and
 13103 thus our estimates of N can be improved. To that extent, we can envision the data as
 13104 arising from repeated sampling over seasons or years (or *primary* periods) within which
 13105 one or more samples (e.g., trap nights) might be taken (*secondary* periods). Fig. 16.1
 13106 demonstrates the sampling process graphically. Comparing this with all of our previous
 13107 work, the sample occasions (e.g., trap nights, weeks, etc...) described in the closed popu-
 13108 lation chapters are called *secondary* sampling occasions in the context of open population
 13109 models.

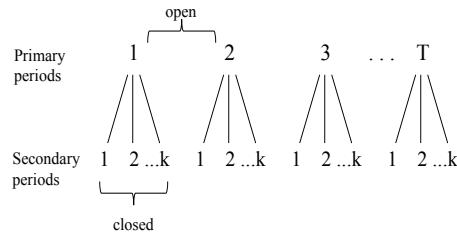


Figure 16.1. Schematic of the robust design with T primary sampling periods and K secondary periods. The populations are considered open between primary periods and closed within each.

13110 We can easily formulate a non-spatial JS model using the robust design. We define y_{ikt}
 13111 as the encounter history for individual i at secondary occasion k during primary occasion
 13112 t . If we have a Bernoulli encounter process then we can describe the observation model,
 13113 specified conditional on the “alive state”, $z(i, t)$, for individual i at primary time t , as:

$$y_{ikt}|z(i, t) \sim \text{Bernoulli}(p_t z(i, t)).$$

13114 (Note: throughout this chapter we will focus on changes in the alive state, so we will
 13115 index z using parentheses in order to make the notation easier to read, where $z(i, t)$ is
 13116 equivalent to z_{it}). Thus, if individual i is alive at time t ($z(i, t) = 1$), then the observations
 13117 are Bernoulli with detection probability p as before. Conversely, if the individual is not
 13118 alive ($z(i, t) = 0$), then the observations must be fixed zeros with probability 1. Note
 13119 our distinct use of the variable z here as representing the state of individuals (alive/dead)
 13120 instead of our previous use of z as the data augmentation variable.

13121 Survival and recruitment in the open population are manifest in a model for the
 13122 latent state variables $z(i, t)$ describing individual mortality and recruitment events. An
 13123 important aspect of the hierarchical formulation of the model that we adopt here is that the

model for the state variables is described conditional on the total number of individuals ever alive during the study (a parameter which we label N) based on T periods, as in Schwarz and Arnason (1996). Data augmentation induces a special interpretation on the latent state variables $z(i, t)$. In particular, “not alive” includes individuals that have died, or individuals that have not yet been recruited. Royle and Dorazio (2008) showed that using this formulation simplifies the state model and also allows it to be implemented directly in the **BUGS** language. For example, considering the case $T = 2$, the state model is composed of the following two components: First, the initial state is described by:

$$z(i, 1) \sim \text{Bernoulli}(\psi)$$

and then a model describing the transition of individual states from $t = 1$ to $t = 2$:

$$z(i, 2) \sim \text{Bernoulli}(\phi z(i, 1) + \gamma(1 - z(i, 1))).$$

If $z(i, 1) = 1$, then the individual may survive to time $t = 2$ with probability ϕ whereas, if $z(i, 1) = 0$, then the “pseudo-individual” may be recruited with probability γ .

We can then generalize this model for $T > 2$ time periods and allow survival and recruitment to be time dependent. Initialize the model for time $T = 1$ as we have done above and then the model describing the transition of individual states from t to $t + 1$ is:

$$z(i, t + 1) \sim \text{Bernoulli}(\phi_t z(i, t) + \gamma_t(1 - z(i, t))).$$

This parameterization results in $T - 1$ survival and recruitment parameters. The main difference here from the CJS model, described below, is that we include recruitment and are interested in estimating N for each t . Since this state model described above is conditional-on- N , we must deal with the fact that N is unknown, which is done through data augmentation similar to how we used it in the closed population models.

16.2.2 Data augmentation for the Jolly-Seber model

The fundamental challenge in carrying out a Bayesian analysis of this model is that the parameter N (the total number of individuals alive during the study) is not known. We have discussed and demonstrated data augmentation in many previous chapters; however, with the open population model, we have to take care that two issues are addressed: (1) the data augmentation is large enough to accommodate all potential individuals alive in the population during the entire study and (2) that individuals cannot die and then re-enter the population. Royle and Dorazio (2008) (see also Kéry and Schaub (2012)) describe this formulation for open population models, including the non-spatial JS and robust design models.

To begin, let’s consider the role of recruitment, γ , in the model when we use data augmentation to estimate N . Data augmentation formally reparameterizes the model, replacing N , the number of individuals ever alive with M , where we assume $N \sim \text{Binomial}(M, \psi)$. That is, the expected value of N under the model is equal to ψM . As a result of this reparameterization, the recruitment parameters γ_t are also relative to the number of “available recruits” on the data augmented list of size M , and not directly related to the population size. This can be dealt with by deriving N_t , and R_t , the population size and

13161 number of recruits in year t , as a function of the latent state variables $z(i, t)$. For example,
 13162 the total number of individuals alive at time t is

$$N_t = \sum_{i=1}^M z(i, t)$$

13163 and the number of recruits is

$$R_t = \sum_{i=1}^M (1 - z(i, t-1)) z(i, t)$$

13164 which is the number of individuals *not* alive at time $t-1$ but alive at time t .

13165 In the case of just two primary periods, this process is straightforward. When the
 13166 number of primary sample occasions is greater than 2, we must formulate the model for
 13167 recruitment by introducing another latent variable, in order to ensure that an individual
 13168 can only be recruited once into the population. Here, this formulation of the model uses
 13169 a set of latent indicator variables, which we label $A(i, t)$, which describe the time interval
 13170 $(t-1, t)$ at which individual i is recruited into the population. Let $A(i, t) = 1$ if individual
 13171 i is recruited in time interval $(t-1, t)$ otherwise $A(i, t) = 0$. To construct the recruitment
 13172 process we make use of the standard conditional binomial construction of a removal process
 13173 (Royle and Dorazio 2008). The initial state is given by:

$$A(i, 1) \sim \text{Bernoulli}(\gamma_1)$$

13174 for $i = 1, 2, \dots, N$. Then, for $t > 1$

$$A(i, t) | A(i, t-1) \dots A(i, 1) \sim \text{Bernoulli}\left((1 - \sum_{\tau=1}^{t-1} A(i, \tau)) \gamma_t\right)$$

13175 where γ_1 is equivalent to ψ in the description of the 2 primary occasion version open
 13176 population model above and τ is just a counter for times 1 to $t-1$. Each recruitment
 13177 variable is conditional on whether the individual was ever previously recruited and this
 13178 construction forces the recruitment variable after initial recruitment to be degenerate
 13179 (have a sample size of 0). This ensures that an individual cannot be recruited again after
 13180 initial recruitment. Then, we can describe the state variables $z(i, t)$ by a 1st order Markov
 13181 process. For $t = 1$, the initial states are fixed:

$$z(i, 1) \equiv A(i, 1)$$

13182 and, for subsequent states, we have

$$z(i, t) | z(i, t-1), A(i, t) \sim \text{Bernoulli}(\phi_t z(i, t-1) + A(i, t)).$$

13183 Thus, if an individual is in the population at time t (i.e., $z(i, t) = 1$), then that individual's
 13184 status at time $t+1$ is the outcome of a Bernoulli random variable with parameter (survival
 13185 probability) ϕ_t . If the individual, however, is not in the population at time t (i.e., $z(i, t) =$
 13186 0), then the outcome is a Bernoulli random variable with probability γ_t , a parameter that
 13187 is related to *per capita* recruitment. Recall that we use A to describe if an individual is
 13188 available for recruitment and this is directly related to z as described above. We carry

13189 out this process in **JAGS** by using the `sum()` and `step()` functions together to ascertain
 13190 if a particular individual i was ever previously alive. The `step()` function is a logical test
 13191 in **JAGS** for $x \geq 0$ such that `step($x \geq 0$)` returns a 1, otherwise 0. Individuals that
 13192 were ever previously alive are no longer eligible to be “recruited” into the population. The
 13193 implementation of this model in **JAGS** is shown in panel 16.1.

```
model{

  psi ~ dunif(0,1)
  phi ~ dunif(0,1)
  p.mean ~ dunif(0,1)

  for(t in 1:T){
    N[t] <- sum(z[1:M,t])
    gamma[t] ~ dunif(0,1)
  }

  for(i in 1:M){
    z[i,1] ~ dbern(psi)           # Alive state for the first year
    cp[i,1] <- z[i,1]*p.mean
    Y[i,1] ~ dbinom(cp[i,1], K)   # Y are the number of encounters
    A[i,1] <- (1-z[i,1])

    for(t in 2:T){               # For loop for years 2 to T
      a1[i,t] <- sum(z[i, 1:t])  # Sum over the alive states from 1 to t
      A[i,t] <- 1-step(a1[i,t] - 1)
      # A is the indicator if an individual is available to be recruited
      mu[i,t]<- (phi*z[i,t-1]) + (gamma[t]*A[i,t-1])
      # Alive state at t is dependent on phi and gamma
      z[i,t] ~ dbern(mu[i,t])
      cp[i,t] <- z[i,t]*p.mean
      Y[i,t] ~ dbinom(cp[i,t], K)
    }
  }
}
```

Panel 16.1: **JAGS** model specification for the non-spatial Jolly-Seber model using data augmentation.

13194 **Ovenbird mist-netting study**
 13195 We now return to the ovenbird data collected using mist-nets at Patuxent Wildlife
 13196 Research Center. We introduced these data in Chaps. 9 and 14, and they are provided

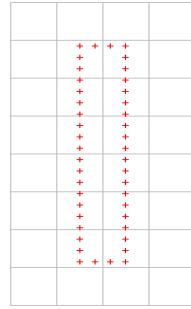


Figure 16.2. Arrangement of the mist nests in the ovenbird study. The nets are arranged in a 600-m by 100m rectangle, spaced 30 m apart.

with the **secr** package (see, Efford et al. (2004); Borchers and Efford (2008)). To refresh your memory: 44 mist nets spaced 30 m apart on the perimeter of a 600-m x 100-m rectangle (see Fig. 16.2) were operated on 9 or 10 non-consecutive days in late May and June for 5 years from 2005-2009.

In Chaps. 9 and 14, we dealt with this dataset as a type of spatial “multi-session” model where abundances in each year, N_t , were regarded as independent random variables either with a Poisson prior (as implemented in **secr**, or a binomial prior if analyzed using **BUGS** with data augmentation. This is the simplest approach for modeling data collected over multiple years, but it does not allow for inference about demographic processes, as does the JS model.

In the spatial multi-session model (S-MS) we were not interested in individual identity across years; however, we need to maintain the order of individuals across years to estimate the survival and recruitment of individuals into the population. We organize the data set so that each row in the array represents just one individual across all primary periods. For the ovenbird dataset, we can organize the data by creating a master list of all individuals captured during the entire study. From this list, we can assign each individual a unique row in our dataset (in the **R** commands, we do this by using the **unique()** function on the row names for each year of our 3-D array and use **pmatch()** to associate the data to the correct column). The resulting array is individual by secondary occasion by primary occasion, $M \times K \times T$. The **R** commands to organize the data in a way suitable for fitting a Jolly-Seber type model are included in the **scrbook** package using the function **ovenbirds.js()** and are not shown here. The key difference between our model and organization of the data here and that in Chapt. 9 is that, here, we have to preserve individual identity across years (in the model and data structure).

The data augmentation must be large enough to include individuals alive during any of the time periods and to account for that, we set $M = 200$. There were 70 unique individuals captured over the 5 year period. For this example, we hold survival constant but allow recruitment to be time dependent (since γ is essentially a function of the data

13225 augmentation process as described above, it does not make sense to hold recruitment
 13226 constant and we therefore make it time specific).

13227 To implement the model in Panel 16.1, the following commands are used:

```
13228 # Set initial values for the alive state, z
13229 > zst <- c(rep(1,M/2),rep(0,M/2))
13230 > zst <- cbind(zst,zst,zst,zst,zst)
13231
13232 > inits <-function(){list(z=zst,sigma=runif(1,25,100),gamma=runif(5,0,1))}
13233 > parameters <- c("psi","N","phi", "p.mean", "gamma")
13234 > data <- list (K=10,Y=Ybin,M=M)
13235
13236 > library("rjags")
13237 > out1 <- jags.model("modelNSJS.txt",data,inits,n.chains=3,n.adapt=500)
13238 > out2NSJS <- coda.samples(out1,parameters,n.iter=20000)
```

13239 In this non-spatial JS model, N_t is estimated to be between about 22 and 33 for each
 13240 of the 5 years (see Table 16.1 for results). The posterior mean for detection (`p.mean` in the
 13241 model) was 0.14. We did not include `p.mean` in the table because the SCR models do not
 13242 have a parameter that directly corresponds to it. Instead, SCR models have a detection
 13243 function that is related to distance.

13244 **Shortcomings of the traditional JS models**

13245 One of the biggest shortcomings of the non-spatial JS model is that we estimate N but
 13246 have no explicit spatial area associated with it (so, in Table 16.1, the density estimate from
 13247 the non-spatial JS model is listed as NA). Ignoring the spatial information in the data
 13248 makes the estimation of density an informal process. As we saw in the closed models, the
 13249 explicit incorporation of spatial information in the model will allow us to make an explicit
 13250 estimate of density. This improvement should also carry through in our estimation of other
 13251 demographic parameters such as survival and recruitment as the movement of individuals
 13252 is directly accounted for in the model.

13253 **16.2.3 Spatial Jolly-Seber models**

13254 To parameterize the spatial JS models, we follow all of the same steps as the non-spatial
 13255 model but also include the trap location information into our detection function. Basically,
 13256 we are using the closed population SCR model to estimate the detection parameters and
 13257 initial population size, and the open component is carried out in the process of how we
 13258 model the transition of $z(i,t)$ to $z(i,t+1)$ which is the same as in the non-spatial JS
 13259 model. To do so, we describe the Bernoulli observation model, specified conditional on
 13260 $z(i,t)$, as has been done throughout the book:

$$y_{ijk|t}|z(i,t) \sim \text{Bernoulli}(p_{ijk}z(i,t)).$$

13261 with

$$p_{ijk} = p_0 * \exp(-\alpha_1 d_{ij}^2) \quad (16.2.1)$$

13262 where $d_{ij} = ||\mathbf{x}_j - \mathbf{s}_i||$, the distance between activity center \mathbf{s}_i and trap \mathbf{x}_j . As before,
 13263 p_0 is the baseline encounter probability, for an individual with home range center located

13264 precisely at a trap, and $\alpha_1 = (1/(2\sigma^2))$ where σ is the scale parameter in this Gaussian
 13265 encounter probability model.

13266 If individual i is alive at time t ($z(i,t) = 1$), then the observations are Bernoulli.
 13267 Conversely, if the individual is not alive ($z(i,t) = 0$), then the observations must be fixed
 13268 zeros with probability 1. As always, other observation models can be considered in the
 13269 context of a fully open JS type model, such as the Poisson or multinomial models described
 13270 in Chapt. 9, and we can consider many alternative models of encounter probability.

13271 We initialize the model for time $T = 1$ and then model the transition of individual
 13272 states from t to $t + 1$ as:

$$z(i,t+1) \sim \text{Bern}(\phi_t z(i,t) + \gamma_t(1 - z(i,t))).$$

13273 Previously, in sec. 16.2.2, it was described how this formulation of the model uses a set
 13274 of latent indicator variables $A(i,t)$ which describes if an individual is recruited into the
 13275 population during time interval $(t-1, t)$. We apply the same approach here, so that, as
 13276 before, $A(i,t) = 1$ if individual i is recruited in time interval $(t-1, t)$; otherwise $A(i,t) = 0$.

13277 The number of recruits into the population is calculated based on the alive state
 13278 of the previous time steps $(1, 2, \dots, t-1)$ and the current time step (t) . For example,
 13279 to estimate the number of recruits from time period 1 to 2, we count those individuals
 13280 not in the population at time 1 ($z(i,1) = 0$) but alive at time 2 ($z(i,2) = 1$). We can
 13281 determine if individual i has entered the population at time $t = 2$ by using the formula:
 13282 $R_{i,2} = (1 - z(i,1))z(i,2)$ and then sum $R_{i,2}$ over M to get the total number of recruits.
 13283 We can do this for all the primary periods in our study, as shown in the **JAGS** code in
 13284 Panel 16.2.

13285 Ovenbird mist-netting study

13286 In the previous analysis of the ovenbird data, we did not make use of the spatial
 13287 location for each net the ovenbirds were captured in. However, there were 44 mist nets
 13288 operational during each of the sampling occasions. We already organized the data above
 13289 so that our 3-D encounter histories are set up. The data set is then $M = 200$ individuals
 13290 by $K = 10$ secondary occasions by $T = 5$ primary occasions. In the non-spatial version,
 13291 we reduced the data to captured or not-captured; however, the encounter history array,
 13292 **Yarr**, contains the number of the net that each individual was captured in and contains a
 13293 45 if the individual was not captured. The code above describes how the encounter history
 13294 array is created, so we do not reproduce this piece of code here. To call the model, use
 13295 the following **R** code which sets the initial values for **z[i,t]**, the parameters to monitor,
 13296 and calls **JAGS**. The code is also available in the **ovenbirds.js()** function.

```
13297 > zst <- c(rep(1,n),rep(0,M-n))
13298 > zst <- cbind(zst,zst,zst,zst,zst)
13299
13300 > inits <- function(){list(z=zst,sigma=runif(1,25,100),
13301           gamma=runif(5,0,1), S=Sst,alpha0=runif(1,-2,-1))}
13302 > parameters <- c("psi", "alpha0", "alpha1", "sigma", "N",
13303           "D", "phi", "gamma", "R")
13304 > data <- list(X=as.matrix(X[[1]]), K=10, Ycat=Yarr,
13305           M=M, ntraps=ntraps, ylim=ylim, xlim=xlim)
```

```

model {
  psi ~ dunif(0,1)      # Prior distributions
  phi ~ dunif(0,1)
  alpha0 ~ dnorm(0,10)
  sigma ~ dunif(0,200)
  alpha1 <- 1/(2*sigma*sigma)
  A <- ((xlim[2]-xlim[1]))*((ylim[2]-ylim[1]))  # Area of state-space

  for(t in 1:T){
    N[t] <- sum(z[1:M,t])  # Calculate abundance for each year
    D[t] <- N[t]/A          # Calculate density for each year
    R[t] <- sum(R[1:M,t])   # Calculate the recruits for each year
    gamma[t] ~ dunif(0,1)   # Prior for time specific recruitment parameter
  }

  for(i in 1:M){
    z[i,1] ~ dbern(psi)
    R[i,1] <- z[i,1]        # To estimate the number of recruits
    R[i,2] <- (1-z[i,1])*z[i,2]
    R[i,3] <- (1-z[i,1))*(1-z[i,2])*z[i,3]
    R[i,4] <- (1-z[i,1))*(1-z[i,2))*(1-z[i,3])*z[i,4]
    R[i,5] <- (1-z[i,1))*(1-z[i,2))*(1-z[i,3))*(1-z[i,4])*z[i,5]

    for(t in 1:T){
      # Independent activity centers for each year
      S[i,1,t] ~ dunif(xlim[1],xlim[2])
      S[i,2,t] ~ dunif(ylim[1],ylim[2])
      for(j in 1:ntraps){
        d[i,j,t] <- pow(pow(S[i,1,t]-X[j,1],2) + pow(S[i,2,t]-X[j,2],2),1)
      }
      for(k in 1:K){
        for(j in 1:ntraps){
          lp[i,k,j,t] <- exp(alpha0 - alpha1*d[i,j,t])*z[i,t]
          cp[i,k,j,t] <- lp[i,k,j,t]/(1+sum(lp[i,,t]))
        }
        cp[i,k,ntraps+1,t] <- 1-sum(cp[i,k,1:ntraps,t])
        # Here, the last cell indicates not captured
        Ycat[i,k,t] ~ dcat(cp[i,k,,t])
      }
    }
    A[i,1]<-(1-z[i,1])
    for(t in 2:T){           # For loop for years 2 to T
      a1[i,t] <- sum(z[i, 1:t]) # Sum over alive states from 1 to t
      A[i,t] <- 1-step(a1[i,t] - 1)
      # A indicates if individual is available to be recruited at time t
      mu[i,t] <- (phi*z[i,t-1]) + (gamma[t]*A[i,t-1])
      # Alive state at t is dependent on phi and gamma
      z[i,t] ~ dbern(mu[i,t])
    }
  }
}

```

Table 16.1. Posterior mean of model parameters for the non-spatial Jolly-Seber model (NS-JS), the spatial Jolly-Seber model (S-JS), and the spatial multi-session model (S-MS) fitted to the ovenbird data set. Density shown in individuals per hectare.

	NS-JS	S-JS	S-MS
D[1]	NA	0.96	0.93
D[2]	NA	1.00	1.00
D[3]	NA	1.10	1.20
D[4]	NA	1.10	0.89
D[5]	NA	0.79	0.76
N[1]	26.5	33	32.4
N[2]	30.2	36	35.8
N[3]	33.1	39	42.1
N[4]	29.5	37	30.8
N[5]	21.7	28	26.2
alpha0	NA	-2.9	-2.88
alpha1	NA	1.2e-04	1.22e-04
sigma	NA	6.4	6.44
gamma[1]	0.50	0.50	NA
gamma[2]	0.09	0.09	NA
gamma[3]	0.11	0.13	NA
gamma[4]	0.13	0.16	NA
gamma[5]	0.07	0.08	NA
phi	0.48	0.53	NA
psi	0.14	0.17	NA
R2	NA	15	NA
R3	NA	19	NA
R4	NA	8.3	NA
R5	NA	8.3	NA

```

13306 > library("rjags")
13307 > out1 <- jags.model("modelJS.txt", data, inits, n.chains=3,
13308   n.adapt=500)
13309 > out2JS <- coda.samples(out1,parameters,n.iter=10000)

```

13310 Our results for density, α_0 , and α_1 are rather similar to those found in the multi-
 13311 season analysis from Chapt. 9. Since all of our parameters including α_0 and α_1 are
 13312 shared between seasons, we would expect these results to be similar between the multi-
 13313 season model and the JS model (see Table 16.1). There are some slight differences in the
 13314 parameter estimates, for example, the density is lower in year 4 in the multi-season model
 13315 than in the JS model. This may be due to a smaller sample size in that year; due to the
 13316 Markovian relationship between abundances, the JS model is able to make use of the data
 13317 more efficiently. Because we have defined the same state space for the spatial JS model
 13318 and multi-season, our estimates of N_t are directly comparable. However, the estimates
 13319 of N_t under the non-spatial JS model are not directly comparable as we do not have a
 13320 well-defined effective trapping area. We see from Table 16.1 that N_t is smallest for the
 13321 non-spatial JS model across all years. This suggests that the actual effective trapping area
 13322 is smaller than our state-space, but we cannot know how much relative to the state-space
 13323 to make useful comparisons between the N_t s.

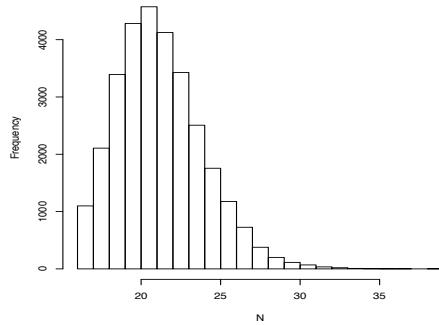


Figure 16.3. Posterior distribution of N_5 from the spatial JS model for the ovenbird dataset. This figure suggests that there is no truncation of the posterior of N_5 by M .

In the JS formulation of the model, we also estimate the recruitment for each year, and we can look at our derived values for recruitment (R_2, R_3, R_4 , and R_5). R_2 is the number of new recruits from primary period 1 to 2; R_3 is the number of new recruits from primary period 2 to 3; and so forth. R_2 and R_3 are almost double that of R_4 and R_5 , suggesting that less animals were recruited into the population in the latter years of the study. The density in the last year of the study was lower than previous years. It is good to check your results when you see a pattern like this – the number of recruits declining each year – because this could be an indication that the data augmentation was not large enough. In this example, we checked to make sure that $M=200$ was sufficiently large by examining the recruitment parameter, γ . If γ is close to 1 during any of the time periods, then there are not enough augmented individuals in the overall dataset. In this case, the 97.5% quantile of γ_5 , the recruitment probability in the final year of the study, was 0.14 and none of the other γ 's were close to 1 either. You can also look at the posterior distributions of N_t to make sure they are not truncated, Fig. 16.3 shows that the posterior distribution of N_5 is not truncated. The posterior mean for survival, ϕ , was 0.53. Although we did not do that in this example, it should be easy to see that we could allow survival to vary by time, as we did with recruitment. Our estimates of survival seem reasonable when compared with the ovenbird literature. Some studies have found annual male ovenbird survival to be around 0.62 (Porneluzi and Faaborg, 1999; Bayne and Hobson, 2002), whereas female ovenbird survival is much lower (0.21, Bayne and Hobson (2002)). With more individuals, we could run this model with survival estimated for each sex separately. However, researchers should be careful not to over-parameterize models based on the amount of data available. The results indicate that the posterior mean estimate of ϕ was greater in the SCR model (0.53) than the non-spatial model (0.48) which suggests that the SCR model is starting to separate movement from survival in order to estimate the true rather than apparent survival.

16.3 CORMACK-JOLLY-SEBER MODELS

16.3.1 Traditional CJS models

13351 Cormack-Jolly-Seber (CJS) models are used extensively to estimate survival probabilities.
 13352 There are two common ways to fit these models, using either a multinomial likelihood
 13353 approach (Lebreton et al., 1992) or a state-space formulation of the model (Gimenez
 13354 et al., 2007; Royle, 2008). The multinomial likelihood approach is based on summarizing
 13355 the data to counts of unique encounter histories, which have a multinomial distribution.
 13356 The data are summarized over individuals and so it is not feasible to build general models
 13357 that contain individual covariates. On the other hand, the state-space formulation of
 13358 the model preserves individual identity and, therefore, parameters can be modeled at the
 13359 individual level, and individual effects (covariates or heterogeneity) can be included. In the
 13360 present context of spatial capture-recapture models, we naturally think about including
 13361 individual locations, or activity centers, as individual covariates.

13362 We can adopt a simple state-space parameterization of the basic single state, non-
 13363 spatial CJS model in which the observation model is described conditional on the latent
 13364 state variables $z(i, t)$ – the “alive state” which indicates whether individual i is alive
 13365 ($z(i, t) = 1$) or not ($z(i, t) = 0$) during each of $t = 1, 2, \dots, T$ primary periods. Let y_{it}
 13366 indicate the observed encounter data of individual i in primary period t . The model,
 13367 specified conditional on $z(i, t)$, is:

$$y_{it}|z(i, t) \sim \text{Bernoulli}(p_t z(i, t)).$$

13368 Analogous to the JS model, if individual i is alive at time t ($z(i, t) = 1$), then the obser-
 13369 vations are Bernoulli with probability of detection p_t .

13370 If the individual is not alive ($z(i, t) = 0$), then the observations must be fixed zeros
 13371 with probability 1. Contrary to the JS model, in the CJS model we condition on first
 13372 capture, which means that $z(i, t)$ will be 1 when t is the primary period individual i
 13373 is first captured in. We denote this $z(i, f_i)$, where f_i indicates the primary occasion in
 13374 which individual i is first captured, which can vary from $1 \dots T$. This ensures that each
 13375 individual is alive upon entering the model, and is also the reason that recruitment is not
 13376 estimated in the model.

13377 The “alive state” at time t for each individual is a function of the state at the previous
 13378 time step $t - 1$. Because we condition on the first capture, the initial state is set to one:

$$z(i, f_i) = 1$$

13379 where f_i indicates the primary occasion in which individual i is captured and the model
 13380 for the transition of individual states from t to $t + 1$ for all $t > f_i$ is

$$z(i, t) \sim \text{Bernoulli}(\phi z(i, t - 1)).$$

13381 Because we start with $z(i, f_i) = 1$, the individual survives with probability ϕ to time
 13382 $f_i + 1$ and so forth. Once an individual leaves the population (i.e., $z(i, t) = 0$), there
 13383 is no mechanism for the individual to return. This means that under this specification
 13384 individuals cannot temporarily emigrate. In the CJS model we are not estimating N_t , so
 13385 we do not need to make use of data augmentation here in order to account for uncaptured
 13386 individuals (remember: we explicitly condition on first capture). This version of the model

13387 is easy to construct in the **BUGS** (or **JAGS**) language which is shown in Panel 16.3.
 13388 Variations on this basic model and associated code for fitting the model in **BUGS** are
 13389 described in detail in Kéry and Schaub (2012, Chaps. 7-9).

```
model{
  phi ~ dunif(0,1) # Survival (constant over time)

  for(t in 1:T){
    p[t] ~ dunif(0, 1) # Detection (varies with time)
  }

  for(i in 1:M){
    z[i,first[i]] ~ dbern(1)
    for (t in (first[i]+1):T){
      tmp[i,t] <- z[i,t]*p[t]
      y[i,t] ~ dbern(tmp[i,t])
      phiUP[i,t] <- z[i,t-1]*phi
      z[i,t] ~ dbern(phiUP[i,t])
    }
  }
}
```

Panel 16.3: **JAGS** model specification for the non-spatial basic Cormack-Jolly-Seber (CJS) model. Note that the first alive state of each individual, $z[i,first[i]]$, is not stochastic. It is equal to 1 with probability 1.

13390 **Movement and survival of American shad in the Little River**

13391 As an example for the CJS model, we use data collected on American shad (*Alosa*
 13392 *sapidissima*) in the Little River in North Carolina, U.S.A. (see photo in Fig. 16.4). The
 13393 Little River is a tributary to the Neuse River and the confluence is near Goldsboro,
 13394 North Carolina about 212 river kilometers from the Pamlico Sound. The motivation for
 13395 this example stems from an interest in better understanding survival and movement of
 13396 migratory fish. American shad are an anadromous fish that use rivers for spawning. The
 13397 data were collected and analyzed as described in Raabe (2012). Using a resistance board
 13398 weir near the river mouth, 315 fish were tagged with passive integrated transponders
 13399 (PIT) in the spring of 2010. An array of 7 upstream PIT antennas passively recaptured
 13400 individuals during upstream and downstream migrations. Each time a fish passed over
 13401 the antenna, it was recorded and the data were summarized weekly for 12 weeks. The fish
 13402 do not necessarily move past all antennae and may remain in the river between antennae
 13403 for more than a week, thus they are not all detected at each time period. The antennae
 13404 do not always operate perfectly either and fish that pass may not be recorded at some
 13405 times.



Figure 16.4. American shad caught in North Carolina, U.S.A. Credit: Joshua Raabe, North Carolina State University

13406 To apply the basic CJS model, we create the encounter history for each individual for
 13407 the 12 weeks and we also create a vector to indicate the period (week) of first capture.
 13408 The code is not shown here but is available in the `scrbook` package within the function
 13409 `shad.cjs()`. This function contains all of the code to fit the non-spatial, multi-state, and
 13410 spatial CJS models to the American shad dataset.

13411 Table 16.2 shows the estimated detection probabilities for each of the 12 primary
 13412 periods in the study. The posterior mean for detection probabilities ranges from 0.126 to
 13413 0.880, which could potentially be due to variation in water flow, stream depth, storms,
 13414 etc... The weekly survival probability, ϕ , had a posterior mean estimate of 0.824. This
 13415 estimate could be considered low for a weekly probability, but is likely due to the fact that
 13416 the migration upstream can be quite energetically taxing and the fish are likely to only
 13417 feed minimally in rivers (Leggett and Carscadden., 1978; Leonard and McCormick, 1999).
 13418 Additionally, the CJS model is only estimating apparent survival and some fish may have
 13419 left the stream temporarily or permanently heading back to the ocean or possibly to other
 13420 tributaries that are not monitored. We demonstrate in Panel 16.3 how to allow p to vary
 13421 by time, but we could also allow survival, ϕ to vary by time by implementing it exactly
 13422 as we do for p . As we move into the multi-state model, we can test for movement and
 13423 survival by state, which allows more specific biological questions to be addressed.

13424 **16.3.2 Multi-state CJS models**

13425 The basic version of the CJS model only allows for estimation of survival and detection
 13426 probabilities. However, researchers are often interested in addressing other ecological
 13427 questions such as age-dependent survival rates, habitat based movements, etc. Multi-
 13428 state models allow researchers to directly address such questions by incorporating more
 13429 than one state that an individual may potentially be in (Arnason, 1972, 1973; Brownie
 13430 et al., 1993). These possible states can be geographic location, age class, or reproductive
 13431 status among many others. Instead of just having an encounter history for an individual,

Table 16.2. Results of the basic non-spatial CJS model for the American shad dataset.

	Mean	SD	2.5 %	50 %	97.5 %
p[1]	0.499	0.289	0.026	0.499	0.975
p[2]	0.627	0.058	0.511	0.628	0.738
p[3]	0.762	0.036	0.689	0.763	0.829
p[4]	0.880	0.025	0.828	0.882	0.925
p[5]	0.548	0.043	0.465	0.548	0.633
p[6]	0.259	0.038	0.190	0.258	0.337
p[7]	0.126	0.031	0.072	0.124	0.194
p[8]	0.236	0.045	0.155	0.234	0.332
p[9]	0.237	0.049	0.148	0.234	0.341
p[10]	0.589	0.072	0.447	0.590	0.728
p[11]	0.834	0.063	0.700	0.839	0.942
p[12]	0.468	0.072	0.330	0.466	0.614
ϕ	0.824	0.011	0.802	0.825	0.846

we will also have auxiliary information on the state of that individual at capture (e.g., breeder or non-breeder). Since our interest is in movement of individuals, here we will consider states that represent spatial units or geographic locations. Generally speaking, we might think that the transition rates between locations could be due to habitat features (or quality) and we can use multi-state models to help us address such a question. In addressing movement through a multi-state modeling approach, the movement is often parameterized as random or Markovian between patches (Arnason, 1972, 1973; Schwarz et al., 1993).

In the simplest version of the multi-state model we have just two states. Thus, individuals can be marked and recaptured in one of two states (we'll call them A and B here). We will assume that the two "states" are different geographic sites. In the single-state model above, an individual i was either alive ($z(i, t) = 1$) at time t or dead ($z(i, t) = 0$). Now, we must consider that the individual could be alive in a given state or dead and that individuals can transition between states. An easy way to think about this is to look at the state transition matrix in Table 16.3. Here, ϕ^A is the probability of surviving in State A from time t to $t + 1$ and ϕ^B is the analogous parameter for State B. The movement parameters are ψ^{AB} and ψ^{BA} , where ψ^{AB} is the probability that an individual, survives from t to $t + 1$ and moves to State B just before $t + 1$ and vice versa for ψ^{BA} . The movement could also be defined as occurring before the survival; i.e., ψ^{AB} is the probability that an individual moves from State A to State B shortly after time t and then survives to time $t + 1$ in State B.

Table 16.3. Transition matrix for a multi-state model with just two states.

	State A	State B	Dead
State A	$\phi^A(1 - \psi^{AB})$	$\phi^A\psi^{AB}$	$1 - \phi^A$
State B	$\phi^B\psi^{BA}$	$\phi^B(1 - \psi^{BA})$	$1 - \phi^B$
Dead	0	0	1

Because individuals are not necessarily observed in their given state, detection should

be estimated separately for each of the states. Hence, we also have p^A and p^B , the probability of detecting an individual in State A and State B respectively. Also, at this point, we assume that there is no error in observed State (i.e., if the animal is observed, then the State is recorded correctly).

In the next few paragraphs, we show how the formulation of the 2 state multi-state capture recapture presented above can be related directly to SCR models. To start, define \mathbf{s} as the index of which state an individual is actually in and u_{it} as the state in which individual i was observed during sample t . In this two state example, u_{it} can only take on values for being observed in A or B (i.e., 1 or 2). We can define a simplistic model as follows:

$$u_{it} \sim \text{dcat}(\psi)$$

where ψ is a constant vector.

We observe an individual with probability p_0 , that is:

$$\Pr(y_{it} = 1 | u_{it}) = p_0$$

The state-transition probabilities are constant.

In an alternative formulation of this model, we can define \mathbf{s} as the index of which state an individual is in and then condition the observed locations, u_{it} as a function of the actual state, \mathbf{s} . This means that whether an individual moves or not, or where it moves to, is a function of where it is located. In this case, successive movement outcomes are *iid* and we can write the model according to:

$$u_{it} \sim \text{dcat}(\psi(\mathbf{s}_i))$$

Conditional on the state in which individual i is located, the probability of observing the individual is the same as above, p_0 . However, in this formulation of the model, the state-transition probabilities are constant, conditional on \mathbf{s} . Other models for these transition probabilities are possible and we will discuss those later.

A slight modification of this model would define \mathbf{s} as a “home area” for each individual. Then the region the animal goes to is a function not of where it was last time, but which region is its home area. This model is only slightly different from the Markovian model and, as was shown in Chapt. 9 for closed populations models, is how we make the technical transition from multi-state models to SCR models. Essentially increasing to a large number of strata, this formulation of the multi-state model becomes an SCR model where the “area of activity” \mathbf{s} becomes the “activity center” for each individual. In this case, the vector $\psi(\mathbf{s}_i)$ is a $J \times 1$ vector, corresponding to the probability of observation in each trap, given the individual’s activity center. Therefore, SCR models are closely related to classical multi-state models where the state-variable is “space.”

To describe this model for **JAGS**, we use a slightly different formulation which combines u_{it} and y_{it} as defined above into one observation matrix such that $y_{it} = 1, 2$, or 3 where 3 indicates “not observed”. Additionally, we use $z(i, t)$ to indicate the true state of individual i such that $z(i, t) = 1, 2$, or 3 where 1 indicates alive and in state 1, 2 indicates alive and in state 2, and 3 indicates “not alive”. Using this delineation, we just need to set up the transition matrix based on Table 16.3 and define each item within the model specification, shown in Panel 16.4. Note that this can become quite cumbersome when dealing with models that have many states.

```

model{

# r is an index for state (excluding the 'not alive' state)
for(r in 1:2){
    phi[r] ~ dunif(0,1)
    psi[r] ~ dunif(0,1)
    p[r] ~ dunif(0,1)
}

for (i in 1:M){
    z[i,first[i]] <- y[i, first[i]]
    for (t in (first[i]+1):T){
        z[i,t] ~ dcat(ps[z[i,t-1], i, ])
        y[i,t] ~ dcat(po[z[i,t], i, ])
    }
    ps[1, i, 1] <- phi[1] * (1-psi[1])
    ps[1, i, 2] <- phi[1] * psi[1]
    ps[1, i, 3] <- 1-phi[1]
    ps[2, i, 1] <- phi[2] * (1-psi[2])
    ps[2, i, 2] <- phi[2] * psi[2]
    ps[2, i, 3] <- 1-phi[2]
    ps[3, i, 1] <- 0
    ps[3, i, 2] <- 0
    ps[3, i, 3] <- 1

    po[1, i, 1] <- p[1]
    po[1, i, 2] <- 0
    po[1, i, 3] <- 1-p[1]
    po[2, i, 1] <- 0
    po[2, i, 2] <- p[2]
    po[2, i, 3] <- 1-p[2]
    po[3, i, 1] <- 0
    po[3, i, 2] <- 0
    po[3, i, 3] <- 1
}
}

```

Panel 16.4: **JAGS** model specification for a two-state version of the multi-state CJS model. Code modified from (Kéry and Schaub, 2012, Chapt. 9).

Table 16.4. Results of the multi-state CJS model for the migratory fish example. p^A is the detection probability in the first state (A), which in this case is the down stream area. ϕ^A is the weekly survival probability in state A and ψ^{AB} is the probability that an individual, which survived from t to $t + 1$ in Site A, moves to State B just before $t + 1$.

	Mean	SD	2.5 %	50 %	97.5 %
p^A	0.777	0.045	0.689	0.777	0.866
p^B	0.434	0.027	0.382	0.434	0.489
ϕ^A	0.850	0.022	0.807	0.851	0.893
ϕ^B	0.782	0.019	0.743	0.782	0.820
ψ^{AB}	0.421	0.034	0.356	0.421	0.489
ψ^{BA}	0.927	0.014	0.897	0.937	0.952

13494 Movement and survival of American shad in the Little River

13495 Previously, we analyzed the American shad data using a basic (i.e., non-spatial) CJS
 13496 model. However, the researchers were interested in movement of fish during migration and
 13497 so we classified the stream into 2 states (regions) – “downstream” and “upstream”. Each
 13498 antenna was assigned to a state based on the location, those below 20 river kilometers
 13499 were considered in the downstream state. Each fish has an encounter history including
 13500 whether or not the fish was detected during each week of the 12 week study, but also
 13501 the “state” of capture (“downstream” or “upstream”). A vector to indicate the period
 13502 of first capture was also created. Fish captured in more than one state during the week
 13503 were assigned the state in which they were captured most during that week. And the
 13504 model assumes that individuals observed in a state at consecutive primary periods did
 13505 not move from that state within the primary period. The data manipulation and model
 13506 specification for the multi-state CJS model is provided in the `scrbook` package under
 13507 the function `shad.cjs()`.

13508 Survival between the two areas was quite different (see Table 16.4). This might suggest
 13509 that fish moving further upstream are expending more energy and are more likely to die.
 13510 While survival in the two states was different, it is intuitive that the average of the survival
 13511 probabilities for A and B is essentially the same as that from the basic non-spatial CJS
 13512 ($\phi = 0.82$, see Table 16.2). Also, it should be noted that ψ^{BA} was very high, indicating
 13513 that fish in this study are returning downstream after spawning in the upstream area.
 13514 These results highlight the utility in using a multi-state model to understand movement
 13515 between states; here, we used spatial states, but age, class, breeding status, etc. are all
 13516 possibilities. We did have to reduce the dataset however to fit this model and information
 13517 on exact spatial location of detections was lost in creating just two states, downstream
 13518 and upstream. Losing information is one potential effect of using a multi-state model;
 13519 additionally when states are hidden or unknown (e.g., when animals are in a region not
 13520 exposed to sampling or the state is misclassified), these models can be difficult to fit (see
 13521 ? for an overview of these problems). Not losing information and unknown states are two
 13522 issues that can be resolved using the fully spatial CJS model. Misclassification of state (or
 13523 even individual) is a difficult problem to solve and current approaches (Link et al., 2010;
 13524 McClintock et al., In press) are in development for SCR models.

13525 16.3.3 Spatial CJS models

13526 In Chapt. 9, we suggested that SCR models are essentially a type of multi-state model
 13527 with spatially structured transition probabilities. As we noted, individuals can appear
 13528 in > 1 states simultaneously, which is not directly analogous to a standard multi-state
 13529 model. However, building on the state-space and multi-state CJS models, we can explicitly
 13530 incorporate individual movement as an individual covariate model (Royle, 2009a). To
 13531 move from the basic and multi-state CJS models to the SCR version, we need only make a
 13532 few changes to the model. We will not have discrete states and thus the biggest difference
 13533 is that individuals do not “transition” between a finite set of states, but instead are allowed
 13534 to move in continuous space.

13535 We may consider the same basic encounter models as described previously (i.e., Pois-
 13536 son, Bernoulli, or multinomial). In particular, let y_{ijkt} indicate the observed encounter
 13537 data of individual i in trap j , during interval (secondary period or sub-sample) $k =$
 13538 $1, 2, \dots, K$ and primary period t . We note that in some cases we may have intervals
 13539 ($K = 1$) which correspond to the design underlying a standard CJS or JS models whereas
 13540 the case $K > 1$ corresponds to the “robust design” (Pollock 1982). The Poisson observa-
 13541 tion model, specified conditional on $z(i, t)$, is:

$$y_{ijkt} | z(i, t) \sim \text{Poisson}(\lambda_0 g_{ij} z(i, t))$$

13542 where λ_0 is the baseline encounter rate and g_{ij} is the detection model as a function of
 13543 distance. If the individual is not alive ($z(i, t) = 0$), then the observations must be fixed
 13544 zeros with probability 1. Remember that in the CJS formulation, we condition on first
 13545 capture which means that $z(i, t)$ will be 1 when t is the first primary period of capture.
 13546 As before in the non-spatial CJS model, we can denote this as $z(i, f_i)$ where f_i indicates
 13547 the primary occasion in which individual i is first captured.

13548 Modeling time-effects either within or across primary periods is straightforward. For
 13549 that, we define $\lambda_0 \equiv \lambda_0(k, t)$ and then develop models for $\lambda_0(k, t)$ as in our closed SCR
 13550 models (we note that trap-specific effects could be modeled analogously).

13551 We follow the same model for survival as described in the non-spatial version of the
 13552 CJS. The model is initialized by setting the alive state at first capture to one:

$$z(i, f_i) = 1$$

13553 and for the transition of an individual’s alive state from t to $t + 1$, for all $t > f_i$, we have

$$z(i, t) \sim \text{Bernoulli}(\phi z(i, t - 1)).$$

13554 An individual survives with probability ϕ from one time step to the next. It is easy to see
 13555 that we can let survival be time specific by allowing ϕ to vary with each time step:

$$z(i, t) \sim \text{Bernoulli}(\phi_t z(i, t - 1)).$$

13556 In either case, once an individual leaves the population (i.e., $z(i, t) = 0$), there is no
 13557 recruitment so individuals cannot return. Again, we are not estimating N_t in this model,
 13558 hence we do not need any data augmentation. This conveniently makes the model run
 13559 faster too!

Table 16.5. Results of the spatial Cormack-Jolly-Seber model fitted to the American shad data set.

	Mean	SD	2.5 %	50 %	97.5 %
lam0[1]	5.555	0.224	5.125	5.553	6.003
lam0[2]	4.442	0.155	4.143	4.437	4.752
lam0[3]	1.892	0.068	1.763	1.891	2.031
lam0[4]	1.126	0.055	1.021	1.125	1.238
lam0[5]	0.949	0.058	0.838	0.948	1.067
lam0[6]	0.359	0.040	0.284	0.357	0.443
lam0[7]	0.188	0.031	0.133	0.186	0.254
lam0[8]	0.309	0.044	0.230	0.307	0.402
lam0[9]	0.363	0.052	0.269	0.361	0.471
lam0[10]	0.627	0.072	0.493	0.625	0.777
lam0[11]	1.611	0.109	1.408	1.607	1.835
lam0[12]	0.939	0.139	0.697	0.929	1.241
ϕ	0.784	0.012	0.760	0.785	0.807
σ	13.954	0.197	13.573	13.950	14.350

13560 Movement and survival of American shad in the Little River

13561 Going back to our American shad example, we can consider that this is exactly a
 13562 spatial capture recapture problem. In stream networks, the placement of PIT antennas
 13563 along the stream mimics the type of spatial data collected in terrestrial passive detector
 13564 arrays such as camera traps, hair snares, acoustic recording devices, etc. The difference is
 13565 that for fish and aquatic species, the stream constrains the movement of individuals to a
 13566 linear network. Using the data from the array of 7 PIT antennas and the number of times
 13567 each fish passed over the antenna, we can apply the SCR CJS model to evaluate movement
 13568 up and downstream of these fish. When we look at the individuals encountered at each
 13569 antenna for each of the primary periods, the dimensions of the data are 315 individuals by
 13570 7 antennas by 12 sample occasions. Individuals can encounter any antenna any number
 13571 of times during the week, which means we just sum the encounters over the week and
 13572 eliminate any need for explicit secondary occasions in the model. The result is a 3-D
 13573 array instead of a 4-D array. Given the structure of the encounters, we use a Poisson
 13574 encounter model in this example shown in 16.5. The code to carry out this model is
 13575 provided in the `scrbook` package using the function `shad.cjs()`.

13576 The baseline encounter rate, λ_0 , was allowed to vary by week and ranged from 0.188
 13577 to 5.555. We use the Poisson encounter model in this spatial CJS example rendering λ_0
 13578 not directly comparable to p_0 from the non-spatial and multi-state versions, which arises
 13579 as the detection probability under the Binomial encounter model. The posterior mean for
 13580 ϕ was 0.784 (see Table 16.5), again showing that the weekly survival probability is rather
 13581 low, just as we saw in the two previous example analyses of these data. Here, we are
 13582 modeling survival probability as constant, but there is reason to believe that it might vary
 13583 by time (similar to detection) and we might consider this additional parameterization in a
 13584 more complete analysis of the data set. The other parameter of interest is σ , the movement
 13585 parameter, which had a posterior mean of 13.954. Stream locations are recorded in river
 13586 kilometers (RKM), so σ is in units of kms. Our system here is linear, so we do not think
 13587 of fish as having a home range radius. However, σ can still inform us about the linear

```

model {
# Priors
sigma ~ dunif(0,80)
sigma2 <- sigma*sigma
lam0 ~ dgamma(0.1, 0.1)
phi ~ dunif(0, 1)  # Survival (constant across time)
tauv~dunif(0, 30)
tau<-1/(tauv*tauv)

for (i in 1:M){
z[i,first[i]] <- 1
S[i,first[i]] ~ dunif(0,50)  #Fish enter the stream at 0, thus the
#first AC is set to the lower stream end

for(j in 1:nantenna) {
D2[i,j,first[i]] <- pow(S[i,first[i]]-antenna.loc[j], 2)
lam[i,j,first[i]]<- lam0*exp(-D2[i,j,first[i]]/(2*sigma2))
tmp[i,j,first[i]] <- lam[i,j,first[i]]
y[i,j,first[i]] ~ dpois(tmp[i,j,first[i]])
}

for (t in first[i]+1:T) {
S[i,t] ~ dunif(xl, xu)
for(j in 1:nantenna) {
D2[i,j,t] <- pow(S[i,t]-antenna.loc[j], 2)
lam[i,j,t] <- lam0 * exp(-D2[i,j,t]/(2*sigma2))
tmp[i,j,t] <- z[i,t]*lam[i,j,t]
y[i,j,t] ~ dpois(tmp[i,j,t])
}
phiUP[i,t] <- z[i,t-1]*phi
z[i,t] ~ dbern(phiUP[i,t])
}
}
}

```

Panel 16.5: **JAGS** model specification for the spatial Cormack-Jolly-Seber (CJS) model for the American shad dataset. Note that the first alive state of each individual, $z[i, \text{first}[i]]$, is not stochastic. It is equal to 1 with probability 1.

13588 distance fish are moving. One final note about this example, we have simplified the dataset
 13589 for analysis here and some parameter estimates are different than found in Raabe (2012).

16.4 MODELING MOVEMENT AND DISPERSAL DYNAMICS

13590 To better understand the dynamics of a population, it is important to consider how the
 13591 locations of activity centers evolve over time. It is known that home ranges and territories
 13592 of animals can shift and other types of movement (migration, dispersal) can take place. In
 13593 the framework of open population SCR models, these dynamics of individual locations can
 13594 be reflected in appropriate models for the distribution of the activity centers. To begin,
 13595 a plausible “null model” for the distribution of individual activity centers is to assume
 13596 they are static over time and do not change across periods, i.e., $\mathbf{s}_i \sim \text{Uniform}(\mathcal{S})$. This
 13597 model may be appropriate for territorial species where the primary sampling periods are
 13598 relatively close together in time or the overall time frame of the study is limited. It might
 13599 seem more likely that activity centers change over time but are independent from year to
 13600 year for a given individual such $\mathbf{s}_i \sim \text{Uniform}(\mathcal{S})$. This is also how the spatial version of
 13601 the JS and CJS models were formulated above and this might a reasonable model when
 13602 there are large time lags between surveys, or if the individuals redistribute themselves
 13603 frequently in the study population.

13604 An intermediate option would be to assume that $\mathbf{s}(i, t) \sim \text{Normal}(\mathbf{s}(i, t - 1), \tau^2 \mathbf{I})$ for
 13605 $t > 1$ so that individual home range centers are perturbed randomly from their previous
 13606 value. This is possibly the the most realistic model for many cases. For example, many
 13607 migratory passerines, like the ovenbird, return to the same location, or nearly so each
 13608 year.

13609 We could use also use models of the activity centers to look at patterns of animal dis-
 13610 tribution with regard to habitat. For example, if the primary period is a season, it may be
 13611 expected that individuals move as the available food sources change. Using telemetry data
 13612 and/or capture recapture models a number of developments have been made to under-
 13613 stand animal movement patterns relative to habitat or dynamic systems (e.g., Jonsen et al.
 13614 (2005); Hooten and Wikle (2010)). Similarly, if we have an indicator of habitat that varies
 13615 by season, then in SCR models we can model the location of activity centers as a function
 13616 of the change in habitat. There are a number of options for modeling variation in activity
 13617 centers or animal locations as a function of covariates such as habitat, season, or behavior.
 13618 Other approaches to analyzing movement in a mark-recapture framework include but are
 13619 not limited to diffusion and auto-regressive models (Ovaskainen, 2004; Ovaskainen et al.,
 13620 2008)), agent-based (Grimm et al., 2005; Hooten et al., 2010) and dispersal kernels (Fu-
 13621 jiwara et al., 2006). For example, we define \mathbf{u}_{ikt} as the individual's observed location
 13622 at secondary period k in primary period t . Then $\mathbf{u}_{ikt} \sim \text{Normal}(\mathbf{s}(i, t), \Sigma_t)$ where Σ_t is
 13623 the variance-covariance matrix at time t . This is a model we have used quite frequently
 13624 throughout the book, i.e., that individual observed locations are assumed to follow a bi-
 13625 variate normal distribution about the activity center, \mathbf{s} . This is similar to the Guassian
 13626 and Laplace dispersal kernels. We could further allow the observed locations to follow an
 13627 auto-regressive model such that $\mathbf{u}_{ikt} \sim \text{Normal}(\rho(\mathbf{u}_{i,k,t-1} - \mathbf{s}(i, t - 1)), \Sigma_t^*)$. These are
 13628 just a few simple examples; as more information becomes available and data are collected
 13629 over longer time periods, we will be able to use more complex movement models in open
 13630 SCR models.

Cautionary note:

Using such Markovian models for the change in location of activity centers across primary periods, activity centers are no longer bound by the limits of the state-space. Imagine an individual living at the very edge of S at time t – there is some probability that, under the perturbation model, the location of \mathbf{s} at $t+1$ could be outside S . When activity centers are no longer bound to the state-space, the way we have determined density (i.e., $D = N/||S||$) no longer directly applies. This is not an issue in the CJS models where only survival and detection are of interest. But in the JS model, if individuals can move outside the state-space and remain alive, then the density must be recalculated such that we only count those individuals with activity centers *within* the state-space. This was previously not an issue because the prior on the activity centers constrained all individuals to the state-space at all times.

16.4.1 Thoughts on movement of American shad

In our American shad example above, we had reason to believe that individual movement is directly related to stream flow. When the stream flow is low, we might expect that the fish move very little, and when the stream flow is high, they might move upstream to spawn. In this case, we could model the effect of stream flow in two ways. First, we might allow σ to be a function of flow and to vary for each primary occasion, according to:

$$\log(\sigma_t) = \mu_\sigma + \alpha_2 \text{Flow}_t$$

But if we think that the change in activity centers between primary periods might be related to the general pattern of fish migrating upstream more during high flow or staying closer to the same location in low flow, then we could allow the correlation in activity centers to be a function of flow. In this case, for example, a low flow period might indicate that activity centers are more correlated to the previous time period because fish are not actively migrating during such a time. This means that we assume the activity centers are correlated so we have

$$\mathbf{s}(i, t) \sim \text{Normal}(\mathbf{s}(i, t-1), \tau^2 \mathbf{I})$$

where

$$\log(\tau) = \mu_\tau + \alpha_2 \text{Flow}_t.$$

These are just a few thoughts on simple ways to model movement as a function of habitat variables which we have only started exploring on these data. As we discussed in the previous section, there are many other movement models that could be used.

16.4.2 Modeling dispersal

Dispersal is a well studied area in population ecology and is often of heightened interest because it relates directly to population regulation, habitat quality, and linking of local populations. However, studying dispersal with capture-recapture data can be difficult for a few reasons. One common issue with using capture-recapture data for dispersal estimation is that short distances are sampled more frequently than long distances. This is particularly true if we consider that most trap arrays are not large relative the potential

13667 dispersal distances of animals. In some cases, such as with small mammals, we may be able
 13668 to capture both short and long distance dispersals in one trap array; in other cases, we may
 13669 have discrete study sites set up across a larger area which capture individuals within and
 13670 between sites. Either way, data are likely to be sparse for long distance dispersal events
 13671 and this is particularly true if there are different habitat types which are sampled with
 13672 different levels of effort (Ovaskainen et al., 2008), thus causing more difficulty in fitting
 13673 models to data where much information is missing. In addition to that, determining if an
 13674 individual has left an area or died can be difficult if the sampling does not cover the area
 13675 an individual has moved to or if the sampling method has failed (e.g., a band or tag falls
 13676 off or a mark is lost).

13677 Regardless of these common sampling limitations, let's look at an optimal the situation
 13678 where we have the trap array large enough to observe some dispersal events (or possibly
 13679 multiple trap arrays on the landscape where an individual is observed in different arrays). We sketch out a possible dispersal model but note that this is a simple example.
 13680 In this case, each individual could have some probability of dispersing, say η where
 13681 $pd_{i,t} \sim \text{Bernoulli}(\eta)$ indicates if an individual disperses at time t and then

$$s_{i,t+1,1} = s_{i,t,1} + pd_{i,t}(ds_{i,t}\cos(\theta_{i,t})) \\ 13683 \quad s_{i,t+1,2} = s_{i,t,2} + pd_{i,t}(ds_{i,t}\sin(\theta_{i,t}))$$

13684 where ds_i is the dispersal distance for individual i and θ is the dispersal direction (in
 13685 radians). Thus when $pd_i = 0$, then the activity centers remain the same as the previous
 13686 time step and if $pd_i = 1$ then the individual disperses to a new activity center. For this
 13687 specification, we have to provide a model for dispersal distance. One option is to let $ds_{i,t} \sim$
 13688 exponential(L) where L is the mean dispersal distance for individuals dispersing and let
 13689 $\theta_{i,t} \sim \text{Uniform}(-\pi, \pi)$ where π is not a parameter in this case, but the mathematical
 13690 constant(i.e., $\pi = 3.14159\dots$). If all individuals are expected to move some distance
 13691 between periods, then the pd indicator could be removed. A number of distributions
 13692 exists for fitting these parameters (e.g., the von Mises is commonly used for angles) and
 13693 more complex models with components like weighted directional movement and various
 13694 movement states could be fit (see Jonsen et al. (2005); ?); McClintock et al. (2012))

16.5 SUMMARY AND OUTLOOK

13695 In this chapter we have described a framework for making inference not only about spatial
 13696 and temporal variation in population density, but also demographic parameters including
 13697 survival, recruitment, and movement. The ability to model population vital rates is es-
 13698 sential for ecology, management, and conservation; and the models described here allow
 13699 researchers to examine the spatial and temporal dynamics governing those population
 13700 parameters. While we have covered a lot of ground in this chapter, but there are many
 13701 variations of the basic JS and CJS models, such as dead recovery models or models that
 13702 address transiency that we have not explicitly 'converted' to a spatial framework, and
 13703 these areas provide a broad field of further model development.

13704 As open models are further developed, mechanisms for dealing directly with dispersal
 13705 and transients will provide improved inference frameworks for understanding movement
 13706 as well as the potential to estimate *true* survival instead of only *apparent survival*. This
 13707 is a function of explicitly modeling movement, which means we can separate movement

13708 from mortality, as we sketched out in the model above for dispersal, providing a huge
13709 advantage over traditional models. Also, models of individual dispersal can be used to
13710 examine dynamics of population dynamics relative to habitat, density-dependence, or
13711 climatic events.

13712 Birth and death processes, as well as movement, all have the potential to be related
13713 to the space usage of animals in the landscape. Understanding the impact of spatially
13714 varying density on survival and recruitment will provide insights into the basic ecology
13715 of species. With the advent of non-invasive techniques, like camera trapping and genetic
13716 analysis of tissue, we can start to understand the population dynamics of species that are
13717 rarely observed in the wild. As more and more data are collected, we can use the models
13718 to explore the spatio-temporal patterns of survival, recruitment, density, and movement
13719 of species, providing incredibly useful biological and ecological information as we face
13720 broad changes in climate, land-use, habitat fragmentation, etc. Rathbun and Cressie
13721 (1994) articulate a model for marked point processes where they separate out the spatial
13722 birth, growth, and survival processes for longleaf pine trees. Because of the application,
13723 these demographic parameters are slightly different than how they are often considered in
13724 wildlife and ecology, but still, there are analogies. Allowing birth, growth, and survival
13725 as well as density to arise from different spatially varying processes is the next stage in
13726 development of the open SCR models.

13727

Part IV

13728

13729

Super-Advanced SCR Models

13730
13731

17

13732
13733

DEVELOPING MARKOV CHAIN MONTE CARLO SAMPLERS

13734 In this chapter we will dive a little deeper into Markov chain Monte Carlo (MCMC)
13735 sampling. We will construct custom MCMC samplers in **R**, starting with easy-to-code
13736 GLMs and GLMMs and moving on to simple CR and SCR models. This material might
13737 seem slightly out of place here, as it does not deal with specific aspects or modifications of
13738 SCR models, but rather, with a particular way of implementing them (and other models,
13739 too). Knowing how to build an MCMC sampler is not essential for any of the SCR models
13740 we have covered so far, but we will need these skills to implement some models that
13741 come up in the last few chapters of this book. The aim of this chapter is to provide you
13742 with some working knowledge of building MCMC samplers. To this end, we will NOT
13743 provide exhaustive background information on the theory and justification of MCMC
13744 sampling – there are entire books dedicated to that subject and we refer you to Robert
13745 and Casella (2004) and Robert and Casella (2010). Rather we aim to provide you with
13746 enough background and technical know-how to start building your own MCMC samplers
13747 for SCR models in **R**. You will find that quite a few topics that come up in this chapter
13748 have already been covered in previous chapters, particularly the introduction into Bayesian
13749 analysis in Chapt. 3. To keep you from having to leaf back and forth we will in some
13750 places briefly review aspects of Bayesian analysis, but we try to focus on the more technical
13751 issues of building MCMC samplers relevant to SCR models.

17.1 WHY BUILD YOUR OWN MCMC ALGORITHM?

13752 The standard programs we have used so far to do MCMC analyses are **WinBUGS** (Gilks
13753 et al., 1994) and **JAGS** (Plummer, 2003). The wonderful thing about these **BUGS**
13754 engines is that they automatically use appropriate and, most of the time, reasonably
13755 efficient forms of MCMC sampling for the model specified by the user.

13756 The fact that we have such a Swiss Army knife type of MCMC machine begs the
13757 question: Why would anyone want to build their own MCMC algorithm? For one, there

13758 are a limited number of distributions and functions implemented in **BUGS**. While **Open-**
 13759 **BUGS** and **JAGS** provide more options, some more complex models may be impossible
 13760 to build within these programs. A very simple example from spatial capture-recapture
 13761 that can give you a headache in **WinBUGS** is when your state-space is an irregular-
 13762 shaped polygon, rather than an ideal rectangle that can be characterized by four pairs
 13763 of coordinates. It is easy to restrict activity centers to any arbitrary polygon in **R** using
 13764 an ESRI shapefile (and we will show you an example in a little bit), but you cannot use
 13765 a shapefile in a **BUGS** model. Similarly, models of space usage that take into account
 13766 ecological distance (Chapt. 12) cannot be implemented in the **BUGS** engines.

13767 Sometimes, implementing an MCMC algorithm in **R** may be faster than in **Win-**
 13768 **BUGS** - especially if you want to run simulation studies where you have hundreds or
 13769 more simulated data sets, several years' worth of data or other large models, this can be
 13770 a big advantage. Further, writing your own sampler gives you more control over which
 13771 kind of updater is used (see following sections). Finally, building your own MCMC al-
 13772 gorithm is a great exercise to understand how MCMC sampling works. So while using
 13773 the **BUGS** language requires you to understand the structure of your model, building an
 13774 MCMC algorithm requires you to think about the relationship between your data, priors
 13775 and posteriors, and how these can be efficiently analyzed and characterized. However,
 13776 if you don't think you will ever sit down and write your own MCMC sampler, consider
 13777 skipping this chapter - apart from coding it will not cover anything SCR-related that is
 13778 not covered by other, more model-oriented chapters as well.

17.2 MCMC AND POSTERIOR DISTRIBUTIONS

13779 MCMC is a class of simulation methods for drawing (correlated) random numbers from
 13780 a target distribution, which in Bayesian inference is the posterior distribution. As a re-
 13781 minder, the posterior distribution is a probability distribution for an unknown parameter,
 13782 say θ , given observed data and its prior probability distribution (the probability distribu-
 13783 tion we assign to a parameter before we observe data). The great benefit of having the
 13784 posterior distribution of θ is that it can be used to make probability statements about
 13785 θ , such as the probability that θ is equal to some value, or the probability that θ falls
 13786 within some range of values. The posterior distribution summarizes all we know about a
 13787 parameter and thus, is the central object of interest in Bayesian analysis. Unfortunately,
 13788 in many if not most practical applications, it is nearly impossible to directly compute the
 13789 posterior. Recall Bayes' theorem:

$$[\theta|y] = \frac{[y|\theta][\theta]}{[y]}, \quad (17.2.1)$$

13790 where θ is the parameter of interest, y is the observed data, $[\theta|y]$ is the posterior, $[y|\theta]$ the
 13791 likelihood of the data conditional on θ , $[\theta]$ the prior probability of θ , and, finally, $[y]$ is the
 13792 marginal probability of the data, defined as

$$[y] = \int [y|\theta][\theta]d\theta$$

13793 This marginal probability is a normalizing constant that ensures that the posterior
 13794 integrates to 1. Often, the integral is difficult or impossible to evaluate, unless you are

13795 dealing with a really simple model. For example, consider a normal model, with a set of
 13796 n observations, $y_i; i = 1, 2, \dots, n$:

$$y_i \sim \text{Normal}(\mu, \sigma),$$

13797 where σ is known and our objective is to estimate μ . To fully specify the model in a
 13798 Bayesian framework, we first have to define a prior distribution for μ . Recall from Chapt.
 13799 3 that for certain data models, certain priors lead to conjugacy, i.e. if you choose a certain
 13800 prior for your parameter, the posterior distribution will be of a known parametric form.
 13801 More specifically, under conjugacy, the prior and posterior distributions are from the same
 13802 parametric family. The conjugate prior for the mean of a normal model is also a normal
 13803 distribution:

$$\mu \sim \text{Normal}(\mu_0, \sigma_0^2).$$

13804 If μ_0 and σ_0^2 are fixed, the posterior for μ has the following form (for some of the algebra
 13805 behind this, see Chapt. 2 in Gelman et al. (2004)):

$$\mu|y \sim \text{Normal}(\mu_n, \sigma_n^2) \quad (17.2.2)$$

13806 where

$$\mu_n = \left(\frac{\sigma^2}{\sigma^2 + n\sigma_0^2} \right) \times \left(\mu_0 + \frac{n\sigma_0^2}{\sigma^2 + n\sigma_0^2} \right) \times \bar{y}$$

13807 and

$$\sigma_n^2 = \frac{\sigma^2 \sigma_0^2}{\sigma^2 + n\sigma_0^2}.$$

13808 We can directly obtain estimates of interest from this normal posterior distribution, such
 13809 as its mean $\hat{\mu}$ (which is equivalent to an estimate of μ_n) and variance; we do not need
 13810 to apply MCMC, since we can recognize the posterior as a parametric distribution, in-
 13811 cluding the normalizing constant $[y]$. But generally we will be interested in more complex
 13812 models with several, say m , parameters. In this case, computing $[y]$ from Eq. 17.2.1 re-
 13813 quires m -dimensional integration, which can be difficult or impossible. Thus, the posterior
 13814 distribution is generally only known up to a constant of proportionality:

$$[\theta|y] \propto [y|\theta][\theta]$$

13815 The power of MCMC is that it allows us to approximate the posterior using simulation
 13816 without evaluating the high dimensional integrals, and to directly sample from the pos-
 13817 terior, even when the posterior distribution is unknown! The price is that MCMC is
 13818 computationally expensive. Although MCMC first appeared in the scientific literature in
 13819 1949 (Metropolis and Ulam, 1949), widespread use did not occur until the 1980s when
 13820 computational power and speed increased (Gelfand and Smith, 1990). It is safe to say that
 13821 the advent of practical MCMC methods is the primary reason why Bayesian inference has
 13822 become so popular during the past three decades.

13823 In a nutshell, MCMC lets us generate sequential draws of θ (the parameter(s) of in-
 13824 terest) from distributions approximating the unknown posterior over T iterations. The
 13825 distribution of the draw at t depends on the value drawn at $t-1$; hence, the draws from a
 13826 Markov chain¹. As T goes to infinity, the Markov chain converges to the desired distri-
 13827 bution, in our case the posterior distribution for $\theta|y$. Thus, once the Markov chain has

¹Remember that for T random samples $\theta^{(1)}, \dots, \theta^{(T)}$ from a Markov chain the distribution of $\theta^{(t)}$ depends only on the immediately preceding value, $\theta^{(t-1)}$.

reached its stationary distribution, the generated samples can be used to characterize the posterior distribution, $[\theta|y]$, and point estimates of θ , its standard error and confidence bounds, can be obtained directly from this approximation of the posterior.

17.3 TYPES OF MCMC SAMPLING

There are several general MCMC algorithms in widespread use, the most popular being Gibbs sampling and Metropolis-Hastings sampling, both of which were briefly introduced in Chapt. 3. We will be dealing with these two classes in more detail and use them to construct MCMC algorithms for SCR models. Also, we will briefly review alternative techniques that are applicable in some situations.

17.3.1 Gibbs sampling

Gibbs sampling was named after the physicist J.W. Gibbs by Geman and Geman (1984), who applied the algorithm to a Gibbs distribution². The roots of Gibbs sampling can be traced back to work of Metropolis et al. (1953), and it is actually closely related to Metropolis sampling (see Chapt. 11.5 in Gelman et al. (2004), for the link between the two samplers). We will focus on the technical aspects of this algorithm, but if you find yourself hungry for more background, Casella and George (1992) provide a more in-depth introduction to the Gibbs sampler.

Let's go back to our simple example from above to understand the motivation and functioning of Gibbs sampling. Recall that for a normal model with known variance and a normal prior for μ , the posterior distribution of $\mu|y$ is also normal. Conversely, with a fixed (known) μ , but unknown variance, the conjugate prior for σ^2 is an inverse-gamma distribution with shape and scale parameters a and b :

$$\sigma^2 \sim \text{Inverse-Gamma}(a, b).$$

With fixed a and b , algebra reveals that the posterior $[\sigma^2|\mu, y]$ is also an inverse-gamma distribution, namely:

$$\sigma^2|\mu, y \sim \text{Inverse-Gamma}(a_n, b_n), \quad (17.3.1)$$

where $a_n = n/2 + a$ and $b_n = (1/2) \sum_{i=1}^n (y_i - \mu)^2 + b$. However, what if we know neither μ nor σ^2 , which is probably the more common case? The joint posterior distribution of μ and σ^2 now has the general structure

$$[\mu, \sigma^2|y] = \frac{[y|\mu, \sigma^2][\mu][\sigma^2]}{\int [y|\mu][\mu][\sigma^2]d\mu d\sigma^2}$$

or

$$[\mu, \sigma^2|y] \propto [y|\mu, \sigma^2][\mu][\sigma^2]$$

²a distribution from physics we are not going to worry about, since it has no immediate connection with Gibbs sampling other than giving its name

13855 This cannot easily be reduced to a distribution we recognize. However, we can con-
 13856 dition μ on σ^2 (i.e., we treat σ^2 as fixed) and remove all terms from the joint posterior
 13857 distribution that do not involve μ to construct the full conditional distribution,

$$[\mu|\sigma^2, y] \propto [y|\mu][\mu]$$

13858 The full conditional of μ again takes the form of the normal distribution shown in Eq.
 13859 17.2.2; similarly, $[\sigma^2|\mu, y]$ takes the form of the inverse-gamma distribution shown in Eq.
 13860 17.3.1, both distributions we can easily sample from. And this is precisely what we do
 13861 when using Gibbs sampling: we break down high-dimensional problems into convenient
 13862 one-dimensional problems by constructing the full conditional distributions for each model
 13863 parameter separately; and we sample from these full conditionals, which, if we choose
 13864 conjugate priors, are known parametric distributions. Let's put the concept of Gibbs
 13865 sampling into the MCMC framework of generating successive samples, using our simple
 13866 normal model with unknown μ and σ^2 and conjugate priors as an example. These are the
 13867 steps you need in order to build a Gibbs sampler:

13868 **Step 0:** Begin with some initial values for θ , say $\theta^{(0)}$. In our example, $\theta = (\mu, \sigma)$, so
 13869 we have to specify initial values for μ and σ , for example by drawing a random number
 13870 from some uniform distribution, or by setting them close to what we think they might be.
 13871 (Note: This step is required in any MCMC sampling; chains have to start from somewhere.
 13872 We will get back to these technical details a little later.)

13873 **Step 1:** For iteration t , Draw $\theta^{(t)}$ from the conditional distribution $[\theta_1^{(t)}|\theta_2^{(t-1)}, \dots, \theta_d^{(t-1)}]$.
 13874 Here, θ_1 is μ , which we draw from the normal distribution in Eq. 17.2.2 using $\sigma^{(t-1)}$ as
 13875 value for σ .

13876 **Step 2:** Draw $\theta_2^{(t)}$ from the conditional distribution $[\theta_2^{(t)}|\theta_1^{(t)}, \theta_3^{(t-1)}, \dots, \theta_d^{(t-1)}]$. Here, θ_2
 13877 is σ , which we draw from the inverse-gamma distribution of Eq. 17.3.1, using the newly
 13878 generated $\mu^{(t)}$ as value for μ .

13879 **Step 3, ..., d:** Draw $\theta_3^{(t)}, \theta_4^{(t)}, \dots, \theta_d^{(t)}$ from their conditional distribution $[\theta_3^{(t)}|\theta_1^{(t)}, \theta_2^{(t)}, \theta_4^{(t-1)},$
 13880 $\dots, \theta_d^{(t-1)}], \dots, [\theta_d^{(t)}|\theta_1^{(t)}, \dots, \theta_{d-1}^{(t)}]$. In our example we have no additional parameters,
 13881 so we only need step 0 through to 2.

13882 **Repeat Steps 1 to d for $T =$** a large number of samples.

13883 Note that the order in which we update the parameters within the Gibbs algorithm
 13884 does not matter. In terms of **R** coding, this means we have to write Gibbs updaters for
 13885 μ and σ^2 and embed them into a loop over T iterations. The final code in the form of an
 13886 **R** function is shown in Panel 17.1.

13887 This is it! You can go ahead and simulate some data, $y \sim \text{Normal}(5, 0.5)$ and then
 13888 use the function **NormGibbs()** in the **R** package **scrbook** to run your first Gibbs sampler
 13889 (note that the **R** function **rnorm** requires you to supply the standard deviation σ and we
 13890 have written **NormGibbs** so that it returns σ instead of σ^2 so you can easily compare your
 13891 input value and parameter estimate).

```
13892 > set.seed(13)
13893
```

```
Norm.Gibbs<-function(y=,mu_0=mu_0,sigma2_0=sigma2_0,a=a,b=b,niter=niter){

ybar<-mean(y)
n<-length(y)
mu<-1           #mean initial value
sigma2<-1        #sigma2 initial value
an<-n/2 + a      #shape parameter of InvGamma of sigma2
out<-matrix(nrow=niter, ncol=2)
colnames(out)<-c('mu', 'sig')

for (i in 1:niter) {

#update mu
mu_n<-((sigma2/(sigma2+n*sigma2_0))*mu_0
+ (n*sigma2_0/(sigma2 + n*sigma2_0))*ybar)
sigma2_n <- (sigma2*sigma2_0)/ (sigma2 + n*sigma2_0)
mu<-rnorm(1,mu_n, sqrt(sigma2_n))

#update sigma2
bn<- 0.5 * (sum((y-mu)^2)) + b
sigma2<-1/rgamma(1,shape=an, rate=bn)
out[i,]<-c(mu,sqrt(sigma2))
}
return(out)
}
```

Panel 17.1: R-code for a Gibbs sampler for a normal model with unknown μ and σ and conjugate priors (normal and inverse-gamma, respectively) for both parameters.

```

13894 #true mean and sd are 5 and 0.5
13895 > y<-rnorm(1000, 5,0.5) #data
13896
13897 > mu_0<-0 #prior mean
13898 > sigma2_0<-100 #prior variance
13899
13900 #inverse-gamma hyperparameters
13901 > a<-0.1
13902 > b<-0.1
13903
13904 > mod=Norm.Gibbs(y, mu_0, sigma2_0, a,b,niter=10000)

```

13905 Your output, `mod`, will be a table with two columns, one per parameter, and T rows,
 13906 one per iteration. For this 2-parameter example you can visualize the joint posterior by
 13907 plotting samples of μ against samples of σ (Fig. 17.1):

```
13908 > plot(out[,1], out[,2])
```

13909 The marginal distribution of each parameter is approximated by examining the samples
 13910 of this particular parameter. You can visualize it by plotting a histogram of the samples
 13911 (Fig. 17.2 upper left and right):

```

13912 > par(mfrow=c(1,2))
13913 > hist(out[,1]); hist(out[,2])

```

13914 Finally, recall an important characteristic of MCMC, namely, that the chain has to
 13915 have converged (reached its stationary distribution) in order to regard samples as coming
 13916 from the posterior distribution. In practice, that means you have to throw out some
 13917 of the initial samples called the burn-in. We will talk about this in more detail when
 13918 we talk about convergence diagnostics. For now, you can use the `plot(out[,1])` or
 13919 `plot(out[,2])` command to make a time series plot of the samples of each parameter and
 13920 visually assess how many of the initial samples you should discard. Fig. 17.2 bottom left
 13921 and right shows plots for the samples of μ and σ from our simulated data set; you see that
 13922 in this simple example the Markov chain apparently reaches its stationary distribution
 13923 very quickly – the chains look ‘grassy’ seemingly from the start. It is hard to discern a
 13924 burn-in phase visually (but we will see examples further on where the burn-in is clearer)
 13925 and you may just discard the first 500 draws to be sure you only use samples from the
 13926 posterior distribution. The mean of the remaining samples are your estimates of μ and σ :

```

13927 > summary(mod[501:10000,])
13928      mu           sig
13929 Min. :4.935   Min. :0.4652
13930 1st Qu.:4.988   1st Qu.:0.4930
13931 Median :4.998   Median :0.5006
13932 Mean   :4.998   Mean   :0.5008
13933 3rd Qu.:5.009   3rd Qu.:0.5084
13934 Max.   :5.062   Max.   :0.5486

```

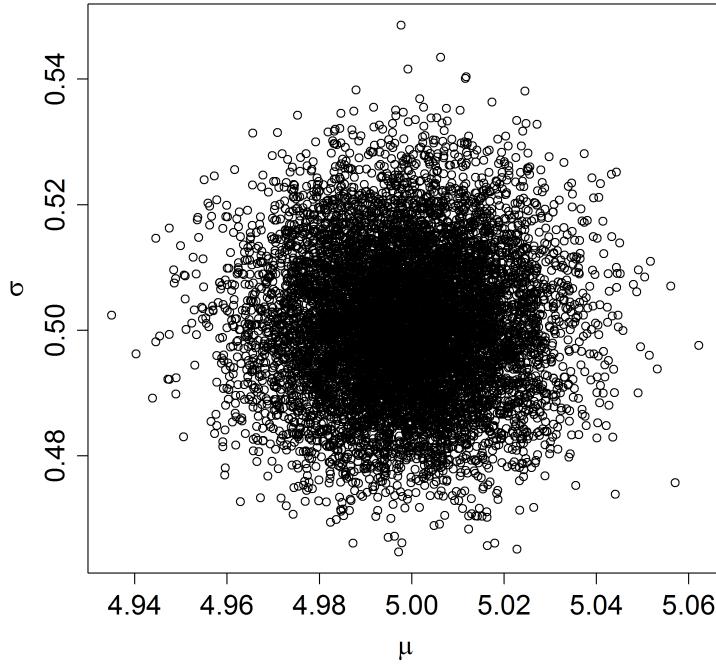


Figure 17.1. Joint posterior distribution of μ and σ from a normal Model

17.3.2 Metropolis-Hastings sampling

Although it is applicable to a wide range of problems, the limitations of Gibbs sampling are obvious: what if we do not want to use conjugate priors or what if we cannot recognize the full conditional distribution as a parametric distribution, or simply do not want to worry about these issues? The most general solution is to use the Metropolis-Hastings (MH) algorithm, which also goes back to the work by Metropolis et al. (1953). You saw the basics of this algorithm in Chapt. 3. In a nutshell, because we do not recognize the posterior $[\theta|y]$ as a parametric distribution, the MH algorithm generates samples from a known proposal distribution, say $h(\theta)$, that depends on the value of θ at the previous time step, $\theta^{(t-1)}$. The candidate value θ^* is accepted with probability

$$r = \min\left(1, \frac{[\theta^*|y]h(\theta^{(t-1)}|\theta^*)}{[\theta^{(t-1)}|y]h(\theta^*|\theta^{(t-1)})}\right)$$

Proposal distributions must be chosen so that reversibility is ensured. That means,

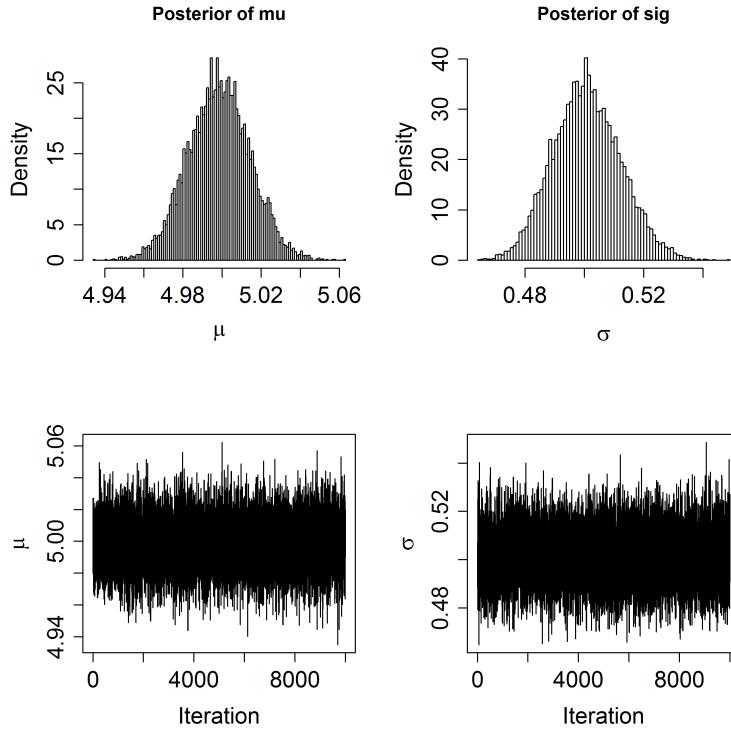


Figure 17.2. Plots of the posterior distributions of μ (upper left) and σ (upper right) from a normal model and time series plots of μ (lower left) and σ (lower right).

it must be possibly to go from any one value to any other. But within that criterion the proposal distribution can be absolutely anything! You can generate candidate values from a Normal(0,1) distribution, from a Uniform(-3455,3455) distribution, or anything of proper support. Note, however, that good choices of $h(\theta)$ are those that approximate the posterior distribution. Obviously if $h(\theta) = [\theta|y]$ (i.e., the posterior) then you always accept the draw and it stands to reason that proposals that are more similar to $[\theta|y]$ will lead to higher acceptance probabilities. Actually, when $h(\theta) = [\theta|y]$ we can draw samples of θ directly from $h(\theta)$, which brings us back to Gibbs sampling. Thus, Gibbs sampling is a special case of Metropolis-Hastings sampling.

The original Metropolis algorithm required $h(\theta)$ to be symmetric so that

$$h(\theta^*|\theta^{(t-1)}) = h(\theta^{(t-1)}|\theta^*)$$

In that case these two terms just cancel out from the MH acceptance probability and r is then just the ratio of the target density evaluated at the candidate value to that evaluated

13958 at the current value. A later development of the algorithm by Hastings (1970) lifted this
 13959 condition. Since using a symmetric proposal distribution makes life a little easier, we are
 13960 going to focus on this specific case. A type of symmetric proposal useful in many situations
 13961 is the so-called *random-walk* proposal distribution where candidate values are drawn from
 13962 a normal distribution with mean equal to the current value and some standard deviation,
 13963 say δ , which is prescribed by the user (see below for further explanation).

13964 **Parameters with bounded support:** Many models contain parameters that have
 13965 bounded support. E.g., variance parameters live on $[0, \infty]$, parameters that represent
 13966 probabilities live on $[0, 1]$, etc.. For such cases, it is sometimes convenient to use a random
 13967 walk proposal distribution that can generate any real number (e.g., a normal random walk
 13968 proposal). Under these circumstances you should not constrain the proposal distribution
 13969 itself, but you can just reject parameters that are outside of the parameter space (sec. 6.4.1
 13970 in Robert and Casella, 2010). You will see plenty of examples of updating parameters with
 13971 bounded support in this chapter.

13972 It is worth knowing that there are alternatives to the random walk MH algorithm.
 13973 For example, in the independent MH, the proposal distribution h does not depend on
 $\theta^{(t-1)}$, while the Langevin algorithm (Roberts and Rosenthal, 1998) aims at avoiding the
 13974 random walk by favoring moves towards regions of higher posterior probability density.
 13975 The interested reader should look up these algorithms in Robert and Casella (2004) or
 13976 Robert and Casella (2010).

13977 Building a MH sampler can be broken down into several steps. We are going to
 13978 demonstrate these steps using a different but still simple and common model: the logit-
 13979 normal or logistic regression model. For simplicity, assume that

$$y|\theta \sim \text{Bernoulli} \left(\frac{\exp(\theta)}{1 + \exp(\theta)} \right)$$

13981 and

$$\theta \sim \text{Normal}(\mu, \sigma).$$

13982 The following steps are required to set up a random walk MH algorithm:

13983 **Step 0:** Choose initial values, $\theta^{(0)}$.

13984 **Step 1:** Generate a proposed value of θ from $h(\theta^*|\theta^{(t-1)})$. We will use the random walk
 13985 MH algorithm, so we draw θ^* from $\text{Normal}(\theta^{(t-1)}, \delta)$, where δ is the standard deviation
 13986 of the normal proposal distribution, the tuning parameter that we have to set.

13987 **Step 2:** Calculate the ratio of posterior densities for the proposed and the original value
 13988 for θ :

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

13989 In our example,

$$r = \frac{\text{Bernoulli}(y|\theta^*) \times \text{Normal}(\theta^*|\mu, \sigma)}{\text{Bernoulli}(y|\theta^{(t-1)}) \times \text{Normal}(\theta^{(t-1)}|\mu, \sigma)}$$

13990 **Step 3:** Set

$$\begin{aligned} \theta^t &= \theta^* \text{ with probability } \min(r, 1) \\ &= \theta^{(t-1)} \text{ otherwise} \end{aligned}$$

13991 We can do this last step by drawing a random number u from a Uniform(0, 1) and
 13992 accept θ^* if $u < r$. This is repeated for $t = 1, 2, \dots, T$ a large number of samples. As for
 13993 Gibbs sampling, the order in which we update parameters does not matter. The **R** code
 13994 for this MH sampler is provided in Panel 17.2.

```
Logreg.MH<-function(y=y, mu0=mu0, sig0=sig0, delta=delta, niter=niter) {
  out<-c()
  theta<-runif(1, -3,3) #initial value
  for (iter in 1:niter){
    theta.cand<-rnorm(1, theta, delta)
    loglike<-sum(dbinom(y, 1, exp(theta)/(1+exp(theta)), log=TRUE))
    logprior <- dnorm(theta,mu0 ,sig0, log=TRUE)
    loglike.cand<-sum(dbinom(y, 1, exp(theta.cand)/(1+exp(theta.cand)),
      log=TRUE))
    logprior.cand <- dnorm(theta.cand, mu0, sig0, log=TRUE)
    if (runif(1)<exp((loglike.cand+logprior.cand)-(loglike+logprior))){
      theta<-theta.cand
    }
    out[iter]<-theta
  }
  return(out)
}
```

Panel 17.2: **R** code to run a Metropolis sampler on a simple logit-normal model.

13995 The reason why in the **R** code we sum the logs of the likelihood and the prior, rather
 13996 than multiplying the original values, is simply computational. The product of small prob-
 13997 abilities can be numbers very close to 0, which computers do not handle well. Thus we
 13998 add the logarithms, sum, and exponentiate to achieve the desired result. Similarly, in
 13999 case you have forgotten, $x/y = \exp(\log(x) - \log(y))$, with the latter being favored for
 14000 computational reasons.

14001 Comparing MH sampling to Gibbs sampling, where all draws from the conditional
 14002 distribution are used, in the MH algorithm we discard a portion of the candidate values,
 14003 which inherently makes it less efficient than Gibbs sampling – the price you pay for its
 14004 increased generality. In Step 1 of the MH sampler we had to choose a variance, δ , for
 14005 the normal proposal distribution. Choice of the parameters that define our candidate

14006 distribution is also referred to as 'tuning', and it is important since adequate tuning will
 14007 make your algorithm more efficient. δ should be chosen (a) large enough so that each step
 14008 of drawing a new proposal value for θ can cover a reasonable distance in the parameter
 14009 space, as otherwise, mixing of the Markov chain is inefficient and chains will tend to have
 14010 strong autocorrelation; and (b) small enough so that proposal values are not rejected too
 14011 often, as otherwise the random walk will 'get stuck' at specific values for too long. As a rule
 14012 of thumb, your candidate value should be accepted in about 40% of all cases. Acceptance
 14013 rates of 20 – 80% are probably ok, but anything below or above may well render your
 14014 algorithm inefficient (this does not mean that it will give you wrong results, only that you
 14015 will need more iterations to converge to the posterior distribution). In practice, tuning
 14016 will require some 'trial-and-error', some common sense and, with enough experience, some
 14017 intuition. Or, one can use an adaptive phase, where the tuning parameter is automatically
 14018 adjusted until it reaches a user-defined acceptance rate, at which point the adaptive phase
 14019 ends and the actual Markov chain begins. This is computationally a little more advanced.
 14020 Link and Barker (2010) discuss this in more detail. It is important that the samples drawn
 14021 during the adaptive phase are discarded.

14022 To illustrate the effects of tuning, we ran the Metropolis-Hastings algorithm in Panel
 14023 17.2 with $\delta = 0.01$, $\delta = 0.2$ and $\delta = 1$. The first 150 iterations for θ are shown in Fig. 17.3.
 14024 We see that for a very small δ (the dashed line) the burn-in is extremely slow - after 150
 14025 iterations the chain isn't even half way there, while for the other two values of δ (solid and
 14026 dotted) the burn-in phase seems to be over after only about 10 iterations. While $\delta = 0.2$
 14027 leads to reasonably good mixing, the chain clearly gets stuck on certain values with $\delta = 1$.
 14028

14029 Other than graphically, you can easily check acceptance rates for the parameters you
 14030 monitor (that are part of your output) using the `rejectionRate()` function of the package
 14031 `coda` (we will talk more about this package a little later on). Do not let the term 'rejection
 14032 rate' confuse you; it is simply $1 - \text{acceptance rate}$. There may be parameters – for example,
 14033 individual values of a random effect or latent variables – that you do not want to save,
 14034 though, and in our next example we will show you a way to monitor their acceptance rates
 14035 with a few extra lines of code.

14036 17.3.3 Metropolis-within-Gibbs

14037 One weakness of the MH sampler is that formulating the joint posterior when evaluating
 14038 whether to accept or reject the candidate values for θ becomes increasingly complex or
 14039 inefficient as the number of parameters in a model increases. As you already saw in Chapt.
 14040 3, in these cases you can simply combine MH sampling and Gibbs sampling. You can use
 14041 the principles of Gibbs sampling to break down your high-dimensional parameter space
 14042 into easy-to-handle one-dimensional conditional distributions and use MH sampling for
 14043 these conditional distributions. Better yet, if you have some conjugacy in your model,
 14044 you can use the more efficient Gibbs sampling for these parameters and one-dimensional
 14045 MH for all the others. You have already seen the basics of how to build both types of
 14046 algorithms, so we can jump straight into an example here and build a Metropolis-within-
 14047 Gibbs algorithm.

14048 **GLMMs: Poisson regression with a random effect** Let's assume a model that gets
 14049 us closer to the problem we ultimately want to deal with - a GLMM. Here, we assume

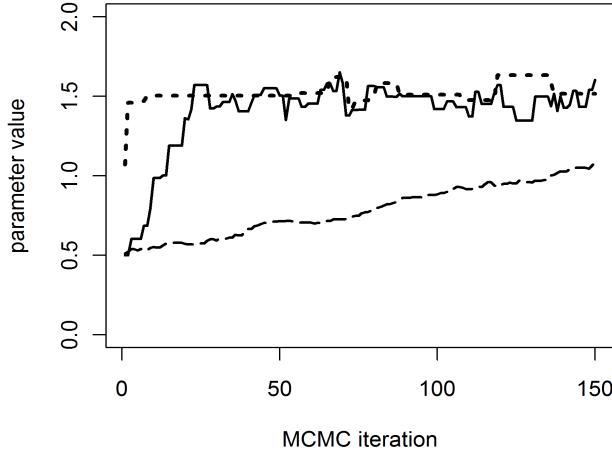


Figure 17.3. Time series plots of θ from a MH algorithm with tuning parameter $\delta = 0.01$ (dashed line), 0.2 (solid line) and 1 (dotted line).

14050 we have Poisson counts, y_{ij} , from $j = 1, 2, \dots, n$ plots in i different study sites, and we
 14051 believe that the counts are influenced by some plot-specific covariate, \mathbf{x} , but that there is
 14052 also a random site effect. So our model is:

$$\begin{aligned} 14053 \quad y_{ij} &\sim \text{Poisson}(\lambda_{ij}) \\ \lambda_{ij} &= \exp(\alpha_i + \beta x_{ij}) \end{aligned}$$

14054 Let's place normal priors on α and β ,

$$\alpha_i \sim \text{Normal}(\mu_\alpha, \sigma_\alpha)$$

14055 and

$$\beta \sim \text{Normal}(\mu_\beta, \sigma_\beta)$$

14056 In this model, we do not specify μ_α and σ_α , but instead, estimate them as well, so we
 14057 have to specify hyperpriors for these parameters:

$$\begin{aligned} \mu_\alpha &\sim \text{Normal}(\mu_0, \sigma_0) \\ \sigma_\alpha^2 &\sim \text{Inverse-Gamma}(a_0, b_0) \end{aligned}$$

14058 Note that for simplicity we assume that β is constant across the i study sites, and for
 14059 analysis we set μ_β and σ_β (i.e., we don't estimate these parameters from the data). With

14060 the model completely specified, we can compile the full conditionals, breaking the multi-dimensional parameter space into one-dimensional components:
 14061

$$\begin{aligned} [\alpha_1 | \alpha_2, \alpha_3, \dots, \alpha_i, \beta, \mathbf{y}_1] &\propto [\mathbf{y}_1 | \alpha_1, \beta][\alpha_1] \\ &\propto \text{Poisson}(\mathbf{y}_1 | \exp(\alpha_1 + \beta \mathbf{x}_1)) \times \text{Normal}(\alpha_1 | \mu_\alpha, \sigma_\alpha), \end{aligned}$$

14062 where $\mathbf{y}_1 = (y_{11}, y_{12}, \dots, y_{1n})$ is the vector of observed counts for site $i = 1$ and, in general,
 14063 \mathbf{y}_i is the vector of all counts for site i ; analogous, \mathbf{x}_i is the vector of all observations of
 14064 the covariate for site i . The other full conditionals for each α_i are constructed similarly:

$$\begin{aligned} [\alpha_2 | \alpha_1, \alpha_3, \dots, \alpha_i, \beta, \mathbf{y}_2] &\propto [\mathbf{y}_2 | \alpha_2, \beta][\alpha_2] \\ &\propto \text{Poisson}(\mathbf{y}_2 | \exp(\alpha_2 + \beta \mathbf{x}_2)) \times \text{Normal}(\alpha_2 | \mu_\alpha, \sigma_\alpha), \end{aligned}$$

14065 and so on for all elements of α . The full-conditional for β is:

$$\begin{aligned} [\beta | \boldsymbol{\alpha}, \mathbf{y}] &\propto [\mathbf{y} | \boldsymbol{\alpha}, \beta][\beta] \\ &\propto \text{Poisson}(\mathbf{y} | \exp(\boldsymbol{\alpha} + \beta \mathbf{x})) \times \text{Normal}(\beta | \mu_\beta, \sigma_\beta). \end{aligned}$$

14066 Finally, we need to update the hyperparameters for the random effects vector α :

$$\begin{aligned} [\mu_\alpha | \boldsymbol{\alpha}] &\propto [\boldsymbol{\alpha} | \mu_\alpha, \sigma_\alpha][\mu_\alpha] \\ [\sigma_\alpha | \boldsymbol{\alpha}] &\propto [\boldsymbol{\alpha} | \mu_\alpha, \sigma_\alpha][\sigma_\alpha] \end{aligned}$$

14068 Note that the likelihood contributions of the counts \mathbf{y} at each site, when conditioned
 14069 on $\boldsymbol{\alpha}$, do not depend on the hyperparameters μ_α and σ_α . As such, the full conditionals
 14070 for these hyperparameters only depend on the collection of all $\boldsymbol{\alpha}$, not the data. Since we
 14071 assumed $\boldsymbol{\alpha}$ to come from a normal distribution, the choice of priors for μ_α (normal) and
 14072 σ_α^2 (inverse-gamma) leads to the same conjugacy we observed in our initial normal model,
 14073 so that both hyperparameters can be updated using Gibbs sampling.

14074 Now let's build the updating steps for these full conditionals. Again, for the MH steps
 14075 that update $\boldsymbol{\alpha}$ and β we use normal proposal distributions with standard deviations δ_α
 14076 and δ_β .

14077 First, we set the initial values $\boldsymbol{\alpha}^{(0)}$ and $\beta^{(0)}$. Then, starting with α_1 , we draw $\alpha_1^{(1)}$
 14078 from $\text{Normal}(\alpha_1^{(0)}, \delta_\alpha)$, calculate the conditional posterior density of $\alpha_1^{(0)}$ and $\alpha_1^{(1)}$ and
 14079 compare their ratios,

$$r = \frac{\text{Poisson}(\mathbf{y}_1 | \exp(\alpha_1^{(1)} + \beta \mathbf{x}_1)) \times \text{Normal}(\alpha_1^{(1)} | \mu_\alpha, \sigma_\alpha)}{\text{Poisson}(\mathbf{y}_1 | \exp(\alpha_1^{(0)} + \beta \mathbf{x}_1)) \times \text{Normal}(\alpha_1^{(0)} | \mu_\alpha, \sigma_\alpha)}$$

14080 and accept $\alpha_1^{(1)}$ with probability $\min(r, 1)$. We repeat this for all $\boldsymbol{\alpha}$.

14081 For β , we draw $\beta^{(1)}$ from $\text{Normal}(\beta^{(0)}, \delta_\beta)$, compare the posterior densities of $\beta^{(0)}$ and
 14082 $\beta^{(1)}$,

$$r = \frac{\text{Poisson}(\mathbf{y} | \exp(\boldsymbol{\alpha} + \beta^{(1)} \mathbf{x})) \times \text{Normal}(\beta^{(1)} | \mu_\beta, \sigma_\beta)}{\text{Poisson}(\mathbf{y} | \exp(\boldsymbol{\alpha} + \beta^{(0)} \mathbf{x})) \times \text{Normal}(\beta^{(0)} | \mu_\beta, \sigma_\beta)},$$

14083 and accept $\beta^{(1)}$ with probability $\min(r, 1)$.

14084 For μ_α and σ_α^2 , we sample directly from the full conditional distributions (Eq. 17.2.2
 14085 and Eq. 17.3.1):

$$\mu_\alpha^{(1)} \sim \text{Normal}(\mu_n, \sigma_n^2)$$

14086 where

$$\mu_n = \frac{\sigma_\alpha^{2(0)}}{\sigma_\alpha^{2(0)} + n_\alpha \sigma_0^2} \times \mu_0 + \frac{n_\alpha \sigma_0^2}{\sigma_\alpha^{2(0)} + n_\alpha \sigma_0^2} \times \bar{\alpha}^{(1)}$$

14087 and

$$\sigma_n^2 = \frac{\sigma_\alpha^{2(0)} \sigma_0^2}{\sigma_\alpha^{2(0)} + n \sigma_0^2}$$

14088 Here, $\bar{\alpha}$ is the current mean of the vector α , which we updated before, and n_α is the
 14089 length of α . For σ_α^2 we use

$$\sigma_\alpha^{2(1)} \sim \text{Inverse-Gamma}(a_n, b_n),$$

14090 where

$$a_n = n_a / 2 + a_0,$$

14091 and

$$b_n = 0.5 \sum_{i=1}^{n_\alpha} (\alpha_i^{(1)} - \mu_\alpha^{(1)})^2 + b_0.$$

14092 We repeat these steps over T iterations of the MCMC algorithm. Call the function
 14093 **PoisGLMM()** in **scrbook** to check out what this algorithm looks like in **R**.

14094 In this example we may not want to save each individual α_i , but are only interested in
 14095 their mean and standard deviation. Since these two parameters will change as soon as the
 14096 value for one element in α changes, their acceptance rates will always be close to 1 and
 14097 are not representative of how well your algorithm performs. To monitor the acceptance
 14098 rates of parameters you do not want to save, you simply need to add a few lines of code
 14099 into your updater to see how often the individual parameters are accepted. The code for
 14100 updating α from our Poisson GLMM below shows one way how to monitor acceptance of
 14101 individual α_i 's.

```
14102 #initiate counter for acceptance rate of alpha
14103 alphaUps<-0
14104
14105 #loop over sites, update intercepts alpha one at a time;
14106 #only data at site i contributes information
14107 #lev is the number of sites i
14108 for (i in 1:lev) {
14109   alpha.cand<-rnorm(1, alpha[i], delta_alpha)
14110   loglike<- sum(dpois (y[site==i], exp(alpha[i] + beta*x[site==i]),
14111     log=TRUE))
14112   logprior<- dnorm(alpha[i], mu_alpha,sig_alpha, log=TRUE)
14113   loglike.cand<- sum(dpois (y[site==i], exp(alpha.cand + beta *x[site==i]),
14114     log=TRUE))
14115   logprior.cand<- dnorm(alpha.cand, mu_alpha,sig_alpha, log=TRUE)
14116   if (runif(1)< exp((loglike.cand+logprior.cand) -(loglike+logprior))) {
```

```

14117 alpha[i]<-alpha.cand
14118 alphaUps<-alphaUps+1
14119 }
14120 }
14121
14122 #lets you check the acceptance rate of alpha at every 100th iteration
14123 if(iter %% 100 == 0) {
14124     cat("    Acceptance rates\n")
14125     cat("        alpha =", alphaUps/lev, "\n")
14126 }
```

14127 17.3.4 Rejection sampling and slice sampling

14128 While MH and Gibbs sampling are probably the most widely applied algorithms for posterior approximation, there are other options that work under certain circumstances and
 14129 may be more efficient when applicable. **WinBUGS** applies these algorithms and we want
 14130 you to be aware that there is more out there to approximate posterior distributions than
 14131 Gibbs and MH. One alternative algorithm is rejection sampling. Rejection sampling is
 14132 not an MCMC method, since each draw is independent of the others. The method can
 14133 be used when the posterior $[\theta|y]$ is not a known parametric distribution but can be ex-
 14134 pressed in closed form. Then, we can use a so-called envelope function, say, $g(\theta)$, that
 14135 we can easily sample from, with the restriction that $[\theta|y] < M \times g(\theta)$. We then sample a
 14136 candidate value for θ from $g(\theta)$, calculate $r = [\theta|y]/M \times g(\theta)$ and keep the sample with
 14137 the probability r . M is a constant that has to be picked so that r lies between 0 and 1, for
 14138 example by evaluating both $[\theta|y]$ and $g(\theta)$ at n points and looking at their ratios. Rejec-
 14139 tion sampling only works well if $g(\theta)$ is similar to $[\theta|y]$, and packages like **WinBUGS** use
 14140 adaptive rejection sampling (Gilks and Wild, 1992), where a complex algorithm is used to
 14141 fit an adequate and efficient $g(\theta)$ based on the first few draws. Though efficient in some
 14142 situations, rejection sampling does not work well with high-dimensional problems, since
 14143 it becomes increasingly hard to define a reasonable envelope function. For an example
 14144 of rejection sampling in the context of SCR models, see Chapt. 11, where we use it to
 14145 simulation inhomogeneous point processes.

14147 Another alternative is slice sampling (Neal, 2003). In slice sampling, we sample uni-
 14148 formly from the area under the plot of $[\theta|y]$. Considering a single univariate θ . Let's define
 14149 an auxiliary variable, $U \sim \text{Unif}(0, [\theta|y])$. Then, θ can be sampled from the vertical slice
 14150 of $[\theta|y]$ at U (Fig. 17.4):

$$\theta|U \sim \text{Unif}(B),$$

14151 where $B = \{\theta : [\theta|y] \geq U\}$

14152 Slice sampling can be applied in many situations; however, implementing an efficient
 14153 slice sampling procedure can be complicated. We refer the interested reader to Robert and
 14154 Casella (2010, Chapt. 7) for a simple example. Both rejection sampling and slice sampling
 14155 can be applied on one-dimensional conditional distributions within a Gibbs sampling setup.

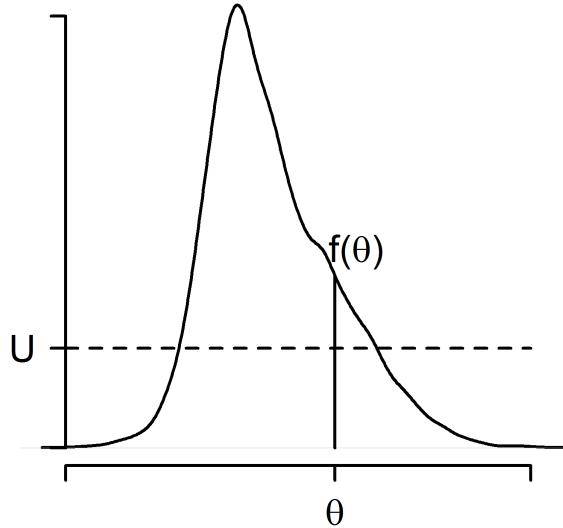


Figure 17.4. Slice sampling. For $U \sim \text{Unif}(0, [\theta|y])$, we can sample θ from the vertical slice of $[\theta|y]$ at U ; $\theta|U \sim \text{Unif}(B)$, where $B = \{\theta : [\theta|y] \geq U\}$.

17.4 MCMC FOR CLOSED CAPTURE-RECAPTURE MODEL M_H

14156 By now you have seen MCMC samplers for some simple generalized (mixed) linear models.
 14157 Now, to ease you into more complex models, we construct our own MCMC algorithm using
 14158 a Metropolis-within-Gibbs sampler for the non-spatial model with individual heterogeneity
 14159 in capture probability, model M_h , developed in Chapt. 4.

14160 To recapitulate: Under the non-spatial model, each of the n observed individuals is
 14161 either detected (1) or not (0) during each of K sampling occasions. We estimate N using
 14162 data augmentation and have a Bernoulli model for the data augmentation variables z_i .

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

14163 The binomial observation model is expressed conditional on the latent variables z_i .

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

14164 Further, we prescribe a distribution for the capture probability p_i . Here we assume

$$\text{logit}(p_i) \sim \text{Normal}(\mu_p, \sigma_p^2)$$

14165 As usual, we have to go through two general steps before we write the MCMC algo-
 14166 rithm:

- 14167 (1) Identify the model with all its components (including priors)
 14168 (2) Recognize and express the full conditional distributions for all parameters

14169 Our model components are as follows: $[y_i|p_i, z_i]$, $[p_i|\mu_p, \sigma_p]$, and $[z_i|\psi]$ for each $i = 1, 2, \dots, M$ and then prior distributions $[\mu_p]$, $[\sigma_p]$ and $[\psi]$. The joint posterior distribution of all unknown quantities in the model is proportional to the joint distribution of all elements y_i, p_i, z_i and also the prior distributions of the prior parameters:

$$\left\{ \prod_{i=1}^M [y_i|p_i, z_i] [p_i|\mu_p, \sigma_p] [z_i|\psi] \right\} [\mu_p, \sigma_p, \psi]$$

14173 For prior distributions, we assume that μ_p, σ_p, ψ are mutually independent and for μ_p
 14174 and σ_p we use improper uniform priors, and $\psi \sim \text{Uniform}(0, 1)$. This is equivalent to
 14175 Beta(1, 1), which will come in handy, as we will see in a moment. Note that the likelihood
 14176 contribution for each individual, when conditioned on p_i and z_i , does not depend on ψ ,
 14177 μ_p , or σ_p . As such, the full-conditional for the structural parameter ψ only depends on the
 14178 collection of data augmentation variables z_i , and that for μ_p and σ_p will only depend on
 14179 the collection of latent variables $p_i; i = 1, 2, \dots, M$ (this is equivalent to what we saw in the
 14180 Poisson regression with random intercept α , where hyperparameters for the distribution
 14181 of α did not depend on the observed data). The full conditionals for all the unknowns are
 14182 as follows:

14183 (1) For p_i :

$$\begin{aligned} [p_i|y_i, \mu_p, \sigma_p, z_i] &\propto [y_i|p_i][p_i|\mu_p, \sigma_p] \text{ if } z_i = 1 \\ &\quad [p_i|\mu_p, \sigma_p] \text{ if } z_i = 0 \end{aligned}$$

14184 (2) for z_i :

$$[z_i|y_i, p_i, \psi] \propto [y_i|z_i \times p_i] \text{Bernoulli}(z_i|\psi)$$

14185 (3) For μ_p :

$$[\mu_p|p_i, \sigma_p] \sim \left\{ \prod_i [p_i|\mu_p, \sigma_p] \right\} \times \text{const}$$

14186 (4) For σ_p :

$$[\sigma_p|p_i, \mu_p] \sim \left\{ \prod_i [p_i|\mu_p, \sigma_p] \right\} \times \text{const}$$

14187 (5) For ψ :

$$[\psi|z_i] \propto \left\{ \prod_i [z_i|\psi] \right\} \text{Beta}(1, 1)$$

14188 Remember that Beta(1,1) is equivalent to Uniform(0,1). The beta distribution is the
 14189 conjugate prior to the binomial and Bernoulli distributions and the general form of a full
 14190 conditional of a beta-binomial model with $x_i \sim \text{Bernoulli}(\theta)$ and $\theta \sim \text{Beta}(a, b)$ is

$$[\theta|\mathbf{x}] \propto \text{Beta}(a + \sum_i x_i, b + n - \sum_i x_i).$$

14191 In our case that means

$$[\psi|z_i] \propto \text{Beta}(1 + \sum z_i, 1 + M - \sum z_i).$$

What we've done here is identify each of the full conditional distributions in sufficient detail to toss them into our Metropolis-Hastings algorithm. The constant terms in the full conditionals for μ_p and σ_p reflect the improper prior we chose for both parameters. Because of the choice of an improper prior, prior probability densities for both parameters $\propto 1$, i.e. constant, and these constants cancel out of the MH acceptance ratio (see updating step below and following example). Below, you see the updating step for the detection parameter \mathbf{p} . Note that (1) we draw candidate values on the logit scale and (2) instead of looping through $1 - M$ individuals to update all p_i , we update all elements of the vector of \mathbf{p} in parallel, for computational efficiency.

```

14192  #### update the logit(p) parameters
14193  lp.cand<- rnorm(M,lp,1) # 1 is a tuning parameter
14194  p.cand<-plogis(lp.cand)
14195  ll<-dbinom(ytot,K,z*p, log=T)
14196  prior<-dnorm(lp,mu,sigma, log=T)
14197  llcand<-dbinom(ytot,K,z*p.cand, log=T)
14198  prior.cand<-dnorm(lp.cand,mu,sigma, log=T)
14199
14200
14201  kp<- runif(M) < exp((llcand+prior.cand)-(ll+prior))
14202  p[kp]<-p.cand[kp]
14203  lp[kp]<-lp.cand[kp]
```

The parameters μ_p and σ_p are also updated using MH steps (see the code for μ_p below). In truth, we could also sample μ_p and σ_p^2 directly with certain choices of prior distributions. For example, if $\mu_p \sim \text{Normal}(0, 1000)$ then the full conditional for μ_p is also normal (see sec. 17.3.1), etc..

```

14212  p0.cand<- rnorm(1,p0,.05)
14213  if(p0.cand>0 & p0.cand<1){
14214    mu.cand<-log(p0.cand/(1-p0.cand))
14215    ll<-sum(dnorm(lp,mu,sigma,log=TRUE))
14216    llcand<-sum(dnorm(lp,mu.cand,sigma,log=TRUE))
14217    if(runif(1)<exp(llcand-ll)) {
14218      mu<-mu.cand
14219      p0<-p0.cand
14220    }
14221  }
```

For ψ we can easily sample directly from the beta distribution:

```
14227  psi<-rbeta(1, sum(z) + 1, M-sum(z) + 1)
```

To update the z_i we have opted for a MH updater (although they could be updated directly from their full-conditional). Since z_i can only take the values of 0 or 1, we generate candidate values using `z.cand<-ifelse(z==1,0,1)`. The updating step for z_i is detailed in the next example. You can check out the full code by invoking `modelMh()` from the **R** package `scrbook`.

17.5 MCMC ALGORITHM FOR MODEL SCR0

14233 Conceptually, but also in terms of MCMC coding, it is only a small step from the non-
 14234 spatial model M_h to a fully spatial capture-recapture model. Next, we will walk you
 14235 through the steps of building your own MCMC sampler for the basic SCR model (i.e.
 14236 without any individual, site or time specific covariates) with both a Poisson and a binomial
 14237 encounter process. As usual, we will have to go through two general steps before we write
 14238 the MCMC algorithm:

- 14239 (1) Identify the model with all its components (including priors)
 14240 (2) Recognize and express the full conditional distributions for all parameters

14241 It is worthwhile to go through all of step 1 for an SCR model, but you have probably
 14242 seen enough of step 2 in our previous examples to get the essence of how to express a full
 14243 conditional distribution. Therefore, we will exemplify step 2 for some parameters and tie
 14244 these examples directly to the respective R code.

14245 **Step 1 – Identify your model** Recall the components of the basic SCR model with
 14246 a Poisson encounter process from Chapt. 9: We assume that individuals i , or rather, their
 14247 activity centers \mathbf{s}_i , are uniformly distributed across the state-space \mathcal{S} ,

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

14248 and that the number of times individual i encounters trap j , y_{ij} , is a Poisson variable
 14249 with mean λ_{ij} ,

$$y_{ij} \sim \text{Poisson}(\lambda_{ij}).$$

14250 The link between individual location, movement and trap encounter rates is made by the
 14251 assumption that λ_{ij} , is a decreasing function of the distance between \mathbf{s}_i and the location
 14252 of j , \mathbf{x}_j , say

$$d_{ij} = \|\mathbf{s}_i - \mathbf{x}_j\|,$$

14253 of the Gaussian (or half-normal) form

$$\lambda_{ij} = \lambda_0 \exp(-d_{ij}^2/2\sigma^2),$$

14254 where λ_0 is the baseline trap encounter rate at $d_{ij} = 0$ and σ is the scale parameter of the
 14255 half-normal function.

14256 As in the non-spatial example for model M_h , we estimate N , here the number of \mathbf{s}_i
 14257 in \mathcal{S} , using data augmentation (sec. 4.2). We create $M - n$ all-zero encounter histories
 14258 and estimate N by summing over the auxiliary data augmentation variables, z_i , which we
 14259 assume is a Bernoulli random variable,

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

14260 To link the two model components, we modify our trap encounter model to

$$\lambda_{ij} = \lambda_0 \times \exp(-d_{ij}^2/2\sigma^2) \times z_i.$$

14261 The model has the following structural parameters, for which we need to specify priors:

14262 ψ : the Uniform(0, 1) is required as part of the data augmentation procedure and in general
 14263 is a natural choice of an uninformative prior for a probability. It will also lead to
 14264 conjugacy as we saw in the example of model M_h , so that we can update ψ directly
 14265 from its full conditional distribution using Gibbs sampling.
 14266 s_i : since s_i is a pair of coordinates it is two-dimensional and we use a uniform prior
 14267 limited by the extent of our state-space over both dimensions.
 14268 σ : we can conceive several priors for σ but let's assume an improper prior, one that is
 14269 Uniform over $(0, \infty)$. As we already saw, this choice is convenient when updating the
 14270 parameter, because the constant prior probability cancels out of the MH acceptance
 14271 ratio.
 14272 λ_0 : analogous, we will use a Uniform($0, \infty$) improper prior for λ_0 .

14273 **Step 2 – Construct the full conditionals:** Having completed step 1, let's look at
 14274 the full conditional distributions for some of these parameters. We saw that with improper
 14275 priors, full conditionals are proportional only to the likelihood of the observations; for
 14276 example, consider σ :

$$[\sigma | s, \lambda_0, z, y] \propto \left\{ \prod_i [y_i | s_i, \lambda_0, z_i, \sigma] \right\}$$

14277 The R code to update σ is shown below. Notice that we automatically reject negative
 14278 candidate values, since σ cannot be < 0 .

```

14279 sig.cand <- rnorm(1, sigma, 0.1) #draw candidate value
14280 if(sig.cand>0){ #automatically reject sig.cand that are <0
14281   lam.cand <- lam0*exp(-(d*d)/(2*sig.cand*sig.cand))
14282   ll<- sum(dpois(y, lam*z, log=TRUE))
14283   llcand <- sum(dpois(y, lam.cand*z, log=TRUE))
14284   if(runif(1) < exp( llcand - ll) ){
14285     ll<-llcand
14286     lam<-lam.cand
14287     sigma<-sig.cand
14288   }
14289 }
```

14290 These steps are analogous for λ_0 and s_i and we will use MH steps for all of these
 14291 parameters. Similar to the random intercepts in our Poisson GLMM, we update each s_i
 14292 individually. Note that to be fully correct, the full conditional for s_i contains both the
 14293 likelihood and prior component, since we did not specify an improper, but a proper uniform
 14294 prior on s_i . However, with a uniform distribution the probability density of any value is
 14295 $1/(\text{upper limit} - \text{lower limit}) = \text{constant}$. Thus, the prior components are identical for
 14296 both the current and the candidate value so that when you calculate the ratio of posterior
 14297 densities, r , the identical prior component appears both in the numerator and denominator
 14298 and cancel each other out.

14299 We still have to update z_i . The full conditional for z_i is

$$[z_i | y_i, \sigma, \lambda_0, s_i] \propto [y_i | z_i, \sigma, \lambda_0, s_i][z_i]$$

14300 and since $z_i \sim \text{Bernoulli}(\psi)$, the term has to be taken into account when updating z_i :

```

14301      zUps <- 0 #set counter to monitor acceptance rate
14302      for(i in 1:M) {
14303          #no need to update seen individuals, since their z =1
14304          if(seen[i])
14305              next
14306          zcand <- ifelse(z[i]==0, 1, 0)
14307          llz <- sum(dpois(y[i,],lam[i,]*z[i], log=TRUE))
14308          llcand <- sum(dpois(y[i,], lam[i,]*zcand, log=TRUE))

14309          prior <- dbinom(z[i], 1, psi, log=TRUE)
14310          prior.cand <- dbinom(zcand, 1, psi, log=TRUE)
14311          if(runif(1) < exp((llcand+prior.cand)-(llz+prior))){
14312              z[i] <- zcand
14313              zUps <- zUps+1
14314          }
14315      }
14316  }
```

14317 The parameter ψ is a hyperparameter of the model, with an uninformative prior distribution of Uniform(0, 1) or Beta(1, 1), so that

$$[\psi|\mathbf{z}] \propto \text{Beta}\left(1 + \sum_i z_i, 1 + M - \sum_i z_i\right).$$

14319 These are all the building blocks you need to write the MCMC algorithm for the spatial
14320 null model with a Poisson encounter process. You can find the full **R** code by calling the
14321 function (**SCR0pois**) in the **R** package **scrbook**.

14322 17.5.1 SCR model with binomial encounter process

14323 The equivalent SCR model with a binomial encounter process is very similar. Here, each
14324 individual i can only be detected once at any given trap j during a sampling occasion k .
14325 Thus

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

14326 Where p_{ij} is some function of distance between \mathbf{s}_i and trap location \mathbf{x}_j . Here we use:

$$p_{ij} = 1 - \exp(-\lambda_{ij})$$

14327 Recall from Chapt. 3 that this is the complementary log-log (cloglog) link function,
14328 which constrains p_{ij} to fall between 0 and 1. For our MCMC algorithm that means that,
14329 instead of using a Poisson likelihood, $\text{Poisson}(y|\sigma, \lambda_0, \mathbf{s}, z)$, we use a binomial likelihood,
14330 $\text{Binomial}(y|\sigma, \lambda_0, \mathbf{s}, z; K)$, in all the conditional distributions. An exemplary updating step
14331 for λ_0 under a binomial encounter model is shown below. The full MCMC code for the
14332 binomial SCR with a cloglog link (**SCR0binom.cl**) can be found in the **R** package **scrbook**.

```

14333      lam0.cand <- rnorm(1, lam0, 0.1)
14334      #automatically reject lam0.cand that are <0
14335      if(lam0.cand >0){
```

```

14336      lam.cand <- lam0.cand*exp(-(d*d)/(2*sigma*sigma))
14337      p.cand <- 1-exp(-lam.cand)
14338      ll<- sum(dbinom(y, K, pmat *z, log=TRUE))
14339      llcand <- sum(dbinom(y, K, p.cand *z, log=TRUE))
14340      if(runif(1) < exp( llcand - ll) ){
14341          ll<-llcand
14342          pmat<-p.cand
14343          lam0<- lam0.cand
14344      }
14345  }

```

Another possibility is to model variation in the individual and site specific detection probability, p_{ij} , directly, without any transformation, such that

$$p_{ij} = p_0 \times \exp(-d_{ij}^2/(2\sigma^2))$$

and $p_0 \in [0, 1]$. This formulation is analogous to how detection probability is modeled in distance sampling under a half-normal detection function; however, in distance sampling p_0 – detection of an individual on the transect line – is assumed to be 1 (Buckland et al., 2001). Under this formulation the updater for p_0 becomes:

```

14352      p0.cand <- rnorm(1, p0, 0.1)
14353      if(p0.cand >0 & p0.cand < 1 ){
14354          #automatically rejects lam0.cand that are not {0,1}
14355          p.cand <- p0.cand*exp(-(d*d)/(2*sigma*sigma))
14356          ll<- sum(dbinom(y, K, pmat *z, log=TRUE))
14357          llcand <- sum(dbinom(y, K, p.cand *z, log=TRUE))
14358          if(runif(1) < exp( llcand - ll) ){
14359              ll<-llcand
14360              pmat<-p.cand
14361              p0<- p0.cand
14362          }
14363      }

```

17.6 LOOKING AT MODEL OUTPUT

Now that you have an MCMC algorithm to analyze spatial capture-recapture data with, let's run an actual analysis so we can look at the output. As an example, we will use the Fort Drum bear data set we first introduced in Chapt. 1 and already analyzed in several preceding chapters. You can load the Fort Drum data (`data(beardata)`), extract the trap locations (`trapmat`) and detection data (`bearArray`) and build the augmented $M \times J$ array of individual encounter histories:

```

14370  > M=700
14371  > trapmat<-beardata$trapmat
14372  #summarizes captures across occasions
14373  > bearmat<-apply(beardata$bearArray, 1:2, sum)
14374  > Xaug<-matrix(0, nrow=M, ncol=dim(trapmat)[1])
14375  > Xaug[1:dim(bearmat)[1],]<-bearmat #create augmented data set

```

14376 In addition to these data, we need to specify the outermost coordinates of the state-
 14377 space. Since bears are wide ranging animals we add a 20-km buffer to the maximum and
 14378 minimum coordinates of the trap array:

```
14379 > xl<- min(trapmat[,1])- 20
14380 > yl<- min(trapmat[,2])- 20
14381 > xu<- max(trapmat[,1])+ 20
14382 > yu<- max(trapmat[,2])+ 20
```

14383 Finally, use the MCMC code for the binomial encounter model with the cloglog link
 14384 (`SCR0binom.cl`) and run 5000 iterations. This should take approximately 25 minutes (in
 14385 real life we would of course run the algorithm a lot longer but for demonstration purposes
 14386 let's stick with a number of iterations that can be run in a manageable amount of time).

```
14387 > set.seed(13)
14388 > mod0<-SCR0binom.cl(y=Xaug, X=trapmat, M=M, xl=xl, xu=xu, yl=yl,
14389 + yu=yu, K=8, delta=c(0.1, 0.05, 2), niter=5000)
```

14390 Before, we used simple **R** commands to look at model results. However, there is a
 14391 specific **R** package to summarize MCMC simulation output and perform some convergence
 14392 diagnostics – package `coda` (Plummer et al., 2006). Download and install `coda`, then
 14393 convert your model output to an `mcmc` object

```
14394 > chain<-mcmc(mod0)
```

14395 which can be used by `coda` to produce MCMC specific output.

14396 17.6.1 Markov chain time series plots

14397 Start by looking at time series plots of your Markov chains using `plot(chain)`. This com-
 14398 mand produces a time series plot and marginal posterior density plots for each monitored
 14399 parameter, similar to what we did before using the `hist()` and `plot()` commands. Fig.
 14400 17.5 shows an example of these plots for σ and λ_0 . Time series plots will tell you several
 14401 things: First, recall from sec. 17.3.2 that the way the chains move through the parameter
 14402 space gives you an idea of whether your MH steps are well tuned. If chains were constant
 14403 over many iterations you would need to decrease the tuning parameter of the (normal)
 14404 proposal distribution. If a chain moves along some gradient to a stationary state very
 14405 slowly, you may want to increase the tuning parameter so that the parameter space is
 14406 explored more efficiently.

14407 Second, you will be able to see if your chains converged and how many initial sim-
 14408 ulations you have to discard as burn-in. In the case of the chains shown in Fig. 17.5,
 14409 we would probably consider the first 750 – 1000 iterations as burn-in, as afterwards the
 14410 chains seem to be fairly stationary.

14411 17.6.2 Posterior density plots

14412 The `plot()` command also produces posterior density plots and it is worthwhile to look
 14413 at those carefully. For parameters with priors that have bounds (e.g. uniform over some

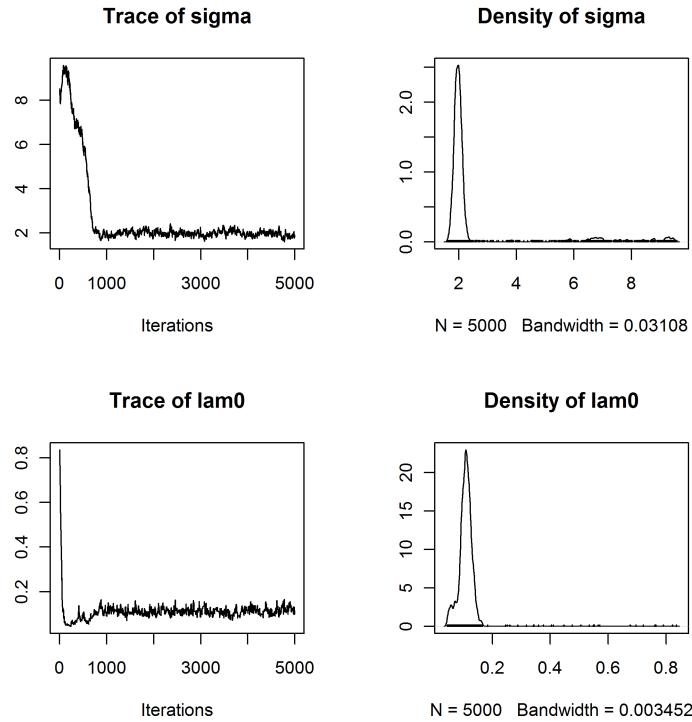


Figure 17.5. Time series and posterior density plots for σ and λ_0 for the Fort Drum black bear data.

interval), you will be able to see if your choice of the prior is truncating the posterior distribution. In the context of SCR models, this will mostly involve our choice of M , the size of the augmented data set. If the posterior of N has a lot of mass concentrated close to M (or equivalently the posterior of ψ has a lot of mass concentrated close to 1), as in the example in Fig. 17.6, we have to re-run the analysis with a larger M . A diffuse posterior plot suggests that the parameter may not be well-identified. There may not be enough information in your data to estimate model parameters and you may have to consider a simpler model. Finally, posterior density plots will show you if the posterior distribution is symmetrical or skewed – if the distribution has a heavy tail, using the mean as a point estimate of your parameter of interest may be biased and you may want to opt for the median or mode instead.

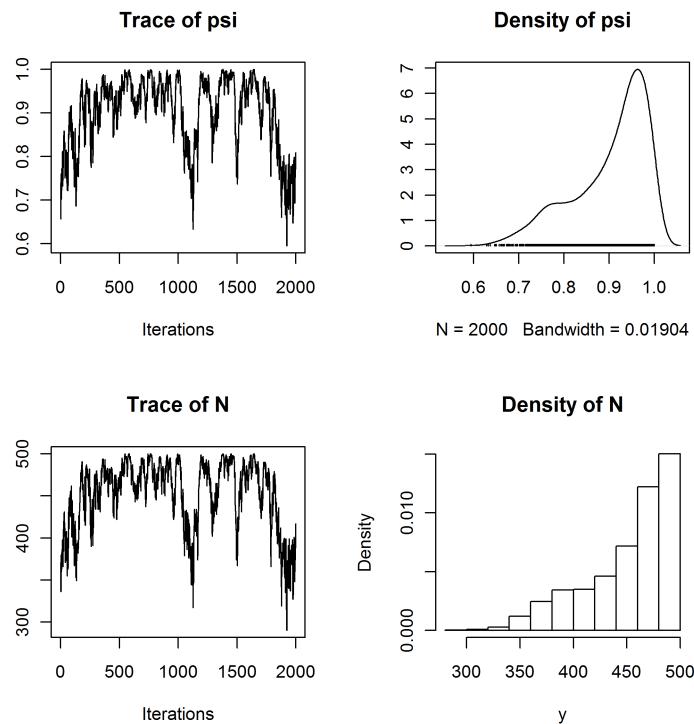


Figure 17.6. Time series and posterior density plots of ψ and N for the Fort Drum black bear data truncated by the upper limit of M (500).

14425 17.6.3 Serial autocorrelation and effective sample size

14426 Checking the degree of autocorrelation in your Markov chains and estimating the effective
 14427 sample size your chain has generated should be part of evaluating your model output.
 14428 If you use **WinBUGS** through the **R2WinBUGS** package, the **print()** command will auto-
 14429 matically return the effective sample size for all monitored parameters. In the **coda**
 14430 package there are several functions you can use to do so. The function **effectiveSize()**
 14431 will directly give you an estimate of the effective sample size for the parameters:

```
14432 > effectiveSize(window(chain, start=1001))
14433   sigma      lam0      psi      N
14434 93.89807 163.72311 51.96443 46.45394
```

14435 Alternatively, you can use the **autocorr.diag()** function, which will show you the
 14436 degree of autocorrelation for different lag values (which you can specify within the function

14437 call, we use the defaults below):

```
14438 > autocorr.diag(window(chain, start=1001))
14439      sigma      lam0      psi      N
14440 Lag 0  1.0000000 1.0000000 1.0000000 1.0000000
14441 Lag 1  0.9316928 0.91464875 0.9745833 0.9663320
14442 Lag 5  0.7603332 0.67445407 0.8525272 0.8500215
14443 Lag 10 0.6065374 0.48724122 0.7514657 0.7530124
14444 Lag 50 0.1122331 0.06564406 0.3811939 0.3823236
```

14445 In the present case we see that autocorrelation is especially high for the parameter ψ and
 14446 effective sample size for this parameter is only 52! This means we would have to run the
 14447 model for much longer to obtain a reasonable effective sample size. Unfortunately, with
 14448 many SCR data sets we observe high degrees of serial autocorrelation. For now, let's
 14449 continue using this small number of samples to look at the output.

14450 17.6.4 Summary results

14451 Now that we checked that our chains apparently have converged and pretending that
 14452 we have generated enough samples from the posterior distribution, we can look at the
 14453 actual parameter estimates. The `summary()` function will return two sets of results: the
 14454 mean parameter estimates, with their standard deviation, the naïve standard error – i.e.
 14455 your regular standard error calculated for T (= number of iterations) samples without
 14456 accounting for serial autocorrelation – and the Time-series SE (in **WinBUGS** and earlier
 14457 in this book referred to as MC error), which accounts for autocorrelation. Remember our
 14458 rule of thumb that this error decreases with increasing chain length and should be 1% or
 14459 less of the parameter estimate. In **WinBUGS** the MC error is only given in the log output
 14460 within **BUGS** itself. You should adjust the `summary()` call by removing the burn-in from
 14461 calculating parameter summary statistics. To do so, use the `window()` command, which
 14462 lets you specify at which iteration to start 'counting'. In contrast to **WinBUGS**, which
 14463 requires you to set the burn-in length before you run the model, this command gives us
 14464 full flexibility to make decisions about the burn-in after we have seen the trajectories of
 14465 our Markov chains. For our example, `summary(window(chain, start=1001))` returns the
 14466 following output:

```
14467 Iterations = 1001:5000
14468 Thinning interval = 1
14469 Number of chains = 1
14470 Sample size per chain = 4000
14471
14472 1. Empirical mean and standard deviation for each variable,
14473 plus standard error of the mean:
14474
14475          Mean        SD  Naive SE Time-series SE
14476 sigma    1.9697  0.12534 0.0019818       0.012792
14477 lam0     0.1124  0.01521 0.0002405       0.001311
14478 psi      0.7295  0.11794 0.0018648       0.015278
```

```

14479 N      510.9190 81.99868 1.2965130      10.580567
14480
14481 2. Quantiles for each variable:
14482
14483      2.5%    25%    50%    75%   97.5%
14484 sigma   1.7288   1.8831   1.9666   2.0517   2.2240
14485 lam0    0.0863   0.1008   0.1112   0.1217   0.1449
14486 psi     0.5100   0.6423   0.7261   0.8170   0.9549
14487 N      359.0000 451.0000 508.0000 572.0000 668.0000

```

14488 Looking at the MC errors (column labeled `Time-series SE`), we see that in spite of the
14489 high autocorrelation, the MC error for σ is below the 1% threshold, whereas for all other
14490 parameters, MC errors are still above, another indication that for a thorough analysis we
14491 should run a longer chain.

14492 Our algorithm gives us a posterior distribution of N , but we are usually interested
14493 in the density, D . Density itself is not a parameter of our model, but we can derive a
14494 posterior distribution for D by dividing each value of N (N at each iteration) by the area
14495 of the state-space (here 3032.719 km^2) and we can use summary statistics of the resulting
14496 distribution to characterize D :

```

14497 > summary(window(chain[,4]/ 3032.719, start=1001))
14498
14499 Iterations = 1001:5000
14500 Thinning interval = 1
14501 Number of chains = 1
14502 Sample size per chain = 4000
14503
14504 1. Empirical mean and standard deviation for each variable,
14505 plus standard error of the mean:
14506
14507      Mean           SD        Naive SE Time-series SE
14508      0.1684690    0.0270380    0.0004275    0.0034888
14509
14510 2. Quantiles for each variable:
14511
14512      2.5%    25%    50%    75%   97.5%
14513 0.1184 0.1487 0.1675 0.1886 0.2203

```

14514 We see that our mean density of $0.17/\text{km}^2$ is very similar to the estimate of $0.18/\text{km}^2$
14515 obtained under the non-spatial model M_0 in Chapt. 4.

14516 17.6.5 Other useful commands

14517 While inspecting the time series plot gives you a first idea of how well you tuned your
14518 MH algorithm, use `rejectionRate()` to obtain the rejection rates (1 – acceptance rates)
14519 of the parameters that are written to your output:

```

14520 > rejectionRate(chain)
14521   sigma      lam0      psi      N
14522 0.42988598 0.78775755 0.00000000 0.03160632

```

14523 Recall (sec. 17.3.2) that rejection rates should lie between 0.2 and 0.8, so our tuning
 14524 seems to have been appropriate here. Draws of the parameter ψ are never rejected since
 14525 we update it with Gibbs sampling, where all candidate values are kept. And since N is
 14526 the sum of all z_i , all it takes for N to change from one iteration to the next are small
 14527 changes in the z-vector, so the rejection rate of N is always low. If you have run several
 14528 parallel chains, you can combine them into a single mcmc object using the `mcmc.list()`
 14529 command on the individual chains (note that each chain has to be converted to an mcmc
 14530 object before combining them with `mcmc.list()`). You can then easily obtain the Gelman-
 14531 Rubin diagnostic (Gelman et al., 2004), in **WinBUGS** called Rhat, using `gelman.diag()`,
 14532 which will indicate if all chains have converged to the same stationary distribution. For
 14533 details on these and other functions, see the `coda` manual, which can be found (together
 14534 with the package) on the CRAN mirror.

17.7 MANIPULATING THE STATE-SPACE

14535 So far, we have constrained the location of the activity centers to fall within the outermost
 14536 coordinates of our rectangular state-space by posing upper and lower bounds for x and y .
 14537 But what if S has an irregular shape – maybe there is a large water body we would like
 14538 to remove from S , because we know our terrestrial study species does not occur there. Or
 14539 the study takes place in a clearly defined area such as an island.

14540 As mentioned before, this situation is difficult to handle in **BUGS** engines. In some
 14541 simple cases we can adjust the state-space by setting one of the coordinates of s_i to be
 14542 some function of the other and reject candidate s_i that do not fall within this modified
 14543 state-space. In this manner, we can cut off corners of the rectangle to approximate the
 14544 actual state-space³. To visualize this approach, plot the following rectangle, representing
 14545 your state-space polygon, and line, representing, for example, the approximation of a shore
 14546 line:

```

14547 > xlim<-c(-5,5)
14548 > ylim<-c(-7,7)
14549 > plot(xlim, ylim, type='n')
14550 > abline(a=4, b=0.4)

```

14551 The Y coordinates limiting your state-space to the habitat that is suitable to the species
 14552 you study can now be expressed as a linear function of the X coordinates, in this case,
 14553 $Y = 4 + 0.4 \times X$. To include this new limit in a **BUGS** model, we need to change the
 14554 following:

```

14555 #draw SX and SY as before
14556 SX[i]~dunif(xlim[1],xlim[2])
14557 SY[i]~dunif(ylim[1],ylim[2])

```

³This idea was pitched to us by Mike Meredith, Biodiversity Conservation Society Sarawak/WCS Malaysia

```

14558 #calculate upper limit for Y given X
14559 ymax[i]<-4+0.4*SX[i]
14560 # use step function to see if location [SX, SY]
14561 # is below the Y limit (Pin = 1) or not (Pin = 0)
14562 Pin[i] <- step(ymax[i] - SY[i])
14563 In[i] ~ dbern(Pin[i])

```

14564 The object `In` is a vector of M 1's, passed as data to the model. If $\text{Pin} = 0$, the likelihood
 14565 will be 0 and the candidate $[\text{SX}, \text{SY}]$ pair will be rejected. If $\text{Pin} = 1$, this bit of the
 14566 likelihood is equal to 1, and whether or not the the candidate pair of coordinates is
 14567 accepted depends only on capture history of i . This approach can be very useful in some
 14568 situations but is clearly restricted by the functional form of the relationship between SX
 14569 and SY that it requires.

14570 In **R**, we are much more flexible, as we can use the actual state-space polygon to
 14571 constrain s_i . To illustrate that, let's look at a camera trapping study of raccoons (*Procyon*
 14572 *lotor*) conducted on South Core Banks, a barrier island within Cape Lookout National
 14573 Seashore, North Carolina (details of the study can be found in Sollmann et al. (2013) and
 14574 in Chapt. 19 where we present the analysis of this data set with spatial mark-resight
 14575 models). Since camera-traps were spread across the entire length of the island, we set
 14576 the state-space to be delineated by the shore line of the island (Fig. 17.7), which clearly
 14577 cannot easily be approximated as a rectangle. Instead, within **R** we can use an actual
 14578 shapefile of the island.

14579 In other circumstances you may still want to create the state-space as before, by adding
 14580 some buffer to your trapping grid, but you may find that the resulting rectangle includes
 14581 water bodies, paved parking lots or any other kind of habitat you know is never used by the
 14582 species you study. In order to precisely describe the state-space, these features need to be
 14583 removed. You can create a precise state-space polygon in **ArcGIS** and read it into **R**, or
 14584 create the polygon directly within **R**, by intersecting two shapefiles – one of the rectangle
 14585 defining the outer limits of your state-space state and one of the landscape feature you
 14586 want to remove. While you will most likely have to obtain the shapefile describing the
 14587 landscape of and around your trapping grid (coastlines, water bodies etc.) from some
 14588 external source, the polygon shapefile buffering your outermost trapping grid coordinates
 14589 can easily be written in **R**.

14590 If `xmin`, `xmax`, `ymin` and `ymax` mark the most extreme x and y coordinates of your
 14591 trapping grid and b is the distance you want to buffer with, load the package **shapefiles**
 14592 (Stabler, 2006) and issue the following **R** commands:

```

14593 > xl= xmin-b
14594 > xu= xmax+b
14595 > yl= ymin-b
14596 > yu= ymax+b
14597
14598         #create data frame with coordinate pairs
14599 > dd <- data.frame(Id=c(1,1,1,1,1),X=c(xl,xu,xu,xl,xl),
14600 +   Y=c(yl,yl,yu,yu,yl))
14601 > ddTable <- data.frame(Id=c(1),Name=c("Item1"))
14602           #convert to shapefile, type polygon

```

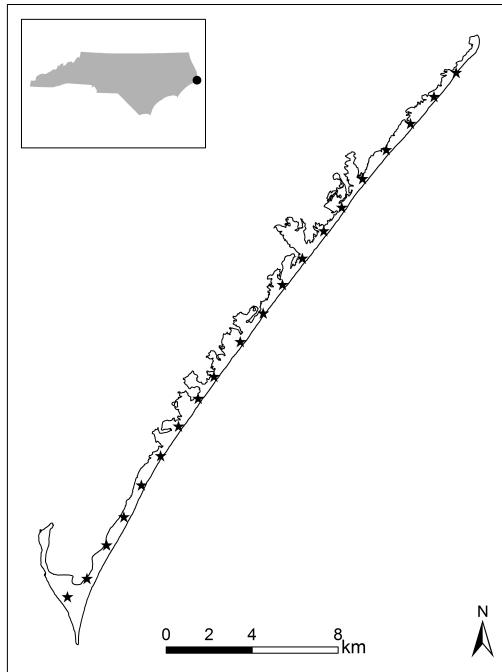


Figure 17.7. Camera traps (stars) set up on South Core Banks, a barrier island within Cape Lookout National Seashore, North Carolina (inset map) to estimate the raccoon population (see Chapt. 19 for details).

```

14603 > ddShapefile <- convert.to.shapefile(dd, ddTable, "Id", 5)
14604     # name and save to location of choice
14605 > write.shapefile(ddShapefile, 'c:/Test', arcgis=T)

```

14606 You can read shapefiles into **R** loading the package **maptools** (Lewin-Koh et al., 2011)
 14607 and using the function **readShapeSpatial()**. Make sure you read in shapefiles in UTM
 14608 format, so that units of the trap array, the movement parameter σ and the state-space
 14609 are all identical. Intersection of polygons can be done in **R** also, using the package **rgeos**
 14610 (Bivand and Rundel, 2011) and the function **gIntersect()**. The area of your (single)
 14611 polygon can be extracted directly from the state-space object **SSp**:

```

14612 > area <- SSp@polygons[[1]]@Polygons[[1]]@area /1000000

```

14613 Note that dividing by 1000000 will return the area in km^2 if your coordinates describing
 14614 the polygon are in UTM. If your state-space consists of several disjunct polygons, you will
 14615 have to sum the areas of all polygons to obtain the size of the state-space. To include
 14616 this polygon into our MCMC sampler we need one last spatial **R** package, **sp** (Pebesma

14617 and Bivand, 2011), which has a function, `over()`, which allows us to check if a pair of
 14618 coordinates falls within a polygon or not.⁴ All we have to do is embed this new check into
 14619 the updating steps for the s_i :

```
14620 #draw candidate value
14621 Scand <- as.matrix(cbind(rnorm(M, S[,1], 2), rnorm(M, S[,2], 2)))
14622     #convert to spatial points on UTM (m) scale
14623 Scoord<-SpatialPoints(Scand*1000)
14624     # check if scand is within the polygon
14625 SinPoly<-over(Scoord,SSp)
14626
14627 for(i in 1:M) {
14628     #if scand falls within polygon, continue update
14629     if(is.na(SinPoly[i])==FALSE) {
14630         ... [rest of the updating step remains the same]
```

14631 Note that it is much more time-efficient to draw all M candidate values for s and check
 14632 once if they fall within the state-space, rather than running the `over()` command for
 14633 every individual pair of coordinates. To make sure that our initial values for s also fall
 14634 within the polygon of \mathcal{S} , we use the function `runifpoint()` from the package `spatstat`
 14635 (Baddeley and Turner, 2005), which generates random uniform points within a specified
 14636 polygon. You'll find this modified MCMC algorithm (`SCR0poisSSp`) in the **R** package
 14637 `scrbook`.

14638 Finally, observe that we are converting candidate coordinates of \mathcal{S} back to meters to
 14639 match the UTM polygon. In all previous examples, for both the trap locations and the
 14640 activity centers we have used UTM coordinates divided by 1000 to estimate σ on a km
 14641 scale. This is adequate for wide ranging species like bears. In other cases you may center
 14642 all coordinates on 0. No matter what kind of transformation you use on your coordinates,
 14643 make sure to always convert candidate values for \mathcal{S} back to the original scale (UTM)
 14644 before running the `over()` command.

17.8 INCREASING COMPUTATIONAL SPEED

14645 Using custom written MCMC algorithms in **R** is not only more flexible but can also be
 14646 faster than using programs such as **JAGS** and especially **WinBUGS**. Also, **R** tends to
 14647 use much less memory than **JAGS**, which can be crucial if you are running a large model
 14648 but only have limited memory available. **WinBUGS** is limited in the amount of memory
 14649 it can access and thus will likely not max out your memory, but as a trade-off, it will take a
 14650 long time to run such models. In this chapter we have provided you with the guidelines to
 14651 write your own MCMC sampler. But beyond the material that we have covered there are
 14652 a number of ways you can make your sampler more efficient, through parallel computing
 14653 or by accessing an alternative computer language such as **C++**. Exploring these options
 14654 exhaustively is beyond the scope of this book; instead, in this section we will give you
 14655 some pointers to get started with these more advanced computational issues.

⁴Remember from Chapt. (6.4.2) that the `over` function takes as its second argument (among others) an object of the class “`SpatialPolygons`” or “`SpatialPolygonsDataFrame`”. The former produces a vector while the latter produces a data frame (e.g., in the example above), which is important for how you index the output.

14656 17.8.1 Parallel computing

14657 If you are using a computer with several cores, you can make use of parallel computing to
 14658 speed up overall computation. In parallel computing we execute commands simultaneously
 14659 on different cores of the computer, instead of running them serially on one single core.
 14660 For example, imagine you have 4 cores available and you want to implement a for-loop in
 14661 **R**; instead of going through the loop iteration by iteration, you can prompt **R** to execute
 14662 iterations 1 to 4 simultaneously on the 4 different cores. The core that finishes first will
 14663 then continue with iteration 5, and so on. There are several packages in **R** that allow you
 14664 to induce parallel computing, such as **snow** (Tierney et al., 2011) and **snowfall** (Knaus,
 14665 2010), and the more current versions of **R** (from 2.14.0 upwards) come with a pre-installed
 14666 set of functions grouped under the name **parallel**.

14667 The MCMC algorithms developed here and in other parts of this book come with plenty
 14668 of opportunities to parallelize computation. In various instances within the algorithm, we
 14669 have for-loops across our augmented data set of size M , or we may have for-loops across
 14670 sampling occasions. We also have for-loops across iterations of the algorithm, but since
 14671 one iteration of the Markov chain depends on the preceding iteration these should always
 14672 be run serially, not in parallel. There is another dimension we can think of, and that is
 14673 running multiple chains of an algorithm to assess convergence. This is a comparatively
 14674 easy implementation of parallel computing and thus provides a good starting point to
 14675 understand how it works in **R**.

14676 Let's go back to the Ft. Drum black bear data we analyzed above with the cloglog
 14677 version of the binomial SCR model (sec. 17.6) and run 3 parallel chains using **snowfall**.
 14678 All we need to do is wrap our function **SCR0binom.cl** within another function that can
 14679 then be executed in parallel, returning a list with one output matrix for each chain (install
 14680 **snowfall** before executing the code below; we assume the data objects are already in your
 14681 workspace from the previous analysis):

```
14682 > library(snowfall)
14683 ## create wrapper function
14684 > wrapper<-function(a){
14685 + out<-SCR0binom.cl(y=Xaug, X=trapmat, M=M, xl=xl, xu=xu, yl=yl,
14686 + yu=yu, K=8, delta=c(0.1, 0.05, 2), niter=5000)
14687 + return(out)
14688 + }
```

14689 After creating the wrapper function we need to initialize the cluster of cores, defining
 14690 that we want computation to be implemented in parallel and how many cores we want it
 14691 to be run on. Here, we assume we have (at least) 3 cores, but if your computer only has 2,
 14692 make sure to adjust the code accordingly (i.e., set **cpus=2**). In that case, 2 of the 3 chains
 14693 will be run in parallel and whichever core finishes first will then pick up the third chain.
 14694 Further, we have to export all **R** libraries and data to all the cores, and set up a random
 14695 number generator, so that we do not get identical results from the different cores:

```
14696 > sfInit( parallel=TRUE, cpus=3 ) #initialize cluster
14697 > sfLibrary(scrbook) #export library scrbook
14698 > sfExportAll() #export all data in current workspace
14699 > sfClusterSetupRNG() #set up random number generator
```

```

14700 > outL=sfLapply(1:3,wrapper) # execute 'wrapper' 3 times

14701 The object outL is a list of length 3, with one out matrix from the function SCRObinom.cl
14702 for each chain. After computation is complete, terminate the cluster using the command
14703 sfStop(). Note that the intermediate output of current values and acceptance rates in the
14704 R console is suppressed when using parallel computing. We can now look at the output
14705 as described previously using the package coda, by first defining outL to be a list of mcmc
14706 objects.

14707 > library(coda)
14708 #turn output into MCMC list
14709 > res<-mcmc.list(as.mcmc(outL[[1]]),as.mcmc(outL[[2]]),as.mcmc(outL[[3]]))
14710 > summary(window(res, start=1001)) #remove first 1000 iterations as burn-in
14711
14712 [... some output removed ...]
14713
14714      Mean       SD  Naive SE Time-series SE
14715 sigma   1.9723  0.13093 0.0011952      0.0087055
14716 lam0    0.1115  0.01535 0.0001401      0.0009003
14717 psi     0.7130  0.10787 0.0009847      0.0077910
14718 N      499.6166 74.74934 0.6823650      5.4232653
14719
14720 2. Quantiles for each variable:
14721
14722      2.5%     25%     50%     75%   97.5%
14723 sigma   1.74339  1.8811  1.9637  2.0530  2.2618
14724 lam0    0.08443  0.1007  0.1105  0.1211  0.1438
14725 psi     0.52046  0.6350  0.7093  0.7814  0.9627
14726 N      366.00000 446.00000 497.00000 547.00000 674.0000

14727 Now that we have parallel chains we can also use the function gelman.diag to evaluate
14728 if chains have converged:
14729 > gelman.diag(window(res, start=1001)) #assess chain convergence
14730
14731 Potential scale reduction factors:
14732
14733      Point est. Upper C.I.
14734 sigma      1.01      1.04
14735 lam0      1.01      1.02
14736 psi       1.07      1.21
14737 N        1.07      1.21
14738
14739 Multivariate psrf
14740
14741 1.05

14742 We can see that estimates are similar to what we observed when running a single
14743 chain (see sec. 17.6) and that all 3 chains appear to have converged, based on their point

```

estimates of the \hat{R} statistic, but, as already noted before, for a real analysis we might want to run this model for quite a bit longer, to bring down the upper confidence interval limits on \hat{R} for ψ and N . If you have 3 cores then running these 3 parallel chains should not have taken longer than running a single chain. Yet if you look at the effective sample size now using `effectiveSize`, you can see that it has roughly tripled, as we would expect:

```
14749 > effectiveSize(window(res, start=1001))
14750
14751   sigma      lam0      psi       N
14752 272.6935 411.8384 167.4192 168.3355
```

17.8.2 Using C++

Parallel computing is a great tool to speed up computations, but its usefulness is limited by how many cores you have available. Even with a decent number of cores, large models may still take a long time to run. A major reason for this is that for-loops in **R** are time consuming, whereas they are handled much more time efficiently in other computer languages such as **C++**. As we saw above, MCMC algorithms consist of for-loops within for-loops, so that it stands to reason that implementing them in a language like **C++** should make those algorithms run much faster. Being avid **R** users, we cannot claim to be fluent in **C++** or to be aware of all the opportunities this language brings for faster computing. It is also beyond the scope of this book to go into the nuts and bolts of how **C++** works or provide a tutorial, and we refer you to the vast amounts of online and print material designed to give the interested user an introduction to **C++**. Just google “introduction **C++**” and you are sure to come across sites such as <http://www.cplusplus.com> that provide step by step instructions to get you started. Here, we only want to point out one approach to linking **R** with **C++**: the packages `inline` (Sklyar et al., 2010) and `RcppArmadillo` (Fran ois et al., 2011). These two packages provide a very convenient interface between the two languages, but there are other ways of calling **C++** functions from within **R**, such as the `.Call` command. If you are interested, we suggest you refer to the package manuals and vignettes, as well as the online document “Writing R extensions” (at <http://cran.r-project.org/doc/manuals/R-exts.html>) for a much more thorough treatment of this topic.

In order to use **C++** you need a compiler such as `g++` that (together with other compilers, for example for **C** and **FORTRAN**) comes with **Rtools**, which you can easily download from the web (at <http://cran.r-project.org/bin/windows/Rtools/>). All of these compilers are part of the GNU compiler collection (<http://gcc.gnu.org/>). Make sure the version of **Rtools** matches your version of **R** or you may run into compilation errors later on. To give you a taste of **C++** we will show you how to write a function that calculates the squared distances of individual activity centers to all traps, as is implemented in the `scrbook` package in the function `e2dist` (to be exact, `e2dist` calculates the distance, not the squared distance), and compare performance between **R** and **C++**. We will refer to these functions as “distance functions”. First, let us set up dummy data – a matrix holding the coordinates of the trap array, outer limits of the state-space and uniformly distributed activity centers for $M = 700$ individuals:

```
14786 > gx<-seq(1,10,1)
```

```

14787 > gy<-seq(1,10,1)
14788 > X<-as.matrix(expand.grid(gx, gy))
14789 > M<-700
14790 > J<-dim(X)[1]
14791 > b<-3
14792 > xl<-min(gx)-b
14793 > xu<-max(gx)+b
14794 > yl<-min(gy)-b
14795 > yu<-max(gy)+b
14796 > S<-cbind(runif(M, xl, xu), runif(M, yl,yu))

```

14797 Next, we can write a “pedestrian” version of `e2dist` and check how long it takes to
14798 calculate the squared distance matrix:

```

14799 > Dfun<-function(M, J, S, X){
14800 + D2<-matrix(0, nrow=M, ncol=J)
14801 + for (i in 1:M){
14802 + for(j in 1:J){
14803 + D2[i,j]<-(S[i,1]-X[j,1])^2 + (S[i,2]-X[j,2])^2
14804 + }
14805 + return(D2)
14806 +
14807
14808 > system.time(
14809 + (D2R<-Dfun(M, J, S, X))
14810 +
14811
14812 user   system elapsed
14813     0.81    0.01    0.82

```

14814 The code to implement the same function in **C++** using the `inline` and `RcppArmadillo`
14815 packages is shown in panel 17.3. These packages allow you to use a range of data formats
14816 such as lists and matrices, and they take care of compiling the code in **C++** and loading
14817 the resulting function into **R**. This is also referred to compiling **C++** code “on the fly”.
14818 You will see that the way the code is set up is reasonably similar to **R**. One difference
14819 that is worthy to point out is that in **C++** indices for vectors range from 0 to $n - 1$, NOT
14820 from 1 to n , as in **R**. Note that with `inline` we only need to write the core of the code and
14821 define the type of the variables we want to pass to the function, while the `cxxfunction`
14822 call takes care of the rest. Once your function is compiled and loaded you should check
14823 out the full **C++** code by calling `DfunArma@code`.

14824 Executing this code shows that it is faster than the **R** version of the distance function
14825 or `e2dist`; in fact it is too fast for the time resolution of the `system.time()` function to
14826 even give us a time estimate:

```

14827 > system.time(
14828 + (out<-DfunArma(M,J,S,X)))
14829
14830 user   system elapsed
14831     0       0       0

```

14832 While speed differences of less than 1 second may seem negligible, remember that
14833 each command has to be executed at each iteration of the Markov chain. Especially with
14834 time-consuming models such as those for open populations (Chapt. 16) or multi-session
14835 models (Chapt. 14) we believe that **C++** holds large potential to make implementation
14836 of such models more feasible.

17.9 SUMMARY AND OUTLOOK

14837 In a nutshell, programs like **JAGS** and **WinBUGS** do all the MCMC-related things that
14838 we went through in this chapter (and quite a bit more). Looking through your model,
14839 they determine which parameters they can use standard Gibbs sampling for (i.e. for
14840 conjugate full conditional distributions). Then, they determine whether to use adaptive
14841 rejection sampling, slice sampling or – in the ‘worst’ case – Metropolis-Hastings sampling
14842 for the other full conditionals (how the sampler is chosen differs among softwares). For
14843 MH sampling, they will automatically tune the updater so that it works efficiently.

14844 Although these programs are flexible and extremely useful to perform MCMC simulations,
14845 it sometimes is more efficient to develop your own MCMC algorithm. Building an
14846 MCMC code follows three basic steps: Identify your model including priors and express
14847 full conditional distributions for each model parameter. If full conditionals are parametric
14848 distributions, use Gibbs sampling to draw candidate parameter values from those dis-
14849 tributions; otherwise use Metropolis-Hastings sampling to draw candidate values from
14850 a proposal distribution and accept or reject them based on their posterior probability
14851 densities.

14852 These custom-made MCMC algorithms give you more modeling flexibility than ex-
14853 isting software packages, especially when it comes to handling the state-space: In **Win-**
14854 **BUGS** and **JAGS** we define a continuous rectangular state-space using the corner coor-
14855 dinates to constrain the uniform priors on the activity centers **s**. But what if a continuous
14856 rectangle is an inadequate description of the state-space? In this chapter we saw that in
14857 **R** it only takes a few lines of code to use any arbitrary polygon shapefile as the state-
14858 space, which is especially useful when you are dealing with coastlines or large bodies of
14859 water that need removing from the state-space. Another example is the SCR **R** package
14860 **SPACECAP** (Gopalamswamy et al., 2012a) that was developed because implementation of an
14861 SCR model with a discrete state-space was inefficient in **WinBUGS**.

14862 Another situations in which using a **BUGS** engine becomes increasingly complicated
14863 or inefficient is when using point processes other than the homogeneous binomial point
14864 process (“uniformity of density”) which underlies the basic SCR model (see sec. 5.10
14865 in Chapt. 5). In Chapt. 11 you already saw an example of an inhomogeneous point
14866 process model and we briefly introduce a different point processes, implemented using a
14867 custom-made MCMC algorithm, in Chapt. 20. Finally, Chapt. 19 deals with partially
14868 marked populations using hand-made MCMC algorithms to handle the (partially) latent
14869 individual encounter histories. While some of these models can be written in the**BUGS**
14870 language, they are painstakingly slow; others (for example the classes of models considered
14871 in Chapt. 12) cannot be implemented in **WinBUGS/JAGS** at all and we have to either
14872 use likelihood based inference or develop our own MCMC algorithms. In conclusion, while
14873 you can certainly get by using **BUGS/JAGS** for standard SCR models, knowing how to
14874 write your own MCMC sampler gives you more flexibility to tailor these models to your

14875 specific needs.

```
### calculate squared distances using RcppArmadillo
library(inline)
library(RcppArmadillo)

#write core of function code
code<-'
/*define input, assign correct class (matrix, vector etc)*/
arma::mat Sn=Rcpp::as<arma::mat>(S);
arma::mat Xn=Rcpp::as<arma::mat>(X);
int Ntot=Rcpp::as<int>(M);
int ntraps=Rcpp::as<int>(J);
/*create matrix to hold squared distances*/
arma::mat D2(Ntot, ntraps);

/*loop over M and J to calculate distances*/
for (int i=0; i<Ntot; i++){
  for(int j=0; j<ntraps; j++){
    D2(i,j)= pow(Sn(i,0)-Xn(j,0), 2) + pow(Sn(i,1)-Xn(j,1), 2);
  }
}
/*return D2 in R format*/
return Rcpp::wrap(D2);
'

# compile and load
DfunArma<-cxxfunction(signature(M="integer", J="integer", S="numeric",
X="numeric"), plugin="RcppArmadillo", body=code)
```

Panel 17.3: Code to compute squared distance between individual activity centers and traps in **C++** from within **R** using **inline** and **RcppArmadillo**

14876
14877

18

14878

UNMARKED POPULATIONS

14879 Traditional capture-recapture models share the fundamental assumption that each individual in a population can be uniquely identified when captured. Often, this can be
14880 accomplished by marking individuals with color bands, ear tags, or some other artificial
14881 mark that subsequently can be read in the field. For other species, such as tigers (*Panthera*
14882 *tigris*) or marbled salamanders (*Ambystoma opacum*), individuals can be identified using
14883 only their natural markings. However, many species do not possess adequate natural mark-
14884 ings and are difficult to capture, making it impractical to use standard capture-recapture
14885 techniques.

14886 Estimating density when individuals are unmarked can be accomplished using a variety
14887 of alternatives to capture-recapture, such as distance sampling (Buckland et al., 2001)
14888 and *N*-mixture models (Royle, 2004b). These methods can be very effective when their
14889 assumptions are met, but when it is not possible to obtain accurate distance data, or
14890 when movement complicates the use of fixed-area plots, these methods may yield biased
14891 estimates of density (Chandler et al., 2011). Furthermore, some species are so rare and
14892 cryptic that it is nearly impossible to collect enough data using traditional survey methods.
14893

14894 In this chapter, we investigate spatially explicit alternatives for estimating density
14895 of unmarked populations, and we highlight the work of Chandler and Royle (In press)
14896 who demonstrated that the “individual recognition” assumption of traditional capture-
14897 recapture models is not a requirement of spatial capture-recapture models. They showed
14898 that, under certain conditions, spatially correlated count data are sufficient for making
14899 inference about animal distribution and density even when no individuals are marked. The
14900 Chandler and Royle (In press) “spatial count model” (hereafter the SC model) requires
14901 neither distance data nor fixed area plots. Instead, the observed data are trap- and
14902 occasion-specific counts, which are modeled as a reduced-information summary of the
14903 *latent* encounter histories. Because the model is formulated in terms of the data we wish
14904 we had, i.e. the typical encounter history data observed in standard capture-recapture
14905 studies of marked animals, the SC model is just a SCR model with a single extension
14906 to account for the fact that the encounter history data are unobserved. However, this
14907 results in a drastically different model than the models typically used for count data in
14908 ecology because the SC model is parameterized in terms of individuals, and specifically,

14909 their locations relative to the sampling device.

14910 The ability to fit SCR models to data from unmarked populations has important
14911 implications. For one, it means that SCR models can be applied to data collected using
14912 methods like points counts in which observers record simple counts of animals at an array
14913 of survey locations. The model can also be fitted to camera trapping data collected on
14914 unmarked animals, representing one of the first formal method for estimating density
14915 from such data (but see ?). So, is the SC model a free lunch? At face value, it sounds
14916 as though it allows for estimation of all the quantities of interest in standard capture-
14917 recapture studies, but with very little data. But of course the answer is no – lunch is
14918 still not free because with this model come new assumptions, and as was demonstrated
14919 by Chandler and Royle (In press), even with “perfect” data, parameter estimates will
14920 typically not be very precise. This should not be surprising given that we are asking so
14921 much from simple count data.

14922 The real value of the SC model is two-fold. First, it demonstrates an important
14923 theoretical result, namely that spatial correlation in count data carries information about
14924 density and distribution; a result that stands in stark contrast to a prevailing view of
14925 spatial correlation as a nuisance to be avoided or modeled out of unsightly residual plots.
14926 The second reason why this model is important is that it provides the basis for numerous
14927 model extensions that *can* yield precise density estimates. We will discuss some of these
14928 possibilities in this chapter, but perhaps the most useful extension – accommodating data
14929 from both marked and unmarked individuals – is treated separately in the next chapter.
14930 Here, we focus on situations in which all individuals are unmarked, and we begin by
14931 presenting the most basic formulation of the model. Then we proceed, by way of a few
14932 examples, to consider extensions of the model in which ancillary information can be used
14933 to increase precision.

18.1 EXISTING MODELS FOR INFERENCE ABOUT DENSITY IN UNMARKED POPULATIONS

14934 When capture-recapture methods are not a viable option, ecologists often collect simple
14935 count data or even binary detection/non-detection data. These data are often treated as
14936 an index of abundance or occurrence and are analyzed using generalized linear models
14937 such as Poisson regression or logistic regression, perhaps with random effects (Zuur et al.,
14938 2009). However, index methods cannot be used to make unbiased inferences about abun-
14939 dience or occurrence unless strong assumptions about constant detection probability are
14940 valid (Williams et al., 2002; ?). In particular, index methods can be highly misleading
14941 when covariates affect both the ecological process of interest and the observation pro-
14942 cess. A classic example is given by Bibby and Buckland (1987) who found that songbird
14943 detection probability was negatively related to vegetation height, whereas density was pos-
14944 itively associated with vegetation height in restocked conifer plantations. This intuitive
14945 phenomenon has been demonstrated repeatedly (Kéry, 2008; Sillett et al., 2012) and has
14946 led to the development of a vast number of models to estimate population size and oc-
14947 currence probability when individuals are unmarked and detected imperfectly (Buckland
14948 et al., 2001; Williams et al., 2002; MacKenzie et al., 2006; Royle and Dorazio, 2008). A
14949 review of these models is beyond the scope of this chapter, but we mention a few deficien-
14950 cies of existing methods that warrant the exploration of alternatives for robust inference

EXISTING MODELS FOR INFERENCE ABOUT DENSITY IN UNMARKED POPULATIONS

when standard capture-recapture methods do not apply.

Distance sampling (Buckland et al., 2001; Buckland, 2004), which we briefly introduced in Chapter 4, is perhaps the most widely used method for estimating population density when individuals are unmarked and detection probability is less than one. This class of methods is known to work impeccably when estimating the number of stakes in a field or the number of duck nests in a wetland. Distance sampling can also work very well in more interesting situations, and it is an extremely powerful method when the assumptions can be met. However, the assumptions that distance data can be recorded without error and that animals are distributed randomly with respect to the transect can be easily violated by common processes such as animal movement and measurement error. Although numerous methods have been proposed to relax some of these assumptions (Royle et al., 2004; Borchers et al., 1998; Johnson, 2010; ?; Chandler et al., 2011), a more important issue is that distance sampling is simply not practical in many settings. For example, many species are so rare and elusive that they can only be reliably surveyed using “indirect” methods such as camera traps or hair snares.

In response to the increasing use of camera traps in studies of threatened species, and the problems associated with commonly-used indices of abundance (?O’Brien, 2011; ?), several density estimators have been developed for situations in which the population being studied is unmarked (??). These estimators assume that (1) cameras are randomly placed with respect to animal density (2) animals neither avoid nor are attracted to the cameras, and (3) detection probability can be either modeled as a function of distance between the animal and the camera or as a function of movement velocity (which must be known or estimated using auxiliary data). Although these methods might represent an important improvement over index-based methods, the assumptions may not hold in many situations, especially when applied to data from standard designs in which camera stations are either baited or placed along trails – issues that can be dealt with directly using SCR models (see Chaps. 12 and 13). Nonetheless, empirical studies have found that the assumptions do hold in some cases (?).

Other common approaches to estimating density when individuals are unmarked include double observer sampling, removal sampling, and repeated counts, for which custom models have been developed (Nichols et al., 2000b; Farnsworth et al., 2002; Royle, 2004b,a; Nichols et al., 2009; Fiske and Chandler, 2011). To obtain reliable density estimates using these methods, the area surveyed must be well defined and closed with respect to movement and demographic processes. Given a sufficiently short sampling interval, such as a 5-min point-count, the closure assumption may be reasonable. However, short sampling intervals limit the number of detections, so observers generally visit each survey location multiple times during a season. But then, animal movement may invalidate the closure assumption, and a model of temporary emigration is required (Kendall et al., 1997; Chandler et al., 2011). Furthermore, distance-related heterogeneity in detection probability can introduce bias in these models, although this bias is negligible when the ratio of plot size to the scale parameter of the detection function is low (Efford and Dawson, 2009).

We mention these issues not to suggest that existing models do not have value – indeed we believe that they can be used to obtain reliable density estimates in many situations – rather, our aim is to highlight the need for alternative methods when the assumptions of existing methods cannot be met and when spatially-explicit inference is the objective.

18.2 SPATIAL CORRELATION IN COUNT DATA

14996 18.2.1 Spatial correlation as information

14997 All of the previous methods require some sort of auxiliary information to model both
14998 abundance and detection. For instance, multiple observers, distance data, or repeated
14999 visits may be required to ensure that model parameters are identifiable (but see Lele
15000 et al., 2012; Sólymos et al., 2012). The same is true for the SC model, but the auxiliary
15001 information comes in the form of spatial correlation, which requires no extra effort to
15002 collect.

15003 It is natural to be suspicious of the claim that spatial correlation is a good thing. In
15004 fact, elaborate methods have been devised to deal with spatial correlation as a nuisance
15005 (Lichstein et al., 2002; Dormann et al., 2007), and ecologists have been admonished for
15006 failing to obtain “real” replicates uncontaminated by spatial correlation (Hurlbert, 1984).
15007 The following heuristic may be helpful for seeing the value of spatial correlation in the
15008 context of density estimation.

15009 Imagine a 10×10 grid of camera traps and a single unmarked individual exposed
15010 to “capture” whose home range center lies in the center of the trapping grid. If the
15011 individual has a small home range size relative to the extent of the trapping grid, we can
15012 envision what the spatial correlation structure of the encounters might look like. If the
15013 animal’s home range is symmetric around the activity center then the number of times
15014 the individual is detected at each trap (the trap count) should decrease with the distance
15015 between the home range center and the trap; i.e., traps with the same distance from the
15016 activity center will yield counts that are more highly correlated with one another than
15017 traps located at different distances from the activity center. Thus, the correlation among
15018 the counts tells us something about the location of the activity center. It is relatively
15019 intuitive that spatial correlation carries information about distribution, but what about
15020 density?

15021 Imagine now that there are two activity centers located in the trapping grid. Using
15022 trap counts alone, is it possible to determine the number and location of these activity
15023 centers? The answer is yes, at least under certain circumstances. Fig. 18.1 shows the
15024 locations of the two hypothetical activity centers, and the total counts obtained at each
15025 trap after 10 survey occasions. Assuming that animals have bivariate normal home ranges,
15026 the fact that there are two areas in the map with high counts that dissipate with distance
15027 suggests that the most likely number of individuals given these data is 2. Furthermore, the
15028 degree to which the counts dissipate from the two areas of highest intensity is information
15029 about the parameter governing home range size. These two pieces of information are
15030 enough to estimate the number of individuals exposed to sampling – again, given that
15031 a bivariate normal home range is a valid assumption. Of course, the data could just as
15032 well have been generated by a single individual whose home range is distinctly bimodal,
15033 and thus *as always* the assumptions of our model need to be carefully examined using our
15034 biological knowledge of the system. If the assumptions do not hold, it is almost always
15035 possible to relax them, for instance by allowing for non-stationary home ranges as we
15036 demonstrated in Chapt. 12 and 13.

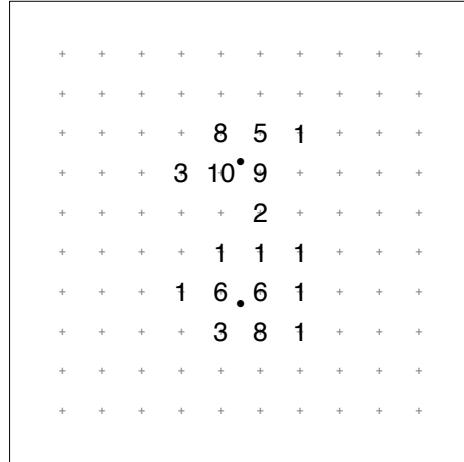


Figure 18.1. Simulated count data at each of 100 camera traps (crosses) after $K = 10$ sampling occasions. The black dots are the locations of two animal activity centers. The spatial count model estimates both the location and number of activity centers exposed to sampling using such spatially-referenced count data.

18.2.2 Types of spatial correlation

The spatial correlation dealt with by the SC model is assumed to arise from animal movement; however, this is just one type of spatial correlation that may exist in ecological count data. Another common type of spatial correlation results from the spatial correlation of environmental covariates. Habitat variables, such as, the percent cover of deciduous forest in North America, will often be patchy rather than randomly distributed, and this can result in spatial correlation in abundance, and hence in count data. Often, this type of spatial correlation can be dealt with by simply including the habitat covariate in the model. For example, a simple, non-spatially-explicit species distribution model with only a few habitat variables can result in a distribution map that reflects the spatial correlation in abundance (Sillett et al., 2012; ?). The point is that the relevant assumption of non-spatial models (e.g. GLMs) is that no spatial correlation exists in the *residuals*, and often, any spatial correlation apparent in the counts can be accounted for using covariates. This may be obvious, but it is a point that seems to be frequently misunderstood.

Of course, sometimes spatial correlation exists in residuals even after including covariate effects. This may be due to unobserved covariates or unobserved processes such as dispersal. When mechanistic models cannot be developed to describe these processes, several options exist for handling spatial correlation as a nuisance (?Zuur et al., 2009; ?). In the context of SCR models, including the SC model dealt with in this chapter, movement-induced spatial correlation is always explicitly modeled, and other sources of spatial correlation can be accounted for as well. For instance, environmentally-induced spatial correlation can be modeled by adopting an inhomogeneous point process model for

15059 the activity centers. That is, the point process intensity can be modeled as a function of
 15060 observed covariates, and theoretically, it should be possible to allow for spatially-correlated
 15061 random effects to deal with unobserved covariates. See Chapt. 11 for details.

18.3 SPATIAL COUNT MODEL

15062 18.3.1 Data

15063 Whereas traditional SCR models require spatially-referenced individual encounter histo-
 15064 ries, the SC model requires simple spatially-referenced count data. Let n_{jk} be the count
 15065 data at sampling location j on occasion k . The entire $J \times K$ matrix of counts will be
 15066 denoted \mathbf{n} . A sampling location in this context could be any device capable of recording
 15067 count data, such as a human observer or a camera trap, and one of the benefits of the SC
 15068 model is that it can be applied to data collected using many different survey methods.
 15069 For ease of presentation, we will refer to sampling devices as traps, but remember that
 15070 a trap is just something capable of recording count data. As in all SCR models, we also
 15071 require the coordinates of the J traps, and we denote the location of trap j by \mathbf{x}_j . In
 15072 some instances, additional data might be available such as trap-specific covariates, state-
 15073 space covariates, information on the identities of a subset of individuals, or perhaps even
 15074 distance data. We consider some of these model extensions in Sec. 18.6, but for the time
 15075 being we ignore these possibilities so that we can focus on the basic model.

15076 18.3.2 Model

15077 The state model is exactly the same as the one we have dealt with throughout this book.
 15078 It is a point process describing the number and distribution of activity centers in the
 15079 state-space \mathcal{S} . Although it might be possible to fit inhomogeneous point process models
 15080 using the methods described in Chapt. 11, given the simplicity of the data, we concentrate
 15081 on a homogeneous point process $\{\mathbf{s}_i, \dots, \mathbf{s}_N\} \sim \text{Uniform}(\mathcal{S})$ where \mathbf{s}_i is the activity center
 15082 of individual i in the population of size N . For the moment, we will assume that N is
 15083 known.

15084 The observation model is the same as in other SCR models in the sense that it de-
 15085 scribes the probability of encountering individual i at trap j , conditional on the location
 15086 of the individual's activity center. The specific encounter process will depend on the sam-
 15087 pling method, and here we consider the standard camera trapping situation in which an
 15088 individual can be encountered at multiple traps during a single occasion, say one night
 15089 during a camera-trapping study, and it can be detected multiple times at a single trap
 15090 during an occasion. This is the Poisson encounter model (a.k.a. the count detector case)
 15091 described in Chapt. 9. Our experience with alternative observation models such as the
 15092 Bernoulli and multinomial models suggests that the parameters of the model may not be
 15093 identifiable in these cases, at least when no additional information is available. This is a
 15094 subject of ongoing research.

15095 As before, we define y_{ijk} as the encounter data for individual i at trap j on occasion
 15096 k , which we model as:

$$y_{ijk} \sim \text{Poisson}(\lambda_{ij}) \quad (18.3.1)$$

15097 where λ_{ij} is the encounter rate. A common encounter rate model is the Gaussian, or
 15098 half-normal, model:

$$\lambda_{ij} = \lambda_0 \exp(-\|\mathbf{x}_j - \mathbf{s}_i\|^2 / 2\sigma^2)$$

15099 in which λ_0 is the baseline encounter rate, $\|\mathbf{x}_j - \mathbf{s}_i\|$ is the Euclidean distance between the
 15100 trap and activity center, and σ is the scale parameter determining the degree to which
 15101 encounter rate decreases with distance. In this context, σ also determines the amount of
 15102 correlation among the counts because if σ is low relative to the trap spacing, then it is
 15103 unlikely that an individual will be detected at multiple traps.

15104 When individuals cannot be uniquely identified, the encounter histories cannot be di-
 15105 rectly observed, which seems like a massively insurmountable problem of epic proportions.
 15106 The solution of Chandler and Royle (In press) is the same one we routinely apply when we
 15107 cannot directly observe the process of interest – we regard the encounter histories as latent
 15108 variables. This leaves the remaining task of specifying the relationship between the count
 15109 data and the encounter histories, i.e. we need a model of $[\mathbf{n}|\mathbf{y}]$ where \mathbf{y} represents the
 15110 entire collection of encounter histories. In this case, there is only one possibility because,
 15111 by definition, the count data are simply a reduced-information summary of the latent en-
 15112 counter histories. That is, they are the sample- and trap-specific totals, aggregated over
 15113 all individuals:

$$n_{jk} = \sum_{i=1}^N y_{ijk}. \quad (18.3.2)$$

15114 So, unlike most model-development problems faced in this book, we don't have to con-
 15115 sider competing probability models for $[\mathbf{n}|\mathbf{y}]$, but instead, we recognize the fact that the
 15116 relationship between the counts and the latent encounter histories is deterministic. This
 15117 deterministic constraint poses some computational challenges, which we discuss below.
 15118 But first we present some alternative formulations of the model.

15119 Recall from Chapt. 2 that the sum of two or more Poisson random variables is also
 15120 a Poisson random variable. Specifically, if $x_1 \sim \text{Poisson}(\lambda_1)$ and $x_2 \sim \text{Poisson}(\lambda_2)$, then
 15121 $(x_1 + x_2) \sim \text{Poisson}(\lambda_1 + \lambda_2)$. Thus, under this Poisson model for the latent encounter
 15122 histories, the count data can be modeled as Poisson:

$$n_{jk} \sim \text{Poisson}(\Lambda_j) \quad (18.3.3)$$

15123 where

$$\Lambda_j = \lambda_0 \sum_i \exp(-\|\mathbf{x}_j - \mathbf{s}_i\|^2 / 2\sigma^2),$$

15124 and because Λ_j does not depend on k , we can aggregate the replicated counts, defining
 15125 $n_{j\cdot} = \sum_k n_{jk}$ and then

$$n_{j\cdot} \sim \text{Poisson}(K\Lambda_j).$$

15126 As such, K and λ_0 serve equivalent roles as affecting baseline encounter rate. Formulating
 15127 the model in terms of the aggregated count data demonstrates that the model can be
 15128 applied to data from a single sampling occasion ($J \equiv 1$) as has been noted elsewhere for
 15129 standard SCR models (Efford et al., 2009b). In the context of studying marked popula-
 15130 tions, the model parameters will only be identifiable in the $J \equiv 1$ case if an animal can be
 15131 captured at multiple traps during a single occasion. The SC model essentially requires the
 15132 same thing, which is to say that it requires correlation in the count data resulting from
 15133 an individual being captured in multiple, closely-spaced traps.

This formulation of the model in terms of the aggregate count also simplifies computations as the latent encounter histories do not need to be updated in the MCMC estimation scheme; however, retaining them in the formulation of the model is important if some individuals are uniquely marked. This is because uniquely identifiable individuals produce observations of some of the y_{ijk} variables, which we elaborate on in the next chapter.

18.3.3 On N being unknown

Even though there are no observed encounter histories in the situation we consider here, we can still use data augmentation (Tanner and Wong, 1987; Liu and Wu, 1999) to resolve the problem that N is unknown. In fact, we are actually using two different types of data augmentation since we first augment the observed data with latent encounter histories, and then we augment this latent data array with a set of all-zero encounter histories. This approach turns out to be very similar to other data augmentation schemes used to model spatial dependence in other contexts (Wolpert and Ickstadt, 1998; Best et al., 2000).

Although the process of data augmentation should be familiar by now, we briefly review the basics. For homogeneous point process models, N is typically modeled as $N \sim \text{Binomial}(M, \psi)$, which is equivalent to a discrete uniform prior on N if $\psi \sim \text{Uniform}(0, 1)$. Since a binomial model is equivalent to a series of M independent Bernoulli trials, we can rewrite $N \sim \text{Binomial}(M, \psi)$ as $z_i \sim \text{Bernoulli}(\psi)$ where z_i is an auxiliary variable indicating if individual i is a member of the population, such that $N = \sum_{i=1}^M z_i$. Having expanded the model to include a prior on N , we can summarize the SC model, with a Gaussian observation model, as follows:

$$\begin{aligned} z_i &\sim \text{Bernoulli}(\psi) \\ y_{ijk} &\sim \text{Poisson}(\lambda_{ijk} z_i) \\ \lambda_{ijk} &= \lambda_0 \exp(-\|\mathbf{x}_j - \mathbf{s}_i\|^2)/(2\sigma^2) \\ n_{jk} &= \sum_{i=1}^M y_{ijk} \end{aligned}$$

18.3.4 Inference

Bayesian analysis can proceed once suitable priors have been put on the hyperparameters ψ , σ , and λ_0 . Chandler and Royle (In press) provided **R** code for fitting the model using MCMC, and they evaluated the model's performance with uniform priors on the three hyperparameters. They also discussed the possibilities and effects of including prior knowledge about σ into the model. In Sec. 18.5, we explain how the model can be implemented using **JAGS**, but first we briefly contemplate the viability of classical analysis of this model.

The obvious challenge faced when conducting a classical analysis of this model is that the number of latent variables is huge. In all SCR models, the activity centers are latent, but now, even the encounter histories are latent. Maximizing likelihoods with latent variables (random effects) involves integrating (or summing) over all possible values of the latent variables. For the activity centers, this is typically accomplished by integrating the conditional-on-s likelihood $[\mathbf{y}_i | \mathbf{s}_i]$ over the two-dimensional state-space \mathcal{S} (Chapt. 6).

However, with the SC model, we have to sum over all possible encounter histories meeting the constraint of Eq. 18.3.2. The number of possible encounter histories will, in general, be too high to make the likelihood tractable, and thus we do not think that maximum likelihood is a viable option for analyzing this model. However, one might be able to obtain approximate maximum likelihood estimates using simulation-based methods (Lele et al., 2010), which will typically be more computationally challenging than the Bayesian analysis.

18.4 HOW MUCH CORRELATION IS ENOUGH?

In Chapt. 10, we noted that if trap spacing is too wide relative to the encounter rate parameter σ , then few spatial recaptures will be realized and the model parameters will be estimated poorly. The same principal applies here – σ shouldn't be too small or too large relative to trap spacing or else the counts will be i.i.d. Poisson random variables. So how much correlation is enough? Phrased differently, what is the ideal ratio of σ to trap spacing to ensure correlation and minimize the variance of the posterior distributions? We see two options for answering this questions, both of which are topics in need of additional research. The first approach is to use the methods described in Chapt. 10, i.e. by either conducting simulation studies with various trap spacing to σ ratios, or to analytically minimize a variance criterion for a given set of sampling conditions and effort. The former approach was used by Chandler and Royle (In press) whose limited simulation study indicated that an ideal ratio is approximately 2. This agrees with findings from previous research on the optimal design of SCR studies (Chapt. 10), as it should.

15181 A second approach that may be of use if a data set has already been collected is to
15182 use standard techniques from spatial statistics to determine if adequate correlation exists
15183 in the counts. For example, one might compute Ripley's K -statistic or generate (semi-
15184)variograms (Illian et al., 2008). We have not studied the utility of such approaches, but
15185 it seems worthy of investigation.

18.5 APPLICATIONS

18.5.1 Simulation example

Simulating data under the SC model proceeds by first simulating standard SCR encounter history data and then collapsing it into count data. The following blocks of **R** code generate data from the model, with parameters $\sigma = 5$, $\lambda_0 = 0.4$, and $N = 50$. The state-space is a $[0, 100] \times [0, 100]$ square, and a grid of 100 traps is centered in the middle. The first block of code generates the trap coordinates X and the $N = 50$ activity centers:

```
15200 [1] 50
15201 > s <- cbind(runif(N, xlim[1], xlim[2]), runif(N, ylim[1], ylim[2]))
```

15202 We could have set $N = 50$ directly, but instead we treated density as a fixed parameter
 15203 ($\mu = 50$) and generated N as a random variable – it just so happens that with the specified
 15204 random seed, N equals 50.

15205 Now we can generate the encounter histories under the Poisson observation model.
 15206 Let's suppose that sampling is conducted over $K = 5$ nights.

```
15207 > sigma <- 5
15208 > lam0 <- 0.4
15209 > J <- nrow(X)
15210 > K <- 5
15211 > y <- array(NA, c(N, J, K))
15212 > for(j in 1:J) {
15213 +   dist <- sqrt((X[j,1]-s[,1])^2 + (X[j,2] - s[,2])^2)
15214 +   lambda <- lam0*exp(-dist^2/(2*sigma^2))
15215 +   for(k in 1:K) {
15216 +     y[,j,k] <- rpois(N, lambda)
15217 +   }
15218 + }
```

15219 The object y is the $N \times J \times K$ array of encounter data, which cannot be directly ob-
 15220 served if the animals are unmarked. Converting the encounter data to count data can be
 15221 accomplished using a single `apply` command.

```
15222 > n <- apply(y, c(2,3), sum)
15223 > dimnames(n) <- list(paste("trap", 1:J, sep=""),
15224 +                         paste("night", 1:K, sep=""))
15225 > n[1:4,]
15226   night1 night2 night3 night4 night5
15227 trap1     1      0      0      0      0
15228 trap2     1      2      2      0      1
15229 trap3     1      0      0      1      0
15230 trap4     0      0      0      0      0
```

15231 This displays the first 4 rows of n , the $J \times K$ matrix of counts.

15232 The question now is: Is it possible to estimate the parameters? In our simulated
 15233 dataset we have $J \times K = 500$ data points, but how many parameters do we need to
 15234 estimate? A frequentist might say that there are only 3 parameters: λ_0 , σ , and N (or
 15235 density μ) because inference about the latent parameters is carried out using prediction
 15236 methods after the 3 hyperparameters have been estimated. However, a Bayesian would
 15237 probably say that each s and each element of the latent encounter array y is a parameter
 15238 in need of a posterior. From this perspective there are far more parameters than data
 15239 points, and thus it would appear as though the situation is dire. Whether or not the
 15240 parameters are actually estimable is a rather difficult question to answer. One simplistic,
 15241 but not definitive, approach for addressing the question is to conduct a simulation study
 15242 and evaluate the frequentist performance of the model by asking how often the data-
 15243 generating values are included in confidence/credible intervals, and how biased are point

estimates. Chandler and Royle (In press) conducted such a simulation study and found that, while the variance of the posterior distribution was high by most standards, the bias of the posterior mode of N was small and the coverage of the credible intervals was close to nominal. Moreover, they found no evidence that the posterior distributions were dominated by the priors, further supporting the conclusion that spatial correlation in the count data is sufficient for estimating density and encounter probability parameters.

At this point in time the SC model can only be fit using one of the **BUGS** engines, or using custom software like the **R** code accompanying Chandler and Royle (In press). Although **BUGS** might provide the most flexible option for fitting the model, it is not straight-forward because of the constraints in the model. In **WinBUGS**, the $n_{jk} = \sum_i y_{ijk}$ constraint can be enforced using the so-called “ones-trick”, but we prefer **JAGS** because it has a distribution called **dsum** that was designed for this type of situation in which the observed data are a sum of random variables. Panel 18.1 shows the **JAGS** code, but we abbreviated the arguments to **dsum** because in practice you need to provide all M of them. The code looks slightly unwieldy if M is large, but you can easily create it using the **paste** function in **R**. Here is an example, with an unrealistically small value of $M = 10$:

```
15261 > paste("y[", 1:10, ",j,k]", sep="", collapse=", ")
15262 [1] "y[1,j,k], y[2,j,k], y[3,j,k], y[4,j,k], y[5,j,k], y[6,j,k],
15263 y[7,j,k], y[8,j,k], y[9,j,k], y[10,j,k]"
```

The **JAGS** model in Panel 18.1 can be used to fit the version of the model in which the latent encounters are updated at each Monte Carlo iteration. One challenge faced when using this version of the model is that **JAGS** cannot auto-generate initial values that honor the constraints in the model, so it is necessary to provide them. The following code presents one fairly general way of creating acceptable starting values and formatting the data for analysis using the **rjags** package:

```
15270 > library(rjags)
15271 > dat1 <- list(n=n, X=X, J=J, K=K, M=200, xlim=xlim, ylim=ylim)
15272 > init1 <- function() {
15273 +   yi <- array(0, c(dat1$M, dat1$J, dat1$K))
15274 +   for(j in 1:dat1$J) {
15275 +     for(k in 1:dat1$K) {
15276 +       yi[sample(1:dat1$M, dat1$n[j,k]),j,k] <- 1
15277 +     }
15278 +   }
15279 +   list(sigma=runif(1, 1, 2), lam0=runif(1),
15280 +       y=yi, z=rep(1, dat1$M))
15281 + }
15282 > pars1 <- c("lam0", "sigma", "N", "mu")
```

The code in Panel 18.1 is useful because it shows how closely this model is related to standard SCR models, and it provides the basis for including data on both marked and unmarked individuals, as will be discussed in the next chapter. However, this model runs very slowly, even when using a fast 64-bit machine with chains run in parallel. The code in Panel 18.2 runs much faster because it does not include the latent encounter histories.

Table 18.1. Posterior summaries from the spatial count (“SC”) model applied to simulated data using **scrbook** and **JAGS**. 25000 samples were generated, but substantial Monte Carlo error is still evident. All parameters were given uniform priors.

Parameter	Mean	SD	2.5%	50%	97.5%
scrUN(..., updateY=FALSE)					
$\sigma = 5$	4.718	0.922	3.239	4.615	6.833
$\lambda_0 = 0.4$	0.500	0.136	0.268	0.489	0.793
$N = 50$	60.653	31.067	21.000	54.000	137.000
scrUN(..., updateY=TRUE)					
σ	4.554	0.784	3.216	4.486	6.264
λ_0	0.489	0.131	0.262	0.479	0.775
N	64.772	30.162	26.000	59.000	140.000
JAGS (without latent encounter histories)					
σ	4.70	0.88	3.24	4.66	6.63
λ_0	0.52	0.14	0.27	0.52	0.80
N	58.55	30.30	20.00	52.00	135.00

15288 An even faster (but perhaps less efficient) alternative is to use the **scrUN** function in
 15289 **scrbook**. The usage is as follows:

15290 > out1 <- scrUN(n=n, X=X, M=300, niter=25000, xlims=xlim, ylims=ylim,
 15291 inits=list(lam0=0.3, sigma=rnorm(1, 5, 0.1)), updateY=TRUE,
 15292 tune=c(0.004, 0.09, 0.35))

15293 where **n** is the matrix of counts, **X** is the trap coordinate matrix, **M** sets the size of the data-
 15294 augmented latent data, **xlims** and **ylims** define the rectangular state-space, **inits** is a list
 15295 of starting values, and **updateY** determines if the latent encounter histories are updated as
 15296 part of the MCMC algorithm. In general, we recommend using the option **updateY=FALSE**
 15297 because the Markov chains tend to mix better. Even so, it can be important to fiddle
 15298 with the tuning parameters until the acceptance rates are between 40–60%. Otherwise,
 15299 the Markov chains will exhibit extremely high autocorrelation. This is one reason to favor
 15300 **JAGS** over our implementation in **scrbook** since **JAGS** finds suitable tuning parameters
 15301 automatically during the adaptive phase (when using Metropolis updates).

15302 We fit the model to the simulated data using both formulations – with and without the
 15303 latent encounter histories – using both **JAGS** and **scrUN**. Table 18.1 shows summaries of
 15304 25000 posterior draws, and suggests that while the true parameter values are easily covered
 15305 by the 95% credible intervals, the intervals are rather wide. This low precision is not just
 15306 a peculiarity of this particular data set – it will generally be low unless the sample size
 15307 is very large, as noted by Chandler and Royle (In press). Furthermore, autocorrelation
 15308 of the samples will typically be high Fig. 18.2, and thus it may take many iterations to
 15309 achieve convergence. The results shown in Fig. 18.2 also indicate that the algorithm that
 15310 includes the latent encounter histories seems to have a hard time exploring the region of
 15311 the posterior in which N is low. Given these technical difficulties, we recommend using
 15312 the **JAGS** implementation (based on Panel 18.2), and it is always a good idea to use
 15313 MCMC diagnostic tools such as those available in the **coda** package (Chapt. 17).

15314 The take-home message is that, even with simulated data, the precision of the posterior
 15315 distributions is low and mixing is poor. This should be expected given that we are asking so

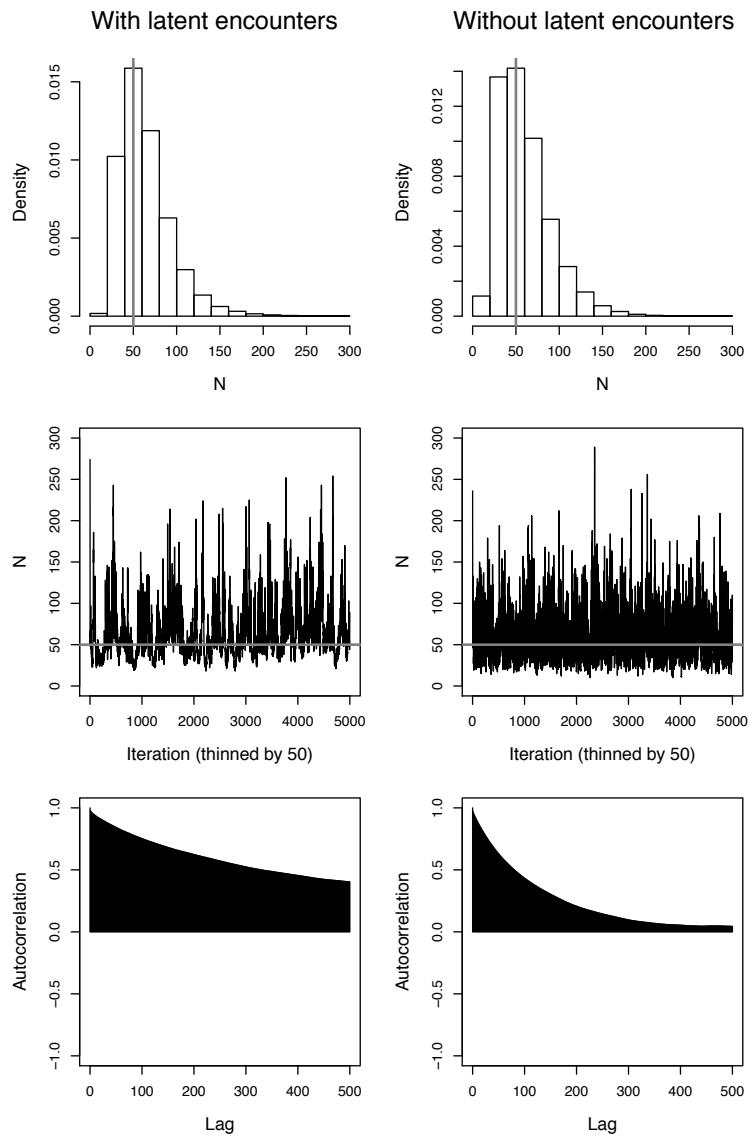
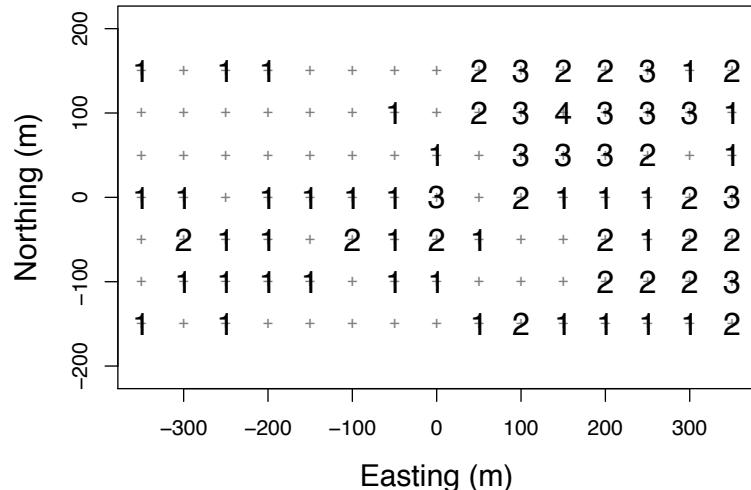


Figure 18.2. MCMC results for the parameter N from the two algorithms (with and without the latent encounter histories). The first row contains the histograms of the posterior distributions, the second row contains the history (or trace) plots, the third row shows the autocorrelation plots.



18.5.2 Northern parula in Maryland

15316 much from so little data. In essence, we are trying to fit a point process model while being
 15317 twice removed from the actual point (activity center) locations. These difficulties may
 15318 warrant the investigation of simpler models at the expense of the mechanistic description
 15319 of the system. Another option is to figure out ways of improving model precision – options
 15320 we discuss in Sec. 18.6. Before doing so, we re-analyze the Northern Parula (*Parula*
 15321 *americana*) data described in Chandler and Royle (In press)

We fit the model using **JAGS** and the code from Panel 18.2, which does not include
 15333 the latent encounter histories. For comparative purposes, we used two sets of priors: (1)
 15334 the improper uniform priors considered by Chandler and Royle (In press) and (2) proper
 15335 Gamma(0.001, 0.001) priors for σ and λ_0 as well as an approximate $N \propto 1/N$ prior, which
 15336 is (almost) the Jeffreys prior (see ? and Link and Barker (2010) for details on Jeffreys

Table 18.2. Posterior summary statistics for spatial count model applied to the northern parula data. Note the sensitivity to the two sets of priors.

Par	Prior	Mean	SD	2.5%	50%	97.5%
N	DUnif(0, 300)	38.474	37.275	1.000	29.000	138.000
λ_0	Unif(0, ∞)	0.310	0.183	0.082	0.269	0.817
σ	Unif(0, ∞)	127.935	99.303	44.760	87.291	438.374
N	$\propto 1/N$	29.591	32.555	1.000	19.000	119.000
λ_0	Gamma(0.001, 0.001)	0.309	0.194	0.078	0.261	0.843
σ	Gamma(0.001, 0.001)	150.183	105.044	48.735	117.069	447.616

prior). The state-space created by buffering the grid of point count locations by 250 m. M was set to 300. To reduce computation time, we used the `parallel` package and distributed 3 chains to 3 separate cores. The entire example can be reproduced using the code on the help page for `nopa` in our **R** package `scrbook`. The following code illustrates the essential elements:

```

15342 > library(scrbook)
15343 > library(rjags)
15344 > dat2 <- list(n = nopa$n, X = nopa$X, M=300, J=nrow(nopa$n), K=ncol(nopa$n),
15345 +           xlim=c(-600, 600), ylim=c(-400, 400))
15346 > init2 <- function() {
15347 +   list(sigma=rnorm(1, 100), lam0=runif(1), z=rep(1, dat2$M))
15348 + }
15349 > cl2 <- makeCluster(3) # Open 3 parallel R instances
15350 > clusterExport(cl2, c("dat2", "init2", "pars1")) # send objects to 3 cores
15351 > out2 <- clusterEvalQ(cl2, { # executes the folowing command on each core
15352 +   library(rjags)
15353 +   jm <- jags.model("nopa2.jag", dat2, init2, n.chains=1, n.adapt=1000)
15354 +   jc <- coda.samples(jm, pars1, n.iter=150000)
15355 +   return(as.mcmc(jc))
15356 + })
15357 > mc2 <- mcmc.list(out2) # put the 3 chains together
15358 > plot(mc2)
15359 > summary(mc2)

```

18.6 EXTENSIONS OF THE SPATIAL COUNT MODEL

18.6.1 Improving Precision

The results of the parula analysis are presented in (Table 18.2). Once again, we see wide credible intervals for N , and also high sensitivity to the priors. These limitations support the conclusions of Chandler and Royle (In press) that researchers should: (1) elicit prior information from the published literature and/or (2) mark a subset of individuals when applying the SC model. Both of these options should be readily accomplished in many studies, especially the first option because extensive information on home range size has

been compiled for many species in diverse habitats (e.g. DeGraaf and Yamasaki, 2001), which can be embodied as a prior distribution for the encounter rate parameter σ in a Bayesian analysis (Chandler and Royle (In press), Chapt. 5).

In some cases, it may not be possible to mark any individuals, and no prior information may exist about encounter parameters; however, it may be possible to improve precision by collecting auxiliary data, such as distance measurements. In fact, in the parula study, detections were classified as either within or beyond 150 m, and it seems sensible to expand the model to accommodate this rudimentary distance sampling data. But if auxiliary data such as distance measurements exist, why bother with the SC model at all since density can be estimated using the distance data alone? This is a good point, and in general, the simplest model that does the job should be preferred. The reasons why one might prefer an expanded SC model over a simple distance sampling model include the ability to model spatial correlation and the ability to model movement. But how exactly can the SC model be extended to accommodate such auxiliary data?

The basic extension that we consider here is to use a type of search-encounter model (Chapt. 15) that includes the activity centers (\mathbf{s}) and the actual locations of individuals (\mathbf{u}). By including both activity centers and actual locations in the model, abundance in any region \mathcal{B} is given by

$$N(\mathcal{B}) = \sum_i I(\mathbf{u}_i \in \mathcal{B}). \quad (18.6.1)$$

Thus, in the context of distance sampling studies in which the distance data are recorded in discrete intervals, the region \mathcal{B} would be the area corresponding to a particular distance interval. The probability of detecting the individuals $N(\mathcal{B})$ would be the average detection probability \bar{p} , which is computed by integrating a distance-based detection function over the distance interval.

In other contexts, such as when conducting removal surveys, the region \mathcal{B} could be a fixed-area plot, such as a stream segment. Again, Eq. 18.6.1 could be used to model local abundance ($N(\mathcal{B})$), and detection probability within the region could be modeled conditional on $N(\mathcal{B})$. A reasonably general description of this model is as follows:

$$\begin{aligned} \mathbf{s}_i &\sim \text{Uniform}(\mathcal{S}) \\ \mathbf{u}_{ik} &\sim \text{BVN}(\mathbf{s}_i, \boldsymbol{\Sigma}) \\ N(\mathcal{B}_{jk}) &= \sum_{i=1}^M I(\mathbf{u}_{ik} \in \mathcal{B}_{jk}) \\ n_{jkl} &\sim \text{Binomial}(N(\mathcal{B}_{jkl}), p) \end{aligned}$$

where $\boldsymbol{\Sigma} = \begin{pmatrix} \tau & 0 \\ 0 & \tau \end{pmatrix}$, with τ governing the size of the bivariate normal home range. The interpretation of the parameter p will depend upon the survey protocol (Nichols et al., 2009).

When plots are far enough apart that individuals cannot move between them, the counts will be uncorrelated and the model can be approximated using a non-spatial N -mixture model allowing for temporary emigration (Chandler et al., 2011). In the next example, we consider data in which the plots are obviously not independent.

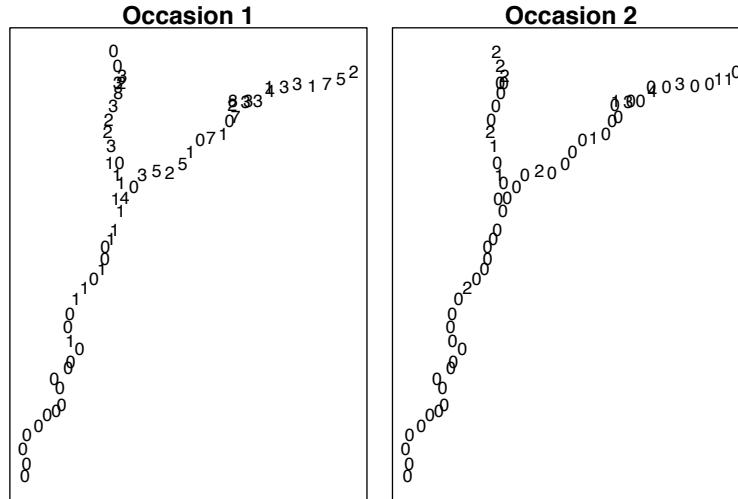


Figure 18.4. Stream segment counts of northern dusky salamanders in the Chesapeake and Ohio National Historic Park, VA/MD. Each number is the count associated with a 25-m stretch in which 3 removal passes were made on 3 occasions each summer (only 2 occasions are shown here). Notice the consistency of the spatial correlation between occasions and the temporal decline in the counts.

18.6.2 Dusky Salamanders in Maryland

The independence assumption of the Chandler et al. (2011) model will not always hold. A prime example is in studies of aquatic species in stream networks. For example, consider the data depicted in Fig. 18.4. The figure shows counts of northern dusky salamanders (*Desmognathus fuscus*) in 25-m stretches on a small stream in the Chesapeake and Ohio National Historic Park. The data were collected by E.H.C. Grant and colleagues with the objective of understanding the spatial and temporal dynamics of salamander populations in response to seasonal and annual variations in stream hydrology. More details the study can be found in ?.

To sample the population, the stream networks were divided into 25-m stretches, and in each stretch, “temporary” removal sampling was conducted, which involves capturing and removing salamanders on 3 consecutive passes. The salamanders are placed in a bucket of water for the brief 10-20 min duration of sampling, and then they are released at the location of capture. The entire process is repeated 3-4 times per season (May-Aug.). In a subset of streams and years, individuals are marked, but in general it is too expensive to mark the entire population, and the data considered here consists entirely of unmarked individuals.

The sampling protocol may be thought of as a “robust design” (Pollock, 1982), with “occasions” (typically 1 day) being the primary period, and secondary samples being the removal passes within the primary periods. An obvious feature of these data is that the

Table 18.3. Posterior summarizes from removal model of salamander counts allowing for movement and decreasing population size over the course of a breeding season.

Parameter	Mean	SD	2.5%	50%	97.5%
N_1	178.393	16.346	151.000	177.000	214.000
N_2	62.322	6.884	51.000	62.000	77.000
N_3	21.202	3.695	15.000	21.000	29.000
ϕ	0.348	0.038	0.275	0.348	0.425
τ	27.427	3.200	21.293	27.173	33.706
p	0.396	0.053	0.294	0.394	0.502

15417 neighboring counts are spatially correlated. In this case, we have reason to believe that
 15418 this correlation is the result of habitat preferences, with individuals actively selecting
 15419 habitat in the upper reaches of the streams. This could be modeled as a function of a
 15420 covariate describing the distance from the mouth of the stream. Another obvious feature
 15421 of this data is that the pattern of spatial correlation remains consistent between occasions,
 15422 but the overall counts decline markedly over the course of the season. These phenomena
 15423 can be explained by the fact that the salamanders have relatively small home ranges, and
 15424 this results in the consistent pattern of correlation among occasions. Furthermore, as the
 15425 season progresses, the streams dry out, and many individuals move underground.

15426 Given the importance of movement within home ranges, which determines the correla-
 15427 tion among occasions, and movement underground, which results in a decreasing number
 15428 of individuals being available for sampling, it would be helpful to have a model that de-
 15429 scribes both processes and allows for evaluation of hypotheses regarding the effects of
 15430 environmental variables. For example, one might ask how stream flow is related to the
 15431 probability that an individual remains active, i.e. aboveground. A model describing this
 15432 process could be used to predict activity levels under future conditions. Although we do
 15433 not investigate covariate effects in this section, we do present a general model allowing for
 15434 movement among occasions, and for decreasing availability over the season.

15435 This expanded model is founded on the one described in the previous section, but it
 15436 also includes a removal model for the observation process, and it includes a basic “open”
 15437 population model to allow for a decline in abundance over time (Chapt. 16). Actually,
 15438 the population is not thought to decline substantially during the season, but rather, the
 15439 number of individuals *available* for detection declines because many individuals move
 15440 underground as the streams dry. Each of these components is included in the **BUGS** de-
 15441 scription of the model presented in Panel 18.3.

15442 We fit this model to the data and obtained the posterior distributions summarized
 15443 in Table 18.3. The results indicate that the population size available for detection did
 15444 decrease rapidly during the season, the rate of which is determined by the ϕ parameter.
 15445 Modeling this parameter as a function of water flow or volume would allow one to predict
 15446 salamander activity under future environmental conditions. Another result of the analysis
 15447 is that the movement parameter, τ , was relatively low, indicating that adult salamanders
 15448 rarely move more than 100 m from their home range center during a season. This explains
 15449 why the distribution of individuals within the stream remains relatively constant over time.

18.7 SUMMARY AND OUTLOOK

15450 Unlike traditional models of count data used in ecology, the SC model is parameterized
15451 in terms of *individuals* – individuals that just so happen not to be marked. Although
15452 developing the model in terms of latent encounters increases model complexity, several
15453 reasons exist for accommodating this latent structure. First, it allows for individual-level
15454 covariates, including the location of each individual in the population. Second, by in-
15455 cluding an underlying point process specific to individual activity centers (and possibly
15456 locations in time), the model allows for modeling continuous variation in density, which
15457 may result in bias when ignored in conventional models of count data. Third, accommo-
15458 dating the latent structure provides a more mechanistic description of ecological systems,
15459 for example by attaching a mechanism (movement) to the widely observed phenomenon
15460 of spatial correlation in count data.

15461 The SC model is a conceptually simple extension of standard SCR models, but in terms
15462 of computational requirements and latent structure, it is perhaps at the extreme end of
15463 what is possible to do with count data. As is always true, the harder we try to mirror
15464 reality with our models, the harder it becomes to estimate the parameters of the system.
15465 In this chapter, we tried to emphasize that as conceptually appealing as the SC model may
15466 be, it is unlikely to produce satisfying results in the absence of additional information.
15467 However, additional information such as home range size estimates will often be available
15468 for many species, and if not, we have provided an alternative method of accommodating
15469 additional data in the form of distance measurements or removal counts. This can greatly
15470 increase precision of estimates from studies designed to make spatially-explicit inferences
15471 about population processes.

15472 Although we focused on the situation in which all individuals are unmarked, one of
15473 the reasons for developing this class of models was to handle the problem of estimating
15474 population size or density when only a subset of individuals are marked. This topic is
15475 the subject of the next chapter, which builds on a rich literature of combining data from
15476 marked and unmarked individuals for purposes of designing efficient studies Bartmann
15477 et al. (1987); Neal et al. (1993); ?; ?.

```

model{
  sigma ~ dunif(0, 200) # Tailor this to your state-space
  lam0 ~ dunif(0, 5)    # consider dgamma() as an alternative
  psi ~ dbeta(1,1)
  for(i in 1:M) {
    z[i] ~ dbern(psi)
    s[i,1] ~ dunif(xlim[1], xlim[2])
    s[i,2] ~ dunif(ylim[1], ylim[2])
    for(j in 1:J) { # Number of traps
      distsq[i,j] <- (s[i,1] - X[j,1])^2 + (s[i,2] - X[j,2])^2
      lam[i,j] <- lam0 * exp(-distsq[i,j] / (2*sigma^2))
      for(k in 1:K) { # Number of occasions
        y[i,j,k] ~ dpois(lam[i,j]*z[i])
      }
    }
  }
  for(j in 1:J) {
    for(k in 1:K) {
      n[j,k] ~ dsum(y[1,j,k], y[2,j,k], ..., y[200,j,k])
    }
  }
  N <- sum(z[]) # Realized population size
  A <- (xlim[2]-xlim[1])*(ylim[2]-ylim[1]) # Area of state-space
  D <- N / A # Realized density
  ED <- (M*psi)/A # Expected density
}

```

Panel 18.1: **JAGS** code defining the spatial count model. This version includes the latent encounter histories. Note the abbreviated arguments to `dsum()`.

```
model{
  sigma ~ dunif(0, 200)
  lam0 ~ dunif(0, 5)
  psi ~ dbeta(1,1)
  for(i in 1:M) {
    z[i] ~ dbern(psi)
    s[i,1] ~ dunif(xlim[1], xlim[2])
    s[i,2] ~ dunif(ylim[1], ylim[2])
    for(j in 1:J) { # Number of traps
      distsq[i,j] <- (s[i,1] - X[j,1])^2 + (s[i,2] - X[j,2])^2
      lam[i,j] <- lam0 * exp(-distsq[i,j] / (2*sigma^2)) * z[i]
    }
  }
  for(j in 1:J) {
    bigLambda[j] <- sum(lam[,j])
    for(k in 1:K) {
      n[j,k] ~ dpois(bigLambda[j])
    }
  }
  N <- sum(z[])
}
```

Panel 18.2: **JAGS** code defining the spatial count model. This version does not include the latent encounter histories, and thus runs much faster than the code in Panel 18.1.

```

model {
  phi ~ dbeta(1,1)      # "availability" parameter
  tau ~ dunif(0, 1000)   # "movement parameter" of Gaussian kernel model
  p ~ dbeta(1,1)         # detection prob
  psi ~ dbeta(1,1)       # data augmentation parameter
  for(i in 1:M) {
    z[i,1] ~ dbern(psi)      # is the individual real?
    z[i,2] ~ dbern(z[i,1]*phi) # and still aboveground?
    z[i,3] ~ dbern(z[i,2]*phi) # ...
    s[i] ~ dcat(PrSeg[])
  }
  for(g in 1:G) {
    PrU[i,g] <- exp(-distmat[s[i],g]^2/(2*tau^2)) # Pr(u | s)
  }
  for(k in 1:K) {
    u[i,k] ~ dcat(PrU[i,])
    for(g in 1:G) {
      y[i,g,k] <- (u[i,k] == g)*z[i,k] # was individual at u==g?
    }
  }
}
for(j in 1:J) {
  for(k in 1:K) {
    NB[j,k] <- sum(y[,seg[j],k]) # Number in seg j at time k
    # removal model:
    n[j,1,k] ~ dbin(p, NB[j,k])
    NB2[j,k] <- NB[j,k] - n[j,1,k]
    n[j,2,k] ~ dbin(p, NB2[j,k])
    NB3[j,k] <- NB2[j,k] - n[j,2,k]
    n[j,3,k] ~ dbin(p, NB3[j,k])
  }
}
N[1] <- sum(z[,1]) # Total abundance, occasion 1
N[2] <- sum(z[,2]) # Total abundance, occasion 2
N[3] <- sum(z[,3]) # Total abundance, occasion 3
}

```

Panel 18.3: **BUGS** description of model for the data shown in Fig. 18.4. The model allows for spatially-explicit temporary emigration, and for a decrease in abundance as individuals move underground throughout the course of the season.

19

SPATIAL MARK-RESIGHT MODELS

15481 Throughout most of this book we have dealt with the situation where all individuals are
15482 identifiable upon encounter because they carry some form of individual mark. In Chapt.
15483 18 we introduced and developed an SCR model for non-identifiable populations, a spatial
15484 *non-capture-recapture* model, if you will. These two extremes are common in the study
15485 of animal populations with non-invasive sampling methods. However, there is also an
15486 intermediate situation where part of the population is tagged or otherwise marked and
15487 can thus be identified upon recapture, while the unmarked portion remains unidentifiable.
15488 In this situation so-called mark-resight models (Bartmann et al., 1987; Arnason et al.,
15489 1991; Neal et al., 1993) can be used to estimate population size and density by combining
15490 data from both the marked and unmarked individuals.

15491 Traditionally, capture-recapture studies involved physical capture and marking of in-
15492 dividuals throughout the study. This methodology is still widely applied in the study
15493 of species that are relatively easy to capture, such as small mammals, but can be very
15494 costly, logically challenging and risky when dealing with larger species. In contrast, in
15495 mark-resight studies a sample of individuals is captured and tagged (or otherwise marked)
15496 during a single marking event. Marking is followed by resighting surveys, upon which both
15497 the detection of marked and unmarked animals is recorded. Resighting surveys are usually
15498 non-invasive (hence the name ‘resighting’), so that they don’t involve handling of animals.
15499 As such, mark-resight models have a major advantage over traditional capture-recapture
15500 models in that they only require individuals to be captured and handled once, during the
15501 initial marking. This reduces field costs and risks for the animals (and potentially the
15502 researchers).

15503 Mark-resight models have a set of underlying assumptions, most of which are identi-
15504 cal to those of capture-recapture models; e.g., demographic population closure (violation
15505 of geographic population closure can be accommodated by some models) and no loss or
15506 misidentification of marks (see also Chapt. 5). Just like standard capture-recapture mod-
15507 els, there are means to incorporate heterogeneity in capture probability. An essential
15508 assumption of mark-resight models is that the marked individuals are a representative
15509 sample of the study population, so that inference about detection can be made for the

15510 whole population from the marked sample. While this is also an implicit assumption of
15511 capture-recapture models, in mark-resight models this means that the process of marking
15512 individuals requires careful consideration in order to produce a random sample. This
15513 assumption is usually addressed by employing a different method for marking than for
15514 resighting.

15515 Owing to the advantages of mark-resight over capture-recapture, especially when dealing
15516 with hard-to-trap species, mark-resight is a popular tool in wildlife population studies.
15517 The method has been applied for decades to a suite of species and survey techniques,
15518 ranging from banding and resighting Canada geese (Hestbeck and Malecki, 1989) to ear-
15519 tagging and camera-trapping grizzly bears (Mace et al., 1994) to paintball marking and
15520 areal resightings of large ungulates (Skalski et al., 2005).

15521 In this chapter we consider mark-resight within a spatial context and develop a spatial
15522 mark-resight (SMR) model. To motivate this model development, imagine you conduct
15523 a live-trapping study during which you capture and mark a number of animals with
15524 individually recognizable tags. Subsequently, you go back out to the field and conduct
15525 resighting surveys on an array of locations, and during these resighting surveys you see
15526 some of your marked individuals, as well as new, unmarked ones. Then, for the marked
15527 animals you obtain the same type of spatially explicit individual encounter histories as you
15528 would in a standard SCR study. In addition, you obtain site (and occasion) specific counts
15529 of individuals you did not mark. SMR models make use of both the encounter history data
15530 from the marked individuals and the counts of unmarked individuals to estimate density
15531 and detection parameters.

15532 In the following sections we first provide some background information on mark-resight
15533 and the types of data such surveys can provide. We will further explore the implications of
15534 the assumption of the marked individuals being a random subset of the population, which,
15535 in the context of SMR models refers to not only the *demographic composition*, but also to
15536 the *spatial distribution* of the marked individuals in the state-space \mathcal{S} . In many real life
15537 sampling situations, this assumption will not hold – animals will most often be marked in
15538 some region that does not represent the entire state-space. As a result, the distribution
15539 of marked individuals will generally *not* follow a homogeneous point process, but their
15540 activity centers will be concentrated in the vicinity of where marking took place. For
15541 the sake of model development, however, throughout the central part of this chapter we
15542 will make the assumption that marked animals are a random sample from the population
15543 in \mathcal{S} . We will show that SMR models are hybrids of standard SCR models and the
15544 models presented in Chapt. 18 for data where individuals cannot be uniquely identified.
15545 We explore models for both known and unknown numbers of marked individuals, and
15546 for imperfect individual identification of marks, and approaches to incorporate telemetry
15547 location data. In the spatial framework, most of the information on model parameters
15548 comes from the marked individuals. But in Sec. 19.5 we will see that, analogous to
15549 the models we developed previously in Chapt. 18, the spatial correlation in counts of
15550 unmarked individuals also contributes information about detection and movement. We
15551 conclude the chapter by presenting some general strategies for addressing a situation where
15552 marked individuals are not a random sample from \mathcal{S} .

19.1 BACKGROUND

15553 Before we start exploring spatial mark-resight approaches in more detail, we need to
15554 establish some terminology and gain a clear understanding of what types of mark-resight
15555 data we can have, in order to appreciate and understand the different flavors of mark-
15556 resight models.

15557 19.1.1 Resighting techniques

15558 As with capture-recapture surveys, there are numerous methods suitable for obtaining
15559 resightings. Common methods are visits to a set of points for resightings by an observer,
15560 or camera-trapping; but resightings need not be restricted to a particular set of locations.
15561 We can just as well envision a search-encounter kind of method, where a certain area is
15562 searched, systematically or opportunistically, for marked animals (see Chapt. 15). In this
15563 chapter we will only deal with fixed location resighting surveys, and we will refer to the set
15564 of resighting locations as the resighting array. In some instances we will also be concerned
15565 with where marked animals were captured, and we refer to these locations as the marking
15566 locations.

15567 19.1.2 Types of mark-resighting data

15568 In general, we have (at least) two sets of data: encounter histories for marked, and thus,
15569 identifiable individuals i at resighting location j and occasion k , y_{ijk} , and counts of un-
15570 marked records, n_{jk} , for each resighting location j and occasion k . Depending on the
15571 sampling technique, we can conceive of three slightly different types of partial ID data.

15572 **(1) Known number of marked individuals:** If you implement a resighting survey
15573 shortly after the marking session, you may be confident that none of the marked individuals
15574 have died or lost their mark. Under these circumstances you know that the number of
15575 marked individuals available for resighting, m , is equal to the number of individuals you
15576 marked. Alternatively, the marking technique might involve radio-transmitters, allowing
15577 you to confirm the presence or absence of marked individuals in the resighting survey area
15578 using radio-telemetry (White and Shenk, 2001). In both cases, you know the number of
15579 marked individuals in the surveyed population. In this situation, even though you may
15580 fail to resight some of the marked individuals, you know how many there are, and so you
15581 can simply assign all-zero encounter histories to the marked individuals not encountered
15582 – in other words, contrary to regular capture-recapture models, in mark-resight models
15583 with a known number of marked individuals, we can observe all-zero encounter histories.
15584 Under these circumstances, estimating N reduces to estimating the number of unmarked
15585 individuals, U .

15586 **(2) Unknown number of marked individuals:** If m is not known, for example because
15587 we suspect that some of the marks may have been lost between tagging and conducting
15588 the resighting surveys, we obtain a slightly different type of mark-resight data. Here, we
15589 do not accurately know the number of marked individuals available for resighting. As a
15590 consequence, individuals have to be resighted at least once for us to know they are still
15591 marked and alive and thus available for resighting. So, contrary to the situation where

15592 we know m and analogous to regular capture-recapture models, we cannot observe all-
15593 zero encounter histories of marked individual. In this situation, estimating N involves
15594 estimating both m and U .

15595 A special case of this kind of data can arise from camera trapping. Even when dealing
15596 with a species that has no spots or stripes, some individuals in the study population can
15597 have natural marks that make them identifiable on pictures, such as scars or a distinct
15598 coloration. In this scenario an individual has to be photographed at least once to be
15599 known. Here, the fact that both the “marking” method and the subsequent resighting
15600 method are the same (although marking in this case does not involve any actual physical
15601 marking) can be cause for concern: our sample of “marked” individuals may not be a
15602 random sample of the population but consist of individuals that for some reason are more
15603 likely to be photographed (e.g., individuals with activity center more interior to the trap
15604 array). In that case, a basic assumption of the mark-resight model is violated.

15605 **(3) Unknown marked status:** Finally, consider a scat or hair snare survey, where only
15606 a part of the sample is analyzed genetically (or DNA can only be extracted from a subset
15607 of samples due to sample quality). In this scenario, n_{jk} can contain both completely
15608 unknown individuals that are not represented at all in the set of encounter histories of
15609 marked animals, Y , but it can also contain samples from individuals that we previously
15610 identified. The difference is that in the first two scenarios, part of the population of
15611 individuals is identifiable, while in the third scenario, part of the sample of individuals
15612 is identifiable. This type of data violates one of the basic assumptions of mark-resight
15613 models, namely, that marked individuals are always correctly identified as such.

15614 To our knowledge there are currently no mark-resight models available that account for
15615 possible misidentification of the marking status of individuals (although some literature is
15616 available on misidentification of individuals in capture-recapture studies, e.g., Yoshizaki
15617 et al., 2009; Lukacs and Burnham, 2005; Link et al., 2010). In this chapter we will ignore
15618 this kind of data and focus instead on types (1) and (2).

15619 For both types of data a slightly different situation arises when we can only tell that
15620 an individual is marked, but not who it is. You may be able to see that an individual is
15621 marked but the identifying feature of the tag (a number or coloration) may have become
15622 unreadable, or may be hidden from view. In this case, in addition to the observed y_{ijk}
15623 and n_{jk} , you also observe a number of sightings of marked but unidentified individuals,
15624 say r_{jk} .

15625 19.1.3 A short history of mark-resight models

15626 Initially, mark-resight methods focused on radio-tagged individuals to estimate popula-
15627 tion size (White and Shenk, 2001). Radio-collars provide a means of determining which
15628 of the animals are in the study area and available for sampling, thus determining the
15629 number of marked individuals in the population. Knowing this number was a prerequisite
15630 for most earlier mark-resight approaches (White, 1996). The oldest mark-resight model
15631 is the good old Lincoln-Petersen estimator, where individuals are marked and a single
15632 resight/recapture occasion is carried out (Krebs, 1999). We need not identify individuals,
15633 but only to tell apart marked from unmarked individuals. Let m be the number of marked
15634 individuals in the population, $m_{(R)}$ the number of marked individuals seen on the resight-
15635 ing occasion, and $n_{(R)}$ the total number of marked and unmarked individuals observed

15636 during resighting. Population size N is then estimated as

$$N = m \times n_{(R)} / m_{(R)}.$$

15637 Mark-resight models using individual capture histories over several resighting occasions
 15638 were developed in the 1980s and 90s and compiled into the program **NOREMARK**
 15639 (White, 1996). Apart from the basic model with known number of marked individuals
 15640 and no individual variation in resighting probabilities (joint hypergeometric maximum
 15641 likelihood estimator) (Bartmann et al., 1987; White and Garrott, 1990; Neal, 1990; Neal
 15642 et al., 1993), **NOREMARK** contains models that account for lack of geographic population
 15643 closure (Neal et al., 1993), individual heterogeneity in resighting rates and sampling
 15644 with replacement (i.e. individuals can be seen more than once on any occasion, (Minta
 15645 and Mangel, 1989; Bowden, 1993)). A first mark-resight model allowing for an unknown
 15646 number of marked individuals was developed by Arnason et al. (1991).

15647 While many of these models perform well under certain situations, they are somewhat
 15648 limited in that they do not allow for combining data across several surveys (McClintock
 15649 et al., 2006) and not all of them are likelihood-based or allow for different parameterizations
 15650 (e.g., including a time effect on detection), so that selection of the most appropriate
 15651 model cannot be based on standard approaches such as AIC, but is largely left up to
 15652 educated guesswork (McClintock et al., 2006). Recently, more flexible and generalized
 15653 likelihood-based mark-resight models have been developed. These models can account
 15654 for individual heterogeneity in detection, unknown number of marked individuals and
 15655 lack of geographical closure, as well as a less than 100% individual identification rate of
 15656 marked individuals; they can be applied to sampling with and without replacement and
 15657 can combine data across several primary sampling occasions in a robust design type of
 15658 analysis (McClintock et al., 2009a,b). Since they are all likelihood-based, model selection
 15659 among different parameterizations and model averaging based on AIC is an option. Most
 15660 of these models have also been incorporated into the program **MARK** (McClintock and
 15661 White, 2012).

15662 For a detailed treatment of these different non-spatial mark-resight models, we refer
 15663 you to the original papers cited in the preceding paragraph. In short, these models are
 15664 based on the joint likelihood of two model components: one describing the resighting
 15665 process of marked individuals and one describing the number of unmarked individuals
 15666 observed. The resighting process of marked individuals can use either a Poisson or a
 15667 Bernoulli observation model, depending on whether sampling is with or without replace-
 15668 ment, and the resighting probabilities can have both fixed effects to model individual
 15669 and environmental covariates, and a random-effect component to accommodate variation
 15670 in detection due to individual heterogeneity. The process describing the number of un-
 15671 marked individuals observed (or, under a Poisson observation model, the number of times
 15672 unmarked individuals are observed), n_t (t here and in the following description denotes
 15673 a primary sampling occasion, for example, a year or a season) is approximated as a nor-
 15674 mal distribution (McClintock et al., 2006), or a normal distribution left-truncated at 0
 15675 (McClintock et al., 2009a):

$$n_t \sim \text{Normal}(\mathbb{E}(n_t), \text{Var}(n_t)).$$

15676 For a single-season study, the t subscript does not need to be included. Although this
 15677 is a simplification of the actual sampling process, McClintock et al. (2006) found this

normal distribution to be a satisfactory approximation, which allows N to enter the model likelihood via $\mathbb{E}(n_t)$ and $\text{Var}(n_t)$.

In the simplest model without any variation in detection, the expected number of resightings of unmarked individuals, $\mathbb{E}(n_t)$, can be written as the number of unmarked individuals times the expected number of detections of a single individual. This is the mean or expected value of the underlying observation model:

$$\mathbb{E}(n_t) = (N - m) * \theta \quad (19.1.1)$$

where $\theta = K \times p$ for a Binomial observation model with K replicates and individual detection probability p , or $\theta = \text{expected individual encounter rate } \lambda$ for a Poisson observation model. Similarly, $\text{Var}(n_t)$ depends on the underlying observation model and is based on the parameters that determine the individual detection probability/encounter rate. Combining these two components, N is directly incorporated into the joint likelihood of the model.

While these mark-resight models are very flexible, they share the shortcomings of traditional capture-recapture models when it comes to estimating population density (e.g., Chaps. 1 and 4). As long as resightings are collected across a number of locations, however, they come with the same spatial information as (re)captures in a standard SCR study. In the following sections we will consider mark-resight sampling in the framework of spatial capture-recapture.

19.1.4 The random sample assumption

In mark-resight studies it is a prerequisite that the marked portion of the studied population is a random sample of the population, so that detection probability for the population can be adequately estimated from the marked subset. If, for example, there is some latent group structure in the population where one group has a higher detection probability than the other, the marked portion of the population should have the same composition with regard to this group structure as the study population. Intuitively, people think of this as a demographic problem. But if you think back to Chapt. 1 and one of the motivations for the development of SCR models, this assumption also has spatial implications. In a non-spatial mark-resight study, if all the marked individuals live on the edge of the resighting array, their exposure to resighting will be lower compared to the exposure of unmarked individuals living in the center of the array, thus artificially deflating estimates of detection. So to obtain a truly random sample of the study population, the *locations of the home ranges* of the marked individuals also have to be a random sample of the home range locations of the entire population. In general, this will be difficult to assess or even to incorporate into study design or analysis, unless the spatial context of sampling is clearly defined. Thus, mark-resight models are, fundamentally, *spatial*.

In the SMR framework, this issue manifests itself more explicitly, for two reasons: (1) we define the spatial context of the population by setting a state-space; (2) we assume a certain distribution or point process for all individuals within that state-space, in most cases a uniform distribution or homogeneous point process (but see Chapt. 11 for models with inhomogeneous spatial point processes). For the marked individuals to follow a homogeneous point process (i.e., be a random spatial subset of the population) in \mathcal{S} , marking must be done uniformly throughout the state-space. When we study a species

15720 where some individuals can be identified based on natural marks, while others do not have
15721 unique marks (for example regular colored versus melanistic leopards), and we can assume
15722 that the distribution of these two groups of individuals across \mathcal{S} are identical, then we
15723 can frame the estimation problem in terms of estimating the density of two homogeneous
15724 point processes, one for the marked and one for the unmarked population. But what if
15725 we actively need to mark individuals in order to distinguish them? Then, we have two
15726 options: (1) if we want to meet the random sample assumption, then the definition of
15727 \mathcal{S} becomes part of our study design (contrary to SCR models, where \mathcal{S} is set after data
15728 collection for analysis purposes); (2) if we don't want to, or cannot, meet the random
15729 sample assumption, we have to specify an alternative model that adequately describes the
15730 distribution of marked and unmarked individuals in \mathcal{S} .

15731 Here is another way to think about this: In SCR models, once the state-space is chosen
15732 large enough, estimates of density are no longer sensitive to the size of \mathcal{S} , because N scales
15733 with the area of \mathcal{S} . In spatial mark-resight, however, our population of individuals consists
15734 of two groups, marked and unmarked. Consider the case where we have a known number
15735 of marks. Because we fix the size of the marked part of our population, total population
15736 size N no longer scales with the area of the state-space. While the number of unmarked
15737 individuals can go up as \mathcal{S} increases in size, m is fixed by design, and thus, as \mathcal{S} increases,
15738 overall density will decrease.

15739 If we want to make sure by design that marked individuals are a random sample from
15740 \mathcal{S} , then, in practical terms, we need to define the state-space, which includes the resighting
15741 array plus a sufficient buffer to include all animals potentially exposed to this array, and
15742 uniformly mark individuals throughout \mathcal{S} . This does not mean that we necessarily have
15743 to achieve complete coverage of \mathcal{S} with our marking effort; alternatively, we could also
15744 randomly distribute traps in \mathcal{S} in order to randomly distribute marks throughout \mathcal{S} . We
15745 can see some sampling situations in which such a scenario might be reasonable, or at least
15746 reasonably approximated. For example, later on in this chapter we present a study where
15747 raccoons were caught and marked throughout an island, the boundaries of which are a
15748 natural limit for the state-space of this particular system.

15749 For many studies, however, this might not be the case. Often, marking is the more
15750 difficult and logistically challenging part of a mark-resight study – think about the diffi-
15751 culty in physically capturing and tagging large carnivores. Especially for rare and cryptic
15752 species, areas over which resighting is conducted might have to be large to accumulate
15753 sufficient data, and marking over an even larger area – \mathcal{S} – would be logically impossi-
15754 ble. So what happens if we capture and mark individuals in a subset of the state-space?
15755 Then, whereas we may well have an overall constant density across \mathcal{S} , we will have a
15756 higher density of marked individuals in the vicinity of the marking locations – live traps,
15757 mist nets, whatever is used to catch animals – and the density of marks will generally go
15758 down as we get further away from the marking locations. As with all methods discussed
15759 in this book, the marking process of mark-resight studies also has a spatial component
15760 and induces a certain spatial distribution of marked individuals in the study area. We
15761 have to account for that when developing an SMR model. Thus, if we want to relax the
15762 assumption that marked animals are a random sample from \mathcal{S} , we need to describe the
15763 distribution of marked individuals' activity centers using an adequate spatial point process
15764 model. Developing a suitable point process model is one of the primary challenges when
15765 fitting SMR models, and one that at this point in time still requires substantial model

development efforts. We provide some ideas on how to approach this problem in the last section of this chapter.

Although it might not be a reasonable assumption for many real life survey situations, for now, we will continue by showing the development of SMR models assuming that the marked animals are, indeed, a random sample of N , following a homogeneous point process in \mathcal{S} . This simplifies the modeling problem substantially, thereby allowing us to focus on the underlying principles and possible useful extensions of SMR models.

19.2 KNOWN NUMBER OF MARKED INDIVIDUALS

We begin the model development with the simplest situation. Here, a known number of individuals constituting a random sample from the population within \mathcal{S} are marked and a series of resight samples are conducted following marking. No marks (or marked animals) are lost between marking and resighting, all individuals are correctly identified as marked or unmarked, and marked individuals are 100% correctly identified to individual level.

Recall from Chapt. 18 that without any individual identity, the observed counts at resighting location j and occasion k , n_{jk} , represent the sum of all latent individual detections at j and k , $\sum_{i=1}^N y_{ijk}$, where y_{ijk} are the latent individual encounter histories.

We can model these counts as

$$n_{jk} \sim \text{Poisson}(\Lambda_j)$$

where

$$\Lambda_j = \sum_{i=1}^M (\lambda_{ij}).$$

Under this formulation, in order to carry out MCMC, we do not need to update the individual y_{ijk} in our model, which is more efficient in terms of computing. However, we can also formulate the model as conditional on the latent y_{ijk} . This is useful because if we have m marked animals in our study population, then y_{ijk} for those m individuals are no longer latent, but fully observed and can easily be included in the analysis to provide information on detection parameters.

The formulation conditional on y_{ijk} basically brings us back to the original SCR model, where individual site and occasion specific counts, y_{ijk} , are modeled as

$$y_{ijk} \sim \text{Poisson}(\lambda_{ij})$$

and

$$\lambda_{ij} = \lambda_0 \exp(-d_{ij}^2 / (2\sigma^2)).$$

in the case of a Poisson encounter model. Unobserved y_{ijk} are treated as missing data and have to be updated as part of the MCMC procedure. We can do that by using their full conditional distribution, which is multinomial with sample size n_{jk} . Let \mathbf{u} be an index vector of the $M - m$ hypothetical unmarked individuals, $u = m + 1, m + 2, \dots, M$, and let \mathbf{y}_{ujk} be the vector of observations of all individuals in \mathbf{u} at j and k . Then

$$\mathbf{y}_{ujk} \sim \text{Multinomial}(n_{jk}, \lambda_{uj})$$

15797 Whereas in the non-spatial mark-resight analysis, known individuals provide information
 15798 about individual detection probability (or rate), in the spatial setting they also inform
 15799 σ , as described in Chapt. 18. Including known individuals into the analysis helps estimate
 15800 model parameters more accurately and precisely. We will address the relationship between
 15801 the number of marked individuals and accuracy of the estimated parameters in Sec. 19.5.

15802 19.2.1 Implementing spatial mark-resight models

15803 Implementing a spatial mark-resight model in **JAGS** is not straightforward, since the
 15804 program does not accept partially observed multivariate nodes (in this case the partially
 15805 observed individual encounter histories which we model as coming from a multinomial
 15806 distribution). We can, however, work around that by separating the marked from the
 15807 unmarked data. The **JAGS** code for the model with a known number of marked indi-
 15808 viduals is shown in Panel 19.1. You see that data augmentation is only applied to the
 15809 unmarked part of the population, and N is the sum of the estimated number of unmarked
 15810 individuals (`sum(z[])`) and the number of marked individuals, which is known. Also, to
 15811 reduce run time, we summed observations of marked individuals across occasions and ac-
 15812 count for that by multiplying λ_{ij} with K . Although the two data sets are separated, both
 15813 parts of the population, marked and unmarked, have the same prior uniform distribution
 15814 of activity centers. A last noteworthy detail in this code is the `dsum()` distribution. This
 15815 distribution is specific to **JAGS** (i.e., you cannot run this model in **BUGS**), and allows
 15816 you to impose a sum constraint on observations. In other words, it allows to model data –
 15817 here the counts of unmarked individuals, n_{jk} – as the sum of a number of latent variables,
 15818 which in this case are the latent encounter histories of unmarked individuals. While it
 15819 can be a pain writing out all the arguments of `dsum()`, it is this function that allows us
 15820 to implement SMR models in **JAGS**.

15821 Alternatively, we can use the technical concepts presented in Chapt. 17 and derive our
 15822 own MCMC algorithm. To do so, we only have to make relatively simple modifications to
 15823 the MCMC code developed for regular SCR models in Chapt. 17. Essentially, since we
 15824 observe individual detections for the marked part of the population, we have to update
 15825 only the unobserved part of the full – augmented – set of encounter histories, \mathbf{Y} , and
 15826 modify the updating steps for z_i and ψ , the parameters introduced by data augmentation,
 15827 to reflect that these only apply to the unmarked part of the population, in other words, to
 15828 the $M - m$ individuals in our data. You can find the full MCMC code in the accompanying
 15829 **R** package `scrbook` by invoking `scrPID`. The **R** code below shows how to simulate SMR
 15830 data using the `scrbook` function `sim.pID.data`, and running an SMR model on the data,
 15831 both in **JAGS** and using `scrPID`. The model file `mknown.jag` in the `jags.model` call should
 15832 contain the code from Panel 19.1.

```
15833 > set.seed(2013)
15834 > N = 80 # pop. size
15835 > m <- 45 # no. marked
15836 > sigma = 0.5
15837 > lam0 = 0.5
15838 > K = 5
15839 # Make resighting array
15840 > gx <- gy <- seq(0,6,1)
```

```

model{

#priors
psi ~ dbeta(1,1)
lam0 ~ dunif(0, 5)
sigma ~ dunif(0, 5)

#marked part
for(i in 1:m) {
  sm[i,1] ~ dunif(xlim[1], xlim[2])
  sm[i,2] ~ dunif(ylim[1], ylim[2])
  for(j in 1:J) {
    distm[i,j] <- sqrt((sm[i,1]-X[j,1])^2 + (sm[i,2]-X[j,2])^2)
    lambdam[i,j] <- lam0*exp(-distm[i,j]^2/(2*sigma^2))
    y[i,j]~dpois(lambdam[i,j]*K)
  }
}

##unmarked part
for(i in 1:M) {
  z[i] ~ dbern(psi)
  s[i,1] ~ dunif(xlim[1], xlim[2])
  s[i,2] ~ dunif(ylim[1], ylim[2])
  for(j in 1:J) {
    dist[i,j] <- sqrt((s[i,1]-X[j,1])^2 + (s[i,2]-X[j,2])^2)
    lambda[i,j] <- lam0*exp(-dist[i,j]^2/(2*sigma^2))
    for(k in 1:K) {
      yu[i,j,k] ~ dpois(lambda[i,j]*z[i])
    }
  }
}

for(j in 1:J) {
  for(k in 1:K) {nU[j,k] ~ dsum(yu[1,j,k],yu[2,j,k],yu[3,j,k],
  [...code shortened...],
  yu[79,j,k],yu[80,j,k])
}
}

N <- sum(z[])+m
}

```

Panel 19.1: **JAGS** model specification for SMR model with known number of marked individuals. In this example, M , the size of the augmented unmarked data set, is 80. Note that the arguments $yu[4,j,k]$ to $yu[78,j,k]$ of the `dsum()` function are omitted from the code.

```

15841 > X <- as.matrix(expand.grid(gx, gy))
15842 > J = dim(X)[1]
15843 > # Limits of S
15844 > xlims <- ylims<-c(-1.5, 7.5)
15845 # Simulate data
15846 > dat = sim.pID.data(N=N, K=K, sigma=sigma, lam0=lam0, knownID=m, X=X, xlims=xlims,
15847 ylims=ylims, obsmod='pois', nmarked='known')
15848
15849 ##### Prep data for analysis in JAGS
15850 > n <- dat$n - apply(dat$Yknown, 2:3, sum)
15851 > y <- apply(dat$Yknown, 1:2, sum)
15852
15853 > M <- 80 # Augmentation only for unmarked
15854
15855 # Initial values for latent y
15856 > yin <- array(0, c(M,J,K))
15857 > for(j in 1:J){
15858 + for(k in 1:K){
15859 + yin[1:M,j,k] <- rmultinom(1, n[j,k], rep(1/M, M))
15860 + }
15861
15862 > data <- list(y=y, nU=n, m=m, M=M, J=J, X=X, xlim=xlims, ylim=ylims, K=K)
15863 > inits <- function(){list(sigma=runif(1), lam0=runif(1),
15864 sm=cbind(runif(m, xlims[1], xlims[2]), runif(m, ylims[1], ylims[2])),
15865 s=cbind(runif(M, xlims[1], xlims[2]), runif(M, ylims[1], ylims[2])),
15866 z=rep(1, M), yu=yin)}
15867 > params <- c('lam0', 'sigma', 'N', 'psi')
15868
15869 # Analysis in JAGS
15870 > library(rjags)
15871 > mod <- jags.model('mknown.jag', data, inits, n.chains=1, n.adapt=800)
15872 > out <- coda.samples(mod, params, n.iter=5000)
15873
15874 > # Analysis with scrbook MCMC code
15875 > library(scrbook)
15876 > library(coda)
15877 > inits2 <- function(){list(psi=runif(1), sigma=0.5, lam0=0.5,
15878 S=cbind(runif(M+m, xlims[1], xlims[2]), runif(M+m, ylims[1], ylims[2]))}
15879 > out2 <- scrPID(n=n, X=X, y=dat$Yknown, M=M+m, obsmod = "pois", niters=5800,
15880 xlims=xlims, ylims=ylims, inits=inits2(), delta=c(0.1,0.1,0.5))


```

15881 You can look at the two sets of output by invoking `summary(out)` for the **JAGS** analysis and `summary(window(mcmc(out2), start=801))` for the custom MCMC algorithm, excluding the first 800 iterations as burn-in. We summarized the results in Table 19.2.1. 15883 The posterior mean of N is slightly higher than the data-generating value of $N = 80$, 15884 but it falls comfortably within the credible intervals. As expected, estimates from both 15885 implementations are very similar; slight differences are probably the result of Monte Carlo 15886

Table 19.1. Posterior summaries of the parameters of a spatial mark-resight model with known number of marks, analyzed in **JAGS** and using **scrPID**.

Implementation	Parameter	Mean	SD	2.5%	50%	97.5%
JAGS	N	88.72	6.75	77	88	103
	λ_0	0.53	0.08	0.39	0.53	70
	σ	1.29	0.02	1.26	1.30	1.32
	ψ	0.47	0.03	0.45	0.47	0.53
scrPID	N	86.01	7.58	73	85	102
	λ_0	0.54	0.08	0.39	0.53	0.72
	σ	0.48	0.03	0.42	0.48	0.53
	ψ	0.51	0.11	0.32	0.51	0.73

15887 error due to the relatively low number of iterations. You will find that sometimes, **JAGS**
 15888 produces an error message upon trying to compile the model, saying that some of the
 15889 observed y are inconsistent with parent nodes at initialization. We have mentioned before
 15890 that **JAGS** cannot always auto-generate acceptable initial values, and we believe this is
 15891 what is happening here. If this error occurs, just repeat the `jags.model` command, usually,
 15892 model compilation is successful on a second attempt (assuming, of course, that you
 15893 followed the code above correctly). We further find that the custom MCMC algorithm
 15894 tends to be faster than **JAGS**, which is why the examples and simulation studies shown
 15895 in the following sections were run solely in **R**.

19.3 UNKNOWN NUMBER OF MARKED INDIVIDUALS

15896 Now let us consider the case where we do *not* know the exact number of marked individuals
 15897 available for resighting so that we have to capture an individual at least once to be sure that
 15898 it is available. Unless we have a direct means of confirming the number of marked animals
 15899 available for resighting, treating this number as unknown is probably more realistic in most
 15900 circumstances. As a consequence of not knowing the exact number of marked individuals,
 15901 we cannot observe all-zero encounter histories. When using maximum likelihood inference,
 15902 this situation requires a model where detection rates of known individuals are modeled
 15903 using a zero-truncated distribution (McClintock et al., 2009a). If we did not account
 15904 for the fact that zeros are unobservable, estimates of detection rates would be artificially
 15905 inflated and estimates of population size would be negatively biased.

15906 Working with zero-truncated distributions in a spatial mark-resight setting is less
 15907 straight-forward than for non-spatial mark-resight. A marked individual only has to show
 15908 up once, anywhere on the resighting array, for us to know that it is there. When resightings
 15909 are pooled across the entire sampling grid, then the total individual counts $\sum_j y_{ijk}$ have
 15910 to be > 0 for all resighted individuals and a zero-truncated distribution can be used to
 15911 model these counts. However, we are concerned with trap-specific encounters, y_{ijk} , which
 15912 can easily be 0 for a resighted individual, as long as a single y_{ij} is > 0 . Thus, the zero-
 15913 truncation does not apply to the individual and trap specific counts we observe, but only
 15914 to the sum of these counts over all traps.

15915 As an alternative to a zero-truncated distribution, in a Bayesian framework, we can
 15916 make use of data augmentation to estimate the number of marked individuals (McClintock

and Hoeting, 2010). In the SMR framework that means that we create two augmented data sets, one for the marked individuals and one for the unmarked, and estimate their number separately, having them share the parameters of the detection model. Sometimes we may know the maximum number that were ever marked before a resighting survey, in which case we can use that number as the data augmentation limit for the marked data set. Panel 19.2 shows the **JAGS** code for the SMR model with unknown number of marks, which is identical to the one in Panel 19.1, but for the augmentation of the marked data set. This introduces both a data augmentation parameter, `psim`, and an auxiliary “alive state” variable, `zm[i]`, into the description of the marked data model. Again, we provide an alternative, **R**-only MCMC algorithm within `scrbook - scrPID.um`.

Note that we could look at the problem of not knowing the number of marked individuals in the study population as a manifestation of a lack of population closure. In other words, marked individuals may have emigrated, died or lost their marks in the time between marking and resighting. If we have information on the rates of these events, or a series of resighting surveys, we could develop an open population model for the marks in our population and estimate their number at a given resighting survey in this fashion. This kind of SMR model remains to be explored.

19.3.1 Canada geese in North Carolina

We applied the spatial mark-resight model with an unknown number of marks and a binomial encounter process to a dataset of Canada goose resightings (Rutledge, 2012). During the molt of 2008, 751 individual geese were captured and marked with neck and leg bands in Greensboro, North Carolina (Fig. 19.1). Geese were resighted at 87 locations on 81 resighting events over a period of 18 months. In addition to the banded geese, the number of unmarked geese was recorded during each resighting event. Here, we only looked at a subset of the data, from mid July to the end of October 2008, which corresponds to the first part of the post-molt season, before migratory Canada geese arrive in North Carolina. We treated this population as closed over this period. During this part of the study, 57 of the resighting sites were visited and $n = 654$ marked geese were resighted 3994 times at 40 different sites. In addition, 7944 sightings of unmarked geese were recorded at 48 sites.

In the model, we allowed σ to vary between males and females. We set the size of the augmented unmarked data set to 7000. We used the total number of marked geese (751) as the upper limit for the augmented marked data set. We ran 50000 MCMC iterations and removed a burn-in of 5000 iterations. To describe the state-space, we buffered the resighting locations by 4.5 km. We assumed that marked geese were a random sample from the state-space, which seems reasonable because (a) marking took place across most of the extent of the resighting array; and (b) marking was done during the molting period, when geese are fairly immobile, and it seems reasonable to assume that, once the molt is complete, the marked geese redistributed themselves. Under this model formulation, estimates of density will be sensitive to the choice of the state-space (19.1.4), but the particular state-space seems like a reasonable choice for this problem. We provide all the data (`data('geesedata')`) and functions (`geeseSMR`) for you to repeat this analysis but be aware that given the large data set it will take days to do so. The **R** code to set up the data and run 5000 iterations of the model for the geese data is given as an example on

```

model{

# Prior distributions
psim ~ dbeta(1,1)
psi ~ dbeta(1,1)
lam0 ~ dunif(0, 5)
sigma ~ dunif(0, 5)

# Marked part of the model
for(i in 1:max) {
  zm[i]~dbern(psim)
  sm[i,1] ~ dunif(xlim[1], xlim[2])
  sm[i,2] ~ dunif(ylim[1], ylim[2])
  for(j in 1:J) {
    distm[i,j] <- sqrt((sm[i,1]-X[j,1])^2 + (sm[i,2]-X[j,2])^2)
    lambdam[i,j] <- lam0*exp(-distm[i,j]^2/(2*sigma^2))*zm[i]
    y[i,j]~dpois(lambdam[i,j]*K*zm[i])
  }
}

# Unmarked part of the model
for(i in 1:M) {
  z[i] ~ dbern(psi)
  s[i,1] ~ dunif(xlim[1], xlim[2])
  s[i,2] ~ dunif(ylim[1], ylim[2])
  for(j in 1:J) {
    dist[i,j] <- sqrt((s[i,1]-X[j,1])^2 + (s[i,2]-X[j,2])^2)
    lambda[i,j] <- lam0*exp(-dist[i,j]^2/(2*sigma^2))
    for(k in 1:K) {
      yu[i,j,k] ~ dpois(lambda[i,j]*z[i])
    }
  }
}

for(j in 1:J) {
  for(k in 1:K) {nU[j,k] ~ dsum(yu[1,j,k],yu[2,j,k],yu[3,j,k],
  [...code shortened...],
  yu[79,j,k],yu[80,j,k])
}
}

Nu <- sum(z[])
Nm<-sum(zm[])
N<-Nu+Nm
}

```

Panel 19.2: JAGS model specification for SMR model with unknown number of marked individuals. In this example, M , the size of the augmented unmarked data set, is 80. Note that the arguments $yu[4,j,k]$ to $yu[78,j,k]$ of the `dsum()` function are omitted from the code for space reasons.

Table 19.2. Posterior summaries of parameters of the spatial mark-resight model for Canada geese in North Carolina. N is the total population size of marked and unmarked individuals; m is the number of marked individuals.

	Mean	SD	2.5%	50%	97.5%
m	739.77	3.24	733	740	746
N	5756.10	90.68	5577	5757	5932
D	13.76	0.19	13.38	13.76	14.14
λ_0	0.19	<0.01	0.18	0.19	0.19
σ , females	1.29	0.02	1.26	1.30	1.32
σ , males	1.06	0.02	1.02	1.06	1.11
ψ , marked	0.99	<0.01	0.98	0.99	0.99
ψ , unmarked	0.72	0.01	0.69	0.72	0.74
ϕ	0.36	0.02	0.32	0.36	0.39

the help page for `geeseSMR`. The model results, including the derived parameter density (D) in individuals per km^2 are shown in Table 19.3.1.

We see that credible intervals of estimates are pretty narrow, surely an effect of the large data set. Estimates of m indicate that most of the 751 geese originally banded are still alive and marked, which is not surprising, given that not much time passed between marking and this first resighting session. The parameter ϕ in this model is the probability of being a male, a measure of the sex ratio of the population, which is slightly biased in favor of females.

19.4 IMPERFECT IDENTIFICATION OF MARKED INDIVIDUALS

Often during resighting, it may be possible to see that an individual is marked but impossible to determine its individual identity. In this situation, in addition to y_{ijk} and n_{jk} , we also have site and occasion specific counts of marked but unidentified individuals, r_{jk} . Here, the individual encounter histories of marked animals are incomplete, and if we used these incomplete data to inform the detection parameter of the model, we would run the risk of underestimating encounter rate and overestimating abundance. Some non-spatial mark-resight models do not require that marked animals be identified individually, as long as the marking status can be observed unambiguously, but ignoring individual level information means that we cannot accommodate heterogeneity in detection (McClintock and White, 2012). In a spatial framework we could ignore marked and unmarked status completely and apply the model by Chandler and Royle (In press) discussed in Chapt. 18. But, that would mean losing important information on individual detection and movement. Therefore, being able to retain the individual identity of records that can be identified while at the same time accounting for imperfect identification of marked individuals is extremely useful.

McClintock et al. (2009a,b) suggest an intuitive means of correcting for this bias in a non-spatial model framework when dealing with a Poisson encounter model (a plausible model when sampling with replacement). When marked but unknown resightings are part of the data, the expected number of records that cannot be identified to the individual

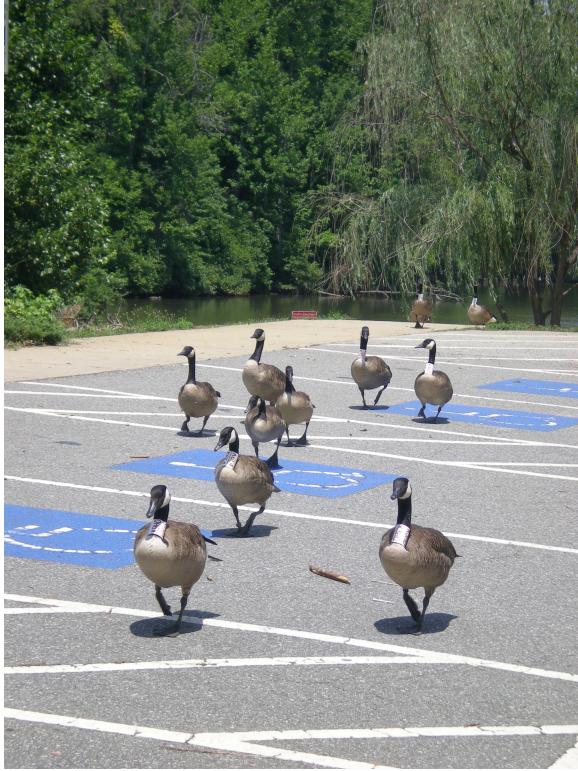


Figure 19.1. Banded and unbanded Canada geese in a parking lot in Greensboro, North Carolina. (Photo credit: M.E. Rutledge, NCSU Canada Goose Project)

15988 level, n , changes from Eq. 19.1.3 to:

$$\mathbb{E}(n) = (N - m)\lambda + \eta/m$$

15989 where λ is the individual encounter rate estimated from the known resighted individu-
 15990 als and η is the number of records of marked but unidentified individuals. So, because
 15991 the observed λ is known to be too low, the average number of unidentified pictures per
 15992 known individual is added as a correction factor. Note that here, n includes both records
 15993 of unmarked individuals and records of marked individuals that could not be identified.
 15994 This procedure assumes that the inability to identify a marked individual occurs at ran-
 15995 dom throughout the population, which seems to be a reasonable assumption under most
 15996 circumstances.

15997 We can translate this same concept to the spatial mark-resight models. In the spatial
 15998 framework we are interested in the individual and trap specific encounter rate, λ_{ij} . Fur-
 15999 ther, we do not look at the sum of all records of unmarked individuals, but formulate the
 16000 model conditional on the latent individual encounter histories. Thus, instead of using η/m

as a correction factor, we need something that applies at the individual and trap level. If we take the sum of all correctly identified records of marked individuals, $\sum y_c$ and divide it by the total number of records of marked individuals, $\sum y_m$, we get the average rate of correct individual identification for marked individuals, say, c :

$$c = \sum y_c / \sum y_m.$$

We can then apply c as a correction factor for λ_0 for the marked individuals.

A more formal, model-based way to specify c is by assuming that

$$\sum y_c \sim \text{Binomial}(\sum y_m, c)$$

and estimating c as another model parameter, so that we account for the uncertainty about it. For the marked individuals we can then multiply λ_0 by c to account for the fact that we observe incomplete individual encounter histories. Since we don't have this identification issue for unmarked individuals, their baseline trap encounter rate remains as before simply λ_0 (or in other words, c for unmarked individuals equals 1).

Incomplete individual identification of marked individuals is easily incorporated into our **JAGS** model, no matter whether m known or unknown, by adding the following two lines of code:

```
16015  c ~ dbeta(1,1) #prior for c
16016  npics[1] ~ dbin(c, npics[2]) #model for c
16017  and modifying the marked observation model description to
16018  y[i,j] ~ dpois(lambdam[i,j]*c*K)
```

Here, the data object **npics** is a vector with the number of correctly identified records of marked individuals and the total number of marked records. Accounting for imperfect identification of marks is also included as an option in the **scrPID** and **scrPID.um** functions. Choosing an uninformative (and conjugate) beta(1, 1) prior for c , within the **scrPID** algorithm we can update c directly from its full conditional distribution, which is $\text{beta}(1 + \sum y_c, 1 + (\sum y_m - \sum y_c))$. We show an example of using c in an analysis in Sec. 19.6.

Observe that now, in addition to assuming that failure to identify marked individuals occurs at random throughout the population, we also assume that it occurs at random throughout space, i.e. our success of identifying a marked individual does not depend on the trap we encounter it in. As long as individuals are identified based on the same type of tags the assumption that failure to identify marked individuals occurs at random throughout the population should be valid. The assumption that failure to identify marked individuals occurs at random in space could be violated, for example when spatially varying habitat conditions influence the ability to recognize individual tags, or when an observer effect influences individual identification rates. While we haven't experimented with it, we believe that the approach described above could readily be extended to account for these differences. For example, identification rates could be calculated separately for different observers, or be modeled as functions of habitat covariates. As an alternative to the approach we present here, model development could explore assigning records of marked

16039 but unidentified individuals to marked individuals in a fashion similar to how unmarked
 16040 records are assigned to hypothetical individuals in this model, namely, based on the loca-
 16041 tion of the record and the estimates of home range centers of marked individuals. While
 16042 this is computationally more advanced it would make full use of the spatial information
 16043 of the unmarked records.

19.5 HOW MUCH INFORMATION DO MARKED AND UNMARKED INDIVIDUALS CONTRIBUTE?

16044 It is intuitive that having marked individuals in the study population should lead to more
 16045 accurate and precise parameter estimates than when no individuals are identifiable. To
 16046 evaluate how strongly adding marked individuals to a population improves parameter
 16047 estimates, Chandler and Royle (In press) performed a simulation study. They used a $15 \times$
 16048 15 resighting grid and simulated detection data of $N = 75$ individuals in a 20×20 units
 16049 state-space over $K = 5$ occasions with $\sigma = 0.5$ and $\lambda_0 = 0.5$. They generated 100 datasets
 16050 each for $m = (0, 5, 15, 25, 35)$ where m is the known number of marked individuals
 16051 randomly sampled from the population.

16052 Without any marked individuals in the population, the posterior distribution of N
 16053 turned out to be highly skewed, but the mode was still an approximately (frequentist)
 16054 unbiased point estimator of N . As anticipated, posterior precision increased substantially
 16055 with the proportion of marked individuals (Table 19.3 and Fig. 19.2). The relative root-
 16056 mean squared error decreased from 0.246 when no individuals were marked to 0.085 when
 16057 35 individuals were marked (Table 19.3). Coverage was nominal for all values of m and
 16058 posterior skew greatly diminished with increasing m (Table 19.3).

Table 19.3. Posterior mean, mode, and associated relative RMSE for simulations in which m of $N=75$ individuals were marked. One hundred simulations of each case were conducted. Table taken from Chandler and Royle (In press)

	Parameter	Mean	rRMSE	Mode	rRMSE	BCI
m=0	N	85.866	0.259	77.720	0.242	0.950
	λ_0	0.506	0.180	0.488	0.182	0.960
	σ	0.495	0.115	0.486	0.113	0.960
m=5	N	80.898	0.184	76.360	0.182	0.970
	λ_0	0.510	0.178	0.494	0.180	0.950
	σ	0.496	0.089	0.488	0.086	0.970
m=15	N	79.028	0.148	76.250	0.147	0.950
	λ_0	0.508	0.163	0.494	0.164	0.950
	σ	0.496	0.073	0.492	0.071	0.970
m=25	N	77.765	0.114	75.810	0.113	0.950
	λ_0	0.511	0.153	0.498	0.157	0.950
	σ	0.496	0.067	0.493	0.065	0.940
m=35	N	76.446	0.085	74.900	0.085	1.000
	λ_0	0.513	0.142	0.501	0.144	0.950
	σ	0.497	0.056	0.493	0.057	0.940

16059 As we saw in the previous chapter, the spatial correlation in unmarked counts can
 16060 be sufficient to obtain estimates of movement and detection parameters. However, only

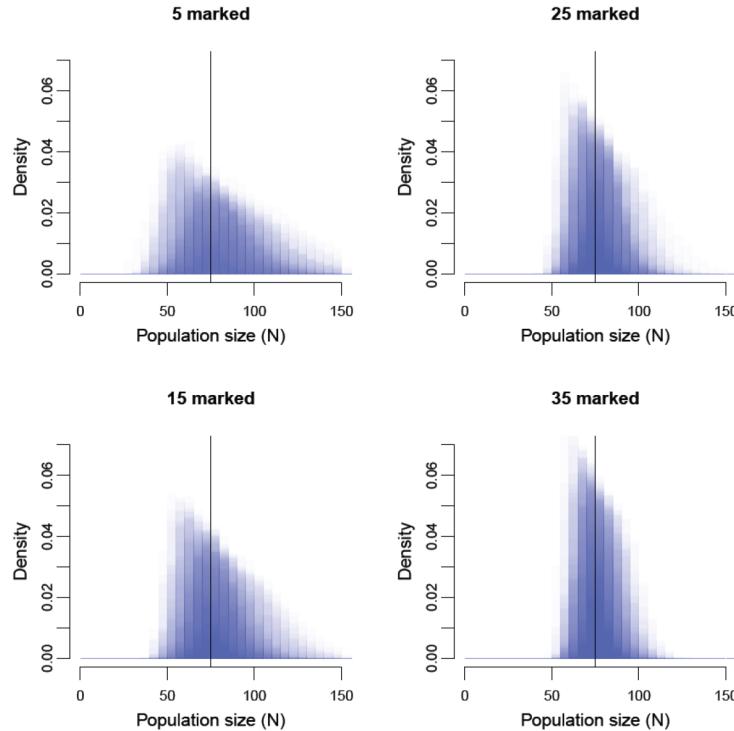


Figure 19.2. Overlaid posterior distributions of N from 100 simulations for four levels of marked individuals.

marked and thus identifiable individuals provide us with direct information about these parameters and may well dominate estimates. To single out the contribution of marked and unmarked individuals to parameter estimates, we re-ran the same simulations but let σ and λ_0 be updated based solely on the data of marked individuals. Results are summarized in Table 19.4. We see that if we update λ_0 and σ based on marked individuals only, estimates of these parameters are more biased and less precise. For estimates of N , especially for $m=5$ and $m=15$, we observe a stronger positive bias, lower accuracy and considerably lower BCI coverage as compared to when both marked and unmarked individuals contribute to parameter estimates (Table 19.4). Thus, unmarked individuals do actually contribute noticeably to estimating model parameters. This stands in contrast to non-spatial mark-resight models, where information about the detection parameter comes solely from the encounter histories of the marked individuals.

Table 19.4. Posterior mean, mode, and associated relative RMSE for simulations in which m of $N=75$ individuals were marked and unmarked individuals did not contribute to estimating λ_0 and σ . One hundred simulations of each case were conducted.

	Parameter	Mean	RMSE	Mode	RMSE	BCI
m=5	N	88.621	0.369	83.139	0.421	0.810
	λ_0	1.255	1.247	0.606	1.148	0.950
	σ	0.472	0.252	0.426	0.333	0.910
m=15	N	81.031	0.192	78.361	0.175	0.820
	λ_0	0.535	0.281	0.476	0.284	0.970
	σ	0.503	0.109	0.490	0.107	0.940
m=25	N	78.206	0.129	76.594	0.123	0.920
	λ_0	0.531	0.204	0.496	0.202	0.960
	σ	0.497	0.081	0.489	0.084	0.950
m=35	N	76.833	0.099	75.422	0.096	0.940
	λ_0	0.528	0.192	0.505	0.186	0.940
	σ	0.499	0.069	0.493	0.070	0.960

19.6 INCORPORATING TELEMETRY DATA

As we expected, parameter estimates of spatial mark-resight models get better the more marked individuals we have in our study population. While this is great advice in theory, it may not be very helpful in practice, especially when dealing with animals that are hard or somewhat dangerous to capture, such as large carnivores. Oftentimes, studies involving the physical capture of such animals will employ telemetry tags in order to learn about the study species' spatial ecology and behavior. In the context of spatial mark-resight models, the actual locations collected by telemetry tags can provide detailed information on individual location and movement, and being able to incorporate this information directly into the SMR model should improve estimates of these parameters, especially when resighting information is sparse.

So how could we combine resighting data and telemetry data in a unified mark-resight model? Recall that the basic SCR model underlying all the SMR models we discuss here uses a Gaussian kernel to describe the trap encounter model. By using this function, we can relate the parameters σ and \mathbf{s}_i directly to those from a bivariate normal model of space usage, with mean = \mathbf{s}_i , and variance-covariance matrix Σ , where the variance in both dimensions is σ^2 and the covariance is 0. Ordinarily, these parameters are estimated directly from the spatial distribution of individual captures/resightings. Telemetry data, however, provide more detailed information on individual location and movement, since the resolution and extent of the data are not limited by the trapping grid and potentially more locations can be accumulated through telemetry than resighting (depending on the monitoring frequency and resighting rates of individuals).

By assuming that the R_i locations of individual i , \mathbf{l}_i (consisting of a pair of x and y coordinates, l_{ix} and l_{iy}), are a bivariate normal (BVN) random variable:

$$\mathbf{l}_i \sim \text{BVN}(\mathbf{s}_i, \Sigma)$$

we can estimate σ as well as \mathbf{s}_i for the collared individuals directly from telemetry locations,

16097 using their full conditional distributions:

$$[\sigma | \mathbf{l}, \mathbf{s}] \propto \left\{ \prod_{i=1}^m \prod_{r=1}^{R_i} \frac{1}{2\pi\sigma^2} \exp \left(-1/2 \left[\frac{(l_{irx} - s_{ix})^2}{\sigma^2} + \frac{(l_{iry} - s_{iy})^2}{\sigma^2} \right] \right) \right\} * [\sigma]$$

16098 and

$$[\mathbf{s}_i | \mathbf{l}, \sigma] \propto \left\{ \prod_{r=1}^{R_i} \frac{1}{2\pi\sigma^2} \exp \left(-1/2 \left[\frac{(l_{irx} - s_{ix})^2}{\sigma^2} + \frac{(l_{iry} - s_{iy})^2}{\sigma^2} \right] \right) \right\} * [\mathbf{s}_i]$$

16099 For the unmarked individuals \mathbf{s}_i are estimated as described before conditional on their
 16100 latent encounter histories. Note that the bivariate normal model assumes that locations
 16101 are independent of each other. If you have frequent telemetry fixes, for example from
 16102 GPS collars that report animal locations every few hours or more, this assumption seems
 16103 unrealistic and it might be advisable to thin your telemetry data (maybe to daily fixes)
 16104 in order to approximate independence. Alternatively, movement models could be used
 16105 that acknowledge the temporal correlation in location data. We suggested some possible
 16106 movement models in Chapt. 15. Not all marked individuals need to be telemetry-tagged,
 16107 but telemetry data should correspond to the period over which resighting surveys were
 16108 conducted (as we discussed in Chapt. 5, both the \mathbf{s}_i and σ should only be interpreted
 16109 against the specific sampling period). Further, telemetry data need to be independent of
 16110 the resighting data.

16111 Again, implementation of this model extension is straight-forward, both in **JAGS** and
 16112 **R**. Take the SMR model description for the case where m is known (Panel 19.1). Then,
 16113 all we have to do is add a description of the bivariate normal model for the telemetry
 16114 locations, here `locs`, into the loop over the m marked individuals:

```
16115 [...parts of model code omitted...]
16116
16117 for(i in 1:m) {
16118
16119   sm[i,1] ~ dunif(xlim[1], xlim[2])
16120   sm[i,2] ~ dunif(ylim[1], ylim[2])
16121
16122   #telemetry model
16123   for (r in off1[i]:off2[i]){
16124     locs[r,1]~dnorm(sm[i,1], 1/(sigma^2))
16125     locs[r,2]~dnorm(sm[i,2], 1/(sigma^2))
16126   }
16127
16128   for(j in 1:J) {
16129     distm[i,j] <- sqrt((sm[i,1]-X[j,1])^2 + (sm[i,2]-X[j,2])^2)
16130     lambdam[i,j] <- lam0*exp(-distm[i,j]^2/(2*sigma^2))
16131     y[i,j]~dpois(lambdam[i,j]*K)
16132   }
16133
16134 [...parts of model code omitted...]
```

16136 The data object `locs` is a table with all $\sum_i^m R_i$ telemetry locations. The two vectors
 16137 `off1` and `off2` describe which subset of this matrix belongs to individual i . So if, say,
 16138 the locations for individual 1 are contained in the first 10 rows of `locs`, `off1` and `off2`
 16139 would be 1 and 10 for $i = 1$; and if the locations of individual 2 are in the following 15
 16140 rows, `off1` and `off2` for $i = 2$ would be 11 and 25, and so on. For the implementation of
 16141 this SMR model with telemetry data in **R**, see the `scrPID.tel` function in `scrbook`. In
 16142 a nutshell, in the MCMC algorithm we replaced the Metropolis-Hastings updating steps
 16143 for σ and activity centers of marked individuals, which were originally conditional on the
 16144 resighting data, with updating steps conditional on the telemetry data. This is not quite
 16145 what the above **JAGS** code does; rather **JAGS** will update these parameters conditional
 16146 on both the telemetry *and* the resighting data. We could easily re-write `scrPID.tel` to do
 16147 that, but believe that for most applications, the information on location and movement
 16148 contained in the telemetry data will outweigh that in the resighting data, so that the
 16149 resulting loss of information should be minimal.

16150 **19.6.1 Raccoons on the Outer Banks of North Carolina**

16151 Sollmann et al. (2013) applied a spatial mark-resight model with telemetry data to a
 16152 camera-trap and radio-telemetry data set from the raccoon population on South Core
 16153 Banks, a barrier island within Cape Lookout National Seashore, North Carolina. Between
 16154 May and September 2007, 131 raccoons were marked with dog collars and large individ-
 16155 ually numbered cattle tags. Individuals were marked throughout the island, so that (a)
 16156 we do not have to deal with sensitivity to choice of the state-space, because it is clearly
 16157 defined by nature; and (b) it is reasonable to assume that marked raccoons are a random
 16158 sample of individuals from this state-space. Of the 131 tagged individuals, 44 were also
 16159 equipped with radio collars. Collared individuals were located using a VHF receiver and
 16160 antenna, and their locations were estimated approximately weekly. Twenty camera traps
 16161 were set up along the length of South Core Banks and camera trapping data collected be-
 16162 tween October 1 2007 to January 22 2008 constituted the resighting data in this analysis.
 16163 During this period 104 marked individuals, 38 radio-collared, were alive and available for
 16164 resighting with camera traps.

16165 The state-space \mathcal{S} was the entire area of South Core Banks island. A change in the
 16166 number of photocaptures over the course of the study suggested a variation of detection
 16167 rate with time. Since date recording in cameras malfunctioned, photographic records
 16168 could only be assigned to the time interval between subsequent trap checks, and these
 16169 intervals between checks are referred to as sampling occasions. These occasions ranged
 16170 from 2 to 43 days; λ_0 was standardized to 7-day intervals and allowed to change with
 16171 sampling occasion. Since not all pictures of marked raccoons could be identified to the
 16172 individual level, the authors applied the correction factor c as described in Sec. 19.4,
 16173 estimated separately for each occasion.

16174 Camera-traps recorded 117 pictures of unmarked raccoons, 33 pictures of 18 marked
 16175 and identifiable raccoons, and 49 records of marked but not individually identifiable in-
 16176 dividuals (Fig. 19.3). An average of 16.32 telemetry locations (SD 4.91) were collected
 16177 for each of the 38 radio-collared individuals. Raccoon abundance on the island was esti-
 16178 mated at 186.71 (SE 14.81) individuals, which translated to a density of 8.29 (SE 0.66)
 16179 individuals per km². Parameter estimates are listed in Table 19.5.



Figure 19.3. Camera trap picture of a raccoon marked with a cattle tag that cannot be read to determine individual identity. Taken on South Core Banks, North Carolina. (Photo credit: Arielle Parsons)

16180 In this study, although a large number of raccoons were tagged, photographic data of
16181 these tagged individuals were surprisingly sparse. Analysis of the photographic data set
16182 without the telemetry data did not render usable estimates as parallel Markov chains did
16183 not converge. One reason for the relatively sparse data was the camera trap study design:
16184 traps were spaced on average 1.77 km apart, which is about 3.5 times σ . Consequently,
16185 very few individual raccoons were photographed at more than one trap. Under these
16186 circumstances, the telemetry data provide the necessary spatial information to estimate
16187 σ and the activity centers of individual animals and thus make other model parameter
16188 estimable. Similarly, in a camera-trapping study on Florida panthers (*Puma concolor*
16189 *coryi*), Sollmann et al. (in revision), including telemetry data from the 3 individuals
16190 that were collared and known to use the study area resulted in density estimates with
16191 considerably higher precision as compared to preliminary estimates *without* telemetry
16192 location data, reducing the width of the 95 % BCI by about 60 %. Such improvements
16193 in precision of estimates is especially important when we are interested in changes in the
16194 population over time.

Table 19.5. Summary statistics of posterior distributions from spatial mark-resight model for raccoon camera trapping and telemetry data, taken from (Sollmann et al., 2013). Baseline trap encounter rate λ_0 was standardized to 7-day intervals; λ_0 and the probability of identifying a picture of a marked individual, c , were allowed to vary among the 6 sampling occasions (t); σ is estimated from telemetry data of 38 radio-collared individuals.

	Mean (SE)	2.5%	50%	97.5%
N	186.71 (14.81)	162	185	220
D	8.29 (0.66)	7.19	8.22	9.77
λ_0 ($t=1$)	0.24 (0.05)	0.16	0.23	0.34
λ_0 ($t=2$)	0.40 (0.08)	0.26	0.39	0.57
λ_0 ($t=3$)	0.11 (0.03)	0.06	0.11	0.17
λ_0 ($t=4$)	0.30 (0.07)	0.17	0.29	0.46
λ_0 ($t=5$)	0.03 (0.01)	0.02	0.03	0.06
λ_0 ($t=6$)	0.03 (0.01)	0.02	0.03	0.05
σ	0.49 (0.01)	0.47	0.49	0.51
c ($t=1$)	0.55 (0.09)	0.38	0.55	0.71
c ($t=2$)	0.39 (0.11)	0.18	0.39	0.62
c ($t=3$)	0.29 (0.11)	0.11	0.29	0.52
c ($t=4$)	0.38 (0.16)	0.10	0.36	0.71
c ($t=5$)	0.38 (0.16)	0.10	0.36	0.71
c ($t=6$)	0.30 (0.14)	0.08	0.29	0.60

19.7 POINT PROCESS MODELS FOR MARKED INDIVIDUALS

As discussed in Sec. 19.1.4, all previously developed SMR models assume that marked individuals are a random sample, both spatially and demographically, from the population of the state-space. For many studies it may not be feasible to strive to meet or approximate the assumption of spatial randomness and it is thus important to generalize SMR models to situations where marking does not take place throughout \mathcal{S} . We already stated that in this situation, we generally cannot assume that activity centers of marked individuals (and unmarked, for that matter) follow a homogeneous point process. In this final section, we will describe two possible approaches to formulating such an inhomogeneous point process model. We will only provide conceptual descriptions, not a full-blown model development, as at the time of writing this book, these approaches are still somewhat experimental.

19.7.1 Homogeneous point process in a subset of \mathcal{S}

Imagine we perform an area search in a square, \mathcal{B} , for some species we want to study, maybe a reptile, and we mark all individuals we encounter. We conduct our sampling in a way that we can assume that marked individuals are randomly sampled within \mathcal{B} , and that there are no marked individuals with activity centers outside of \mathcal{B} . This design entails the assumption that \mathcal{B} can be clearly defined. We will come back to these assumptions in a minute. We then perform resighting surveys of some sort in an area that overlaps \mathcal{B} , so that, when we set a state-space around the resighting locations, \mathcal{B} is completely contained within \mathcal{S} (Fig. 19.4). We further assume that individuals that were marked in \mathcal{B} continue to live within \mathcal{B} when resighting surveys are conducted, i.e. their activity centers do not shift during the complete mark-resight study. This implies population closure across both

the marking and the resighting part of the study, and in this situation we can treat the number of marked animals, m , as known. Let the total population of \mathcal{B} be N_B . Under the conditions specified above, the number of marked animals m can be described as the outcome of a binomial random variable

$$m \sim \text{Binomial}(\theta, N_B)$$

where θ is the probability that an animal living in \mathcal{B} is marked. Under these conditions we can describe the point process for marked individuals as uniform across \mathcal{B} , with zero probability of a marked activity center being located outside of \mathcal{B} . If the combined (or marginal) point process of marked and unmarked individuals is homogeneous across \mathcal{S} , i.e. overall density is constant, then, colloquially speaking, the point process for unmarked individuals is the complement of the marked process: outside of \mathcal{B} , unmarked animals occur at the average density of \mathcal{S} , D , while inside \mathcal{B} they occur at $D * (1 - \theta)$.

The above model is an approach to specifying a spatial reference frame for marked individuals that is independent of \mathcal{S} . Some of the assumptions of the model, however, are reminiscent of traditional capture-recapture and thus, suffer from the same shortcomings. \mathcal{B} needs to be clearly defined as the area the marked individuals live in, but how do we define it? Imagine again that \mathcal{B} is a square search plot. Surely, we could capture an individual at the edge of the plot, whose activity center is located *off* that plot. Not accounting for this effect would overestimate density in \mathcal{B} . This is the equivalent of having to define an effective area sampled in traditional capture-recapture in order to estimate density. Further, we assume that θ , the probability of an individual within the plot being marked, is the same for all individuals in \mathcal{B} . But we discussed early on in this book that this is unlikely to be true, because exposure to sampling depends on an individual's home range overlap with the sampled area. So individuals near the edge of \mathcal{B} are less likely to be marked than those in the center, assuming we dispense marking effort uniformly across \mathcal{B} . In spite of the shortcomings of this approach, we believe it could serve as a reasonable approximation of some study systems. Moreover, it serves as a conceptual device because it presents a relatively simple way of thinking about two overlapping point processes, in the context of SMR the point processes describing the distribution of marked and unmarked individuals in \mathcal{S} .

19.7.2 Inhomogenous point processes

An alternative, and more realistic, point process for marked individuals is one that describes a decline in the density of marks with increasing distance to the marking location(s). We would expect this kind of spatial pattern in marks to arise, because animals living in the center of the marking grid have a higher probability to be marked than those living on the edges, which in turn have a higher marking probability than those living beyond the marking grid. As a consequence, the density of marks is higher in the center of the marking grid and decreases as we move away from it. Imagine that marking of animals takes place across some area or grid, and let C_m be the centroid of that marking area. Then, a plausible model for the distribution of marked animals is a bivariate normal model with mean C_m and variance-covariance matrix, Σ_C , where the variance in both dimensions is σ_C^2 and the covariance is 0 (Fig. 19.5). Of course, there are alternative

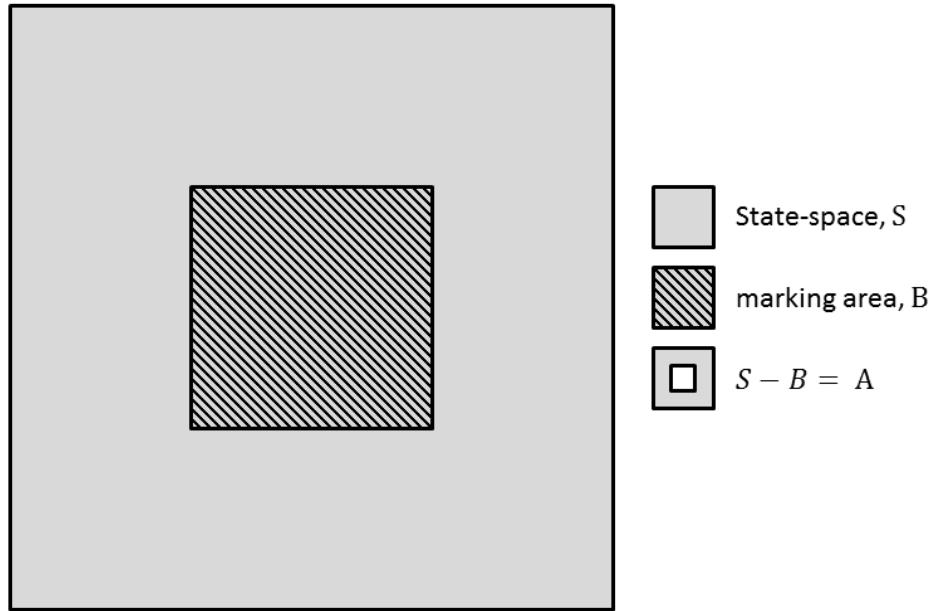


Figure 19.4. Relationship between marking area B and state-space, S .

models to describe a decrease in density, such as a negative-exponential or hazard function. Once the distribution of marked animals conditional on C_m is adequately described, the inhomogeneous point process for the unmarked animals can again be modeled by assuming that the marginal density across \mathcal{S} is constant, similar to the example above. The parameters of the inhomogeneous point process could also be modeled using the methods outlined in Chapt. 18. Density under such a model should be invariant to the size of \mathcal{S} , as is the case with standard SCR models, because the marked individuals are probabilistically constrained to live in the vicinity of the marking locations, no matter how large we choose \mathcal{S} .

While this model formulation is more realistic and general, it has some minor drawbacks. Based on very limited experience, the variance parameter, σ_C^2 , is barely estimable under realistic sample sizes. This is not surprising, given that this parameter attempts to describe the distribution of the activity centers, which are themselves latent. Further, the approach does entail some design constraints for marking. We implicitly assume that the centroid of the marking array is the area of highest probability of being marked, which means that marking effort has to be somewhat constant across the marking area. The assumption seems reasonable for systematic or random marking arrays (Fig. 19.5), but, for example, for a “hollow grid” the centroid would not be an adequate representation of the point of highest marking probability. The take-home message here is that an adequate

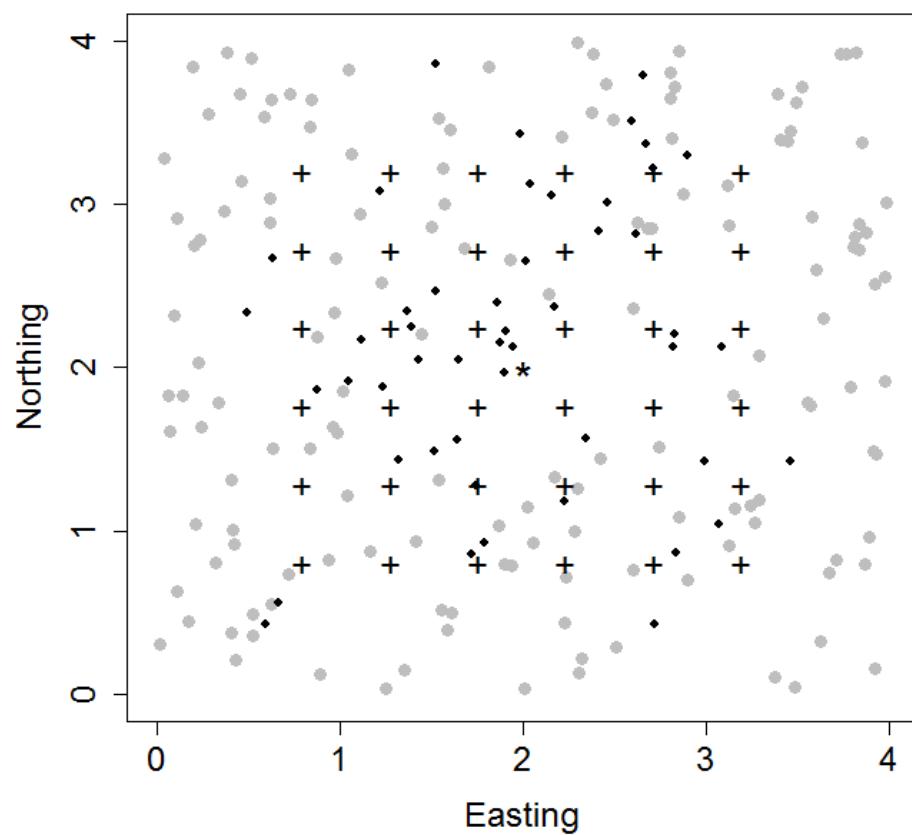


Figure 19.5. Plot of activity centers of marked (black dots) and unmarked (gray dots) individuals, with marked individuals following a bivariate normal distribution around the centroid (asterisk) of the marking locations (+); the marginal point process of both marked and unmarked individuals combined is homogeneous.

16276 point process model for the distribution of marks depends on the spatial context of the
16277 marking process.

16278 **Moving activity centers:** One issue we have not addressed explicitly in this section is
16279 what to do when enough time passes between marking and resighting so that animals may
16280 have rearranged themselves spatially. In example 1 this could mean, marked animals have
16281 shifted their activity center outside of the marking plot \mathcal{B} . In the second example, marked
16282 animals may no longer follow that initial distribution around the centroid of the marking
16283 array. Irrespectively, we will have to make some assumption about how marked individuals
16284 are distributed in space. We saw such a case in the Canada geese example in Sec. 19.3.1. In
16285 that example, we defined a reasonable state-space and assumed that animals redistributed
16286 themselves randomly across that state-space between marking and resighting, because they
16287 were captured while molting. This seems like a sensible solution for that particular study
16288 system, but it leads to a model formulation where estimates of density are sensitive to
16289 choice of \mathcal{S} . Alternatively, we could again use a bivariate normal distribution around the
16290 centroid of the original marking array – if movement of activity centers is random with
16291 respect to C_m it seems plausible that the overall underlying distribution of s would still
16292 be adequately described with a bivariate normal model. It should be clear that choice of
16293 a model for the inhomogeneous point process of marked individuals depends both on the
16294 spatial context of marking and what we know (or believe) about the biology of the study
16295 species.

19.8 SUMMARY AND OUTLOOK

16296 In this chapter we combined SCR models (for marked individuals) with a model for un-
16297 marked individuals, to derive a spatial mark-resight (SMR) model, which accommodates
16298 that only part of the population is individually identifiable, usually through artificial tags.
16299 Under the assumption that marked individuals are a random sample, both demographi-
16300 cally and spatially, from the state-space, the basic model with known number of marked
16301 individuals and perfect individual identification of marks is easily modified for situations
16302 where the number of marked individuals is unknown, or where marked animals can some-
16303 times not be identified to individual level. As expected, having marked individuals in the
16304 study population improved accuracy and precision of parameter estimates when compared
16305 to fully unmarked populations, but we also saw that the spatial counts of unmarked in-
16306 dividuals still contribute information to parameter estimates. We further presented an
16307 approach to incorporate telemetry location data into the spatial mark-resight model to in-
16308 form estimates of σ and activity centers. Just as in SCR, the spatial mark-resight models
16309 can account for a variety of factors that may influence individual movement and detection,
16310 as well as survey-related parameters, and we saw one example for the Canada geese, where
16311 σ was sex-specific, and another for raccoons, where λ_0 was time-dependent.

16312 Many details of SMR models remain to be explored and we noted a few of those
16313 topics throughout this chapter. For example, we mentioned the assignment of marked but
16314 unidentified records to actual marked individuals based on their spatial location, which
16315 provides some (though imperfect) information of their identity (Sec. 19.4). Similarly,
16316 records where the marked status cannot be determined could potentially be included in
16317 the model as some form of overall correction factor on detection. GPS telemetry devices
16318 and their ability to collect location data with much higher frequency offer the opportunity

16319 to assign records of collared animals to individuals based on how close to a given camera the
16320 collared individuals were, both in space and time. In this scenario, individual identity itself
16321 could be expressed probabilistically, leading to an SMR model accounting for potential
16322 misidentification. All these possible extensions can tailor SMR models to specific survey
16323 techniques.

16324 A fundamental assumption of the SMR models developed in this chapter was that
16325 marked animals are a random sample from \mathcal{S} . This simplifies the model as we can assume a
16326 homogeneous point process for both the marked and the unmarked part of the population.
16327 While this is a convenient situation, it is neither likely to arise often in real life, nor strictly
16328 necessary. If marked animals are not a random sample from \mathcal{S} , we need to describe their
16329 distribution in the state-space using an adequate point process that will almost always be
16330 inhomogeneous across \mathcal{S} . We mention two possible approach – a uniform (homogeneous)
16331 point process over a smaller area within \mathcal{S} , and an inhomogeneous point process where
16332 the intensity decreases as the distance to the centroid of the marking locations increases.
16333 Both formulations effectively attempt to describe the distribution of marks in space as a
16334 consequence of the spatial nature of the marking process. We believe that another way
16335 to approach this problem is to combine spatially explicit models for the marking process
16336 and the resighting process. Where marked animals were captured carries information on
16337 their spatial distribution, and it should be possible to make use of this information by
16338 formulating an integrated spatial capture-mark-resighting model. Such approaches have
16339 been developed in a non-spatial CR framework (??), but to our knowledge have not yet
16340 been addressed in the context of SCR.

16341 Spatial mark-resight models are a recent development, and work on how to relax
16342 the spatial component of the random sample assumption and formulate adequate point
16343 process models for the distribution of marks is ongoing. While there is still a lot of work
16344 to be done, we believe that SMR modeling holds the potential to address a wide range
16345 of population estimation problems when dealing with animals that cannot be identified
16346 based on natural marks, a situation that is typical of a majority of animal population
16347 studies.

16348
16349

20

16350
16351

2012: A SPATIAL CAPTURE-RECAPTURE ODYSSEY

16352 You've finally made it to the last chapter and we realize it's been a long journey to get here.
16353 Congratulations! (and thank you!) We hope this book has provided you with many ideas
16354 on how to conduct ecological studies and address specific questions that were previously
16355 thought difficult or impossible to answer, and given you a solid foundation for carrying out
16356 SCR analyses using either Bayesian or classical methods of statistical inference. However,
16357 we believe this journey is only just beginning, and we leave you now with a few thoughts
16358 on what we see as the future of SCR methods.

16359 Let us first briefly consider how we got here. Over a century ago, around 1786 in
16360 France, Pierre-Simon Laplace and others first developed capture-recapture methods and
16361 introduced the study of populations. This was of course regarding human population
16362 demography, but still, the foundation of how we would go on to study animal populations
16363 was being laid out then and there. The Lincoln-Petersen method was articulated by the
16364 1930s and development of capture-recapture models began to grow rapidly starting in
16365 the 1950s. Soon, capture-recapture methods had become a cornerstone of ecological and
16366 wildlife modeling and analysis. Today, spurred on by the advent and rapid development
16367 of non-invasive technologies like DNA sampling, camera trapping, acoustic sampling, and
16368 other methods, capture-recapture is more relevant and widely used than ever before. These
16369 new survey methods allow researchers to use capture-recapture for species that could not
16370 be studied efficiently even a few years ago, especially those that are difficult to capture or
16371 handle including most felids (Fig. 20.3), bears, mustelids such as fishers (*Martes pennanti*,
16372 Fig. 20.1) or weasels (e.g., long-tailed weasel *Mustela frenata*, Fig. 20.2), and many other
16373 species.

16374 With these new sampling techniques, like many commonly used capture-recapture
16375 sampling methods, spatial information about location of capture is collected. Classical
16376 capture-recapture models ignore this information, and in doing so fail to provide a formal
16377 method for modeling spatial variation in density and encounter probability. It was these
16378 deficiencies that motivated the development of SCR models, starting around 2003 - 2004.

16379 We have seen a great increase in the number of papers that use or cite SCR models,



Figure 20.1. Fisher assaulting tree # 8-12, outfitted with a baited hair snare. *Photo credit: NYSDEC (New York State Department of Environmental Conservation), A Fuller/NYSDEC camera trap and hair snare study of fishers in southern NY*

16380 and to articulate and quantify this growth, we did a Google Scholar search on March 6,
16381 2013 using the terms:

16382 “spatial capture recapture” OR “spatially explicit capture recapture”

16383 The results from this literature search are shown in Table 20.1. We see the number of
16384 citations involving SCR rapidly increasing, with growth in citation counts after 2004 fueled
16385 by publication of Efford (2004) and the release of the software DENSITY (Efford et al.,
16386 2004). In 2012 there were 84 articles published and 27 through the first 9 weeks of 2013.
16387 The results, we think, suggest a bright future for the development and application of
16388 spatial capture-recapture models. Most (but not all) of these papers are about the type of
16389 SCR models discussed in this book, although a handful had to do with other types of
16390 spatial analysis as related to capture-recapture models.

16391 We believe that use and growth of SCR modeling in conservation biology, management,
16392 wildlife, fisheries, and many other disciplines that we place under the general umbrella
16393 of ecology will only continue. This prediction is based our belief that SCR provides



Figure 20.2. A long-tailed weasel taking bait on a hair snare, A. Fuller southern NY fisher study *Photo credit: Marty DeLong.*

16394 a flexible framework for studying spatial and temporal variation in ecological processes
 16395 while acknowledging the fact that these processes are almost always observed imperfectly.
 16396 The “big idea” of SCR, if you could distill the whole thing into one idea, is based on
 16397 extending closed population models by augmenting them with a point process model
 16398 that describes the distribution of individuals (Efford, 2004) in space. In a sense, that is
 16399 really all there is to it. It seems like a little thing, a minor addition to a model, some
 16400 incremental advance or “e-improvement” of existing technology. But the relevance is much
 16401 bigger and more profound because, once we have made space explicit in the model, we
 16402 can think about building population models that embody explicit spatial processes and
 16403 using those models to improve our understanding of population biology and ecology, and
 16404 to test explicit hypotheses about mechanisms that govern populations.

16405 We covered many ecological processes that can be studied using SCR, such as land-
 16406 scape connectivity, resource selection, and spatial variation in density. These are all by
 16407 themselves profound extensions of the basic capture-recapture method, and they broaden
 16408 and expand the relevance and utility of capture-recapture for studying animal populations.
 16409 Although we filled almost 600 book pages (mostly) with SCR methods, there remains much
 16410 to be done in the continued development of SCR models. In the following section, we high-
 16411 light some emerging topics that show promise or might be in need of further development.
 16412 Finally, we end with a few remaining thoughts on the use of SCR models in the future.



Figure 20.3. Canada Lynx, ear-tagged and radio collared, producing high quality data in the name of science. *Photo credit: A Fuller, Cornell University*

20.1 EMERGING TOPICS

16413 In this book, we provided an overview and synthesis of capture-recapture methods as
16414 known to us around the end of 2012. There are many emerging topics which we have
16415 not covered either because of lack of technical knowledge, lack of time for satisfactory
16416 development, or lack of a good framework for implementation. Here we present some of
16417 those topics. This is not a complete list by any means, just a subset of topics that we or
16418 our colleagues are currently working on, or that we think might make good PhD, Masters
16419 or other research projects.

16420 20.1.1 Modeling territoriality

16421 In currently developing work, Reich et al. (2012) propose a model that accounts for spa-
16422 tial variation in density and potential interactions between individuals' territory centers.
16423 Under this model, the territory centers follow an inhomogeneous Strauss process (Strauss,
16424 1975), which includes a parameter that determines the strength of repulsion between ter-
16425 ritory centers. The idea is based on the notion that territorial species would have well
16426 defined (and defended) territories and thus activity centers may be more regular on the
16427 landscape than predicted by a homogeneous point process. A simulation study demon-
16428 strated that properly accounting for interactions between individuals can substantially

Table 20.1. Google Scholar citations by year based on a search of ‘‘spatial capture recapture’’ OR ‘‘spatially explicit capture recapture’’ conducted on March 6, 2013. The estimated growth rate of this population of papers was 33.4%.

Time period	Cumulative cites	Cites in year previous
since 2002	274 cites	
since 2003	274 cites	0 articles published in 2002
since 2004	271 cites	3 articles published in 2003
since 2005	269 cites	2 articles published in 2004
since 2006	264 cites	5 articles published in 2005
since 2007	261 cites	3 articles published in 2006
since 2008	253 cites	8 articles published in 2007
since 2009	242 cites	11 articles published in 2008
since 2010	222 cites	20 articles published in 2009
since 2011	176 cites	46 articles published in 2010
since 2012	111 cites	65 articles published in 2011
since 2013	27 cites	84 articles published in 2012
		27 published so far in 2013, since March 6

16429 improve population size estimates in terms of bias and precision relative to the usual
 16430 independence model.

16431 While the Strauss model is intuitive and shows great potential, it presents computa-
 16432 tional challenges. The first challenge is that the likelihood includes a high-dimensional
 16433 integral that has no closed form. To address this issue, Reich et al. (2012) developed an
 16434 approximation to the Strauss likelihood which allows for posterior sampling, extending
 16435 related work for categorical Markov random fields (Green and Richardson, 2002; Smith
 16436 and Smith, 2006). The second challenge is that N is treated as an unknown parameter to
 16437 be updated and hence N varies and so does the dimension of the posterior distribution. In
 16438 this case, the dimension-changing problem can be overcome by using data augmentation,
 16439 as we have done in many situations in this book.

16440 20.1.2 Combining data from different surveys

16441 In some instances, researchers apply different survey techniques to the population of in-
 16442 terest, because they yield complementary information. For example, camera trapping is
 16443 the prime tool for estimating population size/density and other demographic parameters
 16444 for uniquely marked species, while genetic surveys can yield additional information on the
 16445 genetic diversity and health of a population that cannot be studied using camera traps.
 16446 At the same time, genetic surveys, when samples are analyzed to the individual level, also
 16447 yield spatial capture recapture data (see Chapt. 15). In this situation, we have two data
 16448 sets at hand that carry information on animal density, and we should be able to get more
 16449 precise estimates of density if we combine these two data sets into a single SCR model.

16450 Gopalaswamy et al. (2012a) developed two approaches to combining data from different
 16451 survey types. In the first case, both surveys are carried out at the same time, so that we
 16452 can assume that they both sample the same – closed – animal population, i.e., there are
 16453 no possible changes in population density between the two surveys. For camera trapping
 16454 and genetic surveys, we cannot match records of individuals between the two data sets.

16455 However, models for the distinct sample methods may share some parameters (e.g., σ of
16456 the encounter probability model) and, if the studies were conducted simultaneously, they
16457 share a common population size N .

16458 A second approach of using information from one survey in the analysis of a second
16459 survey (that maybe does not yield quite as much data as the primary survey) is by ana-
16460 lyzing the primary data set alone, then taking the posterior distribution of a parameter
16461 both surveys share and using it as an informative prior distribution in the analysis of
16462 the second data set. Gopalaswamy et al. (2012a) refer to this as the stepwise approach,
16463 and they implemented this approach by equating the mean and variance of the posterior
16464 distribution of ψ and σ from the photographic survey to the mean and variance of a beta
16465 and a gamma prior for these parameters, respectively, for the genetic survey. The authors
16466 found that this approach produced almost identical density estimates compared to the
16467 combined model approach described above.

16468 In summary, no matter which approach is chosen, combining data across surveys
16469 can help researchers obtain more precise population size or density estimates, which is
16470 especially valuable when dealing with rare and elusive species like big cats that almost
16471 always will produce sparse individual data sets. The paper by Gopalaswamy et al. (2012a)
16472 considers the situation where we have two SCR data sets, but we can imagine combining
16473 SCR data with other sources of information, such as telemetry data (see Chapt. 19
16474 and Chapt. 13 for examples), and possibly opportunistic observations, although to our
16475 knowledge this latter issue has not been tackled in the context of SCR, yet.

16476 20.1.3 Misidentification

16477 Imperfect identification of individuals can happen in a variety of ways. In genetic surveys
16478 there is usually some probability of misidentification due to genotyping error (e.g. Lukacs
16479 and Burnham (2005)). In camera trap survey a different type of imperfect identification
16480 can occur when only the only one flank of an animal is recorded in a detection event and
16481 cannot be matched to any of the individuals identified by both flanks. In that case, we can
16482 match single-flank pictures with the same side flank pictures, but not with opposite side
16483 flank pictures and thus cannot construct definite encounter histories for these single-flank
16484 individuals (a right flank and a left flank picture could be the same individual, or could
16485 be from two distinct individuals). Finally, in Chapt. 19, in the context of mark-resight
16486 models, we discussed the case where individuals can either not definitely be identified as
16487 marked or not – a violation of a basic mark-resight assumption, and developed an approach
16488 to dealing with the situation where we can always tell if an animal is marked or not, but
16489 we are not always able to ascertain its individual identity.

16490 In non-spatial capture recapture some efforts have been made to formally deal with
16491 misidentification. Stevick et al. (2001) address this problem by double-sampling to de-
16492 rive an error rate for genetic identification, and then including this error rate as a known
16493 constant into a Lincoln-Petersen estimator of abundance. Lukacs and Burnham (2005)
16494 develop an approach that includes an additional parameter in the model – the probability
16495 of a genotype being identified correctly, which is estimated as part of the model likelihood.
16496 Link et al. (2010) developed an approach toward solving the same problem implemented
16497 in a Bayesian framework that relaxes some of the assumptions of the initial approach.
16498 Yoshizaki et al. (2009) deal with misidentification from camera trap pictures due to evolv-

16499 ing marks (i.e., natural marks that change over time, such as scars). This situation is
16500 different from the genotyping error one. Here, a change in marks creates a supposedly
16501 ‘new’ individual that can be recaptured several times, while the original individual is never
16502 captured again (its mark is no longer in the population). In contrast, in genotyping er-
16503 ror it is assumed that misidentification creates a ‘new’ individual that is never observed
16504 again, because each error leads to a new unique genotype. Yoshizaki et al. (2009) approach
16505 this situation similarly, by including a parameter describing the probability of correctly
16506 identifying an individual upon recapture (the parameter can also be interpreted as the
16507 probability that a mark does not change between capture occasions). Because of the de-
16508 pendencies between true and false detection histories (when a ‘new’ individual is created,
16509 the ‘real’ one can no longer be recaptured), the standard multinomial approach to coming
16510 up with a model likelihood does not work and implementing the model in a maximum
16511 likelihood framework is difficult. The authors instead demonstrate an implementation of
16512 the model based on minimizing a function of the squared differences between the observed
16513 and expected frequencies of the observed capture histories.

16514 To our knowledge no attempts have been made to deal with misidentification in an
16515 SCR framework. While all of the mis-ID cases described above require distinct approaches,
16516 we believe that there is one unifying theme to all of them: the capture locations of the
16517 potentially mis-identified records should be informative about identity. For example, a
16518 right flank and a left flank camera trap picture that are taken at two neighboring camera
16519 traps should be more likely to belong to the same individual than a right and a left
16520 flank picture taken at cameras located at opposing ends of the trap array, especially if
16521 animal movement is smaller than the extent of the trap array. SCR models provide a
16522 natural way of using this additional information to reduce the uncertainty arising from
16523 misidentification.

16524 **20.1.4 Gregarious species**

16525 One of the key assumptions of the SCR models that we described throughout this book
16526 is that the activity centers are independent of one another, but this assumption will be
16527 violated for species that associate in pairs, family groups, or any other type of aggregation.
16528 However, we believe that general models can be developed for use in studies of gregarious
16529 species.

16530 The two issues that must be addressed are that (1) detections are not independent
16531 – a trap that catches one individual of given group is likely to capture others in the
16532 same group, and (2) the activity centers s_i should appear clustered or, in fact, completely
16533 redundant in some cases. A possible way to account for this is to change our definition of
16534 s_i from the location of an individual’s activity center, to the location of a group’s activity
16535 center (Russell et al., 2012). Ideally, to accommodate unknown group size, the SCR model
16536 would be expanded to include a model component for group size, so that formal estimation
16537 of both group density and group size would be possible.

16538 **20.1.5 Single Catch Traps**

16539 In Chapt. 9 we covered multinomial models in which an individual’s probability of being
16540 captured in a trap is independent of all other individuals. This is the multi-catch type of

device in which traps never fill-up, but an individual can only be caught in one trap in any given occasion. We suggested (following Efford et al. (2009a)) that the multi-catch independent multinomial model could be used for “single catch” traps (traps that hold a single individual or “fill up”) and that bias associated with mis-specifying the model would be low under certain conditions (i.e., when the proportion of occupied traps is low).

As discussed in Chapt. 9, Sec. , we recognize that the *time*, or order, of capture of an individual in any trapping interval will affect the encounter probability of subsequently captured individuals. Thus if the order of capture was known, then this information could be used to write the likelihood of the detection model exactly. In practice, the order of capture is almost never known, but it should be possible to regard capture order as a latent variable and consider all possible orderings. This would be computationally intense and so we are working on a solution that selects an arbitrary ordering of the captures as a practical approximation to the single-catch process. This will hopefully lead to a formal model for the the single catch trap problem.

20.1.6 Model Fit and Selection

Evaluation of model adequacy or “fit” is an important part of any applied analysis. In Chapt. 8, we offered up a number of ideas based on standard considerations and adapted and applied them to SCR models. However, these ideas have not been widely applied, or evaluated, and much work needs to be done. In particular, some basic analysis of their power under meaningful alternatives would increase their relevance and possibly lead to insights for devising better methods. This applies to both Bayesian and likelihood-based methods, for which there are even fewer published applications of goodness-of-fit assessment.

Similarly, we discussed model selection strategies using more-or-less conventional ideas based on AIC/DIC, and model indicator variables using the Kuo and Mallick (1998) method. Calibration of these methods under alternatives is needed, along with some analysis of sensitivity to density estimates to misspecification of certain model components.

20.1.7 Explicit movement models

We briefly discussed the topics of dispersal, transience, and migration in Chaps. ?? and 16 and sketched out a few ideas that allow for dynamics related to movement or migration. Temporary emigration and transiency are two topics where a significant amount of work has been accomplished in non-spatial closed and open capture-recapture models (Kendall et al., 1997; Pradel et al., 1997; Hines et al., 2003; Clavel et al., 2008; Gilroy et al., 2012; Chandler et al., 2011). Additionally, models for dispersal (e.g., Clobert et al. (2001); Ovaskainen (2004); Ovaskainen et al. (2008) and and other forms of movement (e.g., Jonsen et al. (2005); ?; McClintock et al. (2012)) have received quite a bit of attention and development in ecology.

With the recent development of SCR models, the framework is in place to provide a formal integration of the movement dynamics governing the processes of dispersal, emigration, and transiency. Further, the availability of SCR models that allow for explicit population dynamics (survival, recruitment) (Gardner et al., 2010a) now sets the stage to integrate models of movement dynamics directly with models of population demography,

16583 and parameterize interactions among population processes. What remains as an area of
16584 fruitful research is the development of realistic models of movement dynamics, dispersal,
16585 temporary emigration, and transiency that can be effectively fitted given typical sparse
16586 individual encounter history data generated from capture-recapture studies. Dispersal and
16587 emigration can also be related to the life stage of an individual in a certain population.
16588 Ultimately, combining multi-state models, where the states are age classes or breeding
16589 status categories, with open population SCR models and explicitly modeling patterns
16590 of movement and dispersal as a function of state (e.g., age or size class) seems like an
16591 important area of development.

20.2 FINAL REMARKS

16592 Everything in ecology is spatial, and now so too are capture-recapture models, models
16593 which have been the cornerstone of ecological research on populations for decades. His-
16594 torically, the main use of capture-recapture was to obtain population size estimates, but
16595 SCR models move the problem from one of estimation to one of formalizing hypotheses
16596 about spatial and temporal variation in ecological processes. These processes include re-
16597 source selection, landscape connectivity, and how individuals organize themselves in space.
16598 SCR models allow for this formalization by borrowing methods from spatial statistics, but
16599 unlike many spatial models, SCR models include key demographic parameters such as
16600 density and survival and thus allow for mechanistic rather than just phenomenological
16601 descriptions of natural variation. For these reasons, we believe SCR models will continue
16602 to be developed and extended, and their use will continue to grow.

16603 However, much work still needs to be done to improve computational feasibility, to
16604 address many technical or methodological holes in the literature, and to make these meth-
16605 ods more accessible to practitioners. We look forward to these developments and hope
16606 that this book will help catalyze further exploration on this nascent odyssey.

16607

Part V

16608

16609

Appendices

16610 APPENDIX I - USEFUL SOFTWARE AND
16611 R PACKAGES

16612 Throughout this book we have used a suite of software and R packages, all of which are
16613 freely available online. To make life a little easier for you, here we provide you with a list
16614 of all software and R packages, download links and some (hopefully) helpful tips regarding
16615 their installation.

20.3 WINBUGS

16617 Although **WinBUGS** (Gilks et al., 1994) is becoming increasingly obsolete with the faster
16618 and more flexible **OpenBUGS** and **JAGS**, there are still situations in which the pro-
16619 gram comes in handy. The .exe file can be downloaded from <http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/contents.shtml>. On 32 bit machines you can just go ahead and
16620 double-click on the .exe file and follow the installation instructions on the screen. On 64
16621 bit machines, according to the BUGS project you should download a zip file (from the
16622 same page) and unzip it into a folder of your choice. There are a couple of additional steps
16623 to make BUGS run. First, you need to obtain a key (which is free and valid for life) here:
16624 http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/WinBUGS14_immortality_key.txt. The
16625 key comes with instructions on how to activate it. Second, you need to update the ba-
16626 sic **WinBUGS** version to the most current one (which is from August 2007) following
16627 the instructions given here: http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/WinBUGS14_cumulative_patch_No3_06_08_07_RELEASE.txt. **WinBUGS** is ready to use after quit-
16628 ting and re-opening it. Remember that **WinBUGS** only runs on Windows machines.
16629 Also, there appears to be a problem installing the program in Vista, although we have no
16630 personal experience with this.

20.3.1 WinBUGS through R

16631 While you can run **WinBUGS** as a standalone application, we recommend you access
16632 it from within **R** using the package **R2WinBUGS** (Sturtz et al., 2005), so you can conve-
16633 niently process your output, make graphs etc. **R2WinBUGS** also allows you to run mod-
16634 els in **OpenBUGS** (see below). You can install the package from within **R** directly
16635 from a cran mirror. In addition to the usual package help document (<http://cran.r-project.org/web/packages/R2WinBUGS/R2WinBUGS.pdf>) you can also download a short
16636 manual with some examples (http://voteviwer.com/bayes_beach/R2WinBUGS.pdf).

20.4 OPENBUGS

16641 **OpenBUGS** is the up-to-date version of **WinBUGS** and can be downloaded here: <http://www.openbugs.info/w/Downloads> (Windows, Mac and Linux versions are available).
16642 The name **OpenBUGS** refers to the software being open source, so users do not need
16643 to download a license key, like they have to for **WinBUGS** (although the license key
16644 for **WinBUGS** is free and valid for life). For Windows, install by double-clicking on the
16645 .exe file and following the instructions on the installer screen. Compared to **WinBUGS**,
16646 **OpenBUGS** has more built-in functions. The method of how to determine the right
16647 updater for each model parameter has changed and the user can manually control the
16648 MCMC algorithm used to update model parameters. Several other changes have been
16649 implemented in **OpenBUGS** and a detailed list of differences between the two **BUGS**
16650 versions, can be found at <http://www.openbugs.info/w/OpenVsWin>. We have encountered
16651 convergence problems with simple scr models in this program. There is an extensive
16652 help archive for both **WinBUGS** and **OpenBUGS** and you can subscribe to a mailing
16653 list, where people pose and answer questions of how to use these programs at <http://www.mrc-bsu.cam.ac.uk/bugs/overview/list.shtml>
16654
16655

16656 20.4.1 OpenBUGS through R

16657 Like **WinBUGS**, **OpenBUGS** can be used as a standalone application or through **R**.
16658 There are several packages that allow **R** to interface with **OpenBUGS**, all of which can
16659 be installed directly from a cran mirror:

16660 **R2WinBUGS**: One of the options in the `bugs()` call is `program`, which lets you specify either
16661 **WinBUGS** or **OpenBUGS**. This is a convenient option because after having worked
16662 through some of this book you will likely be familiar with the format of `bugs()` output
16663 and other functions of the **R2WinBUGS** package.

16664 **R2openBUGS**: **R2openBUGS** (Sturtz et al., 2005) is very similar to, and actually based on,
16665 **R2WinBUGS** and it is unclear to us what can be gained by using the former over the latter.
16666 Arguments of the `bugs()` call differ slightly between the two packages and given that
16667 **R2WinBUGS** allows for the use of both **OpenBUGS** and **WinBUGS** it is probably easiest
16668 to stick with it.

16669 **BRugs**: **BRugs** (Thomas et al., 2006) can be installed from within **R** directly from a
16670 cran mirror. In addition to the help document at http://www.biostat.umn.edu/~brad/software/BRugs/BRugs_9_21_07.pdf there is a **WinBUGS** style manual you can access
16671 at <http://www.rni.helsinki.fi/openbugs/OpenBUGS/Docu/BRugs%20Manual.html>.
16672 **BRugs** has the convenient feature that all pieces of a **BUGS** analysis can be run from within
16673 **R**, including checking the model syntax, something that requires opening the **BUGS** GUI
16674 with other packages.

20.5 JAGS

16676 **JAGS** (Just Another Gibbs Sampler) (Plummer, 2003) runs scr models considerably faster
16677 than **WinBUGS**, does not have the convergence problem with simple scr models we have

16678 encountered in **OpenBUGS** but similar to the latter program, is flexible and constantly
16679 updated. Writing a **JAGS** model is virtually identical to writing a **WinBUGS** model.
16680 However, some functions may have slightly different names and you can look up available
16681 functions and their use in the **JAGS** manual. One potential downside is that **JAGS** can
16682 be very particular when it comes to initial values. These may have to be set as close to
16683 truth as possible for the model to start. Although **JAGS** lets you run several parallel
16684 Markov chains, this characteristic interferes with the idea of using overdispersed initial
16685 values for the different chains. Also, we have found that when running models, sometimes
16686 **JAGS** crashes for unclear reasons, taking **R** down with it. Oftentimes, in order to make
16687 it run again you'll have to go through downloading and installing it again (remove the
16688 non-functioning version first).

16689 **JAGS** has a variety of functions that are not available in **WinBUGS**. For example,
16690 **JAGS** allows you to supply observed data for some deterministic functions of unobserved
16691 variables. In **BUGS** we cannot supply data to logical nodes. Another useful feature is
16692 that the adaptive phase of the model (the burn-in) is run separately from the sampling
16693 from the stationary Markov chains. This allows you to easily add more iterations to the
16694 adaptive phase if necessary without the need to start from 0. There are other, more
16695 subtle differences and there is an entire manual section on differences between **JAGS** and
16696 **OpenBUGS**.

16697 **JAGS** is available for download at <http://sourceforge.net/projects/mcmc-jags/files/>, together with the R package **rjags** (Plummer, 2011), which allows running **JAGS**
16698 through **R**, user and installation manuals and examples. At this site **JAGS** is available for
16699 Windows and Mac; Linux binaries are distributed separately and you can find links to various
16700 sources here: <http://mcmc-jags.sourceforge.net/>. **JAGS** comes with a 32 bit and
16701 a 64 bit version and can be installed by double-clicking on the .exe file and following the
16702 instructions on the installer screen. For questions and problems concerning **JAGS** there is a
16703 forum online at <http://sourceforge.net/projects/mcmc-jags/forums/forum/610037>.

16705 20.5.1 JAGS through R

16706 Unlike the two **BUGS** programs, **JAGS** does not have a GUI interface but a command
16707 line interface that can be used to run the program as a standalone application. **JAGS**
16708 will solely perform the MCMC simulation; analyzing and summarizing the output has to
16709 be done outside of **JAGS**. To run **JAGS** through **R** you have two options.

16710 **rjags**: As mentioned above, **rjags** (Plummer, 2011) can be found together with **JAGS**
16711 and was developed/is being maintained by the inventor of **JAGS**, which means it is
16712 guaranteed to stay up to date when/as **JAGS** changes. The package can be installed
16713 from a cran mirror and the help document can be accessed at <http://cran.r-project.org/web/packages/rjags/rjags.pdf>

16715 **R2jags**: Alternatively, the package **R2jags** (Su and Yajima, 2011) provides a means of
16716 accessing **JAGS** through **R**. We prefer **rjags** for the reason named above, as well as because
16717 it stores data in a more memory-efficient way and has better **plot()** and **summary()**
16718 methods.

20.6 R

16719 At the time of the preparation of this list, **R** for Windows is at version 2.15.0, which can
16720 be downloaded at [url`http://cran.r-project.org/bin/windows/base/`](http://cran.r-project.org/bin/windows/base/). This site also contains
16721 helpful tips on how to install **R** in Windows Vista, how to update **R** packages etc. Instal-
16722 lation of **R** in Windows is straightforward: download the .exe file, double-click on it and
16723 follow the instructions of the Windows installer. The later versions of **R** come with versions
16724 for both 64 bit and 32 bit machines. The **R** site (<http://mirrors.softliste.de/cran/>)
16725 has an extensive FAQ section Hornik (2011), which includes instructions on how to install
16726 R on Unix and Mac computers.

16727 **20.6.1 R packages**

16728 This section provides an alphabetical list of useful **R** packages. There is a large number
16729 of **R** packages and by no means is this list intended to be complete in terms of what is
16730 useful. Rather, we list packages that we are familiar with and that we employ at one point
16731 or the other in this book. Unless explicitly stated otherwise, all packages can be installed
16732 directly from within **R** trough a cran mirror.

16733 **adapt**: `adapt` (Genz et al., 2007) is a package for multidimensional numerical integration.
16734 The package has been removed from the CRAN repository but can be obtained from
16735 <http://cran.r-project.org/src/contrib/Archive/adapt/>.

16736 **coda**: `coda` (Plummer et al., 2006) lets you summarize and perform diagnostics on mcmc
16737 output. For a list and description of functions, see the manual at <http://cran.r-project.org/web/packages/coda/coda.pdf>.

16739 **gdistance**: `gdistance` (van Etten, 2011) is a package for calculating distances and routes
16740 on geographical grids and can be used to calculate least cost path surfaces. Manual at
16741 <http://cran.r-project.org/web/packages/gdistance/gdistance.pdf>.

16742 **igraph**: `igraph` (Csardi and Nepusz, 2006) provides routines for graphs and network
16743 analysis. Manual at <http://cran.r-project.org/web/packages/igraph/igraph.pdf>.

16744 **inline**: `inline` (Sklyar et al., 2010) allows the user to define R functions with in-lined C,
16745 C++ or Fortran code. Manual at <http://cran.r-project.org/web/packages/inline/inline.pdf>.

16747 **maps**: `maps` (Becker et al., 2012) is a library for the display of maps. Manual at <http://cran.r-project.org/web/packages/maps/index.html>.

16749 **maptools**: `maptools` (Bivand and Lewin-Koh, 2013) provides a set of tools for read-
16750 ing and manipulating spatial data, especially ESRI shapefiles. Manual at <http://cran.r-project.org/web/packages/maptools/maptools.pdf>.

16752 **mvtnorm**: `mvtnorm` (Genz et al., 2013) computes multivariate normal and t probabilities,
16753 quantiles, random deviates and densities. Manual at <http://cran.r-project.org/web/packages/mvtnorm/mvtnorm.pdf>.

16755 **parallel**: **parallel** contains a suite of functions for parallel computing on multiple com-
16756 puter cores and comes with **R** versions 2.14.0 or higher. More information about the
16757 package can be found at <http://stat.ethz.ch/R-manual/R-devel/library/parallel/doc/parallel.pdf>.
16758

16759 **R2cuba**: **R2cuba** (Hahn et al., 2010) is another package for multidimensional integration.
16760 Manual at <http://cran.r-project.org/web/packages/R2Cuba/R2Cuba.pdf>.

16761 **raster**: **raster** (Hijmans and van Etten, 2012) provides functions for geographic analysis
16762 and modeling with raster data. Manual at <http://cran.r-project.org/web/packages/raster/raster.pdf>.
16763

16764 **Rcpp**: **Rcpp** (Eddelbuettel and François, 2011) provides R functions as well as a C++ li-
16765 brary which facilitate the integration of **R** and C++. Manual at <http://cran.r-project.org/web/packages/Rcpp/Rcpp.pdf>.
16766

16767 **RcppArmadillo**: **RcppArmadillo** (François et al., 2011) is a templated C++ linear algebra
16768 library, integrating the **Armadillo** library and **R**. Manual at <http://cran.r-project.org/web/packages/RcppArmadillo/RcppArmadillo.pdf>.
16769

16770 **reshape**: **reshape** (Wickham and Hadley, 2007) allows you to easily manipulate, summa-
16771 rize and reshape data. Manual at <http://cran.r-project.org/web/packages/reshape/reshape.pdf>.
16772

16773 **rgeos**: **rgeos** (Bivand and Rundel, 2011) provides many useful functions for spatial
16774 operations such as intersecting or buffering spatial features. Manual at <http://cran.r-project.org/web/packages/rgeos/rgeos.pdf>.
16775

16776 **SCRbayes**: (Russell et al., 2012). XXXXXXXX Manual at XXXXX.

16777 **secr**: **secr** (Efford et al., 2009a) is an allround package for fitting a wide array of SCR mod-
16778 els in a frequentist framework. Manual at <http://cran.r-project.org/web/packages/secr/secr.pdf>.
16779

16780 **shapefiles**: **shapefiles** (Stabler, 2006) allows you to read and write ESRI shapefiles
16781 (i.e. shapefiles you would use in **ArcGIS**). Manual at <http://cran.r-project.org/web/packages/shapefiles/shapefiles.pdf>.
16782

16783 **snow**, **snowfall**: **snow** (Tierney et al., 2011) and **snowfall** (Knaus, 2010) provide func-
16784 tionality for parallel computing. The latter is a more user-friendly wrapper around the for-
16785 mer. Manuals at <http://cran.r-project.org/web/packages/snowfall/snowfall.pdf>
16786 and <http://cran.r-project.org/web/packages/snow/snow.pdf>.
16787

16788 **sp**: **sp** (Pebesma and Bivand, 2011) is a package for plotting, selecting, subsetting etc.
16789 spatial data. **sp** and **spatstat** (see below) are complementary in many ways and data
16790 formats can be easily converted between the two packages. Manual at <http://cran.r-project.org/web/packages/sp/sp.pdf>.
16790

16791 **SPACECAP**: SPACECAP (Gopalaswamy et al., 2012a) provides a user friendly GUI interface
16792 to fit SCR models with a Binomial observation model in a Bayesian framework. Manual
16793 at <http://www.icesi.edu.co/CRAN/web/packages/SPACECAP/SPACECAP.pdf>.

16794 **spatstat**: spatstat (Baddeley and Turner, 2005) is an extensive package for analyzing
16795 spatial data. We use it, for example, to generate random points within a state space
16796 that cannot be described as a rectangle but consists of a (or several) arbitrary polygon(s).
16797 Manual at <http://cran.r-project.org/web/packages/spatstat/spatstat.pdf>.

16798 **unmarked**: unmarked (Fiske and Chandler, 2011) fits hierarchical models of animal abun-
16799 dance and occurrence to data collected using a range of predominantly direct observa-
16800 tion based methods. Manual at <http://cran.r-project.org/web/packages/unmarked/unmarked.pdf>.

BIBLIOGRAPHY

- 16804 Adriaensen, F., Chardon, J. P., De Blust, G., Swinnen, E., Villalba, S., Gulnick, H.,
 16805 and Matthysen, E. (2003), "The application of 'least-cost' modelling as a functional
 16806 landscape model," *Landscape and urban planning*, 64, 233–247.
- 16807 Agresti, A. (2002), *Categorical data analysis*, vol. 359, John Wiley and Sons.
- 16808 Aitkin, M. (1991), "Posterior bayes factors," *Journal of the Royal Statistical Society. Series B (Methodological)*, 53, 111–142.
- 16810 Alho, J. M. (1990), "Logistic regression in capture-recapture models," *Biometrics*, 46,
 16811 623–635.
- 16812 Alpízar-Jara, R. and Pollock, K. H. (1996), "A combination line transect and capture-
 16813 recapture sampling model for multiple observers in aerial surveys," *Environmental and Ecological Statistics*, 3, 311–327.
- 16814 Amstrup, S. C., McDonald, T. L., and Manly, B. F. J. (2005), *Handbook of capture-
 16815 recapture analysis*, Princeton Univ Pr.
- 16816 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.
- 16817 Arnason, A., Schwarz, C., and Gerrard, J. (1991), "Estimating closed population size and
 16818 number of marked animals from sighting data," *Journal of Wildlife Management*, 55,
 16819 716–730.
- 16820 Arnason, N. A. (1972), "Parameter estimates from mark-recapture experiments on two
 16821 populations subject to migration and death," *Researches on Population Ecology*, 13,
 16822 97–113.
- 16823 — (1973), "The estimation of population size, migration rates and survival in a stratified
 16824 population," *Researches on Population Ecology*, 15, 1–8.
- 16825 Baddeley, A. and Turner, R. (2005), "Spatstat: an R package for analyzing spatial point
 16826 patterns," *Journal of Statistical Software*, 12, 1–42, ISSN 1548-7660.
- 16827 Bales, S., Hellgren, E., Leslie Jr, D., and Hemphill Jr, J. (2005), "Dynamics of a recolonizing
 16828 population of black bears in the Ouachita Mountains of Oklahoma," *Wildlife Society Bulletin*, 33, 1342–1351.
- 16829 Balme, G. A., Slotow, R., and Hunter, L. T. B. (2010), "Edge effects and the impact of non-
 16830 protected areas in carnivore conservation: leopards in the Phinda–Mkuze Complex,
 16831 South Africa," *Animal Conservation*, 13, 315–323.
- 16832 Bartmann, R. M., White, G. C., Carpenter, L. H., and Garrott, R. A. (1987), "Aerial
 16833 Mark-Recapture Estimates of Confined Mule Deer in Pinyon-Juniper Woodland," *The Journal of Wildlife Management*, 51, 41–46.
- 16834 Bayne, E. and Hobson, K. (2002), "Annual survival of adult American Redstarts and
 16835 Ovenbirds in the southern boreal forest," *The Wilson Bulletin*, 114, 358–367.

- 16841 Becker, R. A., Wilks, A. R., Brownrigg, R., and Minka, T. P. (2012), *maps: Draw Geographical Maps*, r package version 2.3-0.
- 16842 Beier, P., Majka, D. R., and Spencer, W. D. (2008), "Forks in the road: choices in procedures for designing wildland linkages," *Conservation Biology*, 22, 836–851.
- 16843 Belant, J., Van Stappen, J., and Paetkau, D. (2005), "American black bear population size and genetic diversity at Apostle Islands National Lakeshore," *Ursus*, 16, 85–92.
- 16844 Berger, J., Liseo, B., and Wolpert, R. (1999), "Integrated likelihood methods for eliminating nuisance parameters," *Statistical Science*, 14, 1–28.
- 16845 Besag, J. and Kooperberg, C. (1995), "On conditional and intrinsic autoregressions," *Biometrika*, 82, 733–746.
- 16846 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.
- 16847 Bibby, C. J. and Buckland, S. T. (1987), "Bias of bird census results due to detectability varying with habitat," *Acta Ecologica*, 8, 103–112.
- 16848 Bibby, C. J., Burgess, N. D., and Hill, D. A. (1992), "Bird census techniques: Academic Press," London, UK, 257.
- 16849 Bivand, R. and Lewin-Koh, N. (2013), *maptools: Tools for reading and handling spatial objects*, r package version 0.8-23.
- 16850 Bivand, R. and Rundel, C. (2011), *rgeos: Interface to Geometry Engine - Open Source (GEOS)*, r package version 0.1-8.
- 16851 Blair, W. (1940), "A study of prairie deer-mouse populations in southern Michigan," *American Midland Naturalist*, 24, 273–305.
- 16852 Bolker, B. (2008), *Ecological models and data in R*, Princeton Univ Pr.
- 16853 Bondrup-Nielsen, S. (1983), "Density estimation as a function of live-trapping grid and home range size," *Canadian Journal of Zoology*, 61, 2361–2365.
- 16854 Borchers, D. L. (2012), "A non-technical overview of spatially explicit capture–recapture models," *Journal of Ornithology*, 152, 1–10.
- 16855 Borchers, D. L., Buckland, S. T., and Zucchini, W. (2002), *Estimating animal abundance: closed populations*, vol. 13, Springer Verlag.
- 16856 Borchers, D. L. and Efford, M. G. (2008), "Spatially explicit maximum likelihood methods for capture–recapture studies," *Biometrics*, 64, 377–385.
- 16857 Borchers, D. L., Zucchini, W., and Fewster, R. M. (1998), "Mark-recapture models for line transect surveys," *Biometrics*, 54, 1207–1220.
- 16858 Boulanger, J. and McLellan, B. (2001), "Closure violation in DNA-based mark-recapture estimation of grizzly bear populations," *Canadian Journal of Zoology*, 79, 642–651.
- 16859 Bowden, D. C. (1993), "A simple technique for estimating population size. Technical Report 93/12." Tech. rep., Department of Statistics, Colorado State University, Fort Collins, Colorado, USA.
- 16860 Box, G. E. P. and Draper, N. R. (1959), "A basis for the selection of a response surface design," *Journal of the American Statistical Association*, 54, 622–654.
- 16861 — (1987), *Empirical model-building with response surfaces*, Wiley, New York.
- 16862 Boyce, M. S. and McDonald, L. L. (1999), "Relating populations to habitats using resource selection functions," *Trends in Ecology & Evolution*, 14, 268–272.
- 16863 Brooks, S. P., Catchpole, E. A., and Morgan, B. J. T. (2000), "Bayesian Animal Survival Estimation," *Statistical Science*, 15, 357–376.

- 16887 Brownie, C., Hines, J. E., Nichols, J. D., and Pollock, K. H. and Hestbeck, J. B. (1993),
16888 "Capture-recapture studies for multiple strata including non-Markovian transitions."
16889 *Biometrics*, 49, 1173–1187.
- 16890 Buckland, S., Anderson, D., Burnham, K., Laake, J., Borcher, D., and L, T. (2001), *Introduction to distance sampling: estimating abundance of biological populations*, Oxford,
16891 UK: Oxford University Press.
- 16892 Buckland, S. T. (2004), *Advanced distance sampling*, Oxford University Press, USA.
- 16893 Burnham, K., Anderson, D., and Laake, J. (1980), "Estimation of density from line tran-
16894 sect sampling of biological populations," *Wildlife monographs*, 3–202.
- 16895 Burnham, K., Anderson, D., White, G., Brownie, C., and Pollock, K. (1987), "Design and
16896 analysis methods for fish survival experiments based on release-recapture," *American
16897 Fisheries Society Monograph 5. American Fisheries Society, Bethesda Maryland. 1987.
16898 737.*
- 16899 Burnham, K. P. (1997), "Distributional results for special cases of the Jolly-Seber model."
16900 *Communications in Statistics*, 26, 1395 – 1409.
- 16901 Burnham, K. P. and Anderson, D. R. (2002), *Model selection and multimodel inference:
16902 a practical information-theoretic approach*, Springer Verlag.
- 16903 Burnham, K. P. and Overton, W. S. (1978), "Estimation of the size of a closed population
16904 when capture probabilities vary among animals," *Biometrika*, 65, 625.
- 16905 Burt, W. (1943), "Territoriality and home range concepts as applied to mammals," *Journal
16906 of mammalogy*, 24, 346–352.
- 16907 Casella, G. and Berger, R. L. (2002), *Statistical inference*, Duxbury Press.
- 16908 Casella, G. and George, E. I. (1992), "Explaining the Gibbs sampler," *American Statisti-
16909 cian*, 46, 167–174.
- 16910 Caswell, H. (1989), *Matrix population models*, Sinauer assoc. Sunderland.
- 16911 Caswell, H. and Werner, P. A. (1978), "Transient behavior and life history analysis of
16912 teasel (*Dipsacus sylvestris* Huds.)," *Ecology*, 59, 53–66.
- 16913 Chandler, R. B. and Royle, J. A. (2013), "Spatially-explicit models for inference about
16914 density in unmarked or partially marked populations," *Annals of Applied Statistics*,
16915 XXXX–XXXX.
- 16916 Chandler, R. B., Royle, J. A., and King, D. (2011), "Inference about density and tempo-
16917 rary emigration in unmarked populations," *Ecology*, 92, 1429–1435.
- 16918 Clavel, J., Robert, A., Devictor, V., and Julliard, R. (2008), "Abundance estimation
16919 with a transient model under the robust design," *Journal of Wildlife Management*, 72,
16920 1203–1210.
- 16921 Clobert, J., Danchin, E., Dhondt, A., and Nichols, J. (2001), *Dispersal*, Oxford.
- 16922 Cochran, W. (2007), *Sampling techniques*, United States of America: John Wiley & Sons.
- 16923 Compton, B. W., McGarigal, K., Cushman, S. A., and Gamble, L. R. (2007), "A resistant-
16924 kernel model of connectivity for amphibians that breed in vernal pools," *Conservation
16925 Biology*, 21, 788–799.
- 16926 Conde, D., Colchero, F., Zarza, H., Christensen, N., Sexton, J., Manterola, C., Chávez,
16927 C., Rivera, A., Azuara, D., and Ceballos, G. (2010), "Sex matters: Modeling male
16928 and female habitat differences for jaguar conservation," *Biological Conservation*, 143,
16929 1980–1988.
- 16930 Conn, P. B. and Cooch, E. G. (2009), "Multistate capture - recapture analysis under im-
16931 perfect state observation: an application to disease models," *Journal of Applied Ecology*,

- 16933 46, 486–492.
- 16934 Conroy, M. J. and Carroll, J. P. (2009), *Quantitative Conservation of Vertebrates*, Wiley-
16935 Blackwell.
- 16936 Conroy, M. J., Runge, J. P., Barker, R. J., Schofield, M. R., and Fonnesbeck, C. J. (2008),
16937 “Efficient estimation of abundance for patchily distributed populations via two-phase,
16938 adaptive sampling,” *Ecology*, 89, 3362–3370.
- 16939 Converse, S., White, G., and Block, W. (2006a), “Small mammal responses to thinning
16940 and wildfire in ponderosa pine-dominated forests of the southwestern United States,”
16941 *Journal of Wildlife Management*, 70, 1711–1722.
- 16942 Converse, S., White, G., Farris, K., and Zack, S. (2006b), “Small mammals and forest fuel
16943 reduction: national-scale responses to fire and fire surrogates,” *Ecological Applications*,
16944 16, 1717–1729.
- 16945 Converse, S. J. and Royle, J. A. (2012), “Dealing with incomplete and variable detectabil-
16946 ity in multi-year, multi-site monitoring of ecological populations,” in *Design and Anal-*
16947 *ysis of Long-term Ecological Monitoring Studies*, eds. Gitzen, R. R., Millspaugh, J. J.,
16948 Cooper, A. B., and Licht, D. S., Cambridge University Press, pp. 426–442.
- 16949 Cooch, E. and White, G. (2006), “Program MARK: a gentle introduction,” *available
16950 online with the MARK programme*, 7.
- 16951 Cormack, R. M. (1964), “Estimates of survival from the sighting of marked animals,”
16952 *Biometrika*, 51, 429–438.
- 16953 Coull, B. A. and Agresti, A. (1999), “The Use of Mixed Logit Models to Reflect Hetero-
16954 geneity in Capture-Recapture Studies,” *Biometrics*, 55, 294–301.
- 16955 Cox, D. (1955), “Some statistical methods connected with series of events,” *Journal of
16956 the Royal Statistical Society. Series B (Methodological)*, 17, 129–164.
- 16957 Cressie, N. (1991), *Statistics for spatial data*, Wiley Series in Probability and Mathematical
16958 Statistics.
- 16959 Csardi, G. and Nepusz, T. (2006), “The igraph software package for complex network
16960 research,” *InterJournal*, Complex Systems, 1695.
- 16961 Cushman, S. A., Compton, B. W., and McGarigal, K. (2010), “Habitat fragmentation
16962 effects depend on complex interactions between population size and dispersal ability:
16963 Modeling influences of roads, agriculture and residential development across a range
16964 of life-history characteristics,” in *Spatial complexity, informatics, and wildlife conser-*
16965 *vation*, eds. Cushman, S. A. and Huettmann, F., New York: Springer, chap. 20, pp.
16966 369–385.
- 16967 Cushman, S. A., McKelvey, K. S., Hayden, J., and Schwartz, M. K. (2006), “Gene flow in
16968 complex landscapes: testing multiple hypotheses with causal modeling,” *The American
16969 Naturalist*, 168, 486–499.
- 16970 Dawson, D. K. and Efford, M. G. (2009), “Bird population density estimated from acoustic
16971 signals,” *Journal of Applied Ecology*, 46, 1201–1209.
- 16972 DeGraaf, R. M. and Yamasaki, M. (2001), *New England wildlife: habitat, natural history,
16973 and distribution*, University Press of New England.
- 16974 DeSante, D. F., Burton, K. M., Saracco, J. F., and Walker, B. L. (1995), “Productiv-
16975 ity indices and survival rate estimates from MAPS, a continent-wide programme of
16976 constant-effort mist-netting in North America,” *Journal of Applied Statistics*, 22, 935–
16977 948.
- 16978 Dice, L. R. (1938), “Some census methods for mammals,” *Journal of Wildlife Management*,

- 16979 2, 119–130.
- 16980 — (1941), “Methods for estimating populations of mammals,” *Journal of Wildlife Management*, 5, 398–407.
- 16981
- 16982 Diggle, P. J. (2003), *Statistical analysis of spatial point processes*, London: Arnold, 2nd ed.
- 16983
- 16984 Dijkstra, E. W. (1959), “A note on two problems in connexion with graphs,” *Numerische mathematik*, 1, 269–271.
- 16985
- 16986 Dillon, A. and Kelly, M. (2007), “Ocelot *Leopardus pardalis* in Belize: the impact of trap spacing and distance moved on density estimates,” *Oryx*, 41, 469–477.
- 16987
- 16988 Dixon, P. (2002), “Bootstrap resampling,” *Encyclopedia of environmetrics*.
- 16989
- 16990 Dorazio, R. M. (2007), “On the choice of statistical models for estimating occurrence and extinction from animal surveys,” *Ecology*, 88, 2773–2782.
- 16991
- 16992 Dorazio, R. M. and Royle, J. A. (2003), “Mixture models for estimating the size of a closed population when capture rates vary among individuals,” *Biometrics*, 59, 351–364.
- 16993
- 16994 Dormann, C. F., McPherson, J. M., Araújo, M., Bivand, R., Bolliger, J., Carl, G.,
- 16995 G Davies, R., Hirzel, A., Jetz, W., Daniel Kissling, W., et al. (2007), “Methods to account for spatial autocorrelation in the analysis of species distributional data: a review,” *Ecography*, 30, 609–628.
- 16996
- 16997 Durban, J. and Elston, D. (2005), “Mark-recapture with occasion and individual effects: abundance estimation through Bayesian model selection in a fixed dimensional parameter space,” *Journal of agricultural, biological, and environmental statistics*, 10, 291–305.
- 16998
- 16999 Eddelbuettel, D. and François, R. (2011), “Rcpp: Seamless R and C++ Integration,” *Journal of Statistical Software*, 40, 1–18.
- 17000
- 17001 Efford, M. (2011a), *secr: Spatially explicit capture-recapture models*, R package version 2.3.1.
- 17002
- 17003 Efford, M. G. (2004), “Density estimation in live-trapping studies,” *Oikos*, 106, 598–610.
- 17004
- 17005 — (2011b), “Estimation of population density by spatially explicit capture-recapture analysis of data from area searches,” *Ecology*, 92, 2202–2207.
- 17006
- 17007 — (2012), “Spatially explicit capture-recapture for bear researchers and managers,” Western Black Bear Workshop, Coeur d’Alene, ID.
- 17008
- 17009 Efford, M. G., Borchers, D. L., and Byrom, A. E. (2009a), “Density estimation by spatially explicit capture–recapture: likelihood-based methods,” *Modeling demographic processes in marked populations*, 255–269.
- 17010
- 17011 Efford, M. G. and Dawson, D. K. (2009), “Effect of distance-related heterogeneity on population size estimates from point counts,” *The Auk*, 126, 100–111.
- 17012
- 17013 — (2010), “SECR for acoustic data,” .
- 17014
- 17015 Efford, M. G., Dawson, D. K., and Borchers, D. L. (2009b), “Population density estimated from locations of individuals on a passive detector array,” *Ecology*, 90, 2676–2682.
- 17016
- 17017 Efford, M. G., Dawson, D. K., and Robbins, C. S. (2004), “DENSITY: software for analysing capture-recapture data from passive detector arrays,” *Animal Biodiversity and Conservation*, 27, 217–228.
- 17018
- 17019 Efford, M. G. and Fewster, R. M. (2012), “Estimating population size by spatially explicit capture–recapture,” *Oikos*.
- 17020
- 17021 Efford, M. G., Warburton, B., Coleman, M. C., and Barker, R. J. (2005), “A field test of two methods for density estimation,” *Wildlife Society Bulletin*, 33, 731–738.
- 17022
- 17023 Epps, C. W., Palsbøll, P. J., Wehausen, J. D., Roderick, G. K., Ramey II, R. R., and
- 17024

- 17025 McCullough, D. R. (2005), "Highways block gene flow and cause a rapid decline in
17026 genetic diversity of desert bighorn sheep," *Ecology Letters*, 8, 1029–1038.
- 17027 Epps, C. W., Wehausen, J. D., Bleich, V. C., Torres, S. G., and Brashares, J. S. (2007),
17028 "Optimizing dispersal and corridor models using landscape genetics," *Journal of Applied
17029 Ecology*, 44, 714–724.
- 17030 Farnsworth, G. L., Pollock, K. H., Nichols, J. D., Simons, T. R., E., H. J., and R.,
17031 S. J. (2002), "A removal model for estimating detection probabilities from point-count
17032 surveys," *The Auk*, 119, 414–425.
- 17033 Fedorov, V. V. (1972), *Theory of Optimal Experiments*, Academic Press, New York.
- 17034 Fedorov, V. V. and Hackl, P. (1997), *Model-oriented design of experiments*, Springer.
- 17035 Fienberg, S. E., Johnson, M. S., and Junker, B. W. (1999), "Classical multilevel and
17036 Bayesian approaches to population size estimation using multiple lists," *Journal of the
17037 Royal Statistical Society of London A*, 163, 383–405.
- 17038 Fiske, I. J. and Chandler, R. B. (2011), "unmarked: An R package for fitting hierarchical
17039 models of wildlife occurrence and abundance," *Journal Of Statistical Software*, 43, 1–23.
- 17040 Forester, J. D., Im, H. K., and Rathouz, P. J. (2009), "Accounting for animal movement
17041 in estimation of resource selection functions: sampling and data analysis," *Ecology*, 90,
17042 3554–3565.
- 17043 Forester, J. D., Ives, A. R., Turner, M. G., Anderson, D. P., Fortin, D., Beyer, H. L., Smith,
17044 D. W., and Boyce, M. S. (2007), "State-space models link elk movement patterns to
17045 landscape characteristics in Yellowstone National Park," *Ecological Monographs*, 77,
17046 285–299.
- 17047 Fortin, D., Beyer, H. L., Boyce, M. S., Smith, D. W., Duchesne, T., and Mao, J. S. (2005),
17048 "Wolves influence elk movements: behavior shapes a trophic cascade in Yellowstone
17049 National Park," *Ecology*, 86, 1320–1330.
- 17050 Foster, R. J. and Harmse, B. J. (2012), "A critique of density estimation from camera-
17051 trap data," *The Journal of Wildlife Management*, 76, 224–236.
- 17052 Fowler, C. (1981), "Density dependence as related to life history strategy," *Ecology*, 62,
17053 602–610.
- 17054 François, R., Eddelbuettel, D., and Bates, D. (2011), *RcppArmadillo: Rcpp integration
17055 for Armadillo templated linear algebra library*, r package version 0.2.25.
- 17056 Frary, V., Duchamp, J., Maehr, D., and Larkin, J. (2011), "Density and distribution of
17057 a colonizing front of the American black bear *Ursus americanus*," *Wildlife Biology*, 17,
17058 404–416.
- 17059 Fujiwara, M., Anderson, K., Neubert, M., and Caswell, H. (2006), "On the estimation of
17060 dispersal kernels from individual mark-recapture data," *Environmental and Ecological
17061 Statistics*, 13, 183–197.
- 17062 García-Alaníz, N., Naranjo, E. J., and Mallory, F. F. (2010), "Hair-snares: A non-invasive
17063 method for monitoring felid populations in the Selva Lacandona, Mexico," *Tropical
17064 Conservation Science*, 3, 403–411.
- 17065 Gardner, B., Reppucci, J., Lucherini, M., and Royle, J. (2010a), "Spatially explicit in-
17066 ference for open populations: estimating demographic parameters from camera-trap
17067 studies," *Ecology*, 91, 3376–3383.
- 17068 Gardner, B., Royle, J. A., and Wegan, M. T. (2009), "Hierarchical models for estimating
17069 density from DNA mark-recapture studies," *Ecology*, 90, 1106–1115.
- 17070 Gardner, B., Royle, J. A., Wegan, M. T., Rainbolt, R. E., and Curtis, P. D. (2010b),

- 17071 “Estimating black bear density using DNA data from hair snares,” *The Journal of*
17072 *Wildlife Management*, 74, 318–325.
- 17073 Garshelis, D. L. and Hristienko, H. (2006), “State and provincial estimates of American
17074 black bear numbers versus assessments of population trend,” *Ursus*, 17, 1–7.
- 17075 Gelfand, A. and Smith, A. (1990), “Sampling-based approaches to calculating marginal
17076 densities,” *Journal of the American statistical association*, 85, 398–409.
- 17077 Gelman, A. (2006), “Prior distributions for variance parameters in hierarchical models,”
17078 *Bayesian analysis*, 1, 515–533.
- 17079 Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004), *Bayesian data analysis*,
17080 second edition., Boca Raton, Florida, USA: CRC/Chapman & Hall.
- 17081 Gelman, A., Meng, X. L., and Stern, H. (1996), “Posterior predictive assessment of model
17082 fitness via realized discrepancies,” *Statistica Sinica*, 6, 733–759.
- 17083 Geman, S. and Geman, D. (1984), “Stochastic relaxation, Gibbs distributions, and the
17084 Bayesian restoration of images,” *IEEE Transactions on Pattern Analysis and Machine
17085 Intelligence*, PAMI-6, 721–741.
- 17086 Genz, A., Bretz, F., Miwa, T., Mi, X., Leisch, F., Scheipl, F., and Hothorn, T. (2013),
17087 *mvtnorm: Multivariate Normal and t Distributions*, r package version 0.9-9994.
- 17088 Genz, A. S., Meyer, M. R., Lumley, T., and Maechler, M. (2007), “The adapt Package. R
17089 package version 1.0-4.” .
- 17090 Gerlach, G. and Musolf, K. (2000), “Fragmentation of landscape as a cause for genetic
17091 subdivision in bank voles,” *Conservation Biology*, 14, 1066–1074.
- 17092 Gilks, W. and Wild, P. (1992), “Adaptive rejection sampling for Gibbs sampling,” *Applied
17093 Statistics*, 41, 337–348.
- 17094 Gilks, W. R., Thomas, A., and Spiegelhalter, D. J. (1994), “A Language and Program for
17095 Complex Bayesian Modelling,” *Journal of the Royal Statistical Society. Series D (The
17096 Statistician)*, 43, 169–177, ArticleType: primary_article / Issue Title: Special Issue:
17097 Conference on Practical Bayesian Statistics, 1992 (3) / Full publication date: 1994 /
17098 Copyright 1994 Royal Statistical Society.
- 17099 Gilroy, J., Virzi, T., Boulton, R. L., and Lockwood, J. (2012), “A new approach to
17100 the “apparent survival” problem: estimating true survival rates from mark-recapture
17101 studies,” *Ecology*, 93, 1509–1516.
- 17102 Gimenez, O., Rossi, V., Choquet, R., Dehais, C., Doris, B., Varella, H., Vila, J. P., and
17103 Pradel, R. (2007), “State-space modelling of data on marked individuals,” *Ecological
17104 Modelling*, 206, 431–438.
- 17105 Gopalaswamy, A. M. (2012), “Capture-recapture models, spatially explicit,” in *Encyclo-
17106 pedia of Environmentrics Second Edition*, eds. El-Shaarawi, A. H. and Piegorsch, W.,
17107 John Wiley and Sons Ltd, Chichester, UK.
- 17108 Gopalaswamy, A. M., Royle, A. J., Hines, J., Singh, P., Jathanna, D., Kumar, N. S., and
17109 Karanth, K. U. (2012a), “Program SPACECAP: software for estimating animal density
17110 using spatially explicit capturerecapture models,” *Methods in Ecology and Evolution*,
17111 online early, r package version 1.0.4.
- 17112 Gopalaswamy, A. M., Royle, J. A., Delampady, M., Nichols, J. D., Karanth, K. U., and
17113 Macdonald, D. W. (2012b), “Density estimation in tiger populations: combining infor-
17114 mation for strong inference,” *Ecology*, 93, 1741–1751.
- 17115 Grant, E. H. C., Nichols, J. D., Lowe, W. H., and Fagan, W. F. (2010), “Use of multiple
17116 dispersal pathways facilitates amphibian persistence in stream networks,” *Proceedings*

- 17117 *of the National Academy of Sciences*, 107, 6936–6940.
- 17118 Green, P. and Richardson, S. (2002), “Hidden Markov Models and Disease Mapping,”
17119 *Journal of the American Statistical Association*, 97, 1055–1070.
- 17120 Greig-Smith, P. (1964), *Quantitative plant ecology*, Butterworths (Washington).
- 17121 Grimm, V., Revilla, E., Berger, U., Jeltsch, F., Mooij, W., Railsback, S., Thulke, H.-
17122 H., Weiner, J., Wiegand, T., and DeAngelis, D. (2005), “Pattern-oriented modeling of
17123 agent-based complex systems: Lessons from ecology.” *Science*, 310, 987–991.
- 17124 Hahn, T., Bouvier, A., and Kiêu, K. (2010), *R2Cuba: Multidimensional Numerical Inte-*
17125 *gration*, R package version 1.0-6.
- 17126 Hall, R. J., Henry, P. F. P., and Bunck, C. M. (1999), “Fifty-year trends in a box turtle
17127 population in Maryland,” *Biological Conservation*, 88, 165–172.
- 17128 Hanks, E. M. and Hooten, M. B. (in press), “Circuit Theory and Model-Based Inference
17129 for Landscape Connectivity,” *Journal of the American Statistical Association*.
- 17130 Hanski, I. A. (1999), *Metapopulation Ecology*, Oxford Univiversity Press.
- 17131 Hardin, R. H. and Sloane, N. J. A. (1993), “A New Approach to the Construction of
17132 Optimal Designs,” *Journal of Statistical Planning and Inference*, 37, 339–369.
- 17133 Hastings, W. (1970), “Monte Carlo sampling methods using Markov chains and their
17134 applications,” *Biometrika*, 57, 97–109.
- 17135 Hawkins, C. E. and Racey, P. A. (2005), “Low population density of a tropical forest
17136 carnivore, Cryptoprocta ferox: implications for protected area management,” *Oryx*, 39,
17137 35–43.
- 17138 Hayes, R. J. and Buckland, S. T. (1983), “Radial distance models for the line transect
17139 method.” *Biometrics*, 39, 29–42.
- 17140 Hayne, D. (1950), “Apparent home range of Microtus in relation to distance between
17141 traps,” *Journal of mammalogy*, 31, 26–39.
- 17142 Hayne, D. W. (1949), “An Examination of the Strip Census Method for Estimating Animal
17143 Populations,” *The Journal of Wildlife Management*, 13, 145–157.
- 17144 Hedley, S. L., Buckland, S. T., and Borchers, D. L. (1999), “Spatial modelling from line
17145 transect data,” *Journal of Cetacean Research and Management*, 1, 255–264.
- 17146 Hendriks, I., Tenan, S., Tavecchia, G., Marbá, N., Jordá, G., Deudero, S., Álvarez, E., and
17147 Duarte, C. M. (2013), “Boat anchoring impacts coastal populations of the pen shell,
17148 the largest bivalve in the Mediterranean,” *Biological Conservation*, 160, 105–113.
- 17149 Hestbeck, J. B. and Malecki, R. A. (1989), “Mark-resight estimate of Canada Goose
17150 midwinter numbers,” *Journal of Wildlife Management*, 53, 749–752.
- 17151 Hijmans, R. J. and van Etten, J. (2012), *raster: Geographic analysis and modeling with*
17152 *raster data*, R package version 1.9-67.
- 17153 Hines, J. E., Kendall, W. L., and Nichols, J. D. (2003), “On the use of the robust design
17154 with transient capture-recapture models,” *The Auk*, 120, 1151–1158.
- 17155 Hobbs, N. T. (2011), “An Ecological Modeler’s Primer on JAGS,” .
- 17156 Holdenried, R. (1940), “A population study of the long-eared chipmunk (*Eutamias quadri-*
17157 *maculatus*) in the central Sierra Nevada,” *Journal of Mammalogy*, 21, 405–411.
- 17158 Hooten, M., Johnson, D., Hanks, E., and Lowry, J. (2010), “Agent-based inference for
17159 animal movement and selection,” *Journal of agricultural, biological, and environmental*
17160 *statistics*, 15, 523–538.
- 17161 Hooten, M. and Wikle, C. (2010), “Statistical agent-based models for discrete spatio-
17162 temporal systems.” *Journal of the American Statistical Association*, 105, 236–248.

- 17163 Hornik, K. (2011), "The R FAQ," ISBN 3-900051-08-9.
- 17164 Huggins, R. M. (1989), "On the statistical analysis of capture experiments," *Biometrika*,
17165 76, 133.
- 17166 Hurlbert, S. H. (1984), "Pseudoreplication and the design of ecological field experiments,"
17167 *Ecological monographs*, 54, 187–211.
- 17168 Illian, J., Penttinen, A., Stoyan, H., and Stoyan, D. (2008), *Statistical analysis and mod-
elling of spatial point patterns*, Wiley.
- 17170 Ivan, J. (2012), "Density, demography, and seasonal movements of snowshoe hares in
17171 central Colorado," Ph.D. thesis, Colorado State University.
- 17172 Ivan, J., White, G., and Shenk, T. (2013a), "Using auxiliary telemetry information to
17173 estimate animal density from capture-recapture data," *Ecology*.
- 17174 — (2013b), "Using simulation to compare methods for estimating density from capture-
17175 recapture data," *Ecology*.
- 17176 Jackson, R., Roe, J., Wangchuk, R., and Hunter, D. (2006), "Estimating Snow Leop-
17177 ard Population Abundance Using Photography and Capture-Recapture Techniques,"
17178 *Wildlife Society Bulletin*, 34, 772–781.
- 17179 Jennelle, C. S., Runge, M. C., and MacKenzie, D. I. (2002), "The use of photographic rates
17180 to estimate densities of tigers and other cryptic mammals: A comment on misleading
17181 conclusions," *Animal Conservation*, 5, 119–120.
- 17182 Jett, D. A. and Nichols, J. D. (1987), "A Field Comparison of Nested Grid and Trapping
17183 Web Density Estimators," *Journal of Mammalogy*, 68, 888–892.
- 17184 Jhala, Y. V., Qureshi, Q., Gopal, R., and R., S. P. (2011), "Status of tigers, co-predators
17185 and prey in India." Tech. rep., National Tiger Conservation Authority, Govt. of India,
17186 New Delhi and Wildlife Institute of India, Dehradun.
- 17187 Johnson (2010), "A Model-Based Approach for Making Ecological Inference from Distance
17188 Sampling Data," *Biometrics*, 66, 310–318.
- 17189 Johnson, D. (1980), "The comparison of usage and availability measurements for evaluat-
17190 ing resource preference," *Ecology*, 61, 65–71.
- 17191 Johnson, D. H. (1999), "The insignificance of statistical significance testing," *The journal
17192 of wildlife management*, 63, 763–772.
- 17193 Johnson, D. S., London, J. M., Lea, M. A., and Durban, J. (2008a), "Continuous-time
17194 correlated random walk model for animal telemetry data," *Ecology*, 89, 1208–1215.
- 17195 Johnson, D. S., Thomas, D. L., Ver Hoef, J. M., and Christ, A. (2008b), "A general
17196 framework for the analysis of animal resource selection from telemetry data," *Biometrics*,
17197 64, 968–976.
- 17198 Jolly, G. M. (1965), "Explicit estimates from capture-recapture data with both death and
17199 dilution-stochastic model," *Biometrika*, 52, 225–247.
- 17200 Jonsen, I. D., Flemming, J. M., and Myers, R. A. (2005), "Robust state-space modeling
17201 of animal movement data," *Ecology*, 86, 2874–2880.
- 17202 Kadane, J. B. and Lazar, N. A. (2004), "Methods and criteria for model selection," *Journal
17203 of the American Statistical Association*, 99, 279–290.
- 17204 Karanth, K. U. (1995), "Estimating tiger *Panthera tigris* populations from camera-trap
17205 data using capture–recapture models," *Biological Conservation*, 71, 333–338.
- 17206 Karanth, K. U. and Nichols, J. D. (1998), "Estimation of Tiger Densities in India Using
17207 Photographic Captures and Recaptures," *Ecology*, 79, 2852–2862.
- 17208 — (2000), "Ecological status and conservation of tigers in India," *WCS, US Fish and*

- 17209 *Wildlife Service. Centre for Wildlife Studies. Bangalore, India*, 123.
- 17210 — (2002), *Monitoring tigers and their prey: a manual for researchers, managers and*
17211 *conservationists in {T}ropical {A}sia*, Bangalore, India: Centre for Wildlife Studies.
- 17212 Kass, R. and Wasserman, L. (1996), "The selection of prior distributions by formal rules,"
17213 *Journal of the American Statistical Association*, 91, 1343–1370.
- 17214 Kays, R. W., Slanson, K. M., Long, R. A., MacKay, P., Zielinski, W. J., and Ray, J. C.
17215 (2008), "Remote cameras." *Noninvasive survey methods for carnivores*, 110–140.
- 17216 Kelly, M., Noss, A., Di Bitetti, M., Maffei, L., Arispe, R., Paviolo, A., De Angelo, C., and
17217 Di Blanco, Y. (2008), "Estimating puma densities from camera trapping across three
17218 study sites: Bolivia, Argentina, and Belize," *Journal of Mammalogy*, 89, 408–418.
- 17219 Kendall, K. C., Stetz, J. B., Boulanger, J., Macleod, A. C., Paetkau, D., and White, G. C.
17220 (2009), "Demography and genetic structure of a recovering grizzly bear population,"
17221 *The Journal of Wildlife Management*, 73, 3–16.
- 17222 Kendall, W. L. (1999), "Robustness of Closed Capture-Recapture Methods to Violations
17223 of the Closure Assumption," *Ecology*, 80, 2517–2525.
- 17224 Kendall, W. L., Nichols, J. D., and Hines, J. E. (1997), "Estimating Temporary Emigration
17225 Using Capture-Recapture Data with Pollock's Robust Design," *Ecology*, 78, 563–578.
- 17226 Kéry, M. (2008), "Estimating abundance from bird counts: binomial mixture models
17227 uncover complex covariate relationships," *The Auk*, 125, 336–345.
- 17228 — (2010), *Introduction to WinBUGS for Ecologists: Bayesian Approach to Regression,*
17229 *ANOVA, Mixed Models and Related Analyses*, Academic Press.
- 17230 Kéry, M. (2011), "Towards the modelling of true species distributions," *Journal of Bio-*
17231 *geography*, 38, 617–618.
- 17232 Kéry, M., Gardner, B., and Monnerat, C. (2010), "Predicting species distributions from
17233 checklist data using site-occupancy models," *Journal of Biogeography*, 37, 1851–1862.
- 17234 Kéry, M., Gardner, B., Stoeckle, T., Weber, D., and Royle, J. A. (2011), "Use of Spatial
17235 Capture-Recapture Modeling and DNA Data to Estimate Densities of Elusive Animals,"
17236 *Conservation Biology*, 25, 356–364.
- 17237 Kéry, M., Royle, J. A., and Schmid, H. (2005), "Modeling avian abundance from replicated
17238 counts using binomial mixture models," *Ecological Applications*, 15, 1450–1461.
- 17239 Kéry, M. and Schaub, M. (2012), *Bayesian Population Analysis Using WinBugs*, Academic
17240 Press.
- 17241 Kiefer, J. (1959), "Optimal experimental designs (with discussion)," *Journal of the Royal*
17242 *Statistical Society, Series B*, 272–319.
- 17243 King, R. and Brooks, S. (2001), "On the Bayesian analysis of population size," *Biometrika*,
17244 88, 317–336.
- 17245 King, R., Brooks, S., and Coulson, T. (2008), "Analyzing Complex Capture–Recapture
17246 Data in the Presence of Individual and Temporal Covariates and Model Uncertainty,"
17247 *Biometrics*, 64, 1187–1195.
- 17248 Knape, Jonas & de Valpine, P. (2012), "Are patterns of density dependence in the Global
17249 Population Dynamics Database driven by uncertainty about population abundance?"
17250 *Ecology Letters*, 15, 17–23.
- 17251 Knaus, J. (2010), *snowfall: Easier cluster computing (based on snow)*., r package version
17252 1.84.
- 17253 Koehler, G. M. and Pierce, D. J. (2003), "Black Bear Home-Range Sizes in Washington:
17254 Climatic, Vegetative, and Social Influences," *Journal of Mammalogy*, 84, 81–91, Article-

- 17255 Type: research-article / Full publication date: Feb., 2003 / Copyright 2003 American
17256 Society of Mammalogists.
- 17257 Kohn, M., York, E., Kamradt, D., Haught, G., Sauvajot, R., and Wayne, R. (1999),
17258 “Estimating population size by genotyping faeces,” *Proceedings of the Royal Society of*
17259 *London. Series B: Biological Sciences*, 266, 657–663.
- 17260 Krebs, C. J. (1999), *Ecological methodology*, Menlo Park, CA: Benjamin/Cummings.
- 17261 Kucera and Barrett (2011), “missing,” missing, missing.
- 17262 Kuo, L. and Mallick, B. (1998), “Variable selection for regression models,” *Sankhy: The*
17263 *Indian Journal of Statistics, Series B*, 60, 65–81.
- 17264 Laird, N. M. and Ware, J. H. (1982), “Random-effects models for longitudinal data,”
17265 *Biometrics*, 38, 963–974.
- 17266 Langtimm, C. A., Dorazio, R. M., Stith, B. M., and Doyle, T. J. (2011), “New aerial
17267 survey and hierarchical model to estimate manatee abundance,” *The Journal of Wildlife*
17268 *Management*, 75, 399–412.
- 17269 Le Cam, L. (1990), “Maximum likelihood: an introduction,” *International Statistical Re-*
17270 *view/Revue Internationale de Statistique*, 58, 153–171.
- 17271 Lebreton, J. D., Burnham, K. P., Clobert, J., and Anderson, D. R. (1992), “Modeling
17272 Survival and Testing Biological Hypotheses Using Marked Animals: A Unified Approach
17273 with Case Studies,” *Ecological Monographs*, 62, 67–118.
- 17274 Leggett, W. C. and Carscadden, J. E. (1978), “Latitudinal variation in reproductive
17275 characteristics of American shad (*Alosa sapidissima*): evidence for population specific
17276 life history strategies in fish.” *Journal of the Fisheries Research Board of Canada*, 35,
17277 1469–1477.
- 17278 Lele, S. R. and Keim, J. L. (2006), “Weighted distributions and estimation of resource
17279 selection probability functions,” *Ecology*, 87, 3021–3028.
- 17280 Lele, S. R., Moreno, M., and Bayne, E. (2012), “Dealing with detection error in site
17281 occupancy surveys: What can we do with a single survey?” *Journal of Plant Ecology*.
- 17282 Lele, S. R., Nadeem, K., and Schmuland, B. (2010), “Estimability and likelihood inference
17283 for generalized linear mixed models using data cloning,” *Journal of the American*
17284 *Statistical Association*, 105, 1617–1625.
- 17285 Leonard, J. B. K. and McCormick, S. D. (1999), “Effects of migration distance on whole-
17286 body and tissue-specific energy use in American shad (*Alosa sapidissima*),” *Canadian*
17287 *Journal of Fisheries and Aquatic Sciences*, 56, 1159–1171.
- 17288 Lewin-Koh, N. J., Bivand, R., contributions by Edzer J. Pebesma, Archer, E., Baddeley,
17289 A., Bibiko, H.-J., Dray, S., Forrest, D., Friendly, M., Giraudoux, P., Golicher, D., Rubio,
17290 V. G., Hausmann, P., Hufthammer, K. O., Jagger, T., Luque, S. P., MacQueen, D.,
17291 Niccolai, A., Short, T., Stabler, B., and Turner, R. (2011), *maptools: Tools for reading*
17292 *and handling spatial objects*, r package version 0.8-10.
- 17293 Lichstein, J. W., Simons, T. R., Shriner, S. A., and Franzreb, K. E. (2002), “Spatial
17294 autocorrelation and autoregressive models in ecology,” *Ecological Monographs*, 72, 445–
17295 463.
- 17296 Link, W. (2003), “Nonidentifiability of Population Size from Capture-Recapture Data
17297 with Heterogeneous Detection Probabilities,” *Biometrics*, 59, 1123–1130.
- 17298 Link, W. A. (In review), “A cautionary note on the discrete uniform prior for the binomial
17299 N ,” .
- 17300 Link, W. A. and Barker, R. J. (1994), “Density Estimation Using the Trapping Web

- 17301 Design: A Geometric Analysis," *Biometrics*, 50, 733–745.
- 17302 — (2005), "Modeling Association among Demographic Parameters in Analysis of Open
17303 Population Capture-Recapture Data," *Biometrics*, 61, 46–54.
- 17304 — (2006), "Model weights and the foundations of multimodel inference," *Ecology*, 87,
17305 2626–2635.
- 17306 — (2010), *Bayesian Inference: With Ecological Applications*, London, UK: Academic
17307 Press.
- 17308 Link, W. A. and Eaton, M. J. (2011), "On thinning of chains in MCMC," *Methods in
17309 Ecology and Evolution*, 3, 112–115.
- 17310 Link, W. A., Yoshizaki, J., Bailey, L. L., and Pollock, K. H. (2010), "Uncovering a latent
17311 multinomial: analysis of markrecapture data with misidentification," *Biometrics*, 66,
17312 178–185.
- 17313 Liu and Wu (1999), "Parameter expansion for data augmentation," *J Am Stat Assoc*, 94,
17314 1264–1274.
- 17315 Lukacs, P. M. and Burnham, K. P. (2005), "Estimating population size from DNA-based
17316 closed capture-recapture data incorporating genotyping error," *Journal of Wildlife
17317 Management*, 69, 396–403.
- 17318 Lunn, D., Spiegelhalter, D., Thomas, A., and Best, N. (2009), "The BUGS project: Evo-
17319 lution, critique, and future directions," *Statistics in Medicine*, 28, 3049–3067.
- 17320 Lunn, D. J., Thomas, A., Best, N., and Spiegelhalter, D. (2000), "WinBUGS-a Bayesian
17321 modelling framework: concepts, structure, and extensibility," *Statistics and Computing*,
17322 10, 325–337.
- 17323 Mace, R., Minta, S., Manley, T., and Aune, K. (1994), "Estimating grizzly bear population
17324 size using camera sightings," *Wildlife Society Bulletin*, 22, 74–83.
- 17325 MacEachern, S. N. and Berliner, L. M. (1994), "Subsampling the Gibbs sampler," *Amer-
17326 ican Statistician*, 48, 188–190.
- 17327 MacKay, P., Smith, D. A., Long, R. A., Parker, M., Long, R. A., MacKay, P., Zielinski,
17328 W. J., and Ray, J. C. (2008), "Scat detection dogs," *Noninvasive survey methods for
17329 carnivores*, 183–222.
- 17330 MacKenzie, D. I., Nichols, J. D., Lachman, G. B., Droege, S., Royle, J. A., and Langtimm,
17331 C. A. (2002), "Estimating site occupancy rates when detection probabilities are less than
17332 one," *Ecology*, 83, 2248–2255.
- 17333 MacKenzie, D. I., Nichols, J. D., Royle, J. A., Pollock, K. H., Bailey, L. L., and Hines,
17334 J. E. (2006), *Occupancy estimation and modeling: inferring patterns and dynamics of
17335 species occurrence*, Academic Press.
- 17336 Maffei, L. and Noss, A. J. (2008), "How small is too small? Camera trap survey areas and
17337 density estimates for ocelots in the Bolivian Chaco," *Biotropica*, 40, 71–75.
- 17338 Magoun, A. J., Long, C. D., Schwartz, M. K., Pilgrim, K. L., Lowell, R. E., and Valken-
17339 burg, P. (2011), "Integrating motion-detection cameras and hair snags for wolverine
17340 identification," *The Journal of Wildlife Management*, 75, 731–739.
- 17341 Manel, S., Schwartz, M. K., Luikart, G., and Taberlet, P. (2003), "Landscape genetics:
17342 combining landscape ecology and population genetics," *Trends in Ecology & Evolution*,
17343 18, 189–197.
- 17344 Manly, B., McDonald, L., Thomas, D., McDonald, T., and Erickson, W. (2002), *Resource
17345 selection by animals: statistical design and analysis for field studies*, Springer, 2nd ed.
- 17346 Marques, T., Buckland, S., Borchers, D., Tosh, D., and McDonald, R. (2010), "Point

- transect sampling along linear features," *Biometrics*, 66, 1247–1255.
- Marques, T., Thomas, L., Ward, J., DiMarzio, N., and Tyack, P. (2009), "Estimating cetacean population density using fixed passive acoustic sensors: An example with Blainvilles beaked whales," *The Journal of the Acoustical Society of America*, 125, 1982.
- Marques, T. A., Thomas, L., and Royle, J. A. (2011), "A hierarchical model for spatial capture-recapture data: Comment," *Ecology*, 92, 526–528.
- Matechou, E., Morgan, B. J. T., Pledger, S., Collazo, J. A., and Lyons, J. E. (2013), "Integrated Analysis of Capture-Recapture-Resighting Data and Counts of Unmarked Birds at Stop-Over Sites," *Journal of Agricultural, Biological, and Environmental Statistics*.
- Matthysen, E. (2005), "Density-dependent dispersal in birds and mammals," *Ecography*, 28, 403–416.
- McCarthy, M. A. (2007), *Bayesian Methods for Ecology*, Cambridge: Cambridge University Press.
- McClintock, B. and Hoeting, J. (2010), "Bayesian analysis of abundance for binomial sighting data with unknown number of marked individuals," *Environmental and Ecological Statistics*, 17, 317–332.
- McClintock, B., King, Thomas, Matthiopoulos, McConnell, and Morales (2012), "A general discrete-time modeling framework for animal movement using multi-state random walks," *Ecological Monographs*, 82, 335–349.
- McClintock, B. and White, G. (2012), "From NOREMARK to MARK: software for estimating demographic parameters using mark-resight methodology," *Journal of Ornithology*, 152, 641–650.
- McClintock, B., White, G., Antolin, M., and Tripp, D. (2009a), "Estimating abundance using mark-resight when sampling is with replacement or the number of marked individuals is unknown," *Biometrics*, 65, 237–246.
- McClintock, B., White, G., and Burnham, K. (2006), "A robust design mark-resight abundance estimator allowing heterogeneity in resighting probabilities," *Journal of agricultural, biological, and environmental statistics*, 11, 231–248.
- McClintock, B. T., Conn, P., Alonso, R., and Crooks, K. R. (In press), "Integrated modeling of bilateral photo-identification data in mark-recapture analyses." *Ecology*, <http://dx.doi.org/10.1890/12-1613.1>.
- McClintock, B. T. and White, G. C. (2009), "A less field-intensive robust design for estimating demographic parameters with markresight data," *Ecology*, 90, 313–320.
- McClintock, B. T., White, G. C., Burnham, K. P., and Pryde, M. A. (2009b), "A generalized mixed effects model of abundance for mark-resight data when sampling is without replacement," in *Modeling demographic processes in marked populations*, ed. Thomson, D., New York: Springer, pp. 271–289.
- McCullagh, P. and Nelder, J. A. (1989), *Generalized linear models*, Chapman & Hall/CRC.
- McRae, B. H. and Beier, P. (2007), "Circuit theory predicts gene flow in plant and animal populations," *Proceedings of the National Academy of Sciences*, 104, 19885–19890.
- McRae, B. H., Dickson, B. G., Keitt, T. H., and Shah, V. B. (2008), "Using circuit theory to model connectivity in ecology, evolution, and conservation," *Ecology*, 89, 2712–2724.
- Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., Teller, E., et al. (1953), "Equation of state calculations by fast computing machines," *The journal of chemical physics*, 21, 1087–1092.

- 17393 Metropolis, N. and Ulam, S. (1949), "The Monte Carlo method," *Journal of the American
17394 Statistical Association*, 44, 335–341.
- 17395 Meyer, R. K. and Nachtsheim, C. J. (1995), "The coordinate-exchange algorithm for
17396 construction exact optimal experimental designs," *Technometrics*, 37(1), 60–69.
- 17397 Millar, R. B. (2009), "Comparison of hierarchical Bayesian models for overdispersed count
17398 data using DIC and Bayes' Factors," *Biometrics*, 65, 962–969.
- 17399 Mills, L. S., Citta, J. J., Lair, K. P., Schwartz, M. K., and Tallmon, D. A. (2000), "Es-
17400 timating animal abundance using noninvasive DNA sampling: promise and pitfalls,"
17401 *Ecological Applications*, 10, 283–294.
- 17402 Minta, S. and Mangel, M. (1989), "A Simple Population Estimate Based on Simulation
17403 for Capture-Recapture and Capture-Resight Data," *Ecology*, 70, 1738–1751.
- 17404 Mitchell, T. J. (1974), "An algorithm for the construction of 'D-optimal' experimental
17405 designs," *Techometrics*, 203–210.
- 17406 Mohr, C. (1947), "Table of equivalent populations of North American small mammals,"
17407 *American midland naturalist*, 37, 223–249.
- 17408 Molinari-Jobin, A., Kéry, M., Marboutin, E., Marucco, F., Zimmermann, F., Molinari, P.,
17409 Frick, H., Wölfl, S., Bled, F., Breitenmoser-Würsten, C., Fuxjäger, C., Huber, T., I.,
17410 K., Kos, I., Manfred Wölfl, M., and Breitenmoser, U. (2013), "Mapping range dynamics
17411 from opportunistic data: spatio-temporal distribution modeling of lynx Lynx lynx L.
17412 in the Alps," *Biological Conservation*, xx, xxxx–xxxx.
- 17413 Mollet, P., Kéry, M., Gardner, B., Pasinelli, G., and A, R. J. (In review), "Population size
17414 estimation for capercaillie (*Tetrao urogallus L.*) using DNA-based individual recognition
17415 and spatial capture-recapture models," missing, missing, missing.
- 17416 Morrison, M. L., Strickland, M. D., Block, W. M., Collier, B. A., and Peterson, M. J.
17417 (2008), *Wildlife Study Design*, Springer.
- 17418 Müller, W. G. (2007), *Collecting spatial data: Optimum design of experiments for random
17419 fields*, Springer.
- 17420 Murdoch, W. W. (1994), "Population redulation in theory and practice," *Ecology*, 75,
17421 271–287.
- 17422 Neal, A., White, G., Gill, R., Reed, D., and Olterman, J. (1993), "Evaluation of mark-
17423 resight model assumptions for estimating mountain sheep numbers," *Journal of Wildlife
17424 Management*, 57, 436–450.
- 17425 Neal, A. K. (1990), "Evaluation of mark-resight population estimates using simulations
17426 and field data from mountain sheep." M.S. thesis, Colorado State University, Fort
17427 Collins, Colorado, USA.
- 17428 Neal, R. (2003), "Slice sampling," *Annals of Statistics*, 31, 705–741.
- 17429 Nelder, J. A. and Wedderburn, R. W. M. (1972), "Generalized linear models," *Journal of
17430 the Royal Statistical Society. Series A (General)*, 135, 370–384.
- 17431 Nichols, J. and Pollock, K. (1990), "Estimation of recruitment from immigration versus
17432 in situ reproduction using Pollock's robust design," *Ecology*, 21–26.
- 17433 Nichols, J., Thomas, L., and Conn, P. (2009), "Inferences about landbird abundance from
17434 count data: recent advances and future directions," *Modeling demographic processes in
17435 marked populations*, 201–235.
- 17436 Nichols, J. D., Hines, J. E., Lebreton, J.-D., and Pradel, R. (2000a), "Estimation of con-
17437 tributions to population growth: a reverse-time capture-recapture approach," *Ecology*,
17438 81 (2), 3362–3376.

- 17439 Nichols, J. D., Hines, J. E., Pollock, K. H., Hinz, R. L., and Link, W. A. (1994), "Estimating Breeding Proportions and Testing Hypotheses about Costs of Reproduction
17440 with Capture-Recapture Data," *Ecology*, 75, 2052–2065.
17441
- 17442 Nichols, J. D., Hines, J. E., Sauer, J. R., Fallon, F. W., Fallon, J. E., and Heglund,
17443 P. J. (2000b), "A double-observer approach for estimating detection probability and
17444 abundance from point counts," *The Auk*, 117, 393–408.
17445 Nichols, J. D. and Karanth, K. U. (2002), "Statistical concepts: assessing spatial distri-
17446 butions," in *Monitoring tigers and their prey: a manual for researchers, managers and*
17447 *conservationists in Tropical Asia*, eds. Karanth, K. U. and Nichols, J. D., Bangalore,
17448 India: Centre for Wildlife Studies, pp. 29–38.
17449 Niemi, A. and Fernández, C. (2010), "Bayesian Spatial Point Process Modeling of Line
17450 Transect Data," *Journal of Agricultural, Biological, and Environmental Statistics*, 15,
17451 327–345.
17452 Norris, J. L. and Pollock, K. H. (1996), "Nonparametric MLE under two closed capture-
17453 recapture models with heterogeneity," *Biometrics*, 52, 639–649.
17454 Nowak, R. M. (1999), *Walker's Mammals of the World, 6th Edition, Volume I*, Baltimore:
17455 John's Hopkins University Press, first printing ed.
17456 Nychka, D., Yang, Q., and Royle, J. A. (1997), "Constructing spatial designs for moni-
17457 toring air pollution using regression subset selection," *Statistics for the Environment 3: Pollution*
17458 *Assessment and Control*, 131–154.
17459 O'Brien, T. (2011), "Abundance, density and relative abundance: A conceptual frame-
17460 work," in *Camera traps in animal ecology: methods and analyses*, eds. O'Connell, A.
17461 F. J., Nichols, J. D., and Karanth, U., Tokyo, Japan: Springer Verlag, pp. 71–96.
17462 O'Connell, A. F., Nichols, J. D., and Karanth, U. K. (2010), *Camera traps in animal*
17463 *ecology: Methods and analyses*, Springer.
17464 O'Hara, R. and Sillanpää, M. (2009), "A review of Bayesian variable selection methods:
17465 what, how and which," *Bayesian Analysis*, 4, 85–118.
17466 Otis, D. L., Burnham, K. P., White, G. C., and Anderson, D. R. (1978), "Statistical
17467 inference from capture data on closed animal populations," *Wildlife monographs*, 3–
17468 135.
17469 Ovaskainen, O. (2004), "Habitat-specific movement parameters estimated using mark-
17470 recapture data and a diffusion model," *Ecology*, 85, 242–257.
17471 Ovaskainen, O., Rekola, H., Meyke, E., and Arjas, E. (2008), "Bayesian methods for
17472 analyzing movements in heterogeneous landscapes from mark-recapture data," *Ecology*,
17473 89, 542–554.
17474 Parmenter, R. R. and MacMahon, J. A. (1989), "Animal Density Estiamtion Using a
17475 Trapping Web Design: Field Validation Experiments," *Ecology*, 70, 169–179.
17476 Parmenter, R. R., Yates, T. L., Anderson, D. R., Burnham, K. P., Dunnum, J. L., Franklin,
17477 A. B., Friggens, M. T., Lubow, B. C., Miller, M., Olson, G. S., and Others (2003),
17478 "Small-mammal density estimation: A field comparison of grid-based vs. web-based
17479 density estimators," *Ecological Monographs*, 73, 1–26.
17480 Patterson, T., Thomas, L., Wilcox, C., Ovaskainen, O., and Matthiopoulos, J. (2008),
17481 "State-space models of individual animal movement," *Trends in ecology & evolution*,
17482 23, 87–94.
17483 Paviolo, A., De Angelo, C., Di Blanco, Y., and Di Bitetti, M. (2008), "Jaguar *Panthera*
17484 *onca* population decline in the Upper Paraná Atlantic Forest of Argentina and Brazil,"

- 17485 *Oryx*, 42, 554.
- 17486 Paviolo, A., Di Blanco, Y., De Angelo, C., and Di Bitetti, M. (2009), "Protection af-
17487 fects the abundance and activity patterns of pumas in the Atlantic Forest," *Journal of*
17488 *Mammalogy*, 90, 926–934.
- 17489 Pebesma, E. and Bivand, R. (2011), *Package 'sp'*, r package version 0.9-91.
- 17490 Pledger, S. (2004), "Unified maximum likelihood estimates for closed capture–recapture
17491 models using mixtures," *Biometrics*, 56, 434–442.
- 17492 Pledger, S., Efford, M., Pollock, K., Collazo, J., and Lyons, J. (2009), "Stopover duration
17493 analysis with departure probability dependent on unknown time since arrival," *Modeling*
17494 *demographic processes in marked populations*, 3, 349–363.
- 17495 Plummer, M. (2003), "JAGS: A program for analysis of Bayesian graphical models us-
17496 ing Gibbs sampling," in *Proceedings of the 3rd International Workshop on Distributed*
17497 *Statistical Computing (DSC 2003)*. March, pp. 20–22.
- 17498 — (2009), "rjags: Bayesian graphical models using mcmc. R package version 1.0 3-12," .
- 17499 — (2011), "rjags: Bayesian graphical models using mcmc. R package version 3-5," .
- 17500 Plummer, M., Best, N., Cowles, K., and Vines, K. (2006), "CODA: Convergence Diagnosis
17501 and Output Analysis for MCMC," *R News*, 6, 7–11.
- 17502 Pollock, K. H. (1982), "A Capture-Recapture Design Robust to Unequal Probability of
17503 Capture." *Journal of Wildlife Management*, 46, 752–757.
- 17504 Pollock, K. H., Nichols, J. D., Brownie, C., and Hines, J. E. (1990), "Statistical inference
17505 for capture-recapture experiments," *Wildlife monographs*, 3–97.
- 17506 Porneluzi, P. A. and Faaborg, J. (1999), "Season long fecundity, survival, and viability
17507 of Ovenbirds in fragmented and unfrangmented landscapes," *Conservation Biology*, 13,
17508 1151–1161.
- 17509 Pradel, R. (1996), "Utilization of Capture-Mark-Recapture for the Study of Recruitment
17510 and Population Growth Rate," *Biometrics*, 52, 703–709.
- 17511 Pradel, R., Hines, J. E., Lebreton, J. D., and Nichols, J. D. (1997), "Capture-Recapture
17512 Survival Models Taking Account of Transients," *Biometrics*, 53, 60–72.
- 17513 Raabe, J. (2012), "Factors Influencing Distribution and Survival of Migratory Fishes Fol-
17514 lowing Multiple Low-Head Dam Removals on a North Carolina River. PhD disserta-
17515 tion." Ph.D. thesis, North Carolina State University, Raleigh, NC.
- 17516 Rathbun, S. (1996), "Estimation of Poisson intensity using partially observed concomitant
17517 variables," *Biometrics*, 52, 226–242.
- 17518 Rathbun, S. and Cressie, N. (1994), "A space-time survival point process for a longleaf
17519 pine forest in southern Georgia," *Journal of the American Statistical Association*, 89,
17520 1164–1174.
- 17521 Rathbun, S., Shiffman, S., and Gwaltney, C. (2007), "Modelling the effects of partially
17522 observed covariates on Poisson process intensity," *Biometrika*, 94, 153–165.
- 17523 Reich, B. J., Gardner, B., and Wilting, A. (2012), "A spatial capture-recapture model for
17524 territorial species," *Biometrics in review*, xx, xx–xxx.
- 17525 Robert, C. P. and Casella, G. (2004), *Monte Carlo statistical methods*, New York, USA:
17526 Springer.
- 17527 — (2010), *Introducing Monte Carlo Methods with R*, New York, USA: Springer.
- 17528 Roberts, G. O. and Rosenthal, J. S. (1998), "Optimal scaling of discrete approximations
17529 to Langevin diffusions," *Journal of the Royal Statistical Society: Series B (Statistical*
17530 *Methodology*), 60, 255–268.

- 17531 Rowcliffe, J., Carbone, C., Jansen, P. A., Kays, R., and Kranstauber, B. (2011), "Quantifying the sensitivity of camera traps: an adapted distance sampling approach," *Methods in Ecology and Evolution*, 2, 464–476.
- 17532
- 17533
- 17534 Rowcliffe, J. M., Field, J., Turvey, S. T., and Carbone, C. (2008), "Estimating animal density using camera traps without the need for individual recognition," *Journal of Applied Ecology*, 45, 1228–1236.
- 17535
- 17536
- 17537 Royle, J., Kéry, M., and Guélat, J. (2011a), "Spatial capture-recapture models for search-encounter data," *Methods in Ecology and Evolution*, 2, 602–611.
- 17538
- 17539 Royle, J. A. (2004a), "Generalized estimators of avian abundance from count survey data," *Animal Biodiversity and Conservation*, 27, 375–386.
- 17540
- 17541 —— (2004b), "N-Mixture Models for Estimating Population Size from Spatially Replicated Counts," *Biometrics*, 60, 108–115.
- 17542
- 17543 —— (2006), "Site occupancy models with heterogeneous detection probabilities," *Biometrics*, 62, 97–102.
- 17544
- 17545 —— (2008), "Modeling individual effects in the Cormack–Jolly–Seber model: a state–space formulation," *Biometrics*, 64, 364–370.
- 17546
- 17547 —— (2009a), "Analysis of capture–recapture models with individual covariates using data augmentation," *Biometrics*, 65, 267–274.
- 17548
- 17549 —— (2009b), "Analysis of capture–recapture models with individual covariates using data augmentation," *Biometrics*, 65, 267–274.
- 17550
- 17551 Royle, J. A. and Chandler, R. B. (2012), "Integrating Resource Selection Information with Spatial Capture-Recapture," *arXiv preprint arXiv:1207.3288*.
- 17552
- 17553 Royle, J. A., Chandler, R. B., Gazenski, K. D., and Graves, T. A. (2013), "Spatial capture–recapture for jointly estimating population density and landscape connectivity," *Ecology*, to appear.
- 17554
- 17555
- 17556 Royle, J. A., Chandler, R. B., Sun, C., and Fuller, A. (2012a), "Integrating Resource Selection Information with Spatial Capture-Recapture," *needs updated MEE*.
- 17557
- 17558 Royle, J. A., Chandler, R. B., Yackulic, C., and Nichols, J. D. (2012b), "Likelihood analysis of species occurrence probability from presence-only data for modelling species distributions," *Methods in Ecology and Evolution*, 3, 545–554.
- 17559
- 17560
- 17561 Royle, J. A. and Converse, S. J. (in review), "Hierarchical spatial capture-recapture models: Modeling population density from replicated capture-recapture experiments," *Ecology*.
- 17562
- 17563
- 17564 Royle, J. A., Converse, S. J., and Link, W. A. (2012c), "Data Augmentation for Hierarchical Capture-recapture Models," *arXiv preprint arXiv:1211.5706*.
- 17565
- 17566 Royle, J. A., Dawson, D. K., and Bates, S. (2004), "Modeling abundance effects in distance sampling," *Ecology*, 85, 1591–1597.
- 17567
- 17568 Royle, J. A. and Dorazio, R. M. (2006), "Hierarchical models of animal abundance and occurrence," *Journal of Agricultural, Biological, and Environmental Statistics*, 11, 249–263.
- 17569
- 17570
- 17571 —— (2008), *Hierarchical modeling and inference in ecology: the analysis of data from populations, metapopulations and communities*, Academic Press.
- 17572
- 17573 —— (2012), "Parameter-expanded data augmentation for Bayesian analysis of capture–recapture models," *Journal of Ornithology*, 152, S521–S537.
- 17574
- 17575 Royle, J. A., Dorazio, R. M., and Link, W. A. (2007), "Analysis of multinomial models with unknown index using data augmentation," *Journal of Computational and Graphical*
- 17576

- 17577 *Statistics*, 16, 67–85.
- 17578 Royle, J. A. and Dubovsky, J. A. (2001), “Modeling spatial variation in waterfowl band-recovery data,” *The Journal of wildlife management*, 65, 726–737.
- 17580 Royle, J. A. and Gardner, B. (2011), “Hierarchical models for estimating density from trapping arrays,” in *Camera traps in animal ecology: methods and analyses*, eds. O’Connel, A. F. J., Nichols, J. D., and Karanth, U., Tokyo, Japan: Springer Verlag, pp. 163–190.
- 17583 Royle, J. A., Karanth, K. U., Gopalaswamy, A. M., and Kumar, N. S. (2009a), “Bayesian inference in camera trapping studies for a class of spatial capture-recapture models,” *Ecology*, 90, 3233–3244.
- 17586 Royle, J. A. and Kéry, M. (2007), “A Bayesian state-space formulation of dynamic occupancy models.” *Ecology*, 88, 1813–1823.
- 17588 Royle, J. A. and Link, W. A. (2006), “Generalized site occupancy models allowing for false positive and false negative errors,” *Ecology*, 87, 835–841.
- 17590 Royle, J. A., Magoun, A. J., Gardner, B., Valkenburg, P., and Lowell, R. E. (2011b), “Density estimation in a wolverine population using spatial capture–recapture models,” *The Journal of Wildlife Management*, 75, 604–611.
- 17593 Royle, J. A. and Nichols, J. D. (2003), “Estimating abundance from repeated presence-absence data or point counts,” *Ecology*, 84, 777–790.
- 17595 Royle, J. A., Nichols, J. D., Karanth, K. U., and Gopalaswamy, A. M. (2009b), “A hierarchical model for estimating density in camera-trap studies,” *Journal of Applied Ecology*, 46, 118–127.
- 17598 Royle, J. A. and Nychka, D. (1998), “An algorithm for the construction of spatial coverage designs with implementation in SPLUS,” *Computers and Geosciences*, 24, 479–488.
- 17600 Royle, J. A. and Young, K. V. (2008), “A Hierarchical Model For Spatial Capture-Recapture Data,” *Ecology*, 89, 2281–2289.
- 17602 Russell, R. E., Royle, J. A., Desimone, R., Schwartz, M. K., Edwards, V. L., Pilgrim, K. P., and McKelvey, K. S. (2012), “Estimating abundance of mountain lions from unstructured spatial samples,” *Journal of Wildlife Management*.
- 17605 Rutledge, M. (2013), “Impacts of Resident Canada Geese in a Suburban Environment,” Ph.D. thesis, North Carolina State University.
- 17607 Sacks, J., Welch, W. J., Mitchell, T. P., and Wynn, H. (1989), “Design and analysis of computer experiments,” *Statistical Science*, 4, 409–435.
- 17609 Sæther, B.-E. and Bakke, Ø. (2000), “Avian life history variation and contribution of demographic traits to the population growth rate,” *Ecology*, 81, 642653.
- 17611 Saether, B. E., Engen, S., and Matthysen, E. (2002), “Demographic characteristics and population dynamical patterns of solitary birds,” *Science*, 295, 2070–2073.
- 17613 Saïd, S. and Servany, S. (2005), “The influence of landscape structure on female roe deer home-range size,” *Landscape ecology*, 20, 1003–1012.
- 17615 Salom-Pérez, R., Carrillo, E., Sáenz, J., and Mora, J. (2007), “Critical condition of the jaguar *Panthera onca* population in Corcovado National Park, Costa Rica,” *Oryx*, 41, 51.
- 17618 Sanathanan, L. (1972), “Estimating the size of a multinomial population,” *The Annals of Mathematical Statistics*, 43, 142–152.
- 17620 Sauer, J. R. and Link, W. A. (2002), “Hierarchical modeling of population stability and species group attributes from survey data,” *Ecology*, 83, 1743–1751.
- 17622 Schofield, M. and Barker, R. (2008), “A unified capture-recapture framework,” *Journal of*

- 17623 *agricultural, biological, and environmental statistics*, 13, 458–477.
- 17624 Schwartz, M. K., Copeland, J. P., Anderson, N. J., Squires, J. R., Inman, R. M., McKelvey,
17625 K. S., Pilgrim, K. L., Waits, L. P., and Cushman, S. A. (2009), “Wolverine gene flow
17626 across a narrow climatic niche,” *Ecology*, 90, 3222–3232.
- 17627 Schwartz, M. K. and Monfort, S. L. (2008), *Genetic and endocrine tools for carnivore*
17628 *surveys*, Island Press Washington, DC, USA.
- 17629 Schwarz, C. J. and Arnason, A. N. (1996), “A General Methodology for the Analysis of
17630 Capture-Recapture Experiments in Open Populations,” *Biometrics*, 52, 860–873.
- 17631 —— (2005), “Jolly-Seber Models in MARK,” in *Program MARK: A Gentle Introduction*,
17632 5th ed., eds. Cooch, E. G. and White, G.
- 17633 Schwarz, C. J., Bailey, R. E., Irvine, J. R., and Dalziel, F. C. (1993), “Estimating salmon
17634 spawning escapement using capture-recapture methods.” *Canadian Journal of Fisheries*
17635 *and Aquatic Sciences*, 50, 1181–1191.
- 17636 Seber, G. A. F. (1965), “A Note on the Multiple-Recapture Census,” *Biometrika*, 52,
17637 249–259.
- 17638 —— (1982), *The estimation of animal abundance and related parameters*, Macmillan Pub-
17639 lishing Co.
- 17640 Sepúlveda, M. A., Bartheld, J. L., Monsalve, R., Gómez, V., and Medina-Vogel, G. (2007),
17641 “Habitat use and spatial behaviour of the endangered Southern river otter (*Lontra*
17642 *provocax*) in riparian habitats of Chile: conservation implications,” *Biological Conserva-*
17643 *tion*, 140, 329–338.
- 17644 Shirk, A. J., Wallin, D. O., Cushman, S. A., Rice, C. G., and Warheit, K. I. (2010),
17645 “Inferring landscape effects on gene flow: a new model selection framework,” *Molecular*
17646 *ecology*, 19, 3603–3619.
- 17647 Sillett, S., Chandler, R. B., Royle, J. A., Kéry, M., and Morrison, S. (2012), “Hierarchical
17648 distance sampling models to estimate population size and habitat-specific abundance
17649 of an island endemic,” *Ecological Applications*.
- 17650 Sillett, T., Rodenhouse, N., and Holmes, R. (2004), “Experimentally reducing neighbor
17651 density affects reproduction and behavior of a migratory songbird,” *Ecology*, 85, 2467–
17652 2477.
- 17653 Skalski, J. R., Millspaugh, J. J., and Spencer, R. D. (2005), “Population estimation and
17654 biases in paintball, mark-resight surveys of elk,” *Journal of Wildlife Management*, 69,
17655 1043–1052.
- 17656 Skaug, H. J. and Schweder, T. (1999), “Hazard models for line transect surveys with
17657 independent observers,” *Biometrics*, 55, 29–36.
- 17658 Sklyar, O., Murdoch, D., Smith, M., Eddelbuettel, D., and François, R. (2010), *inline:*
17659 *Inline C, C++, Fortran function calls from R*, r package version 0.3.8.
- 17660 Smith, D. and Smith, M. S. (2006), “Estimation of binary Markov Random Fields using
17661 Markov chain Monte Carlo,” *Journal of Computational and Graphical Statistics*, 15,
17662 1–21.
- 17663 Smith, M. H., Blessing, R., Chelton, J. G., Gentry, J. B., Golley, F. B., and McGinnis,
17664 J. T. (1971), “Determining density for small mammal populations using a grid and
17665 assessment lines,” *Acta Theriologica*, 16, 105–125.
- 17666 Soisalo, M. K. and Cavalcanti, S. M. C. (2006), “Close-up Space in ”Radio-telemetry”,”
17667 *Biological Conservation*, 129, 487–496.
- 17668 Sollmann, R., Furtado, M. M., Gardner, B., Hofer, H., Jacomo, A. T. A., Trres, N. M., and

- 17669 Silveira, L. (2011), "Improving density estimates for elusive carnivores: Accounting for
17670 sex-specific detection and movements using spatial capture-recapture models for jaguars
17671 in central Brazil," *Biological Conservation*, 144, 1017–1024.
- 17672 Sollmann, R., Gardner, B., and Belant, J. L. (2012), "How Does Spatial Study Design
17673 Influence Density Estimates from Spatial Capture-Recapture Models?" *PloS one*, 7,
17674 e34575.
- 17675 Sollmann, R., Gardner, B., Chandler, R. B., Shindle, D., Onorato, D. P., Royle, J. A., and
17676 O'Connell, A. F. (in revision), "Using multiple data sources provides density estimates
17677 for endangered Florida panther," *Journal of Applied Ecology*.
- 17678 Sollmann, R., Gardner, B., Parsons, A., Stocking, J., McClintock, B., Simons, T., Pollock,
17679 K., and O'Connell, A. (2013a), "A spatial mark-resight model augmented with telemetry
17680 data," *Ecology*.
- 17681 Sollmann, R., Mohamed, A., Samejima, H., and Wilting, A. (2013b), "Risky business or
17682 simple solution—Relative abundance indices from camera-trapping," *Biological Conser-*
17683 *vation*, 159, 405–412.
- 17684 Sólymos, P., Lele, S., and Bayne, E. (2012), "Conditional likelihood approach for analyzing
17685 single visit abundance survey data in the presence of zero inflation and detection error,"
17686 *Environmetrics*, 23, 197–205.
- 17687 Spiegelhalter, D., Best, N., Carlin, B., and Van Der Linde, A. (2002), "Bayesian mea-
17688 sures of model complexity and fit," *Journal of the Royal Statistical Society: Series B*
17689 (*Statistical Methodology*), 64, 583–639.
- 17690 Stabler, B. (2006), *shapefiles: Read and Write ESRI Shapefiles*, r package version 0.6.
- 17691 Stanley, T. and Burnham, K. (1999), "A closure test for time-specific capture-recapture
17692 data," *Environmental and Ecological Statistics*, 6, 197–209.
- 17693 Stanley, T. and Richards, J. (2013), "Software review: a program for testing capture-
17694 recapture data for closure," *Wildlife Society Bulletin*, 33, 782–785.
- 17695 Stevens Jr, D. and Olsen, A. (2004), "Spatially balanced sampling of natural resources,"
17696 *Journal of the American Statistical Association*, 99, 262–278.
- 17697 Stevick, P. T., Palsbøll, P. J., Smith, T. D., Bravington, M. V., and Hammond, P. S.
17698 (2001), "Errors in identification using natural markings: rates, sources, and effects on
17699 capturerecapture estimates of abundance," *Canadian Journal of Fisheries and Aquatic
17700 Sciences*, 58, 1861–1870.
- 17701 Stoyan, D. and Penttinen, A. (2000), "Recent Applications of Point Process Methods in
17702 Forestry Statistics," *Statistical Science*, 15, 61–78.
- 17703 Strauss, D. (1975), "A model for clustering." *Biometrika*, 63, 467 – 475.
- 17704 Sturtz, S., Ligges, U., and Gelman, A. (2005), "R2WinBUGS: A Package for Running
17705 WinBUGS from R," *Journal of Statistical Software*, 12, 1–16.
- 17706 Su, Y.-S. and Yajima, M. (2011), *R2jags: A Package for Running jags from R*, r package
17707 version 0.02-17.
- 17708 Sun, C. C. (in prep), "Population estimation, genetic diversity, and structure of black
17709 bears in southwestern New York," Master's thesis, Cornell University.
- 17710 Taberlet, P. and Bouvet, J. (1992), "Bear conservation genetics," *Nature*, 358, 197–197.
- 17711 Takemura, A. (1999), "Some superpopulation models for estimating the number of pop-
17712 ulation uniques," in *Proceedings of the International Conference on Statistical Data
17713 Protection SDP*, Citeseer, vol. 98, pp. 45–58.
- 17714 Tanner, M. A. and Wong, W. H. (1987), "The calculation of posterior distributions by

- 17715 data augmentation," *J Am Stat Assoc*, 82, 528–540.
- 17716 Thomas, A., O'Hara, B., Ligges, U., and Sturtz, S. (2006), "Making BUGS Open," *R
17717 News*, 6, 12–17.
- 17718 Thompson, C., Royle, J. A., and Garner, J. (2012), "A framework for inference about
17719 carnivore density from unstructured spatial sampling of scat using detector dogs," *The
17720 Journal of Wildlife Management*, 76, 863–871.
- 17721 Thompson, S. K. (2002), *Sampling*, New York: Wiley.
- 17722 Tierney, L., Rossini, A. J., Li, N., and Sevcikova, H. (2011), *snow: Simple Network of
17723 Workstations*, r package version 0.3–7.
- 17724 Tilman, D. and Kareiva, P. (1997), *Spatial ecology: the role of space in population dynamics
17725 and interspecific interactions*, vol. 30, Princeton University Press.
- 17726 Tischendorf, L. and Fahrig, L. (2000), "On the usage and measurement of landscape
17727 connectivity," *Oikos*, 90, 7–19.
- 17728 Tischendorf, L., Grez, A., Zaviezo, T., and Fahrig, L. (2005), "Mechanisms affecting
17729 population density in fragmented habitat," *Ecology and Society*, 10, 7.
- 17730 Tobler, M., Carrillo-Percastegui, S., Leite Pitman, R., Mares, R., and Powell, G. (2008),
17731 "An evaluation of camera traps for inventorying large-and medium-sized terrestrial
17732 rainforest mammals," *Animal Conservation*, 11, 169–178.
- 17733 Tobler, M. W., Hibert, F., Debeir, L., and Hansen, C. (2012), "Density and sustainable
17734 harvest estimates for the lowland tapir in the Amazon of French Guiana using a spatial
17735 capture-recapture model," *none yet*, none, none.
- 17736 Tracy, J. A. (2006), "Individual-based movement modeling as a tool for conserving con-
17737 nectivity," in *Connectivity conservation*, eds. Crooks, K. and Sanjayan, M., Cambridge
17738 University Press, pp. 343–368.
- 17739 Trolle, M. and Kéry, M. (2003), "Estimation of ocelot density in the Pantanal using
17740 capture-recapture analysis of camera-trapping data," *Journal of Mammalogy*, 84, 607–
17741 614.
- 17742 — (2005), "Camera-trap study of ocelot and other secretive mammals in the northern
17743 Pantanal," *Mammalia*, 69, 409–416.
- 17744 Tufto, J., Andersen, R., and Linnell, J. (1996), "Habitat use and ecological correlates
17745 of home range size in a small cervid: the roe deer," *Journal of Animal Ecology*, 65,
17746 715–724.
- 17747 Tyre, A. J., Tenhumberg, B., Field, S. A., Niejalke, D., Parris, K., and Possingham,
17748 H. P. (2003), "Improving precision and reducing bias in biological surveys: estimating
17749 false-negative error rates," *Ecological Applications*, 13, 1790–1801.
- 17750 Valiere, N. and Taberlet, P. (2000), "Urine collected in the field as a source of DNA for
17751 species and individual identification," *Molecular Ecology*, 9, 2150–2152.
- 17752 van Etten, J. (2011), *Package gdistance*, R package version 1.1–2.
- 17753 Venables, W. and Ripley, B. (2002), *Modern applied statistics with S*, Springer verlag.
- 17754 Venables, W., Smith, D., and Team, R. D. C. (2012), "An introduction to R," .
- 17755 Ver Hoef, J. and Boveng, P. (2007), "Quasi-poisson vs. negative binomial regression: How
17756 should we model overdispersed count data?" *Ecology*, 88, 2766–2772.
- 17757 Ver Hoef, J. M. (2012), "Who Invented the Delta Method?" *The American Statistician*,
17758 66, 124–127.
- 17759 Wallace, R. B., Gomez, H., Ayala, G., and Espinoza, F. (2003), "Camera trapping for
17760 jaguar (*Panthera onca*) in the Tuichi Valley, Bolivia," *Journal of Neotropical Mammal-*

- 17761 *ogy*, 10, 133–139.
- 17762 Wegan, M., Curtis, P., Rainbolt, R., and Gardner, B. (2012), “Temporal sampling frame
17763 selection in DNA-based capture mark-recapture investigations.” *Ursus*, 23, 42 – 51.
- 17764 Wegan, M. T. (2008), “Aversive conditioning, population estimation, and habitat prefer-
17765 ence of black bears (*Ursus americanus*) on Fort Drum Military Installation in northern
17766 New York,” Ph.D. thesis, Cornell University, Jan.
- 17767 Wegge, P., Pokheral, C. P., and Jnawali, S. R. (2004), “Effects of trapping effort and trap
17768 shyness on estimates of tiger abundance from camera trap studies,” *Animal Conserva-*
17769 *tion*, 7, 251–256.
- 17770 White, G. (1996), “NOREMARK: population estimation from mark-resighting surveys,”
17771 *Wildlife Society Bulletin*, 24, 50–52.
- 17772 White, G. and Bennetts, R. (1996), “Analysis of frequency count data using the negative
17773 binomial distribution,” *Ecology*, 77, 2549–2557.
- 17774 White, G. and Shenk, M. (2001), “Poplation estimation with radio-marked inividuials.” in
17775 *Radio tracking adn animal populations*, eds. Millspaugh, J. and Marzluff, J., San Diego,
17776 USA: Academic Press, pp. 329–350.
- 17777 White, G. C., Anderson, D. R., Burnham, K. P., and Otis, D. (1982), *Capture-recapture*
17778 *and removal methods for sampling closed populations*, Los Alamos: Los Alamos National
17779 Laboratory.
- 17780 White, G. C. and Garrot, R. (1990), *Analysis of wildlife radiolocation data*, New York,
17781 USA: Academic Press.
- 17782 White, G. C. and Shenk, T. M. (2000), *Population estimation with radio-marked animals*,
17783 San Diego, California: Academic Press.
- 17784 Whitman, J., Ballard, W., and Gardner, C. (1986), “Home range and habitat use by
17785 wolverines in southcentral Alaska,” *The Journal of wildlife management*, 460–463.
- 17786 Wickham and Hadley (2007), “Reshaping data with the reshape package,” *Journal of*
17787 *Statistical Software*, 21.
- 17788 Wikle, C. K. (2010), “Hierarchical Modeling with spatial data,” in *Handbook of Spatial*
17789 *Statistics*, eds. Gelfand, A., Diggle, P., Fuentes, M., and Guttorp, P., Chapman and
17790 Hall, pp. 89–106.
- 17791 Williams, B. K., Nichols, J. D., and Conroy, M. J. (2002), *Analysis and management of*
17792 *animal populations: modeling, estimation, and decision making*, Academic Pr.
- 17793 Wilson, K. R. and Anderson, D. R. (1985a), “Evaluation of a Density Estimator Based
17794 on a Trapping Web and Distance Sampling Theory,” *Ecology*, 66, 1185–1194.
- 17795 — (1985b), “Evaluation of Two Density Estimators of Small Mammal Population Size,”
17796 *Journal of Mammalogy*, 66, 13–21.
- 17797 With, K. and Crist, T. (1995), “Critical thresholds in species’ responses to landscape
17798 structure,” *Ecology*, 76, 2446–2459.
- 17799 Wolpert, R. L. and Ickstadt, K. (1998), “Poisson/gamma random field models for spatial
17800 statistics,” *Biometrika*, 85, 251 –267.
- 17801 Woods, J. G., Paetkau, D., Lewis, D., McLellan, B. N., Proctor, M., and Strobeck, C.
17802 (1999), “Genetic tagging of free-ranging black and brown bears,” *Wildlife Society Bul-*
17803 *letin*, 27, 616–627.
- 17804 Wright, J., Barker, R., Schofield, M., Frantz, A., Byrom, A., and Gleeson, D. (2009), “In-
17805 corporating Genotype Uncertainty into Mark–Recapture-Type Models For Estimating
17806 Abundance Using DNA Samples,” *Biometrics*, 65, 833–840.

- 17807 Wynn, H. P. (1970), "The sequential generation of D-optimum experimental designs,"
17808 *Annals of Mathematical Statistics*, 41, 1655–1664.
- 17809 Yang, H. C. and Chao, A. (2005), "Modeling Animals' Behavioral Response by Markov
17810 Chain Models for Capture–Recapture Experiments," *Biometrics*, 61, 1010–1017.
- 17811 Yoshizaki, J., Pollock, K. H., Brownie, C., and Webster, R. A. (2009), "Modeling misiden-
17812 tification errors in capture-recapture studies using photographic identification of evolv-
17813 ing marks," *Ecology*, 90, 3–9.
- 17814 Zeller, K., McGarigal, K., and Whiteley, A. (2012), "Estimating landscape resistance to
17815 movement: a review," *Landscape Ecology*, 27, 777–797.
- 17816 Zuur, A. F., Ieno, E. N., Walker, N. J., Saveliev, A. A., and Smith, G. M. (2009), *Mixed
17817 effects models and extensions in ecology with R*, Springer Verlag.
- 17818 Zylstra, E., Steidl, R., and Swann, D. (2010), "Evaluating survey methods for monitoring
17819 a rare vertebrate, the Sonoran desert tortoise," *The Journal of Wildlife Management*,
17820 74, 1311–1318.