

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	Preface	12
13	Acknowledgments	13
14	I Background and Concepts	1
15	1 Introduction	3
16	1.1 The Study of Populations by Capture-Recapture	4
17	1.2 Lions and Tigers and Bears, Oh My: Genesis of Spatial Capture-Recapture Data	5
19	1.2.1 Camera trapping	5
20	1.2.2 DNA sampling	6
21	1.2.3 Acoustic sampling	6
22	1.2.4 Search-encounter methods	6
23	1.3 Capture-Recapture for Modeling Encounter Probability	7
24	1.3.1 Example: Fort Drum bear study	7
25	1.3.2 Inadequacy of non-spatial capture-recapture	10
26	1.4 Historical Context: A Brief Synopsis	11
27	1.4.1 Buffering	11
28	1.4.2 Temporary emigration	12
29	1.5 Extension of Closed Population Models	13
30	1.5.1 Towards spatial explicitness: Efford's formulation	14
31	1.5.2 Abundance as the aggregation of a point process	14
32	1.5.3 The activity center concept	15
33	1.5.4 The state-space	15
34	1.5.5 Abundance and density	16
35	1.6 Characterization of SCR models	16
36	1.7 Summary and Outlook	17
37	2 Statistical Models and SCR	21
38	2.1 Random Variables and Probability Distributions	22
39	2.1.1 Stochasticity in ecology	22
40	2.1.2 Properties of probability distributions	24

41	2.2	Common Probability Distributions	27
42	2.2.1	The binomial distribution	27
43	2.2.2	The Bernoulli distribution	29
44	2.2.3	The multinomial and categorical distributions	30
45	2.2.4	The Poisson distribution	31
46	2.2.5	The uniform distribution	32
47	2.2.6	Other distributions	33
48	2.3	Statistical Inference and Parameter Estimation	34
49	2.4	Joint, Marginal, and Conditional Distributions	37
50	2.5	Hierarchical Models and Inference	40
51	2.6	Characterization of SCR Models	42
52	2.7	Summary and Outlook	46
53	3	Closed Population Models	47
54	3.1	The Simplest Closed Population Model: Model M_0	48
55	3.1.1	The core capture-recapture assumptions	50
56	3.1.2	Conditional likelihood	51
57	3.2	Data Augmentation	52
58	3.2.1	DA links occupancy models and closed population models . .	53
59	3.2.2	Model M_0 in BUGS	55
60	3.2.3	Formal development of data augmentation (DA)	56
61	3.2.4	Remarks on data augmentation	58
62	3.2.5	Example: Black bear study on Fort Drum	59
63	3.3	Temporally Varying and Behavioral Effects	63
64	3.4	Models with Individual Heterogeneity	64
65	3.4.1	Analysis of model M_h	66
66	3.4.2	Analysis of the Fort Drum data with model M_h	67
67	3.4.3	Comparison with MLE	69
68	3.5	Individual Covariate Models: Toward Spatial Capture-Recapture .	70
69	3.5.1	Example: Location of capture as a covariate	72
70	3.5.2	Fort Drum bear study	73
71	3.5.3	Extension of the model	76
72	3.5.4	Invariance of density to B	79
73	3.5.5	Toward fully spatial capture-recapture models	79
74	3.6	Distance Sampling: A Primitive SCR Model	80
75	3.6.1	Example: Sonoran desert tortoise study	82
76	3.7	Summary and Outlook	84
77	II	Basic SCR Models	87
78	4	Fully Spatial Capture-Recapture Models	89
79	4.1	Sampling Design and Data Structure	90

80	4.2	The binomial observation model	91
81	4.2.1	Definition of home range center	93
82	4.2.2	Distance as a latent variable	93
83	4.3	The Binomial Point Process Model	94
84	4.3.1	The state-space of the point process	96
85	4.3.2	Connection to model M_h and distance sampling	98
86	4.4	The Implied Model of Space Usage	99
87	4.4.1	Bivariate normal case	101
88	4.4.2	Empirical analysis	102
89	4.4.3	Relevance of understanding space usage	103
90	4.4.4	Contamination due to behavioral response	104
91	4.5	Simulating SCR Data	104
92	4.5.1	Formatting and manipulating real data sets	106
93	4.6	Fitting Model SCR0 in BUGS	107
94	4.7	Unknown N	109
95	4.7.1	Analysis using data augmentation in WinBUGS	111
96	4.7.2	Implied home range area	114
97	4.7.3	Realized and expected density	114
98	4.8	The Core SCR Assumptions	116
99	4.9	Wolverine Camera Trapping Study	118
100	4.9.1	Practical data organization	118
101	4.9.2	Fitting the model in WinBUGS	122
102	4.9.3	Summary of the wolverine analysis	123
103	4.9.4	Wolverine space usage	124
104	4.10	Using a Discrete Habitat Mask	125
105	4.10.1	Evaluation of coarseness of habitat mask	126
106	4.10.2	Analysis of the wolverine camera trapping data	128
107	4.11	Summarizing Density and Activity Center Locations	128
108	4.11.1	Constructing density maps	129
109	4.11.2	Example: Wolverine density map	131
110	4.11.3	Predicting where an individual lives	134
111	4.12	Effective Sample Area	134
112	4.13	Summary and Outlook	135
113	5	Likelihood Analysis of Spatial Capture-Recapture Models	141
114	5.1	MLE with Known N	142
115	5.1.1	Implementation (simulated data)	143
116	5.2	MLE when N is Unknown	147
117	5.2.1	Integrated likelihood under data augmentation	150
118	5.2.2	Extensions	151
119	5.3	Classical Model Selection and Assessment	151
120	5.4	Likelihood Analysis of the Wolverine Camera Trapping Data	152
121	5.4.1	Sensitivity to integration grid and state-space buffer	153

122	5.4.2 Using a habitat mask (Restricted state-space)	154
123	5.5 DENSITY and the R Package secr	156
124	5.5.1 Encounter device types and detection models	158
125	5.5.2 Analysis using the secr package	159
126	5.5.3 Likelihood analysis in the secr package	163
127	5.5.4 Multi-session models in secr	164
128	5.5.5 Some additional capabilities of secr	165
129	5.6 Summary and Outlook	167
130	6 Modeling Encounter Probability	169
131	6.1 Encounter Probability Models	170
132	6.1.1 Bayesian analysis with bear.JAGS	171
133	6.1.2 Bayesian analysis of encounter probability models	172
134	6.2 Modeling Covariate Effects	175
135	6.2.1 Date and time	177
136	6.2.2 Trap-specific covariates	179
137	6.2.3 Behavior or trap response by individual	180
138	6.2.4 Individual covariates	182
139	6.3 Individual Heterogeneity	185
140	6.3.1 Models of heterogeneity	185
141	6.3.2 Heterogeneity induced by variation in home range size	186
142	6.4 Likelihood Analysis in secr	188
143	6.4.1 Notes for fitting standard models	188
144	6.4.2 Sex effects	189
145	6.4.3 Individual heterogeneity	190
146	6.4.4 Model selection in secr using AIC	190
147	6.5 Summary and Outlook	191
148	7 Model Selection and Assessment	193
149	7.1 Model Selection by AIC	194
150	7.1.1 AIC analysis of the wolverine data	195
151	7.2 Bayesian Model Selection	198
152	7.2.1 Model selection by DIC	199
153	7.2.2 DIC analysis of the wolverine data	200
154	7.2.3 Bayesian model averaging with indicator variables	202
155	7.2.4 Choosing among detection functions	205
156	7.3 Evaluating Goodness-of-Fit	206
157	7.4 The Two Components of Model Fit	208
158	7.4.1 Testing uniformity or spatial randomness	208
159	7.4.2 Assessing fit of the observation model	211
160	7.4.3 Does the SCR model fit the wolverine data?	213
161	7.5 Quantifying Lack-of-fit and Remediation	216
162	7.6 Summary and Outlook	217

163	8 Alternative Observation Models	219
164	8.1 Poisson Observation Model	220
165	8.1.1 Poisson model of space usage	221
166	8.1.2 Poisson relationship to the Bernoulli model	222
167	8.1.3 A cautionary note on modeling encounter frequencies	223
168	8.1.4 Analysis of the Poisson SCR model in BUGS	225
169	8.1.5 Simulating data and fitting the model	225
170	8.1.6 Analysis of the wolverine study data	228
171	8.1.7 Count detector models in the secr package	229
172	8.2 Independent Multinomial Observations	229
173	8.2.1 Multinomial resource selection models	231
174	8.2.2 Simulating data and analysis using JAGS	231
175	8.2.3 Multinomial relationship to the Poisson	235
176	8.2.4 Avian mist-netting example	236
177	8.3 Single-catch traps	242
178	8.3.1 Inference for single-catch systems	243
179	8.3.2 Analysis of Efford's possum trapping data	244
180	8.4 Acoustic sampling	247
181	8.4.1 The signal strength model	248
182	8.4.2 Implementation in secr	250
183	8.4.3 Implementation in BUGS	250
184	8.4.4 Other types of acoustic data	251
185	8.5 Summary and Outlook	252
186	9 Sampling Design	255
187	9.1 General Considerations	255
188	9.1.1 Model-based not design-based	255
189	9.1.2 Sampling space or sampling individuals?	255
190	9.1.3 Scope of inference vs. state-space	255
191	9.2 Study design for (spatial) capture-recapture	255
192	9.3 Trap spacing and array size relative to animal movement	255
193	9.3.1 Example: Black bears from Pictured Rocks National Lakeshore:	255
194	9.3.2 Final musings: SCR models, trap spacing and array size	256
195	9.4 Spacing of traps with telemetered individuals	256
196	9.5 Sampling over large scales	256
197	9.6 Model-based Spatial Design	256
198	9.6.1 Formalization of the Design Problem for SCR Studies	256
199	9.6.2 An Optimal Design Criterion for SCR	256
200	9.6.3 Optimization of the criterion	256
201	9.6.4 Illustration	256
202	9.7 Covariate models	256
203	9.8 Summary and Outlook	256

205	III Advanced SCR Models	257
206	10 Modeling Spatial Variation in Density	259
207	10.1 Homogeneous point process revisited	259
208	10.2 Inhomogeneous point processes	259
209	10.3 Observed Point Processes	260
210	10.4 Fitting inhomogeneous point process SCR models	260
211	10.4.1 Continuous space	260
212	10.4.2 Discrete space	260
213	10.5 Ecological Distance and Density Covariates	260
214	10.6 The Jaguar Data	260
215	10.7 Summary and Outlook	260
216	11 Modeling Landscape Connectivity	263
217	11.1 Shortcomings of Euclidean Distance Models	263
218	11.2 Least-Cost Path Distance	263
219	11.2.1 Example of Computing Cost-weighted distance	263
220	11.3 Simulating SCR Data using Ecological Distance	263
221	11.4 Likelihood Analysis of Ecological Distance Models	263
222	11.4.1 Example of SCR with Least-Cost Path	264
223	11.5 Bayesian Analysis	264
224	11.6 Simulation Evaluation of the MLE	264
225	11.6.1 Simulation Results	264
226	11.7 Distance In an Irregular Patch	264
227	11.7.1 Basic Geographic Analysis in R	264
228	11.8 Summary and Outlook	264
229	12 Integrating Resource Selection with Spatial Capture-Recapture Models	267
230	12.1 A Simple Model of Space Usage	267
231	12.1.1 Poisson use model	267
232	12.1.2 Thinning	267
233	12.1.3 Capture-recapture Data	267
234	12.2 The Joint RSF/SCR Likelihood	267
235	12.3 Application: New York Black Bear Study	267
236	12.4 Simulation Study	267
237	12.5 Summary and Outlook	267
238	13 Stratified Populations: Multi-session and Multi-site Data	269
239	13.1 Data Structure	269
240	13.2 Multinomial Abundance Models	269
241	13.2.1 Observation Models	270
242	13.2.2 Simulating group structured capture-recapture data	270

244	13.2.3 Fitting in BUGS	270
245	13.2.4 Approach B modeling ψ	270
246	13.3 Spatial Capture-Recapture	270
247	13.4 Application	270
248	13.4.1 Results	270
249	13.5 Topics in Multi-Session models	272
250	13.5.1 Temporal models	272
251	13.5.2 Dependence – is it a problem?	272
252	13.6 Multi-session models in <code>secr</code>	272
253	13.6.1 Ovenbird data in WinBUGS?	272
254	13.6.2 Converse data in <code>secr</code> ?	272
255	13.7 Summary and Outlook	272
256	14 Models for Search-Encounter Data	273
257	14.1 Search-Encounter sampling designs	273
258	14.2 A Model for Search-Encounter Data	273
259	14.2.1 Ecological process model	273
260	14.2.2 Other stuff	273
261	14.3 Examples	273
262	14.3.1 Hard plot boundaries	273
263	14.3.2 Analysis of other protocols	273
264	14.4 Design 3: Ad hoc implementation of Design 1.	273
265	14.5 Capricailie crap	273
266	14.5.1 model	273
267	14.6 Design 4 – no location info	273
268	14.7 Summary and Outlook	273
269	15 Open Population Models	275
270	15.1 Introduction	275
271	15.1.1 Overview of Population Dynamics	275
272	15.1.2 Animal movement related to population demography	275
273	15.1.3 Basic assumptions of JS and CJS models	275
274	15.2 Traditional Jolly-Seber Models	275
275	15.2.1 Data Augmentation for the Jolly-Seber Model	275
276	15.2.2 Mist-netting example	275
277	15.2.3 Shortcomings of the traditional JS models	275
278	15.3 Spatial Jolly-Seber Models	275
279	15.3.1 Mist-netting example	276
280	15.4 Traditional CJS models	276
281	15.4.1 Migratory fish example	276
282	15.5 Multi-state CJS models	276
283	15.5.1 Migratory fish example	276
284	15.6 Spatial CJS models	276

285	15.6.1 Migratory fish example	276
286	15.7 Moving Activity Centers	276
287	15.7.1 Migratory Fish Example Notes	276
288	15.8 Summary and Outlook	276
289	IV Super-Advanced SCR Models	279
290	16 Developing Markov Chain Monte Carlo Samplers	281
291	16.0.1 Why build your own MCMC algorithm?	281
292	16.1 MCMC and posterior distributions	281
293	16.2 Types of MCMC sampling	281
294	16.2.1 Gibbs sampling	281
295	16.2.2 Metropolis-Hastings sampling	282
296	16.2.3 Metropolis-within-Gibbs	282
297	16.2.4 Rejection sampling and slice sampling	282
298	16.3 MCMC for closed capture-recapture Model Mh	282
299	16.4 MCMC algorithm for model SCR0	282
300	16.4.1 SCR model with binomial encounter process	282
301	16.4.2 Looking at model output	282
302	16.4.3 Posterior density plots	282
303	16.4.4 Serial autocorrelation and effective sample size	282
304	16.4.5 Summary results	282
305	16.4.6 Other useful commands	282
306	16.5 Manipulating the state-space	282
307	16.6 Increasing computational speed	282
308	16.6.1 Parallel computing	282
309	16.6.2 Using C++	282
310	16.7 Summary and Outlook	282
311	17 Spatial Capture-Recapture for Unmarked Populations	285
312	17.1 Existing Models for Inference About Density in Unmarked Populations	285
313	17.2 Spatial Correlation as Information	285
314	17.3 Data	285
315	17.4 Model	285
316	17.5 Northern Parula Example	285
317	17.6 Improving Precision with Prior Information	285
318	17.7 Design issues	285
319	17.7.1 How Much Correlation Is Enough?	285
320	17.7.2 Linear Designs	285
321	17.7.3 Quadrat counts	285
322	17.8 Alternative Observation Models	285
323	17.8.1 Spatial point process models	285

324	17.9 Conclusion	285
325	18 Spatial Mark-Resight Models for partially identifiable populations	287
326	18.1 Background	287
327	18.1.1 Types of partial ID data	287
328	18.1.2 A short history of mark-resight models	287
329	18.2 Known number of marked individuals	287
330	18.2.1 MCMC for a spatial mark-resight model	287
331	18.2.2 Binomial encounter model	287
332	18.3 Unknown number of marked individuals	288
333	18.4 Imperfect identification of marked individuals	288
334	18.5 How much information do marked and unmarked individuals contribute?	288
335	18.6 Incorporating telemetry data	288
336	18.7 Summary and Outlook	288
339	19 2012: A Spatial Capture-Recapture Odyssey	291
340	19.1 10 thesis or dissertation topics	292
341	19.2 Three dimesional space	292
342	19.3 Gregarious species	292
343	V Appendices	293
344	19.4 WinBUGS	295
345	19.4.1 WinBUGS through R	295
346	19.5 OpenBUGS	296
347	19.5.1 OpenBUGS through R	296
348	19.6 JAGS	297
349	19.6.1 JAGS through R	298
350	19.7 R	298
351	19.7.1 R packages	298
352	Bibliography	301

355 Acknowledgements
356

357

Part I

358

359

Background and Concepts

360
361

362

1

INTRODUCTION

363 Space plays a vital role in virtually all ecological processes (Tilman and Kareiva,
364 1997; Hanski, 1999; Clobert et al., 2001). The spatial arrangement of habitat can
365 influence movement patterns during dispersal, habitat selection, and survival. The
366 distance between an organism and its competitors and prey can influence activity
367 patterns and foraging behavior. Further, understanding distribution and spatial
368 variation in abundance is necessary in the conservation and management of popu-
369 lations. The inherent spatial aspect of *sampling* populations also plays an important
370 role in ecology as it strongly affects, and biases, how we observe population struc-
371 ture (Seber, 1982; Buckland et al., 2001; Borchers et al., 2002; Williams et al.,
372 2002). However, despite the central role of space and spatial processes to both
373 understanding population dynamics and how we observe or sample populations, a
374 coherent framework that integrates these two aspects of ecological systems has not
375 been fully realized either conceptually or methodologically.

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

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

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

1.1 THE STUDY OF POPULATIONS BY CAPTURE-RECAPTURE

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

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

414 Capture-recapture has been important in studies of animal populations for many
415 decades, and its importance is growing dramatically in response to technological
416 advances that improve our ability and efficiency to obtain encounter history data.
417 Historically, such information was obtainable using methods requiring physical cap-
418 ture of individuals. However, new methods do not require physical capture or
419 handling of individuals. A large number of passive detection devices produce indi-
420 vidual encounter history data including camera traps (Karanth and Nichols, 1998;
421 O'Connell et al., 2010), acoustic recording devices (Dawson and Efford, 2009), and
422 methods that obtain DNA samples such as hair snares for bears, scent posts for
423 many carnivores, and related methods which allow DNA to be extracted from scat,
424 urine or animal tissue in order to identify individuals. This book is concerned with
425 how such data can be used to carry out inference about animal abundance or den-
426 sity, and other parameters such as survival, recruitment, resource selection, and
427 movement using new classes of capture-recapture models which utilize auxiliary
428 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

429 spatial capture-recapture (SCR) models¹.

430 As the name implies, the primary feature of SCR models that distinguishes
431 them from traditional CR methods is that they make use of the spatial information
432 inherent to capture-recapture studies. Encounter histories that are associated with
433 auxiliary information on the location of capture, are *spatial encounter histories*.
434 This auxiliary information is informative about spatial processes including the spa-
435 tial organization of individuals, variation in density, resource selection and space
436 usage, and movement. As we will see, SCR models allow us to overcome critical
437 deficiencies of non-spatial methods, and integrate ecological theory with encounter
438 history data. As a result, this greatly expands the practical utility and scientific
439 relevance of capture-recapture methods, and studies that produce encounter history
440 data.

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

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

455 1.2.1 Camera trapping

456 Considerable recent work has gone into the development of camera-trapping method-
457 ologies. For a historical overview of this method see Kays et al. (2008) and Kucera
458 and Barrett (2011). Several recent synthetic works have been published includ-
459 ing Nichols and Karanth (2002), and an edited volume by O’Connell et al. (2010)
460 devoted solely to camera trapping concepts and methods. As a method for estimat-
461 ing abundance, some of the earliest work that relates to the use of camera trapping
462 data in capture-recapture models originates from Karanth and colleagues (Karanth,
463 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.

464 In camera trapping studies, cameras are often situated along trails or at baited
465 stations and individual animals are photographed and subsequently identified either
466 manually by a person sitting behind a computer, or sometimes now using specific
467 identification software. Camera trapping methods are widely used for species that
468 have unique stripe or spot patterns such as tigers (Karanth, 1995; Karanth and
469 Nichols, 1998), ocelots (*Leopardus pardalis*; (Trolle and Kéry, 2003, 2005)), leopards
470 (*Panthera pardus*; (Balme et al., 2010)), and many other cat species. Camera traps
471 are also used for other species such as wolverines (*Gulo gulo*; (Magoun et al., 2011;
472 Royle et al., 2011b)), and even species that are less easy to identify uniquely such as
473 mountain lions (*Puma concolor*, (Sollmann et al., in revision)) and coyotes (*Canis*
474 *latrans*, (Kelly et al., 2008)). We note that even for species that are not readily
475 identified by pelage patterns, it might be efficient to use camera traps in conjunction
476 with spatial capture-recapture models to estimate density (see Chaps. 17 and 18).

477 **1.2.2 DNA sampling**

478 DNA obtained from hair, blood or scat is now routinely used to obtain individual
479 identity and encounter history information about individuals (Taberlet and Bouvet,
480 1992; Kohn et al., 1999; Woods et al., 1999; Mills et al., 2000; Schwartz and Monfort,
481 2008). A common method is based on the use of “hair snares” (Fig. 1.2) which are
482 widely used to study bear populations (Woods et al., 1999; Garshelis and Hristienko,
483 2006; Kendall et al., 2009; Gardner et al., 2010b). A sample of hair is obtained as
484 individuals pass under or around barbed-wire (or other physical mechanism) to take
485 bait. Hair snares and scent sticks have also been used to sample felid populations
486 (García-Alaníz et al., 2010; Kéry et al., 2010) and other species. Research has
487 even shown that DNA information can be extracted from urine deposited in the
488 wild (e.g., in snow; see Valiere and Taberlet (2000)) and as a result this may prove
489 another future data collection technique where SCR models are useful.

490 **1.2.3 Acoustic sampling**

491 Many studies of birds (Dawson and Efford, 2009), bats, and whales (Marques et al.,
492 2009) now collect data using devices that record vocalizations. When vocalizations
493 can be identified by individual from multiple recording devices, spatial encounter
494 histories are produced that are amenable to the application of SCR models (Dawson
495 and Efford, 2009; Efford et al., 2009b). Recently, these ideas have been applied to
496 data on direction or distance to vocalizations by multiple simultaneous observers
497 and related problems (D. Borchers, ISEC 2012 presentation).

498 **1.2.4 Search-encounter methods**

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

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

These methods don't seem like normal capture-recapture in the sense that the encounter of individuals is not associated with specific trap location, but SCR models are equally relevant for analysis of such data as we discuss in Chapt. 14.

1.3 CAPTURE-RECAPTURE FOR MODELING ENCOUNTER PROBABILITY

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

1.3.1 Example: Fort Drum bear study

Here we confront the simplest possible capture-recapture problem – but one of great applied interest – estimating density from a standard capture-recapture study. We use this as a way to introduce some concepts and motivate the need for spatial capture-recapture models by confronting technical and conceptual problems that we encounter. The data come from a study to estimate black bear abundance on the Fort Drum Military Installation in upstate New York (Wegan (2008), see also Chapt. 3 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.

538 1.4. Barbed wire traps (see Fig. 1.2) were baited and checked for hair samples
539 each week for eight weeks. Analysis of these data appears in Gardner et al. (2009)
540 and Gardner et al. (2010b), and we use the data in a number of analyses in later
541 chapters.

542 Although each bear was captured, or not, in each of the 38 hair snares, we start
543 by treating this data set as a standard capture-recapture data set and summarize
544 to an encounter history matrix with 47 rows and 8 columns with entries y_{ik} , where
545 $y_{ik} = 1$ if individual i was captured, at any trap, in sample occasion k and $y_{ik} = 0$
546 otherwise. There is a standard closed population model, colloquially referred to
547 as “model M_0 ” (see Chapt. 3), which assumes that encounter probability p is
548 constant for all individuals and sample periods. We fitted model M_0 to the Fort
549 Drum data using traditional likelihood methods, yielding the maximum likelihood
550 estimate (MLE) of $\hat{N} = 49.19$ with an asymptotic standard error (SE) of 1.9.

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

564 The mean female home range radius was estimated (Wegan, 2008) for this study
565 region to be 2.19 km, and the area of the convex hull buffered by 2.19 km is
566 277.01 km^2 . (**R** commands to compute the convex hull, buffer it, and compute the
567 area are given in the **R** package **scrbook** which accompanies the book). Hence,
568 the estimated density here is approximately 0.178 bears/ km^2 using the estimated
569 population size obtained by model M_0 . We could assert that the problem has been
570 solved, go home, and have a beer. But then, on the other hand, maybe we should
571 question the use of the estimated home range radius – after all, this is only the
572 female home range radius and the home ranges change for many reasons. Instead,
573 we may decide to rely on a buffer width based on one-half mean maximum distance
574 moved (MMDM) estimated from the actual hair snare data as is more customary
575 (Dice, 1938). In that case the buffer width is 1.19 km, and the resulting estimated
576 density is increased to 0.225 bears/ km^2 about 27 % larger. But wait – some studies
577 actually found the full MMDM (Parmenter et al., 2003) to be a more appropriate
578 measure of movement (e.g. Soisalo and Cavalcanti (2006)). So maybe we should use
579 the full MMDM which is 2.37 km, pretty close to the telemetry-based estimate and
580 therefore providing a similar estimate of density (0.171 bears/ km^2). So in trying to

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

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

```
597 # traps:  1   2   3   4   5   6   9
598 # bears: 23  13   6   2   1   1   1
```

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

619 So what are we left with? Our density estimates span a range from 0.17 to
 620 0.43 bears/km² depending on which estimator of N we use and what buffer strip
 621 we apply. Should we feel strongly about one or the other? Which buffer should

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

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

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

1.3.2 Inadequacy of non-spatial capture-recapture

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

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

for estimating density, a problem we struggled with in the black bear study. (2) Ordinary CR model and sampling methods *induce* a form of heterogeneity that can only at best be approximated by classical models of latent heterogeneity. SCR models formally accommodate heterogeneity due to the juxtaposition of individuals with the encounter devices. (3) Ordinary CR models do not accommodate trap-level covariates which exist in a large proportion of real studies; (4) Ordinary CR models do not accommodate formal consideration of any spatial process that gives rise to the observed data.

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

1.4 HISTORICAL CONTEXT: A BRIEF SYNOPSIS

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

1.4.1 Buffering

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

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

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

693 The idea of using 1/2 mean maximum distance moved ("MMDM" Wilson and
694 Anderson, 1985b) to create a buffer strip seems to be the standard approach even
695 today, presumably justified by Dice's suggestion to use 1/2 the home range diam-
696 eter, with the mean over individuals of the maximum distance moved being an
697 estimator of home range diameter. Alternatively, some studies have used the full
698 MMDM (e.g. Parmenter et al. (2003)), because the trap array might not provide
699 a full coverage of the home range (home ranges near the edge should be trun-
700 cated) and so 1/2 MMDM should be biased smaller than the home range radius.
701 And, sometimes home range size is estimated by telemetry (Karanth, 1995; Bales
702 et al., 2005). Use of MMDM summaries to estimate home range radius is usually
703 combined with an AIC-based selection from among the closed-population models in
704 Otis et al. (1978) which most often suggests heterogeneity in detection (model M_h).
705 Almost all of these early methods were motivated by studies of small mammals us-
706 ing classical "trapping grids" but, more recently, their popularity in the study of
707 wildlife populations has increased with the advent of new technologies, especially
708 related to non-invasive sampling methods such as camera trapping. In particular,
709 the series of papers by Karanth and Nichols (Karanth, 1995; Karanth and Nichols,
710 1998, 2002) has led to fairly widespread adoption of these ideas.

711 **1.4.2 Temporary emigration**

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

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

732 McLellan (2001) use distance-to-edge (DTE) as a covariate in the Huggins-Alho
733 type of model. Ivan (2012) uses this approach in conjunction with an adjustment
734 to the estimated N obtained by estimating the proportion of time individuals are
735 “on the area formally covered by the grid” using radio telemetry. We do not dwell
736 too much on these different variations but we do note that the use of DTE as an
737 individual covariate amounts to some kind of intermediate model between simple
738 closed population models and fully spatial capture-recapture models, which we
739 address directly in Chapt. 3.

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

1.5 EXTENSION OF CLOSED POPULATION MODELS

749 The deficiency with classical closed population models is that they have no spatial
750 context. N is just an integer parameter that applies equally well to estimating the
751 number of unique words in a book, the size of some population that exists in a
752 computer, or a bucket full of goldfish. The question of *where* the N items belong
753 is central both to interpretation of data and estimates from all capture-recapture
754 studies and, in fact, to the construction of spatial capture-recapture models con-
755 sidered in this book. Surely it must matter whether the N items exist as words in
756 a book, or goldfish in a bowl, or tigers in a patch of forest! That classical closed
757 population models have no spatial context leads to a number of conceptual and
758 methodological problems or limitations as we have encountered previously. More
759 important, ecologists seldom care only about N – space is often central to objec-
760 tives of many population studies – movement, space usage, resource selection, how
761 individuals are distributed in space and in response to explicit factors related to
762 landuse or habitat. Because space is central to so many real problems, this is proba-
763 bly the number 1 reason that many ecologists don’t bother with capture-recapture.
764 They haven’t seen capture-recapture methods as being able to solve their problems.
765 Thus, the essential problem is that classical closed population models are too sim-
766 ple – they ignore the spatial attribution of traps and encounter events, movement
767 and variability in exposure of individuals to trap proximity. These problems can be
768 addressed formally by the development of more general capture-recapture models.

769 1.5.1 Towards spatial explicitness: Efford's formulation

770 The solution to the various issues that arise in the application of ordinary capture-
771 recapture models is to extend the closed population model so that N becomes
772 spatially explicit. Efford (2004) was the first to formalize an explicit model for
773 spatial capture-recapture problems in the context of trapping arrays. He adopted
774 a Poisson point process model to describe the distribution of individuals and essen-
775 tially a distance sampling formulation of the observation model which describes the
776 probability of detection as a function of individual location, regarded as a latent
777 variable governed by the point process model. While earlier (and contemporary)
778 methods of estimating density from trap arrays have been ad hoc in the sense of
779 lacking a formal description of the spatial model, Efford achieved a formalization
780 of the model, describing explicit mechanisms governing the spatial distribution of
781 individuals and how they are encountered by traps, but adopted a more or less
782 ad hoc framework for inference under that spatial model using a simulation based
783 method known as inverse prediction (Gopalaswamy, 2012).

784 Recently, there has been a flurry of effort devoted to formalizing inference un-
785 der this model-based framework for the analysis of spatial capture-recapture data
786 (Borchers and Efford, 2008; Royle and Gardner, 2011; Borchers, 2012; Gopalaswamy,
787 2012). There are two distinct lines of work which adopt the model-based formula-
788 tion in terms of the underlying point process but differ primarily by the manner in
789 which inference is achieved. One approach (Borchers and Efford, 2008) uses classi-
790 cal inference based on likelihood (see Chapt. 5), and the other (Royle and Young,
791 2008) adopts a Bayesian framework for inference (Chaps. 4 and 16).

792 1.5.2 Abundance as the aggregation of a point process

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

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

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

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

822 **1.5.3 The activity center concept**

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

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

845 **1.5.4 The state-space**

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

848 individual \mathbf{s}_i are *unknown*. As a technical matter, we will regard them as random
 849 effects and in order to apply standard methods of statistical inference we need to
 850 provide a distribution for these random effects. In the context of the point process
 851 model, the possible values of the point locations referred to as the “state-space” of
 852 the point process and this is some region or set of points which we will denote by
 853 \mathcal{S} . This is analogous to what is sometimes called the *observation window* for \mathbf{s} in
 854 the point process literature. The region \mathcal{S} serves as a prior distribution for \mathbf{s}_i (or,
 855 equivalently, the random effects distribution). In animal studies, as a description
 856 of where individuals that could be captured are located, it includes our study area,
 857 and should accommodate all individuals that could have been captured in the study
 858 area. In the practical application of SCR models, in most cases estimates of density
 859 will be relatively insensitive to choice of state-space which we discuss further in
 860 Chapt. 4 and elsewhere.

861 **1.5.5 Abundance and density**

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

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

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

1.6 CHARACTERIZATION OF SCR MODELS

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

884 At the same time, there are models of CR data that focus exclusively on *movement*
885 modeling, or models with explicit dynamics (Ovaskainen, 2004; Ovaskainen et al.,
886 2008). Conceptually, these are akin to spatial versions of so-called Cormack-Jolly-
887 Seber (CJS) models in the traditional capture-recapture literature, except they
888 involve explicit mathematical models of movement based on diffusion or Brownian
889 motion. Finally, there are now a very small number of papers that focus on *both*
890 movement and density simultaneously (Royle and Young, 2008; Royle et al., 2011a;
891 Royle and Chandler, 2012) or population dynamics and density (Gardner et al.,
892 2010b).

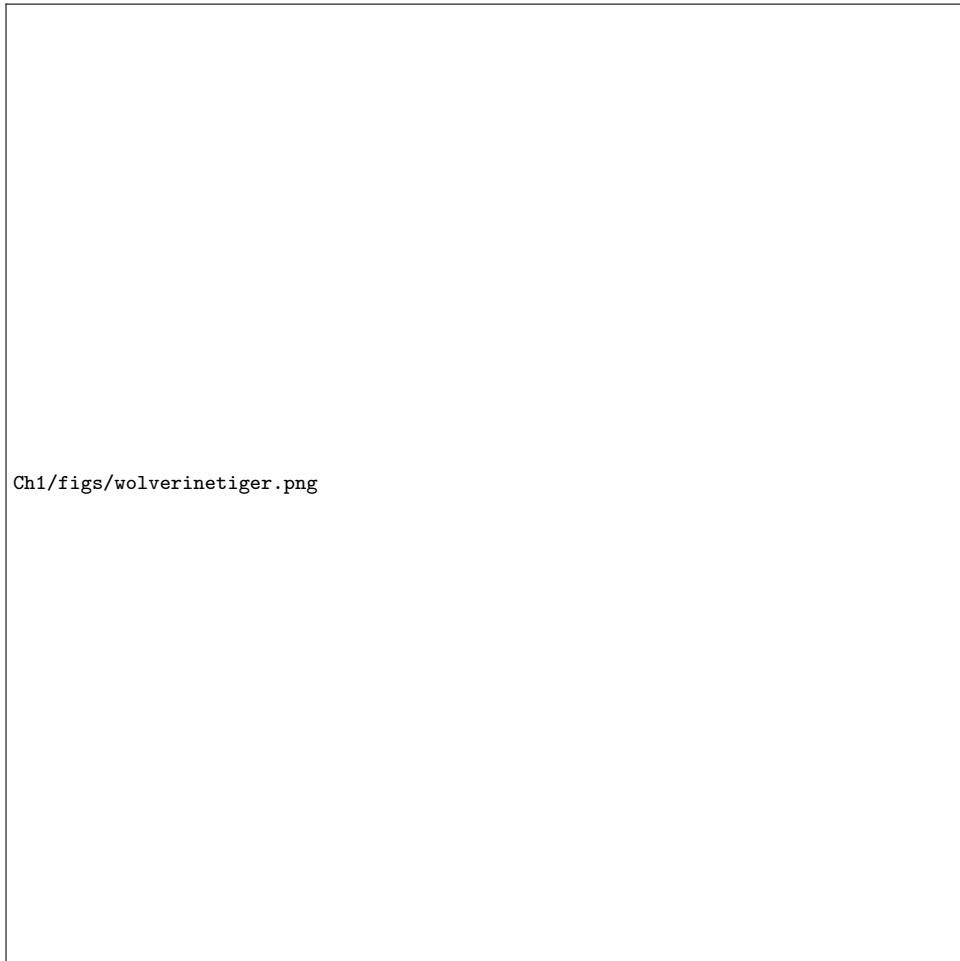
893 A key thing is that these models, whether focused just on density, or just on
894 movement, or both, are similar models in terms of the underlying concepts, the
895 latent structure, and the observation model. They differ primarily in terms of the
896 ecological focus. Understanding movement is an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

1.7 SUMMARY AND OUTLOOK

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

916 As we have shown already, SCR models are not simply an extension of a tech-
917 nique to resolve certain technical problems. Rather, they provide a coherent, flex-
918 ible framework for making ecological processes explicit in models of individual
919 encounter history data, and for studying animal populations processes such as individ-
920 ual movement, resource selection, space usage, population dynamics, and density.
921 Historically, researchers studied these questions independently, using ostensibly un-
922 related study designs and statistical procedures. For example, resource selection
923 function (RSF) models for resource selection, state-space models for movement,

density using closed capture-recapture methods, and population dynamics with various “open” capture-recapture models. SCR can bring all of these problems together into a single unified framework for modeling and inference. Most importantly, spatial capture-recapture models promise the ability to integrate explicit ecological theories directly into the models so that we can directly test hypotheses about either space usage (e.g., Chapt. 12), landscape connectivity (Chapt. 11), movement, or spatial distribution (Chapt. 10). We imagine that, in the near future, SCR models will include point process models that allow for interactions among individuals such as inhibition or clustering (Reich et al., 2012). In the following chapters we develop a comprehensive synthesis and extension of spatial capture-recapture models as they presently exist, and we suggest areas of future development and needed research.



Ch1/figs/wolverinetiger.png

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

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

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

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

936
937

2

938

STATISTICAL MODELS AND SCR

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

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

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

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

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

2.1 RANDOM VARIABLES AND PROBABILITY DISTRIBUTIONS

2.1.1 Stochasticity in ecology

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

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

1004 we need to first mention a few issues related to notation.

1005 Statisticians make things easier for themselves, and more complicated for ev-
 1006 eryone else, by using different notation for probability distributions. Sometimes
 1007 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
 1008 $\pi(X|K, p)$ or $\mathbb{P}(X|K, p)$ or $[X|K, p]$ or even just $[X]!$ Just remember that these
 1009 expressions all have the same meaning—they are all probability distributions that
 1010 tell us the probability of observing any possible realization of the random variable
 1011 X . In this book, we will almost always use bracket notation (the last two examples
 1012 above) to represent arbitrary probability distributions. Hence, from here on out,
 1013 when you see $[X|K, p]$, just remember that this is equivalent to the more traditional
 1014 expression $\Pr(X = x|K, p)$. In addition, from here on, to achieve a more concise
 1015 presentation, we will no longer use uppercase letters to denote random variables
 1016 and lowercase letters for realized values. Rather, we will define a random vari-
 1017 able by some symbol (x , N , etc...) and let the context determine whether we are
 1018 talking about the random variable itself, or realized values of it. In some limited
 1019 cases, we will want upper- and lower-case letters to represent different variables.
 1020 For example, we will often let N denote population size and n denote the number
 1021 of individuals actually detected.

1022 When we wish to be specific about a probability distribution, we will do so in
 1023 one of two ways, one mathematically precise and one symbolic. Before explaining
 1024 these two options, let's choose a specific distribution as a model for the data in our
 1025 example. In this case, the natural choice for $[x|K, p]$ is the binomial distribution,
 1026 the mathematically precise representation of which is

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

1027 The right-hand side of this equation is the binomial pmf (described in more detail
 1028 in Sec. 2.2), and plugging in values for the parameters K , and p will return the
 1029 probability of observing any realized value of the random variable x . This is precise,
 1030 but it is also cumbersome to write repetitively, and it may make the eyes glaze over
 1031 when seen too often. Thus, we will often simplify Eq. 2.1.1 using the symbolic
 1032 notation:

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

1033 The “ \sim ” symbol is meant to represent a stochastic relationship, and can be read
 1034 “is distributed as.” Another reason for using this notation is that it resembles the
 1035 syntax of the **BUGS** language, which we will frequently use to conduct Bayesian
 1036 inference.

1037 Note that once we choose a probability distribution, we have chosen a model. In
 1038 our example, we have specified our model as $x \sim \text{Binomial}(K, p)$, and because we
 1039 are assuming that the parameters are known, we can make probability statements
 1040 about future outcomes. Continuing with our fish example, we might want to know
 1041 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 $p = 0.35$.

1042 outing, assuming that we know $p = 0.35$. Evaluating the binomial pmf returns a
 1043 probability of approximately 0.18, as show using this bit of **R** code:

```
1044 > dbinom(7, 20, 0.35)
1045 [1] 0.1844012
```

1046 By definition, the pmf allows us to evaluate the probability of observing any x given
 1047 $K = 20$ and $p = 0.35$, thus the distribution of the random variable can be visualized
 1048 by evaluating it for all values of x that have non-negligible probabilities, as can be
 1049 easily done in **R**:

```
1050 plot(0:20, dbinom(0:20, 20, 0.35), type="h", ylab="Probability",
1051       xlab="Number of shad caught (X)")
```

1052 the result of which is shown in Fig. 2.1 with some extra details.

1053 The purpose of this little example is to show that once we specify a model for the
 1054 random variable(s) being studied, we can begin drawing conclusions, i.e. making
 1055 inferences, about the processes of interest, even in the face of uncertainty. Prob-
 1056 ability distributions are essential to this process, and thus we need to understand
 1057 them in more depth.

1058 2.1.2 Properties of probability distributions

1059 A pdf or a pmf is a function like any other function in the sense that it has one
 1060 or more arguments whose values determine the result of the function. However,
 1061 probability functions have a few properties that distinguish them from other func-
 1062 tions. The first is that the function must be non-negative for all possible values of
 1063 the random variable, i.e. $[x] \geq 0$. The second requirement is that the integral of
 1064 a pdf must be unity, $\int_{-\infty}^{\infty} [x] dx = 1$, and similarly for a pmf, the summation over
 1065 all possible values is unity, $\sum_x [x] = 1$. The following **R** code demonstrates this for
 1066 the normal and binomial distributions:

```
1067 > integrate(dnorm, -Inf, Inf, mean=0, sd=1)$value
1068 [1] 1
1069 > sum(dbinom(0:5, size=5, p=0.1))
1070 [1] 1
```

1071 This requirement is important to remember when one develops a non-standard
 1072 probability distribution. For example, in Chapt. 10 and 12, we work with resource
 1073 selection functions whose probability density function is not one that is pre-defined
 1074 in software packages such as **R** or **BUGS**.

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

Distribution	Notation	pmf or pmf	Support	Mean $\mathbb{E}(x)$	Variance $\text{Var}(x)$
Discrete random variables					
Poisson	$x \sim \text{Pois}(\lambda)$	$\exp(-\lambda)\lambda^x/x!$	$x \in \{0, 1, \dots\}$	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	$\mathbf{x} \sim \text{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	$(2\pi)^{-k/2} \boldsymbol{\Sigma} ^{-1/2} \exp(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}))$	$x_k \in [-\infty, \infty]$	$\boldsymbol{\mu}$	$\boldsymbol{\Sigma}$

1075 Another feature of probability distributions is that they can be used to compute
 1076 important summaries of random variables. The two most important summaries
 1077 are the expected value, $\mathbb{E}(x)$, and the variance $\text{Var}(x)$. The expected value, or
 1078 mean, can be thought of as the average of a very large sample from the specified
 1079 distribution. For example, one way of approximating the expected values of a
 1080 binomial distribution with $K = 20$ trials and $p = 0.35$ can be implemented in
 1081 **R** using:

```
1082 > mean(rbinom(10000, 20, 0.3))
1083 [1] 6.9865
```

1084 For most probability distributions used in this book, the expected values are known
 1085 exactly, as shown in Table 2.1, and thus we don't need to resort to such Monte Carlo
 1086 approximations. For instance, the expected value of the binomial distribution is
 1087 exactly $\mathbb{E}(x) = Kp = 20 \times 0.35 = 7$. In this case, it happens to take an integer
 1088 value, but this is not a necessary condition, even for discrete random variables.

1089 A more formal definition of an expected value is the average of all possible
 1090 values of the random variable, weighted by their probabilities. For continuous
 1091 random variables, this weighted average is found by integration:

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

1092 For example, if $[x]$ is normally distributed with mean 3 and unit variance, we could
 1093 find the expected value using the following code.

```
1094 > integrate(function(x) x*dnorm(x, 3, 1), -Inf, Inf)
1095 3 with absolute error < 0.00033
```

1096 Of course, the mean *is* the expected value of the normal distribution, so we didn't
 1097 need to compute the integral but, the point is, that Eq. 2.1.3 is generic. For
 1098 discrete random variables, the expected value is found by summation rather than
 1099 integration:

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

1100 where the summation is over all possible values of x . Earlier we approximated the
 1101 expected value of the binomial distribution with $K = 20$ trials and $p = 0.35$ by
 1102 taking a Monte Carlo average. Eq. 2.1.4 let's us find the exact answer, using this
 1103 bit of **R** code:

```
1104 > sum(dbinom(0:100, 20, 0.35)*0:100)
1105 [1] 7
```

1106 This is great. But of what use is it? One very important concept to understand is
 1107 that when we fit models, we are often modeling changes in the expected value of

1108 some random variable. For example, in Poisson regression, we model the expected
 1109 value of the random variable, which may be a function of environmental variables.

1110 The ability to model the expected value of a random variable gets us very far,
 1111 but we also need a model for the variance of the random variable. The variance
 1112 describes the amount of variation around the expected value. Specifically, $\text{Var}(x) =$
 1113 $\mathbb{E}((x - \mathbb{E}(x))^2)$. Clearly, if the variance is zero, the variable is not random as
 1114 there is no uncertainty in its outcome. For some distributions, notably the normal
 1115 distribution, the variance is a parameter to be estimated. Thus, in ordinary linear
 1116 regression, we estimate both the expected value $\mu = \mathbb{E}(x)$, which may be a function
 1117 of covariates, and the variance σ^2 , or similarly the residual standard error σ . For
 1118 other distributions, the variance is not an explicit parameter to be estimated, and
 1119 instead, the mean to variance ratio is fixed. In the case of the Poisson distribution,
 1120 the mean is equal to the variance, $\mathbb{E}(x) = \text{Var}(x) = \lambda$. A similar situation is true
 1121 for the binomial distribution—the variance is determined by the two parameters K
 1122 and p , $\text{Var}(x) = Kp(1 - p)$. In our earlier example with $K = 20$ and $p = 0.35$, the
 1123 variance is 4.55. Toying around with these ideas using random number generators
 1124 may be helpful. Here is some code to illustrate some of these basic concepts:

```
1125 > 20*0.35*(1-0.35)           # Exact variance, Var(x)
1126 [1] 4.55
1127 > x <- rbinom(100000, 20, 0.35)
1128 > mean((x-mean(x))^2)        # Monte Carlo approximation
1129 [1] 4.545525
```

2.2 COMMON PROBABILITY DISTRIBUTIONS

1130 We got a little ahead of ourselves in the previous sections by using the binomial
 1131 and Poisson distributions without describing them in detail. A solid understanding
 1132 of the binomial, Poisson, multinomial, uniform, and normal (or Gaussian) distri-
 1133 butions is absolutely essential throughout the remainder of the book. We will
 1134 occasionally make use of other distributions such as the beta, log-normal, gamma,
 1135 Dirichlet, etc... that can be helpful when modeling capture-recapture data, but
 1136 these distributions can be readily understood once you are comfortable with the
 1137 more commonly used distributions described in this section.

2.2.1 The binomial distribution

1139 The binomial distribution plays a critical role in ecology. It is used for purposes
 1140 as diverse as modeling count data, survival probability, occurrence probability, and
 1141 capture probability, just to name a few. To describe the properties of the binomial
 1142 distribution, and related distributions, we will introduce a new example. Suppose
 1143 we are conducting a bird survey at a site in which $N = 10$ chestnut-sided warblers
 1144 (*Setophaga pensylvanica*) occur, and each of these individuals has a detection prob-
 1145 ability of $p = 0.5$. The binomial distribution is the natural choice for describing

1146 the number of individuals that we would expect to detect (n) in this situation, and
 1147 using our notation, we can write the model as: $n \sim \text{Binomial}(10, 0.5)$. When $p < 1$,
 1148 we can expect that we will observe a different number of warblers on each of K
 1149 replicate survey occasions. To see this, we simulate data under this simple model
 1150 with $K = 3$.

```
1151 > n <- rbinom(3, size=10, prob=0.5) # Generate 3 binomial outcomes
1152 > n                                     # Display the 3 values
1153 [1] 6 4 8
```

1154 The vector of counts will typically differ each time you issue this command; however,
 1155 we know the probability of observing any value of n_k because it is defined by the
 1156 binomial pmf. As we demonstrated earlier, in R this probability can be found using
 1157 the `dbinom` function. For example, the probability of observing $n_k = 5$ is given by:

```
1158 > dbinom(5, 10, 0.5)
```

1159 This simply evaluates the function shown in Table 2.1. We could do the same more
 1160 transparently, but less efficiently, using any of the following:

```
1161 > n <- 5; N <- 10; p <- 0.5
1162 > factorial(N)/(factorial(n)*factorial(N-n))*p^n*(1-p)^(N-n)
1163 > exp(lgamma(N+1) - (lgamma(n+1) + lgamma(N-n+1)))*p^n*(1-p)^(N-n)
1164 > choose(N, n)*p^n*(1-p)^(N-n)
```

1165 Note that the last three lines of code differ only in how they compute the binomial
 1166 coefficient $\binom{N}{n}$, which is the number of different ways we could observe $n = 5$ of
 1167 the $N = 10$ chestnut-sided warblers at the site. The binomial coefficient, which is
 1168 read “N choose n” is defined as

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

1169 Now that we know how to simulate binomial data and compute the probabilities
 1170 of observing any particular outcome n , conditional on the parameters N and
 1171 p , we can contemplate the relevance of the binomial distribution in spatial capture-
 1172 recapture models. One important application of the binomial distribution is as a
 1173 model encounter frequencies. Indeed, one of the most important encounter models
 1174 in SCR will be referred to as the “binomial encounter model”, in which the number
 1175 of times individual i is captured at “trap” j after K survey occasions is modeled as
 1176 $y_{ij} \sim \text{Binomial}(K, p_{ij})$. Here, p_{ij} is the encounter probability determined, in part,
 1177 by the distance between an animal’s activity center and the trap location. This
 1178 binomial encounter model is described in detail in Sec. 6.1. Another important ap-
 1179 plication of the binomial distribution is as a prior for the population size parameter
 1180 in Bayesian analyses, as is discussed in Chapt. 3.

2.2.2 The Bernoulli distribution

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

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

To properly account for variation in p , we could redefine our model for the counts of chestnut-sided warblers as

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

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

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

The Bernoulli pmf is given by $p^n(1 - p)^{1-n}$ and hence we do not need canned functions to facilitate its evaluation. Of course, if you wanted to, you could always

1216 use `dbinom` with the `size` argument set to 1. For example, `dbinom(1, 1, 0.3)`
 1217 returns the Bernoulli probability of observing $n = 1$ given $p = 0.3$.

1218 **2.2.3 The multinomial and categorical distributions**

1219 The binomial distribution is used when we are accumulating a binary response—
 1220 that is, one in which there are two possible categories such as success/failure or
 1221 captured/not-captured. The multinomial distribution is a multivariate extension
 1222 of the binomial used when there are $G > 2$ categories. The multinomial distribution
 1223 can be thought of as a model for placing N items in the G categories, which are
 1224 also called bins or cells. Each bin has its own probability π_g and these probabilities
 1225 must sum to one. In ecology, N is often population size or the number of individuals
 1226 detected, but the definition of the G bins varies among applications. For example,
 1227 in distance sampling, when the distance data are aggregated into intervals, the
 1228 bins are the distance intervals, and the cell probabilities are functions of detection
 1229 probability in each interval (Royle et al., 2004).

1230 The multinomial distribution is widely used to model data from traditional,
 1231 non-spatial capture-recapture studies. Earlier we let y_{ik} denote a binary random
 1232 variable indicating if warbler i was detected on survey k . The vector of observations
 1233 for an individual, \mathbf{y}_i , is often referred to as the individual’s “encounter history”.
 1234 The number of possible encounter histories depends on K , the number of survey
 1235 occasions. Specifically, there are 2^K possible encounter histories¹. If we tabulate the
 1236 number of individuals with each encounter history, the frequencies can be modeled
 1237 using the multinomial distribution.

1238 Going back to our chestnut-sided warbler example, suppose the 10 individuals
 1239 are marked and we make $K = 2$ visits to the site such that there are $2^K = 4$ pos-
 1240 sible encounter histories: (11, 10, 01, 00), where, for example, “10” is the encounter
 1241 history for an individual detected on the first visit but not the second. If $p = 1$,
 1242 then the encounter history for each of the 10 individuals must be “11”. That is, we
 1243 would detect each individual on both occasions. In this case, we the data would be:
 1244 $\mathbf{h} = (10, 0, 0, 0)$, which indicates that all 10 warblers had the first encounter history.
 1245 The corresponding cell probabilities would be $\boldsymbol{\pi} = (1, 0, 0, 0)$. What about the sit-
 1246 uation where $p < 1$, e.g. $p = 0.3$? In this case, the probability of observing the
 1247 capture history “11” (detected on both occasions) is $p \times p = 0.3 \times 0.3 = 0.09$. The
 1248 probability of observing “10” is $p \times (1 - p) = 0.21$. Following this logic, the vector
 1249 of cell probabilities is $\boldsymbol{\pi} = (0.09, 0.21, 0.21, 0.49)$. We can simulate data under this
 1250 model as follows:

```
1251 > caphist.probs <- c("11"=0.09, "10"=0.21, "01"=0.21, "00"=0.49)
1252 > drop(rmultinom(1, 10, caphist.probs))
1253 11 10 01 00
```

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 $2^K - 1$

```
1254 0 3 2 5
```

1255 The result of our simulation is that zero individuals were observed with the capture
 1256 history “11” and 5 individuals were observed with the capture history “00”. The
 1257 other 5 individuals were observed one out of the two occasions. This is not such a
 1258 surprising outcome given $p = 0.3$.

1259 As in non-spatial capture-recapture studies, the multinomial distribution turns
 1260 out to be very important in spatial capture-recapture studies. However, N is not
 1261 defined as population size. Rather, we use the multinomial distribution when an
 1262 individual can only be captured in a single trap during an occasion. Thus $N = 1$
 1263 and the cell probabilities are the probabilities of being captured in each trap. A
 1264 thorough discussion of this point can be found in Chapt. 8. Another application
 1265 of the multinomial distribution in SCR models is discussed in Chapt. 10 where we
 1266 discuss how to model the probability that an individual’s activity center is located
 1267 in one of the cells of a raster defining the spatial region of interest.

1268 Just as the Bernoulli distribution is the elemental form of the binomial distri-
 1269 bution (being the case $N = 1$), the categorical distribution is essentially equivalent
 1270 to the multinomial distribution with size parameter $N \equiv 1$. The only difference is
 1271 that, rather than returning a vector with a single element equal to 1, it returns the
 1272 element *location* where the 1 occurs. For example, if $\mathbf{y} = (0, 0, 1, 0)$ is an outcome
 1273 of a multinomial distribution with $N = 1$, then the categorical outcome would be
 1274 3 because the 1 is located in third position in the vector. Thus, in spatial capture-
 1275 recapture models, we might use either the multinomial distribution with $N = 1$
 1276 or the categorical distribution. The various **BUGS** engines describe the categori-
 1277 cal distribution by the declaration `dcat` and, in **R**, we can simulate categorical
 1278 outcomes using the function `sample` or as so:

```
1279 > which(rmultinom(1, 1, c(0.1, 0.7, 0.2)) == 1)
1280 [1] 2
```

1281 2.2.4 The Poisson distribution

1282 The Poisson distribution is the canonical model for count data in ecology. More
 1283 generally, the Poisson distribution is a model for random variables taking on non-
 1284 negative, integer values. Although it is a simple model having just one parameter,
 1285 $\lambda = \mathbb{E}(x) = \text{Var}(x)$, its applications are highly diverse, including as a model of
 1286 spatial variation in abundance or as a model for the frequency of behaviors over
 1287 time. Just as logistic regression is the standard generalized linear model (GLM)
 1288 used to model binary data, Poisson regression is the default GLM for modeling
 1289 count data and variation in λ .

1290 The Poisson distribution is related to both the binomial and multinomial distri-
 1291 butions, and the following three bits of trivia are occasionally worth knowing. First,
 1292 it is the limit of the binomial distribution as $N \rightarrow \infty$ and $p \rightarrow 0$, which means that
 1293 for high values of N and low values of p , $\text{Poisson}(N \times p)$ is approximately equal

1294 to $\text{Binomial}(N, p)$. Second, if $\{n_1 \sim \text{Poisson}(\lambda_1), \dots, n_K \sim \text{Poisson}(\lambda_K)\}$ then the
 1295 vector of counts is multinomial, $\{n_1, \dots, n_K\} \sim \text{Multinomial}(\sum_k n_k, \{\frac{\lambda_1}{\sum_k \lambda_k}, \dots, \frac{\lambda_K}{\sum_k \lambda_k}\})$.
 1296 Third, the sum of two Poisson random variables $x_1 \sim \text{Poisson}(\lambda_1)$ and $x_2 \sim$
 1297 $\text{Poisson}(\lambda_2)$ is also Poisson: $x_1 + x_2 \sim \text{Poisson}(\lambda_1 + \lambda_2)$.

1298 The Poisson distribution has two important uses in spatial capture-recapture
 1299 models: (1) as a prior distribution for the population size parameter N , and (2) as a
 1300 model for the frequency of captures in a trap. In the first context, the Poisson prior
 1301 for N results in a Poisson point process for the location of the N activity centers
 1302 in the region of interest. This topic is discussed in Chapt. 4 and Chapt 10. The
 1303 second use of the Poisson distribution in spatial capture-recapture is to describe
 1304 data from sampling methods in which an individual can be detected multiple times
 1305 at a trap during a single occasion. For example, in camera trapping studies we
 1306 might obtain multiple pictures of the same individual at a trap during a single
 1307 sampling occasion. Thus, λ in this case would be defined as the expected number
 1308 of detections or captures per occasion.

1309 **2.2.5 The uniform distribution**

1310 The lowly uniform distribution is a continuous distribution whose only two pa-
 1311 rameters are the lower and upper bounds that restrict the possible values of the
 1312 random variable x . These bounds are almost always known, so there is typically
 1313 nothing to estimate. Nonetheless, the uniform distribution is one of the most widely
 1314 used distributions, especially among Bayesians who frequently use it to as a “non-
 1315 informative” prior distribution for a parameter. For example, if we have a capture
 1316 probability parameter p that we wish to estimate, but we have no prior knowl-
 1317 edge of what value it may take in the range $[0,1]$, we will often use the prior
 1318 $p \sim \text{Uniform}(0,1)$. This states that p is equally likely to take on any value between
 1319 zero and one. Prior distributions are described in more detail in the next chapter.

1320 Another common usage of the uniform distribution is as a prior for the coor-
 1321 dinates of points in the real plane, i.e. in two-dimensional space. Such a use of
 1322 the uniform distribution implies that a point process is “homogeneous”, meaning
 1323 that the location of one point does not affect the location of another point and
 1324 that the expected density of points is constant throughout the region. Thus, to
 1325 simulate a realization from a homogeneous Poisson point process in the unit square
 1326 $[0, 1] \times [0, 1]$, we could use the following R code:

```
1327 D <- 100      # points per unit area
1328 A <- 1        # Area of unit square
1329 N <- rpois(1, D*A)
1330 plot(s <- cbind(runif(N), runif(N)))
```

1331 where s is a matrix of coordinates with N rows and 2 columns. We will often
 1332 represent the uniform point process using the following notation:

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

1333 where \mathcal{S} is some specific unit of space called the state-space of the random variable
 1334 s . It would be more correct to somehow distinguish this two-dimensional uniform
 1335 distribution for the univariate one. That is, it might be more clear to use notation
 1336 such as $s \sim \text{Uniform}_2(\mathcal{S})$ instead, but this is somewhat cumbersome, so we will opt
 1337 for the former expression.

1338 2.2.6 Other distributions

1339 The other continuous distributions that are regularly encountered in SCR models
 1340 are primarily used as priors in Bayesian analyses, and thus we will avoid a lengthy
 1341 discussion of their properties. The normal distribution, also called the Gaussian
 1342 distribution, is perhaps the most widely recognized and applied probability model
 1343 in statistics, but it plays only a minor role in SCR models other than as a model for
 1344 signal strength in acoustic SCR models (Efford et al., 2009b; Dawson and Efford,
 1345 2009), and see Sec. 8.4. Nonetheless, it is the canonical prior for any continuous
 1346 random variable with infinite support, and thus it is often used as a prior when
 1347 applying Bayesian methods. One common usage is as a prior for the β coefficients
 1348 of a linear model defining some parameter as a function of covariates (usually on a
 1349 transformed scale). An example, including a cautionary note, is provided in Sec. ??.
 1350 Be aware that although the normal distribution is typically parameterized in terms
 1351 of the variance parameter σ^2 , in the **BUGS** language, the inverse of the variance,
 1352 or precision, is used instead, $\tau = 1/\sigma^2$. In **R**, the `dnorm` function requires the
 1353 standard deviation σ , rather than the variance σ^2 .

1354 The bivariate normal distribution is a generalization of the normal distribution
 1355 and a special case of the multivariate normal distribution whose pdf is shown in
 1356 Table 2.1. The bivariate normal distribution is used to model two (possibly) dependent
 1357 continuous variables whose symmetric variance-covariance matrix is denoted
 1358 Σ . In SCR models, we most often use this model as a rudimentary description of
 1359 movement outcomes about a home range center. If there is no correlation, then the
 1360 model reduces to two independent normal draws along the coordinate axes. The
 1361 following code generates bivariate normal outcomes with no correlation ($\rho = 0$), as
 1362 well as outcomes in which the correlation is $\rho = 0.9$.

```
1363 library(mvtnorm)
1364 set.seed(3)
1365 mu <- c(0,0)
1366 Sigma <- matrix(c(1, .9, .9, 1), 2, 2)
1367 X1 <- cbind(rnorm(50, mu[1], Sigma[1,1]), # No correlation (rho=0)
1368             rnorm(50, mu[2], Sigma[2,2]))
1369 X2 <- rmvnorm(50, mu, Sigma)           # rho=0.9
```

1370 Fig. 2.2 shows the simulated points.

1371 Several of the parameters in capture-recapture models do not have infinite support,
 1372 but instead are probabilities restricted to the range [0, 1], or are positive

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

1373 valued living between zero and ∞ . The beta distribution is the standard prior
 1374 used for probabilities because it can be used to express either a lack of knowledge
 1375 or very precise knowledge about a parameter. For example, a Beta(1,1) distribu-
 1376 tion is equivalent to a Uniform(0,1) distribution. However, unlike the uniform
 1377 distribution, the beta distribution can be used as an informative prior; for exam-
 1378 ple if published estimates of detection probability exist we can choose parameters
 1379 of the beta distribution to reflect that. To gain some familiarity with the beta
 1380 distribution, execute the following R commands:

```
1381 curve(dbeta(x, 1, 1), col="black", ylim=c(0,5))
1382 curve(dbeta(x, 10, 10), col="blue", add=TRUE)
1383 curve(dbeta(x, 10, 20), col="darkgreen", add=TRUE)
```

1384 Other parameters in SCR models are continuous but positive-valued and can be
 1385 modeled using the gamma distribution. As with the beta distribution, the gamma
 1386 distribution is typically favored over the uniform distribution when one is interested
 1387 in using an informative prior. It is also frequently used as a vague prior for the
 1388 inverse of variance parameters, but it is wise to compare this prior to a uniform to
 1389 assess its influence on the posterior.

2.3 STATISTICAL INFERENCE AND PARAMETER ESTIMATION

1390 If the parameters of a statistical model were known with absolute certainty, then it
 1391 would be possible to use pdfs and pmfs to make direct probability statements about
 1392 unknowns such as future outcomes. However, we almost never know the actual
 1393 values of parameters, and instead we have to estimate them from observations
 1394 (i.e., data). Our inferences must then acknowledge the uncertainty associated with
 1395 our imperfect knowledge of the parameters. Doing so is most often accomplished
 1396 using one of two approaches: classical (frequentist) inference or Bayesian inference.
 1397 These two modes of inference regard the uncertainty about parameters in entirely
 1398 different ways. In the next chapter, we will review some of the important concepts
 1399 in Bayesian inference, so here, we will focus on the frequentist perspective.

1400 Suppose we count oak trees at J sites, and the resulting data $\{y_1, \dots, y_J\}$ can
 1401 be assumed to be *iid* outcomes from some distribution, such as the Poisson with
 1402 unknown parameter λ . We want to estimate this parameter. In classical inference,
 1403 the only uncertainty about λ is that attributable to sampling. For instance, we can
 1404 imagine repeatedly sampling the population (sites in this example) and obtaining
 1405 sample-specific estimates of λ . Typically, we entertain the idea that there are an
 1406 infinite number of possible samples and so we could obtain an infinite number of
 1407 estimates: $\{\hat{\lambda}_1, \hat{\lambda}_2, \dots, \hat{\lambda}_\infty\}$. If these estimates are produced using the method

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

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

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

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

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

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

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

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

```

1442 > starting.value <- c('lambda'=1)
1443 > optim(starting.value, negLogLike1)$par # MLE
1444   lambda
1445 3.039844

```

1446 Explicitly maximizing the likelihood, numerically, isn't actually necessary here because the MLE of λ is given by the mean of the observations. A more interesting example is when there are covariates of λ . For example, suppose λ is a function of elevation and vegetation height according to: $\log(\lambda_i) = \beta_0 + \beta_1 \text{ELEV}_i + \beta_2 \text{VEGHT}_i$. This is a standard Poisson regression problem, with likelihood:

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

1451 This likelihood is almost identical to the previous one except that λ is now a function, and so we need to estimate the parameters of the function, i.e. the β 's.
 1452 Some code to fit this model to simulated data is shown here:

```

1454 > nsites <- 100
1455 > elevation <- rnorm(100)
1456 > veght <- rnorm(100)
1457 > beta0 <- 1
1458 > beta1 <- -1
1459 > beta2 <- 0
1460 > lambda <- exp(beta0 + beta1*elevation + beta2*vegght)
1461 > y2 <- rpois(nsites, lambda)
1462 > negLogLike2 <- function(pars) {
1463   +   beta0 <- pars[1]
1464   +   beta1 <- pars[2]
1465   +   beta2 <- pars[3]
1466   +   lambda <- exp(beta0 + beta1*elevation + beta2*vegght)
1467   +   -sum(dpois(y2, lambda, log=TRUE))
1468 }
1469 > starting.values <- c('beta0'=0, 'beta1'=0, 'beta2'=0)
1470 > optim(starting.values, negLogLike2)$par
1471   beta0      beta1      beta2
1472 0.98457756 -1.03025173 -0.01218292

```

1473 We see that the maximum likelihood estimates (MLEs) are very close to the true
 1474 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:

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

1475 This is an example of a mixed effects model or a hierarchical model. How do we
 1476 estimate the parameters of a model that includes random effects? Earlier the like-
 1477 lihood function was written as the product of probabilities determined by a single
 1478 pmf or pdf, $[y|\lambda]$, but now we have an additional random variable, and we are forced
 1479 to think about conditional relationships, because y depends upon β_i and β_i depends
 1480 upon other parameters, specifically μ and σ^2 . This type of conditional dependence
 1481 among parameters is the essence of hierarchical models, and statistical analysis
 1482 of hierarchical models requires that we discuss joint distributions, marginal distri-
 1483 butions and conditional distributions. These concepts will be used extensively in
 1484 Chapt. 5 where we demonstrate how to estimate parameters of hierarchical models
 1485 using maximum likelihood.

2.4 JOINT, MARGINAL, AND CONDITIONAL DISTRIBUTIONS

1486 So far we have restricted our attention to situations in which we wish to make
 1487 inference about a single random variable. However, in ecology, we often are inter-
 1488 ested in multiple random variables and how they are related. Let Y be a random
 1489 variable that may or may not be independent of X (here again we will distinguish
 1490 between random variables and realized values for conceptual clarity). Inference
 1491 about these two random variables can be made using the joint, marginal, or condi-
 1492 tional distributions—or, we may make use of all of them depending on the question
 1493 being asked. In the case of discrete random variables, the joint distribution is the
 1494 probability that X takes on the value x and that Y takes on the value y , which
 1495 is written $[X = x, Y = y]$. To clarify this concept, let's go back to our original
 1496 example where X was the number of fish caught after 20 casts, which we said
 1497 was an *iid* binomial random variable. Now, let's suppose that X depends on the
 1498 random variable Y , which is the number of other fisherman at the hole. Specifi-
 1499 cally, let's say that the probability of catching a fish p is related to Y according
 1500 to $\text{logit}(p) = -0.6 + -2y$. Furthermore, let's make the intuitive assumption that
 1501 the number of fishermen at the hole is a Poisson random variable with mean 0.6,
 1502 i.e. $Y \sim \text{Poisson}(0.6)$. Our model is now fully specified, and so we can answer the
 1503 question: “what is the probability of catching x fish and of there being y fishermen
 1504 at the hole”. This joint distribution is given by the product of the binomial pmf
 1505 (with p determined by y) and the Poisson pmf with $\lambda = 0.6$. The following R code
 1506 creates the joint distribution.

```
1507 > X <- 0:20 # All possible values of X
1508 > Y <- 0:10 # All possible values of Y
```

```

1509 > lambda <- 0.6
1510 > p <- plogis(-0.62 + -2*Y) # p as function of Y
1511 > round(p,2)
1512 [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
1513 > joint <- matrix(NA, length(X), length(Y))
1514 > rownames(joint) <- paste("X=", X, sep="")
1515 > colnames(joint) <- paste("Y=", Y, sep="")
1516 >
1517 > # Joint distribution [X,Y]
1518 > for(i in 1:length(Y)) {
1519 +   joint[,i] <- dbinom(X, 20, p[i]) * dpois(Y[i], lambda)
1520 +
1521 > round(joint,2)
1522     Y=0  Y=1  Y=2  Y=3  Y=4  Y=5  Y=6  Y=7  Y=8  Y=9  Y=10
1523 X=0  0.00 0.08 0.08 0.02  0  0  0  0  0  0  0
1524 X=1  0.00 0.12 0.02 0.00  0  0  0  0  0  0  0
1525 X=2  0.01 0.08 0.00 0.00  0  0  0  0  0  0  0
1526 X=3  0.02 0.04 0.00 0.00  0  0  0  0  0  0  0
1527 X=4  0.04 0.01 0.00 0.00  0  0  0  0  0  0  0
1528 X=5  0.07 0.00 0.00 0.00  0  0  0  0  0  0  0
1529 X=6  0.09 0.00 0.00 0.00  0  0  0  0  0  0  0
1530 X=7  0.10 0.00 0.00 0.00  0  0  0  0  0  0  0
1531 X=8  0.09 0.00 0.00 0.00  0  0  0  0  0  0  0
1532 X=9  0.06 0.00 0.00 0.00  0  0  0  0  0  0  0
1533 X=10 0.04 0.00 0.00 0.00  0  0  0  0  0  0  0
1534 X=11 0.02 0.00 0.00 0.00  0  0  0  0  0  0  0
1535 X=12 0.01 0.00 0.00 0.00  0  0  0  0  0  0  0
1536 X=13 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
1537 X=14 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
1538 X=15 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
1539 X=16 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
1540 X=17 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
1541 X=18 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
1542 X=19 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
1543 X=20 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0

```

1544 This matrix tells us the probability of all possible combinations of x and y , and
 1545 we see that the most likely value is $(X = 1, Y = 1)$, i.e. we will catch 1 fish and
 1546 there will be 1 other fisherman. This matrix also demonstrates the law of total
 1547 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.5. Here is some **R** code to compute the marginal distribution of X , i.e. the probability of catching $X = x$ fish:

```
1548 > margX <- rowSums(joint)
1549 > round(margX, 2)
1550   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
1551 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
1552 X=15  X=16  X=17  X=18  X=19  X=20
1553 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 .

```
1561 > XgivenY <- joint/matrix(margY, nrow(joint), ncol(joint), byrow=TRUE)
1562 > round(XgivenY, 2)
1563   Y=0  Y=1  Y=2  Y=3  Y=4  Y=5  Y=6  Y=7  Y=8  Y=9  Y=10
1564   X=0  0.00 0.25 0.82 0.97  1  1  1  1  1  1  1
1565   X=1  0.00 0.36 0.16 0.03  0  0  0  0  0  0  0
1566   X=2  0.01 0.25 0.02 0.00  0  0  0  0  0  0  0
1567   X=3  0.03 0.11 0.00 0.00  0  0  0  0  0  0  0
```

1570	X=4	0.07	0.03	0.00	0.00	0	0	0	0	0	0
1571	X=5	0.13	0.01	0.00	0.00	0	0	0	0	0	0
1572	X=6	0.17	0.00	0.00	0.00	0	0	0	0	0	0
1573	X=7	0.18	0.00	0.00	0.00	0	0	0	0	0	0
1574	X=8	0.16	0.00	0.00	0.00	0	0	0	0	0	0
1575	X=9	0.12	0.00	0.00	0.00	0	0	0	0	0	0
1576	X=10	0.07	0.00	0.00	0.00	0	0	0	0	0	0
1577	X=11	0.03	0.00	0.00	0.00	0	0	0	0	0	0
1578	X=12	0.01	0.00	0.00	0.00	0	0	0	0	0	0
1579	X=13	0.00	0.00	0.00	0.00	0	0	0	0	0	0
1580	X=14	0.00	0.00	0.00	0.00	0	0	0	0	0	0
1581	X=15	0.00	0.00	0.00	0.00	0	0	0	0	0	0
1582	X=16	0.00	0.00	0.00	0.00	0	0	0	0	0	0
1583	X=17	0.00	0.00	0.00	0.00	0	0	0	0	0	0
1584	X=18	0.00	0.00	0.00	0.00	0	0	0	0	0	0
1585	X=19	0.00	0.00	0.00	0.00	0	0	0	0	0	0
1586	X=20	0.00	0.00	0.00	0.00	0	0	0	0	0	0

1587 Note that we have 11 probability distributions for X , one for each possible value of
 1588 Y , and each pmf sums to unity as it should. Note also that if you show up at the
 1589 hole and there are > 2 fisherman, your chance of catching a fish is very low. Go
 1590 home. These concepts are explained in more detail in other texts such as Casella
 1591 and Berger (2002), Royle and Dorazio (2008), and Link and Barker (2010), but
 1592 hopefully, the code shown here complements the equations and makes it easier for
 1593 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:

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

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

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

1594 From here on out, when you see such notation, you should immediately grasp
 1595 the fact that Y is a random variable independent of X , but X depends upon
 1596 Y through p . Now you have the tools to make probability statements about the
 1597 random variables in this system. The one caveat faced in reality is that we typically
 1598 do not know the values of the parameters, and instead we have to estimate them.
 1599 Maximum likelihood methods for hierarchical models are covered in Chapt. 5.

2.5 HIERARCHICAL MODELS AND INFERENCE

1600 The term hierarchical modeling (or hierarchical model) has become something of
 1601 a buzzword over the last decade with hundreds of papers published in ecological

1602 journals using that term. So then, what exactly is a hierarchical model, anyhow?
 1603 Obviously, this term stems from the root “hierarchy” which means:

1604 **Definition:** *hierarchy* (noun) – a series of ordered groupings of people or things
 1605 within a system;

1606 In the case of a hierarchical model (hierarchical being the adjective form of hi-
 1607 erarchy), the “things” are probability distributions, and they are ordered according
 1608 to their conditional probability structure. Thus, a hierarchical model is *an ordered*
 1609 *series of models, ordered by their conditional probability structure.*

1610 A canonical hierarchical model in ecology is this elemental model of species
 1611 occurrence or distribution (MacKenzie et al., 2002; Tyre et al., 2003; Kéry, 2011):

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

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

1612 where y_i = observation of presence/absence at a site i and z_i = occurrence status
 1613 ($z_i = 1$ if a species occurs at site i and $z_i = 0$ if not). Note that if $p = 1$, then we
 1614 would perfectly observe z and the model would no longer be hierarchical—it would
 1615 be a simple logistic regression model. Note also that this hierarchical model has an
 1616 important conceptual distinction between other types of classical multi-level models
 1617 such as repeated measures on subjects, in that z_i is an actual state of nature. In
 1618 that sense, z is a random variable that is the outcome of a “real” process. Royle
 1619 and Dorazio (2008) used the term *explicit* hierarchical model to describe this type of
 1620 model to distinguish from hierarchical models (*implicit* hierarchical models) where
 1621 the latent variables don’t correspond to an actual state of nature—but rather just
 1622 soak up variation that is unmodeled by explicit elements of the model. At best,
 1623 latent variables in such models are surrogates for something of ecological relevance
 1624 (“time effects”, “space effects” etc.).

1625 With these examples, we expand on our definition of a hierarchical model as we
 1626 will use it in this book:

1627 **Definition:** *Hierarchical Model:* A model with explicit component models that de-
 1628 scribe variation in the data due to (spatial/temporal) variation in *ecological process*,
 1629 and due to *imperfect observation* of the process.

1630 Most models considered in this book describe the encounter of individuals con-
 1631 ditional on the “activity center” of the individual, which is a latent variable (i.e.,
 1632 unobserved random effect). The definition of an activity center will be context-
 1633 dependent as discussed in Chapt. 4, but often it can be thought of as an individual’s
 1634 home range center. The collection of these latent variables represents the outcome
 1635 of an ecological process describing how individuals distribute themselves over the
 1636 landscape. Moreover, how individuals are encountered in traps is, in some cases,

1638 the result of a model governing movement. As such, these models are examples of
 1639 hierarchical models that contain formal model components representing both eco-
 1640 logical process and also the observation of that process. That is, they are explicit
 1641 hierarchical models (Royle and Dorazio, 2008) as opposed to implicit hierarchical
 1642 models.

2.6 CHARACTERIZATION OF SCR MODELS

1643 For the purposes of this book, an SCR model is any “individual encounter model”
 1644 (not just “capture-recapture”!) where auxiliary spatial information is also obtained.
 1645 To be more precise we could as well use the term “spatial capture and/or recap-
 1646 ture” but that is slightly unwieldy and, besides, it also abbreviates to SCR. The
 1647 class of SCR models includes traditional capture-recapture models with auxiliary
 1648 spatial information and even some models that do not even require “recapture”
 1649 (e.g., distance sampling). There is even a class of models (Chapt. 17) which don’t
 1650 require capture or unique identification of individuals.

1651 Conceptually, SCR models involve a collection of random variables, \mathbf{s} , \mathbf{u} and
 1652 y where \mathbf{s} is the activity center, or home range center, \mathbf{u} is the location of the
 1653 individual at the time of sampling, which we may think of as a realization from some
 1654 movement model, and y is the “response variable”—what the observer records. For
 1655 example, $y = 1$ means “detected” and $y = 0$ means “not detected”, but many other
 1656 types of responses are possible (Chapt 8). A broad class of models for estimating
 1657 density are unified by a hierarchical model involving explicit models for animal
 1658 activity centers \mathbf{s} , movement outcomes \mathbf{u} , and encounter data y . In some cases, we
 1659 don’t observe y but rather summaries of y , say $n(y)$, yet it might be convenient
 1660 in such cases to retain an explicit focus on y in terms of model construction. We
 1661 thus introduce a sequence of models—a hierarchical model—to relate these random
 1662 variables, which can be written as

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

1663 Every model we talk about in this book has a subset of these components although
 1664 we never fit the full model because we have not encountered a situation requiring
 1665 that we do so. However, a detailed description of this model and its various com-
 1666 ponents is the subject of this book, and we will not pretend to condense hundreds
 1667 of pages of material into the next few paragraphs. However, we give a cursory
 1668 overview here to whet the appetite and provide some indication of where we are
 1669 going. Don’t worry if some of this material doesn’t sink in just yet—we will walk
 1670 through it slowly in the subsequent chapters.

1671 Let’s begin with the model $[\mathbf{s}]$ that describes the distribution of the activity
 1672 centers of each animal in the spatial region \mathcal{S} (the state-space as we called it previ-
 1673 ously). As will be explained in Chapt. 4 and Chapt. 10, $[\mathbf{s}]$ defines a spatial point
 1674 process, which may be inhomogeneous if there exists spatial variation in density, or

1675 it may be homogeneous if density is constant throughout \mathcal{S} . In the later case, we can
 1676 write $[\mathbf{s}] = \text{Uniform}(\mathcal{S})$, which is to say that the N activity centers are uniformly
 1677 distributed in the polygon \mathcal{S} . A point process is also a model for the number of indi-
 1678 viduals in the population N . So we could write $[\mathbf{s}|\mu]$ where μ is an intensity param-
 1679 eter defined as the number of points per unit area. In other words, μ is population
 1680 density, and we often model population size as either $N \sim \text{Poisson}(\mu A(\mathcal{S}))$, where
 1681 $A(\mathcal{S})$ is the area of the state-space; or, $N \sim \text{Binomial}(M, \psi)$ where $\psi = \mu A(\mathcal{S})/M$
 1682 and M is some large integer used simply as a convenience measure when conducting
 1683 Bayesian analysis. As it turns out, there is very little practical difference in the
 1684 Poisson prior versus a binomial models for N (Chapt. 10).

1685 The model $[\mathbf{u}|\mathbf{s}]$ describes the locations of animals conditional on their activity
 1686 center. In the original formulation of SCR models (Efford, 2004), this model com-
 1687 ponent was intentionally ignored. Indeed when movement is not of direct interest,
 1688 or when \mathbf{s} is defined in a way not related to a home range center, it may be prefer-
 1689 able to ignore this model component (Borchers, 2012). In other cases, we might use
 1690 an explicit model, such as the bivariate normal model (Royle and Young, 2008).

1691 The third component of the model, $[y|\mathbf{u}]$, describes how the observed data—the
 1692 so-called capture-histories—arise conditional on the locations of animals. However,
 1693 as mentioned previously, most SCR models do not contain a movement model, and
 1694 thus, we typically entertain the model $[y|\mathbf{s}]$ instead of $[y|\mathbf{u}]$. This encounter model
 1695 generally has at least two parameters, say p_0 and σ , describing the probability of
 1696 capturing or detecting an individual given the distance between \mathbf{s} and the trap.
 1697 The most basic model is often called the half-normal model, although we typically
 1698 refer to it as the Gaussian model since, in two-dimensional space, it is the kernel
 1699 of a bivariate normal distribution. The model is $p_{ij} = p_0 \exp(-\|\mathbf{x}_j - \mathbf{s}_i\|/(2\sigma^2))$
 1700 where p_0 is the capture probability when the activity center occurs at the trap
 1701 location \mathbf{x}_j , and σ is a spatial scale parameter determining how rapidly capture
 1702 probability declines with distance. One common design leads to the model $[y_{ij}|\mathbf{s}_i] =$
 1703 Bernoulli(p_{ij}). Chapt. 4 and Chapt. 8 describe many other possible encounter
 1704 models.

1705 When individuals are marked by biologists or have natural markings permit-
 1706 ting individual recognition, y_{ij} is the observed data. However, some or all of the
 1707 individuals cannot be uniquely identified, then we cannot record this individual-
 1708 specific encounter history data. Instead, the data might be simply the number of
 1709 detections at a trap or perhaps binary detection/non-detection data at each trap on
 1710 each survey occasion. We call this reduced information data $n(y)$, and Chapt. 17
 1711 and Chapt. 18 describe models for $[n(y)|y]$ that still allow for density estimation.
 1712 The basic strategy is to view y as “missing data” and to use the spatial correlation
 1713 in the counts, or other sources of information, to provide information about these
 1714 latent encounter histories.

1714 Eq. 2.6.1 is a compact description of the the basic components of a SCR model,
 1715 but it is also rather vague. The previous four paragraphs added enough extra detail
 1716 so that we can now describe a specific SCR model. Perhaps the simplest SCR model

is this:

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

1715 These “assumptions” are statistical statements of three basic hypotheses that (1)
 1716 population size N is Poisson distributed (2) activity centers are uniformly dis-
 1717 tributed in two-dimensional space, and (3) capture probability is a function of the
 1718 distance between the activity and the trap. Each of these model components can
 1719 be modified as needed to match specific hypotheses, study designs, and data struc-
 1720 tures. For example, spatial variation in abundance or density can be easily modeled
 1721 as a function of habitat covariates (Chapt. 10).

1722 We realize that many the model description in Eq. 2.6.2 may not be self-evident
 1723 to some ecologists. However, it is absolutely essential that one can understand
 1724 such a model description—not just for being able to read this book, but also for
 1725 understanding any statistical model in ecology. One of the best ways of familiarizing
 1726 oneself with this notation is to translate it into **R** code that simulates outcomes
 1727 from the model. The following code is an example.

```
1728 set.seed(36372)
1729 Area <- 1 # area of state-space (unit square)
1730 x <- cbind(rep(seq(.1,.9,.2), each=5), # trap locations
1731             rep(seq(.1,.9,.2), times=5))
1732 p0 <- 0.3 # baseline capture probability
1733 sigma <- 0.05 # Gaussian scale parameter
1734 mu <- 50 # population density
1735 N <- rpois(1, mu*Area) # population size
1736 s <- cbind(runif(N, 0, 1), # activity centers in unit square
1737             runif(N, 0, 1))
1738 K <- 5
1739 y <- matrix(NA, N, nrow(x)) # capture data
1740 for(i in 1:N) {
1741   d.ij <- sqrt((x[,1] - s[i,1])^2 + # distance between x and s[i]
1742                 (x[,2] - s[i,2])^2)
1743   p.ij <- p0*exp(-d.ij^2 / (2*sigma^2)) # capture probability
1744   y[i,] <- rbinom(nrow(x), K, p.ij) # capture history for animal i
1745 }
```

1746 Fig. 2.3 shows the results of this simulation from a basic, yet very useful, SCR
 1747 model.

1748 Having briefly explained each of the model components in Eq. 2.6.1, and having
 1749 shown how a subset of these components results in a basic SCR model, we can
 1750 now discuss other relevant arrangements. Examples include: (1) Classical distance
 1751 sampling (Buckland et al., 2001; Borchers et al., 2002), (2) Spatial capture-recapture
 1752 models with fixed arrays of traps (Efford, 2004; Borchers and Efford, 2008; Royle

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

et al., 2009a,b; Gardner et al., 2010a; Royle et al., 2011b), and (3) Search-encounter models (Royle and Young, 2008; Royle et al., 2011a). We will now elaborate on some of these distinctions.

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

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

This confounding happens because SCR sampling is spatially biased—restricted to a fixed pre-determined set of locations. Conversely, distance sampling confounds $[\mathbf{u}|\mathbf{s}][\mathbf{s}]$ because, essentially, there is only a single realization of the encounter process. It is probably reasonable to assume that $\Pr(y = 1|\mathbf{u}) = 1$ or at least it is locally constant for most devices (e.g., cameras, etc..), and thus the detection model will have the interpretation in terms of movement (see Chapt. 12 and 11).

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

2.7 SUMMARY AND OUTLOOK

1782 Spatial capture-recapture models are hierarchical models, and hierarchical models
1783 are models of multiple random variables that are conditionally related. It is there-
1784 fore important that the basic rules of modeling random variables are understood,
1785 and we hope that this chapter has made some of the basic concepts accessible to
1786 ecologists with rudimentary background in statistics. If some of this material still
1787 seems difficult to grasp, we recommend working with the provided **R** code, which
1788 is perhaps the best way of making the equations more tangible.

1789 In some respects, it is possible to understand the jist of SCR without knowing
1790 anything about marginal and conditional relationships. One can always fit models
1791 using canned software and interpret the output without understanding the guts of
1792 the model or the details of the estimation process. For some applied ecologists,
1793 this may be perfectly fine, and this book is meant to be useful for both statistical
1794 novices and ecologists with more advanced quantitative skills. In most chapters, we
1795 begin with a basic conceptual discussion, then we explain the technical details that
1796 require an understanding of the concepts in this chapter, and finally we end with
1797 one or more worked examples. For those not interested in the technical details,
1798 we recommend focusing on the chapter introductions and the examples. However,
1799 taking the time to understand the concepts presented in this chapter can only
1800 increase one's ability to tackle the unique and complex problems that often present
1801 themselves when modeling spatial and temporal aspects of population dynamics.

1802
1803

1804

3

CLOSED POPULATION MODELS

1805 In this chapter we introduce ordinary *non-spatial* capture-recapture (CR) models
1806 for estimating population size in closed populations. A closed population is one
1807 whose size, N , does not change during the study. Two forms of closure are often
1808 discussed: demographic closure, meaning that no births or deaths occur, and geo-
1809 graphic closure, which states that no individuals move onto or off of the sampled
1810 area during the study. Although few populations are actually closed except during
1811 very short time intervals, closed population CR models serve as the basis for the
1812 development of the rest of the models presented in this book, including the models
1813 for open populations discussed in Chapt. 15.

1814 We begin with the most basic capture-recapture model, colloquially referred
1815 to as “model M_0 ” (Otis et al., 1978), in which encounter probability is strictly
1816 constant in all respects (across individuals, and replicates). This allows us to high-
1817 light the basic structure of closed population models as binomial GLMs. We then
1818 consider some important extensions of ordinary closed population models that ac-
1819 commodate various types of “individual effects” — either in the form of explicit,
1820 observed covariates (sex, age, body mass) or unstructured “heterogeneity” in the
1821 form of an individual random effect, which represent unobserved or unmeasured co-
1822 variates. A special type of individual covariate models is distance sampling, which
1823 could be thought of as the most primitive spatial capture-recapture model. All of
1824 these different types of closed population models are closely related to binomial
1825 (or logistic) regression-type models. In fact, when N is known, they are precisely
1826 logistic regression models.

1827 We emphasize Bayesian analysis of capture-recapture models and we accomplish
1828 this using a method related to classical “data augmentation” from the statistics
1829 literature (e.g., Tanner and Wong, 1987). This is a general concept in statistics
1830 but, in the context of capture-recapture models where N is unknown, it has a

1831 consistent implementation across classes of capture-recapture models and one that
 1832 is really convenient from the standpoint of doing MCMC (Royle et al., 2007; Royle
 1833 and Dorazio, 2012). We use data augmentation throughout this book and thus
 1834 emphasize its conceptual and technical origins and demonstrate applications to
 1835 closed population models. We refer the reader to Kéry and Schaub (2012, ch. 6)
 1836 for an accessible and complementary development of Bayesian analysis of ordinary,
 1837 i.e., nonspatial closed population models.

3.1 THE SIMPLEST CLOSED POPULATION MODEL: MODEL M_0

1838 To start looking at the simplest capture-recapture model, let's suppose there exists
 1839 a population of N individuals which we subject to repeated sampling, say over K
 1840 "occasions", such as trap nights, where individuals are captured, marked, released,
 1841 and subsequently recaptured. We suppose that individual encounter histories are
 1842 obtained, and these are of the form of a sequence of 0's and 1's indicating capture
 1843 ($y = 1$) or not ($y = 0$) during any sampling occasion. As an example, suppose $K = 5$
 1844 sampling occasions, then an individual captured during occasion 2 and 3 but not
 1845 otherwise would have an encounter history of the form $\mathbf{y} = (0, 1, 1, 0, 0)$. Thus,
 1846 the observation \mathbf{y}_i for each individual ($i = 1, 2, \dots, N$) is a vector having elements
 1847 denoted by y_{ik} for $k = 1, 2, \dots, K$. Usually this is organized as a row of a matrix
 1848 with elements y_{ik} , see Table 3.1. Except where noted explicitly, we suppose that
 1849 observations are independent within individuals and among individuals. Formally,
 1850 this allows us to say that y_{ik} are independent and identically distributed ("iid")
 1851 Bernoulli random variables and we may write $y_{ik} \sim \text{Bernoulli}(p)$. Consequently,
 1852 for this very simple model in which p is constant (i.e., there are no individual or
 1853 temporal covariates that affect p) the original binary detection variables can be
 1854 aggregated into the total number of encounters for each individual¹, $y_i = \sum_k y_{ik}$,
 1855 and the observation model changes from a Bernoulli distribution to a binomial
 1856 distribution based on a sample of size K . That is

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

1857 for every individual in the population $i = 1, 2, \dots, N$, where N is the number of
 1858 individuals in the population (i.e., population size).

1859 We emphasize the central importance of the basic Bernoulli encounter model –
 1860 an individual is either encountered in a sample, or not – which forms the cornerstone
 1861 of almost all of classical capture-recapture models, including many spatial capture-
 1862 recapture models discussed in this book.

1863 Evidently, the basic capture-recapture model is a simplistic version of a logistic-
 1864 regression model with only an intercept term ($\text{logit}(p) = \text{constant}$). To say that all

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..

1865 capture-recapture models are just logistic regressions is a slight over-simplification.
1866 In fact, we are proceeding here as if we knew N . In practice we don't, of course,
1867 and estimating N is actually the central objective. But, by proceeding as if N
1868 were known, we can specify a simple model and then deal with the fact that N is
1869 unknown using standard methods that you are already familiar with (i.e., GLMs -
 see Chapt. ??).

Table 3.1. A toy capture-recapture data set with $n = 6$ observed individuals and $K = 5$ sample occasions. Under a model with constant encounter probability, the binary detection history data can be summarized in the detection frequency (the total number of detections, y_i), which is shown in the right-most column.

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

1870
1871 Assuming individuals in the population are encountered independently, the joint
1872 probability distribution of the observations is the product of N binomials

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

1873 We emphasize that this expression is conditional on N , in which case we get to
1874 observe the $y_i = 0$ observations and the resulting data are just *iid* binomial counts.
1875 Because this is a binomial regression model of the variety described in Chapt. ??,
1876 fitting this model using a **BUGS** engine poses no difficulty.

1877 Equation 3.1.1 can be simplified even further if we reformat the observations
1878 as encounter frequencies. Specifically, let n_k denote the number of individuals
1879 captured exactly k times after K survey occasions, $n_k = \sum_{i=1}^N I(y_i = k)$ where $I()$
1880 is the indicator function evaluating to 1 if its argument is true and 0 otherwise. For
1881 sake of illustration, we converted the data from Table 3.1 to this format (Table 3.2).
1882 What is important to note is that if we know N , then we know n_0 , i.e. the number
1883 of individuals not captured. In this case, an alternative and equivalent expression
1884 to Eq. 3.1.1 is

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

1885 where $\pi_k = \Pr(y = k)$ under the binomial model with parameter p and sample size
1886 K . The essential problem in capture-recapture, however, is that N is *not* known

Table 3.2. Data from Table 3.1 formatted as capture frequencies. Since N is unknown, the number of individuals not captured (n_0) is also unknown.

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

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

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

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

To fit the model in which N is *unknown*, we can regard n_0 as a parameter and maximize the multinomial likelihood directly. Direct likelihood analysis of the multinomial model is straightforward, but that is not always sufficiently useful in practice because we seldom are concerned with models for the aggregated encounter history frequencies, which entail that capture probabilities are the same for all individuals. In many instances, including for spatial capture-recapture (SCR) models, we require a formulation of the model that can accommodate individual-level covariates to account for differences in detection among individuals, which we address subsequently in this chapter, and also in Chapt. 6.

3.1.1 The core capture-recapture assumptions

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

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

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

1918 to be valid, and while they are also assumed by most of the models we present in
 1919 the following chapters, we refrain from repeatedly making such statements. Our
 1920 opinion is that all model assumptions are apparent when a model is clearly specified,
 1921 and it is both redundant and impossible to list all the things not allowed by the
 1922 model. For example, closed population models also assume that other sources of
 1923 error do not occur, but it is not necessary to enumerate each possibility. Rather, it
 1924 is necessary to make clear statements such as

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

1925 This simple model description carries a tremendous amount of information, and
 1926 it leaves very little left to say with respect to assumptions. Although we will not
 1927 always show the *iid* symbol, it will be assumed unless otherwise noted, and this
 1928 assumption is critical for valid inference. It implies that the encounter of one indi-
 1929 vidual does not affect the encounter of another individual, and encounter does not
 1930 affect future encounter. Under this assumption, it is easy to write down the likeli-
 1931 hood of the parameters and obtain parameter estimates; however, whether or not it
 1932 is true depends upon biological and sampling issues. If this assumption is deemed
 1933 false, the model can be discarded in favor of a more realistic alternative. However,
 1934 once we have settled on our model, statistical inference proceeds by assuming the
 1935 model is truth—not an approximation to truth—but actual truth.

1936 In spite of the fact that we assume that all models are truth, but we acknowledge
 1937 that all models are wrong due to their assumptions, assumptions should not be
 1938 viewed as a necessary evil. In fact, one way to view assumptions is as embodiments
 1939 of our ecological hypotheses. If we make these assumptions too complex or too
 1940 specific, then we will never be able to study general phenomena that hold true
 1941 across space and time. Furthermore, in practice, we will rarely have enough data
 1942 to estimate the parameters of highly complex models.

1943 3.1.2 Conditional likelihood

1944 We saw that the closed population model is a simple logistic regression model if N
 1945 is known and, when N is unknown, the model is multinomial with index or sample
 1946 size parameter N . This multinomial model, being conditional on N , is sometimes
 1947 referred to as the “joint likelihood” the “full likelihood” or the “unconditional like-
 1948 lihood” (sometimes “model” in place of “likelihood”) (Sanathanan, 1972; Borchers
 1949 et al., 2002). This formulation differs from the so-called “conditional likelihood”
 1950 approach in which the likelihood of the observed encounter histories is devised con-
 1951 ditional on the event that an individual is captured at least once. To construct this
 1952 likelihood, we have to recognize that individuals appear or not in the sample based
 1953 on the value of the random variable y_i , that is, if and only if $y_i > 0$. The obser-
 1954 vation model is therefore based on $\Pr(y|y > 0)$. For the simple case of model M_0 ,
 1955 the resulting conditional distribution is a “zero truncated” binomial distribution

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

In this book we emphasize Bayesian analysis of capture-recapture models using data augmentation (described in Sec. 3.2 below), which produces yet a third distinct formulation of capture-recapture models based on the zero-*inflated* binomial distribution that we describe in the next section. Thus, there are 3 distinct formulations of the model – or modes of analysis – for analyzing all capture-recapture models based on the (1) binomial model for the joint or unconditional specification; (2) zero-truncated binomial that arises “conditional on n ”; and (3) the zero-inflated binomial that arises under data augmentation. Each formulation has distinct model parameters (shown in Table 3.3 for model M_0).

Table 3.3. Modes of analysis of capture-recapture models. Closed population models can be analyzed using the joint or “full likelihood” which contains N as an explicit parameter, the conditional likelihood which does not involve N , or by data augmentation which replaces N with 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

3.2 DATA AUGMENTATION

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

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

3.2.1 DA links occupancy models and closed population models

There is a close correspondence between so-called “occupancy” models and closed population models (see Royle and Dorazio, 2008, Sec. 5.6). In occupancy models (MacKenzie et al., 2002; Tyre et al., 2003) the sampling situation is that M sites, or patches, are sampled multiple times to assess whether a species occurs at the sites. This yields encounter data such as that illustrated in the left panel of Table 3.4. The important problem is that a species may occur at a site, but go undetected, yielding an all-zero encounter history for the site, which in the case of occupancy studies, are *observed*. However, some of the zero vectors will typically correspond to sites where the species in fact *does* occur. Thus, while the zeros are observed, there are too many of them and, in a sense, the inference problem is to partition the zeros into “structural” (fixed) and “sampling” (or stochastic) zeros, where the former are associated with unoccupied sites and the latter with occupied sites where the species went undetected. More formally, inference is focused on the parameter ψ , the probability that a site is occupied.

In contrast to occupancy studies, in classical closed population studies, we observe a data set as in the middle panel of Table 3.4 where *no* zeros are observed. The inference problem is, essentially, to estimate how many sampling zeros there are – or should be – in a “complete” data set. This objective (how many sampling zeros?) is precisely the same for both types of problems if an upper limit M is specified for the closed population model. The only distinction being that, in occupancy models, M is set by design (i.e., the number of sites in the sample), whereas a natural choice of M for capture-recapture models may not be obvious. However, the choice of M induces a uniform prior for N on the integers $[0, M]$ (Royle et al.,

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

2024 2007). Then, one can analyze capture-recapture models by adding $M - n$ all-zero
 2025 encounter histories to the data set and regarding the augmented data set, essen-
 2026 tially, as a site-occupancy data set, where the occupancy or data augmentation
 2027 parameter (ψ) takes the place of the abundance parameter (N).

2028 Thus, the heuristic motivation of data augmentation is to fix the size of the
 2029 data set by adding *too many* all-zero encounter histories to create the data set
 2030 shown in the right panel of Table 3.4, and then analyze the augmented data set
 2031 using an occupancy type model which includes both “unoccupied sites” (in capture-
 2032 recapture, augmented individuals that are not members of the real population that
 2033 was sampled) as well as “occupied sites” (in capture-recapture, individuals that are
 2034 members of the population but that were undetected by sampling) at which detec-
 2035 tions did not occur. We call these $M - n$ all-zero histories “potential individuals”
 2036 because they exist to be recruited (in a non-biological sense) into the population,
 2037 for example during an analysis by MCMC.

2038 To analyze the augmented data set, we recognize that it is a zero-inflated ver-
 2039 sion of the known- N data set. That is, some of the augmented all-zero rows are
 2040 sampling zeros (corresponding to actual individuals that were missed) and some are
 2041 “structural” zeros, which do not correspond to individuals in the population. For
 2042 a basic closed-population model, the resulting likelihood under data augmentation
 2043 - that is, for the data set of size M - is a simple zero-inflated binomial likelihood.
 2044 The zero-inflated binomial model can be described “hierarchically”, by introducing
 2045 a set of binary latent variables, z_1, z_2, \dots, z_M , to indicate whether each individual
 2046 i is ($z_i = 1$) or is not ($z_i = 0$) a member of the population of N individuals exposed
 2047 to sampling. We assume that $z_i \sim \text{Bernoulli}(\psi)$ where ψ is the probability that
 2048 an individual in the data set of size M is a member of the sampled population –
 2049 in the sense that $1 - \psi$ is the probability of a “structural zero” in the augmented
 2050 data set. The zero-inflated binomial model which arises under data augmentation
 2051 can be formally expressed by the following set of assumptions (we include typical
 2052 priors for a Bayesian analysis):

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

2053 for $i = 1, \dots, M$, where $I(y = 0)$ is a point mass at $y = 0$. It is sometimes
 2054 convenient to express the conditional-on- z observation model concisely in just one
 2055 step:

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

2056 and we understand this to mean, if $z_i = 0$, then y_i is necessarily 0 because its
 2057 success probability is $z_i p = 0$.

2058 Note that, under data augmentation, N is no longer an explicit parameter of
 2059 this model. In its place, we estimate ψ and functions of the latent variables z . In
 2060 particular, under the assumptions of the zero-inflated model, $z_i \stackrel{iid}{\sim} \text{Bernoulli}(\psi)$;
 2061 therefore, N is a function of these latent variables:

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

2062 Further, we note that the latent z_i parameters *can be* removed from the model by
 2063 integration, in which case the joint probability of the data is

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

2064 Interpreted as a likelihood, we can directly maximize this expression to obtain the
 2065 MLEs of the structural parameters ψ and p or those of other more complex models
 2066 (e.g., see Royle, 2006). We could estimate these parameters and then use them
 2067 to obtain an estimator of N using the so-called “Best unbiased predictor” (see
 2068 Royle and Dorazio, 2012). Normally, however, we will analyze the model in its
 2069 “conditional-on- z ” form using methods of MCMC either in the **BUGS** engines or
 2070 using our own MCMC algorithms (see Chapt. 16).

2071 3.2.2 Model M_0 in **BUGS**

2072 It is helpful to understand data augmentation by seeing what its effect is on imple-
 2073 menting model M_0 . For this model, in which we can aggregate the encounter data
 2074 to individual-specific encounter frequencies, the augmented data are given by the
 2075 vector of frequencies $(y_1, \dots, y_n, 0, 0, \dots, 0)$ where the augmented values of $y = 0$
 2076 represent the encounter frequency for potential individuals y_{n+1}, \dots, y_M . The zero-
 2077 inflated model of the augmented data combines the model of the latent variables,
 2078 $z_i \sim \text{Bernoulli}(\psi)$. The **BUGS** model description of the closed population model
 2079 M_0 is shown in Panel 3.1. The last line of the model specification provides the
 2080 expression for computing N from the data augmentation variables z_i . Note that,
 2081 to improve readability of code snippets (especially of large ones), we will sometimes
 2082 deviate from our standard notation a bit. In this case we use `nind` for n (the number
 2083 of encountered individuals), and $M = nind + nz$ is the total size of the augmented
 2084 data set. In other cases we might also use `nocc` in place of K and `ntraps` in place
 2085 of J . We find that word definitions make code easier to understand, especially
 2086 without having to read surrounding text.

2087 Specification of a more general model in terms of the individual encounter obser-
 2088 vations y_{ik} is not much more difficult than for the individual encounter frequencies.
 2089 We define the observation model by a double loop and change the indexing of
 2090 quantities accordingly, i.e.,

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

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

```

2091 for(i in 1:(nind+nz)){
2092   z[i] ~ dbern(psi)
2093   for(k in 1:K){
2094     mu[i,k] <- z[i]*p
2095     y[i,k] ~ dbin(mu[i,k],1)
2096   }
2097 }
```

2098 In this manner, it is straightforward to incorporate covariates on p for both
2099 individuals and sampling occasions (see discussion of this below and also Chapt.
2100 6) as well as to devise other extensions of the model, including models for open
2101 populations (see Chapt. 15).

2102 3.2.3 Formal development of data augmentation (DA)

2103 Use of parameter-expanded data augmentation (PX-DA), or DA for short, for solving
2104 inference problems with unknown N can be justified as originating from the
2105 choice of a uniform prior on N . The Uniform(0, M) prior for N is innocuous in
2106 the sense that the posterior associated with this prior is equal to the likelihood

```

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

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

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

```

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

2107 for sufficiently large M . One way of inducing the $\text{Uniform}(0, M)$ prior on N is by
 2108 assuming the following hierarchical prior:

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

2109 The model assumptions, specifically the multinomial model (Eq. 3.1.3) and Eq.
 2110 3.2.2, may be combined to yield a reparameterization of the conventional model
 2111 that is appropriate for the augmented data set of known size M :

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

2112 This expression arises by removing N from Eq. 3.1.3 by integrating over the bi-
 2113 nomial prior distribution for N . Thus, the models we analyze under data aug-
 2114mentation arise formally by removing the parameter N from the ordinary closed-
 2115 population model, which is conditional on N , by integrating over a binomial prior
 2116 distribution for N .

2117 Note that the $M - n$ unobserved individuals in the augmented data set have
 2118 probability $\psi\pi(0) + (1 - \psi)$, indicating that these unobserved individuals are a
 2119 mixture of individuals that are sampling zeros ($\psi\pi_0$), and belong to the population
 2120 of size N , and others that are “structural zeros” (occurring in the augmented data
 2121 set with probability $1 - \psi$). In Eq. 3.2.3, N has been eliminated as a formal

2122 parameter of the model by marginalization (integration) and replaced with the
2123 new parameter ψ , the data augmentation parameter. However, the full likelihood
2124 containing both N and ψ can also be analyzed (see Royle et al., 2007).

2125 **3.2.4 Remarks on data augmentation**

2126 Data augmentation may seem like a strange and mysterious black-box, and likely
2127 it is unfamiliar to most people, even to many of those with substantial experience
2128 with capture-recapture models. However, it really is just a formal reparameteriza-
2129 tion of capture-recapture models in which N is marginalized out of the ordinary
2130 (conditional-on- N) model (by summation over a binomial prior). As a result, we
2131 could refer to the resulting model as the “binomial-integrated likelihood” to reflect
2132 that an estimator could be obtained from the ordinary likelihood, integrated over
2133 a binomial prior. Other such “integrated likelihood” models are sensible. For ex-
2134 ample, we could place a Poisson prior on N with mean Λ and marginalize N over
2135 the Poisson prior. This produces a likelihood in which Λ replaces N , instead of ψ
2136 replacing N . We note that this type of marginalization (over a Poisson prior) is
2137 done by the **R** package **secr** for analysis of spatial capture-recapture models (see
2138 Sec. 5.5.3).

2139 We emphasize the motivation for data augmentation being that it produces a
2140 data set of fixed size, so that the parameter dimension in any capture-recapture
2141 model is also fixed. As a result, MCMC is a relatively simple proposition using
2142 standard Gibbs Sampling. And, in particular, capture-recapture models become
2143 trivial to implement in **BUGS**. Consider the simplest context—analyzing model
2144 M_0 using the occupancy-type model. In this case, DA converts model M_0 to a
2145 basic occupancy model, and the parameters p and ψ have known full-conditional
2146 distributions (in fact, beta distributions) that can be sampled from directly. Fur-
2147 thermore, the data augmentation variables, i.e., the collection of z ’s, can be sampled
2148 from Bernoulli full conditionals. MCMC is not much more difficult for complicated
2149 models—sometimes the hyperparameters need to be sampled using a Metropolis-
2150 Hastings step (e.g., Chapt. 16), but nothing more sophisticated than that is re-
2151 quired.

2152 Potential sensitivity of parameter estimates to M (especially of N) might be
2153 cause for some concern. The guiding principle is that it should be chosen large
2154 enough so that the posterior for N is not truncated, but it should not be too large
2155 due to the increased computational burden. It seems likely that the properties of
2156 the Markov chains should be affected by M and so some optimal choice of M might
2157 exist (Gopalaswamy, 2012). Formal analysis of this is needed.

2158 There are other approaches to analyzing models with unknown N , using re-
2159 versible jump MCMC (RJMCMC) or other so-called “trans-dimensional” (TD)
2160 algorithms (King and Brooks, 2001; Durban and Elston, 2005; King et al., 2008;
2161 Schofield and Barker, 2008; Wright et al., 2009). What distinguishes DA from RJM-
2162 CMC and related TD methods is that DA is used to create a distinctly new model

2163 that is unconditional on N and we (usually) analyze the unconditional model. The
 2164 various TD/RJMCMC approaches seek to analyze the conditional-on- N model in
 2165 which the dimension of the parameter space is a function of N , and will therefore
 2166 typically vary at each iteration of the MCMC algorithm. TD/RJMCMC approaches
 2167 might appear to have the advantage that one can model N explicitly or consider
 2168 alternative priors for N . However, despite that N is removed as an explicit param-
 2169 eter in DA, it is possible to develop hierarchical models that involve structure on
 2170 N (Converse and Royle, 2012; Royle et al., 2012c; Royle and Converse, in review)
 2171 which we consider in Chapt. 13. Furthermore, data augmentation is often easier
 2172 to implement than RJMCMC, and the details of the DA implementation are the
 2173 same for all capture-recapture problems.

2174 3.2.5 Example: Black bear study on Fort Drum

2175 To illustrate the analysis of model M_0 using data augmentation, we use a data set
 2176 collected at Fort Drum Military Installation in upstate New York by P.D. Curtis and
 2177 M.T Wegan of Cornell University and their colleagues at the Fort Drum Military
 2178 Installation. These data have been analyzed in various forms by Wegan (2008);
 2179 Gardner et al. (2009) and Gardner et al. (2010b). The specific data used here are
 2180 encounter histories on 47 individuals obtained from an array of 38 baited “hair
 2181 snares” (Fig. 3.1) during June and July 2006. Barbed wire traps were baited and
 2182 checked for hair samples each week for eight weeks, thus we distinguished $K = 8$
 2183 weekly sample intervals. The data are provided in the **R** package **scrbook**, can be
 2184 loaded by typing `data(beardata)` at the **R** prompt, and the analysis can be set
 2185 up and run as follows (see `?beardata` for the commands to do the analysis). Here,
 2186 the data were augmented with 128 all-zero encounter histories, resulting in a total
 2187 sample size of $M = 175$.

```
2188 > library(scrbook)
2189 > data(beardata) # load the bear data and extract components
2190 > trapmat <- beardata$trapmat
2191 > nind <- dim(beardata$bearArray)[1]
2192 > K <- dim(beardata$bearArray)[3]
2193 > ntraps <- dim(beardata$bearArray)[2]
2194
2195 > M <- 175
2196 > nz <- M-nind
2197 > Yaug <- array(0, dim=c(M,ntraps,K))
2198
2199 > Yaug[1:nind,,] <- Beardata$bearArray
2200 > y <- apply(Yaug,c(1,3),sum) # summarize by ind x rep
2201 > y[y>1] <- 1 # toss out multiple encounters per occasion
2202 # b/c traditional CR models ignore space
```

2203 The raw data object, `beardata$bearArray` is a 3-dimensional array `nind ×`

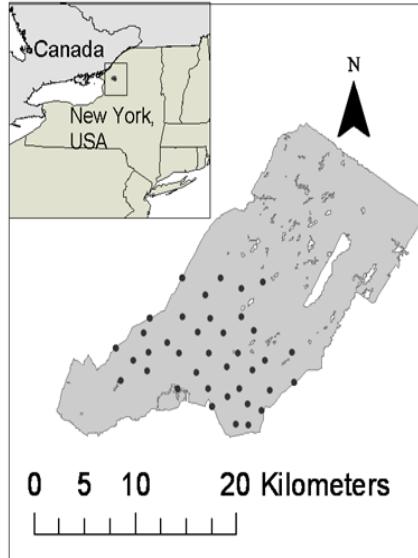


Figure 3.1. Fort Drum Black bear study area and the 38 baited hair snare locations operated for 8 weeks during June and July, 2006.

2204 $n_{traps} \times K$ of individual encounter events (i.e., $y_{ijk} = 1$ if individual i was encountered in trap j during occasion k , and 0 otherwise). For fitting model M_0 (or M_h ,
 2205 see below), it is sufficient to reduce the data to individual encounter frequencies
 2206 which we have re-labeled “y” above. The **BUGS** model file along with commands
 2207 to fit the model are as follows:
 2208

```

2209 > set.seed(2013)                      # to obtain the same results each time
2210 > library(R2WinBUGS)                   # load R2WinBUGS, set-up:
2211 > data0 <- list(y=y, M=M, K=K)        # data ....
2212 > params0 <- c('psi','p','N')          # parameters ....
2213 > zst <- c(rep(1,nind),rbinom(M-nind, 1, .5)) # inits ....
2214 > inits <- function(){ list(z=zst, psi=runif(1), p=runif(1)) }
2215
2216 > cat("                                "
2217 model{
2218
2219   psi ~ dunif(0, 1)
2220   p ~ dunif(0,1)
2221
2222   for (i in 1:M){
2223     z[i] ~ dbern(psi)
```

```

2224   for(k in 1:K){
2225     tmp[i,k] <- p*z[i]
2226     y[i,k] ~ dbin(tmp[i,k],1)
2227   }
2228 }
2229 N<-sum(z[1:M])
2230 }
2231 ",file="modelM0.txt")
2232
2233 ## Run the model:
2234 > fit0 <- bugs(data0, inits, params0, model.file="modelM0.txt", n.chains=3,
2235   n.iter=2000, n.burnin=1000, n.thin=1, debug=TRUE, working.directory=getwd())

```

2236 This produces the following posterior summary statistics:

```

2237 > print(fit0,digits=2)
2238 Inference for Bugs model at "modelM0.txt", fit using WinBUGS,
2239 3 chains, each with 2000 iterations (first 1000 discarded)
2240 n.sims = 3000 iterations saved
2241      mean    sd  2.5%   25%   50%   75% 97.5% Rhat n.eff
2242 psi      0.29  0.04  0.22  0.26  0.29  0.31  0.36    1  3000
2243 p       0.30  0.03  0.25  0.28  0.30  0.32  0.35    1  3000
2244 N       49.94 1.99 47.00 48.00 50.00 51.00 54.00    1  3000
2245 deviance 489.05 11.28 471.00 480.45 488.80 495.40 513.70    1  3000
2246
2247 [... some output deleted ...]

```

2248 **WinBUGS** did well in choosing an MCMC algorithm for this model – we have
2249 $\hat{R} = 1$ for each parameter, and an effective sample size of 3000, equal to the total
2250 number of posterior samples³. We see that the posterior mean of N under this
2251 model is 49.94 and a 95% posterior interval is (48, 54). We revisit these data later
2252 in the context of more complex models.

2253 In order to obtain an estimate of density, D , we need an area to associate
2254 with the estimate of N , and in Chapt. 1 we already went through a number of
2255 commonly used procedures to conjure up such an area, including buffering the trap
2256 array by the home range radius, often estimated by the mean maximum distance
2257 moved (MMDM) (Parmenter et al., 2003), 1/2 MMDM (Dice, 1938) or directly from
2258 telemetry data (Wallace et al., 2003). Typically, the trap array is defined by the
2259 convex hull around the trap locations, and this is what we applied a buffer to. We
2260 computed the buffer by using a telemetry-based estimate of the mean female home
2261 range radius (2.19 km) (Bales et al., 2005) instead of using an estimate based on our
2262 relatively more sparse recapture data. For the Fort Drum study, the convex hull has
2263 an area of 157.135 km², and the buffered convex hull has an area of 277.011 km².
2264 To create this we used functions contained in the **R** package **rgeos** and created a

³This is even a little suspicious....

2265 utility function `bcharea` which is in our **R** package `scrbook`. The commands are
 2266 as follows:

```
2267 > library(rgeos)
2268
2269 > bcharea <- function(buff,traplocs){
2270   p1 <- Polygon(rbind(traplocs,traplocs[1,]))
2271   p2 <- Polygons(list(p1=p1),ID=1)
2272   p3 <- SpatialPolygons(list(p2=p2))
2273   p1ch <- gConvexHull(p3)
2274   bp1 <- (gBuffer(p1ch, width=buf))
2275   plot(bp1, col='gray')
2276   plot(p1ch, border='black', lwd=2, add=TRUE)
2277   gArea(bp1)
2278 }
2279
2280 > bcharea(2.19,traplocs=trapmat)
```

The resulting buffered convex hull is shown in Fig. 3.2.

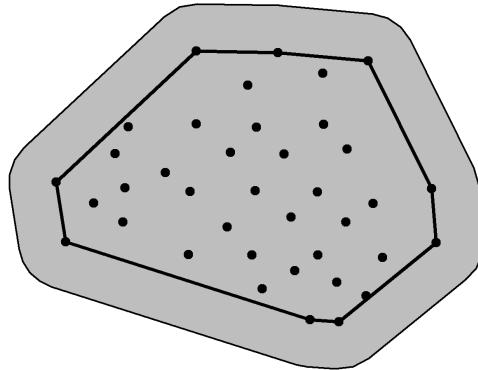


Figure 3.2. Convex hull of the bear hair snare array at Fort Drum, NY, buffered by mean female home range radius (2.19 km).

2281

2282 To conjure up a density estimate under model M_0 , we compute the appropriate
 2283 posterior summary of the ratio of N and the prescribed area (277.011 km^2):

```
2284 > summary(fit0$sims.list$N/277.011)
2285   Min. 1st Qu. Median Mean 3rd Qu. Max.
2286 0.1697 0.1733 0.1805 0.1803 0.1841 0.2130
2287
2288 > quantile(fit0$sims.list$N/277.011,c(0.025,0.975))
2289   2.5% 97.5%
2290 0.1696684 0.1949381
```

2291 which yields a density estimate of about 0.18 ind/km^2 , and a 95% Bayesian con-
 2292 fidence interval of $(0.170, 0.195)$. Our estimate of density should be reliable if we
 2293 have faith in our stated value of the “sampled area”. Clearly though this is largely
 2294 subjective, and not something we can formally evaluate (or estimate) from the data
 2295 based on model M_0 .

3.3 TEMPORALLY VARYING AND BEHAVIORAL EFFECTS

2296 The purpose of this chapter is mainly to emphasize the central importance of the
 2297 binomial model in capture-recapture and so we have considered models for individ-
 2298 ual encounter frequencies—the number of times individuals are captured out of K
 2299 occasions. Sometimes we can’t aggregate the encounter data for each individual,
 2300 such as when encounter probability varies over time among samples. Time-varying
 2301 responses that are relevant in many capture-recapture studies are “effort” such as
 2302 amount of search time, number of observers, or trap nights, or encounter probability
 2303 varying over time, as a function of date or season (Kéry et al., 2010) due to species
 2304 behavior. A common situation in many animal studies is that in which there exists
 2305 a “behavioral response” to trapping (even if the animal is not physically trapped).

2306 Behavioral response is an important concept in animal studies because individ-
 2307 uals might learn to come to baited traps or avoid traps due to trauma related to
 2308 being encountered. There are a number of ways to parameterize a behavioral re-
 2309 sponse to encounter. The distinction between persistent and ephemeral was made
 2310 by Yang and Chao (2005) who considered a general behavioral response model of
 2311 the form:

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

2312 where x_{ik} is a covariate indicator variable of previous capture (i.e., $x_{ik} = 1$ if cap-
 2313 tured in any previous period). Therefore, encounter probability changes depending
 2314 on whether an individual was captured in the immediate previous period (a Marko-
 2315 vian or ephemeral behavioral response; (Yang and Chao, 2005)), described by the
 2316 term $\alpha_1 y_{i,k-1}$ or in *any* previous period (persistent behavioral response), described
 2317 by the term $\alpha_2 x_{ik}$. Because spatial capture-recapture models allow us to include
 2318 trap-specific covariates, we can describe a 3rd type of behavioral response—a lo-
 2319 cal behavioral response that is trap-specific (Royle et al., 2011b). In this local

2320 behavioral response, the encounter probability is modified for an individual trap
 2321 depending on previous capture in that trap. Models with temporal effects are
 2322 easy to describe and analyze in the **BUGS** language and we provide a number of
 2323 examples in Chapt. 6 and elsewhere.

3.4 MODELS WITH INDIVIDUAL HETEROGENEITY

2324 Models in which encounter probability varies by individual have a long history in
 2325 capture-recapture and, indeed, this so-called “model M_h ” is one of the elemental
 2326 capture-recapture models in (Otis et al., 1978). Conceptually, we imagine that
 2327 the individual-specific encounter probability parameters, p_i , are random variables
 2328 distributed according to some probability distribution, $[\theta]$. We denote this basic
 2329 model assumption as $p_i \sim [\theta]$. This type of model is similar in concept to extending
 2330 a GLM to a GLMM but in the capture-recapture context N is unknown. The basic
 2331 class of models is often referred to as “model M_h ” (“h” for heterogeneity), but really
 2332 this is a broad class of models, each being distinguished by the specific distribution
 2333 assumed for p_i . There are many different varieties of model M_h including parametric
 2334 and various non-parametric approaches (Burnham and Overton, 1978; Norris and
 2335 Pollock, 1996; Pledger, 2004). One important practical matter is that estimates of
 2336 N can be extremely sensitive to the choice of heterogeneity model (Fienberg et al.,
 2337 1999; Dorazio and Royle, 2003; Link, 2003). Indeed, Link (2003) showed that in
 2338 some cases it’s possible to find models that yield precisely the same expected data,
 2339 yet produce wildly different estimates of N . In that sense, N for most practical
 2340 purposes is not identifiable across classes of different heterogeneity models, and
 2341 this should be understood before fitting any such model. One solution to this
 2342 problem is to seek to model explicit factors that contribute to heterogeneity, e.g.,
 2343 using individual covariate models (See 3.5 below). Indeed, spatial capture-recapture
 2344 models do just that, by modeling heterogeneity due to the spatial organization
 2345 of individuals in relation to traps or other encounter mechanism. For additional
 2346 background and applications of model M_h see Royle and Dorazio (2008, Chapt. 6)
 2347 and Kéry and Schaub (2012, Chapt. 6).

2348 We will work with a specific type of model M_h here which is a natural extension
 2349 of the basic binomial observation model of model M_0 so that

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

2350 where μ is a fixed parameter (the mean) to be estimated, and η_i is an individual
 2351 random effect assumed to be normally distributed:

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

2352 We could as well combine these two steps and write $\text{logit}(p_i) \sim \text{Normal}(\mu, \sigma_p^2)$. This
 2353 “logit-normal mixture” was analyzed by Coull and Agresti (1999) and elsewhere.
 2354 It is a natural extension of the basic model with constant p , as a mixed GLMM,

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

Model M_h has important historical relevance to spatial capture-recapture situations (Karanth, 1995) because investigators recognized that the juxtaposition of individuals with the array of trap locations should yield heterogeneity in encounter probability, and thus it became common to use some version of model M_h in spatial trapping arrays to estimate N . While this doesn’t resolve the problem of not knowing the effective sample area, it does yield an estimator that accommodates the heterogeneity in p induced by the spatial aspect of capture-recapture studies. To see how this juxtaposition induces heterogeneity, we have to understand the relevance of movement in capture-recapture models. Imagine a quadrat that can be uniformly searched by a crew of biologists for some species of reptile (see Royle and Young (2008)). Figure 3.3 shows a sample quadrat searched repeatedly over a period of time. Further, suppose that the species exhibits some sense of spatial fidelity in the form of a home range or territory, and individuals move about their home range (home range centroids are given by the solid dots) in some kind of random fashion. Heuristically, we imagine that each individual in the vicinity of the study area is liable to experience variable exposure to encounter due to the overlap of its home range with the sampled area - essentially the long-run proportion of times the individual is within the sample plot boundaries, say ϕ . We might model the exposure or *availability* of an individual to capture by supposing that $a_i = 1$ if individual i is available to be captured (i.e., within the survey plot) during any sample, and 0 otherwise. Then, $\Pr(a_i = 1) = \phi$. In the context of spatial studies, it is natural that ϕ should depend on *where* an individual lives, i.e., it should be individual-specific ϕ_i (Chandler et al., 2011). This system describes, precisely, that of “random temporary emigration” (Kendall et al., 1997) where ϕ_i is the individual-specific probability of being “available” for capture.

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

$$p_i = p_0\phi_i$$

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

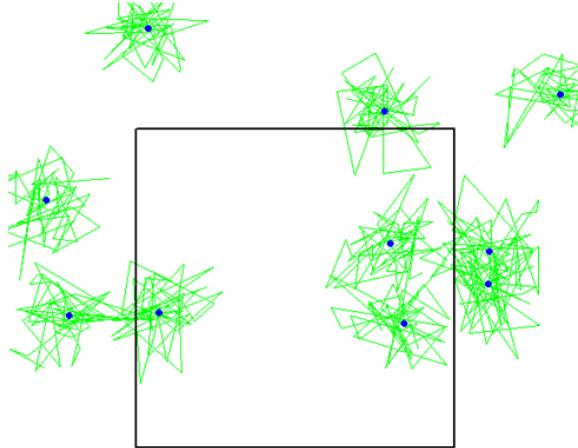


Figure 3.3. A quadrat searched for lizards over some period of time (simulated data). The locations of encounter for each of 10 lizards are connected by lines—the dots are activity centers.

2393 3.4.1 Analysis of model M_h

2394 If N is known, it is worth taking note of the essential simplicity of model M_h as a bi-
 2395 nomial GLMM. This is a type of model that is widely applied throughout statistics
 2396 using standard methods of inference based either on integrated likelihood (Laird
 2397 and Ware, 1982; Berger et al., 1999), which we discuss in Chapt. 5, or standard
 2398 Bayesian methods. However, because N is not known, inference is somewhat more
 2399 challenging. We address that here using Bayesian analysis based on data augmentation.
 2400 Although we use data augmentation in the context of Bayesian methods here,
 2401 we note that heterogeneity models formulated under DA are easily analyzed by
 2402 conventional likelihood methods as zero-inflated binomial mixtures (Royle, 2006)
 2403 and more traditional analysis of model M_h based on integrated likelihood, without
 2404 using data augmentation, has been considered by Coull and Agresti (1999), Dorazio
 2405 and Royle (2003), and others.

2406 As with model M_0 , we have the Bernoulli model for the zero-inflation variables:
 2407 $z_i \sim \text{Bernoulli}(\psi)$ and the model of the observations expressed conditional on these
 2408 latent variables z_i . For $z_i = 1$, we have a binomial model with individual-specific

2409 p_i :

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

2410 and otherwise $y_i|z_i=0 \sim I(y=0)$, i.e., a point mass at $y=0$. Further, we
2411 prescribe a distribution for p_i . Here we assume

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

2412 For prior distributions we assume $p_0 = \text{logit}^{-1}(\mu) \sim \text{Uniform}(0, 1)$ and, for the
2413 standard deviation $\sigma \sim \text{Uniform}(0, B)$ for some large B . Another common default
2414 prior is to assume $\tau = 1/\sigma^2 \sim \text{Gamma}(1, 1)$, although we usually choose $\sigma \sim$
2415 $\text{Uniform}(0, B)$.

2416 **3.4.2 Analysis of the Fort Drum data with model M_h**

2417 Here we provide an analysis of the Fort Drum bear survey data using the logit-
2418 normal heterogeneity model, and we used data augmentation to produce a data
2419 set of $M = 700$ individuals. We have so far mostly used **WinBUGS** but we are
2420 now transitioning to the use of **JAGS** run from within **R** using the useful packages
2421 **R2jags** or **rjags**. The function **jags** from the **R2jags** package runs essentially like
2422 the **bugs** function which we demonstrate here for setting up and running model M_h
2423 for the Fort Drum bear data:

```
2424 [... get data as before ....]
2425
2426 > set.seed(2013)
2427
2428 > cat("
2429 model{
2430   p0 ~ dunif(0,1)           # prior distributions
2431   mup <- log(p0/(1-p0))
2432   sigmap ~ dunif(0,10)
2433   taup <- 1/(sigmap*sigmap)
2434   psi ~ dunif(0,1)
2435
2436   for(i in 1:(nind+nz)){
2437     z[i] ~ dbern(psi)       # zero inflation variables
2438     lp[i] ~ dnorm(mup,taup) # individual effect
2439     logit(p[i]) <- lp[i]
2440     mu[i] <- z[i]*p[i]
2441     y[i] ~ dbin(mu[i],K)    # observation model
2442   }
2443
2444   N<-sum(z[1:(nind+nz)])
2445 }
2446 ",file="modelMh.txt")
```

```

2447 > data1 <- list(y=y, nz=nz, nind=nind, K=K)
2448 > params1 <- c('p0','sigmap','psi','N')
2449 > inits <- function(){ list(z=as.numeric(y>=1), psi=.6, p0=runif(1),
2450           sigmap=runif(1,.7,1.2),lp=rnorm(M,-2)) }
2451 > library(R2jags)
2452 > wabout <- jags(data1, inits, params1, model.file = "modelMh.txt", n.chains = 3,
2453   n.iter = 1010000, n.burnin =10000, working.directory = getwd())

```

2454 We provide an **R** function `modelMhBUGS` in the package `scrbook` which will fit
 2455 the model using either **JAGS** or **WinBUGS** as specified by the user. In addition,
 2456 for fun, we construct our own MCMC algorithm using a Metropolis-within-Gibbs
 2457 algorithm for model M_h in Chapt. 16, where we also develop MCMC algorithms for
 2458 spatial capture-recapture models. Using `modelMhBUGS`, we ran 3 chains of 1 *million*
 2459 iterations (mixing is poor for this model and this data set), which produced the
 2460 posterior distribution for N shown in Fig. 3.4. Posterior summaries of parameters
 2461 are given in Table 3.5.

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

parameter	Mean	SD	2.5%	50 %	97.5%	Rhat	n.eff
p_0	0.072	0.056	0.002	0.060	0.203	1.008	540
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

2462 We used $M = 700$ for this analysis and we note that while the posterior mass
 2463 of N is concentrated away from this upper bound (Fig. 3.4), the posterior has an
 2464 extremely long right tail, with some MCMC draws at the upper boundary $N = 700$,
 2465 suggesting that an even higher value of M may be called for. To characterize the
 2466 posterior distribution of density we produce the relevant summaries of the posterior
 2467 distribution of $D = N/277.11$ (recall the buffered area of the convex hull is 277.11
 2468 km^2):

```

2469 > summary(wabout$sims.list$N/277.11)
2470   Min. 1st Qu. Median Mean 3rd Qu. Max.
2471 0.1696 0.2959 0.3681 0.4428 0.4944 2.5260
2472
2473 > quantile(wabout$sims.list$N/277.11,c(0.025,0.50,0.975))
2474   2.5% 50% 97.5%
2475 0.2237379 0.3680849 1.1511674

```

2476 Therefore, the point estimate, characterized by the posterior median, is around 0.37
 2477 bears per square km and a 95% Bayesian credible interval is (0.224, 1.151).

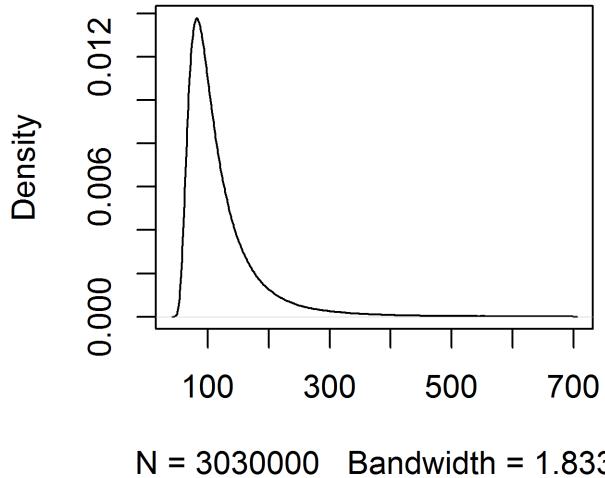


Figure 3.4. Posterior of N for Fort Drum bear study data under the logit-normal version of model M_h .

3.4.3 Comparison with MLE

The posterior of N is highly skewed; therefore, we see that the posterior mean ($N = 122.7$) is considerably higher than the posterior median ($N = 102$). Further, it may be surprising that these posterior summaries do not compare well with the MLE. We used the **R** code contained in Panel 6.1 from Royle and Dorazio (2008) to obtain the MLE of $\log(n_0)$, the logarithm of the number of uncaptured individuals, is $\widehat{\log(n_0)} = 3.86$ and therefore $\hat{N} = \exp(3.86) + 47 = 94.47$, which is larger than the mode shown in Fig. 3.4. To see this, we compute the posterior mode, by finding the posterior value of N with the highest mass. Because N is discrete, we can use the `table()` function in **R** and find the most frequent value⁴. If we want to smooth out some of the Monte Carlo error a bit, we can use a smoother of some sort applied to the tabulated posterior frequencies of N . Here we use a smoothing spline (**R** function `smooth.spline`) with the degree of smoothing chosen by cross-validation (the `cv=TRUE` argument):

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

```

2492 > N <- table(jout$BUGSoutput$sims.list$N)
2493 > xg <- as.numeric(names(N))
2494
2495 > sp <- smooth.spline(xg,N,cv=TRUE)
2496
2497 > sp
2498
2499 Call:
2500 smooth.spline(x = xg, y = N, cv = TRUE)
2501
2502 Smoothing Parameter spar= 0.09339815 lambda= 8.201724e-09 (17 iterations)
2503 Equivalent Degrees of Freedom (Df): 121.1825
2504 Penalized Criterion: 2544481
2505 PRESS: 5903.4

```

2506 We obtain the mode of the smoothed frequencies as follows:

```

2507 sp$x[sp$y==max(sp$y)]
2508 [1] 82

```

2509 We don't dwell too much on the difference between the MLE and features of the
 2510 posterior, but we do note here that the posterior distribution for the parameters of
 2511 this model, for the Fort Drum data set, are very sensitive to the prior distributions.
 2512 In the present case, the use of a Uniform(0, 1) prior for $p_0 = \text{logit}^{-1}(\mu)$ is somewhat
 2513 informative—in particular, it is not at all “flat” on the scale of μ , and this affects
 2514 the posterior. We generally always recommend use of a Uniform(0, 1) prior for
 2515 $\text{logit}^{-1}(\mu)$ in such models. That said, we were surprised at this result, and we
 2516 experimented with other prior configurations including putting a flat prior on μ
 2517 directly. This kind of small sample instability has been widely noted in model
 2518 M_h (Fienberg et al., 1999; Dorazio and Royle, 2003), as has extreme sensitivity to
 2519 the specific form of model M_h (Link, 2003). In summary, while the mode is well-
 2520 defined, the data set is relatively sparse and hence inferences are poor and sensitive
 2521 to model choice.

3.5 INDIVIDUAL COVARIATE MODELS: TOWARD SPATIAL CAPTURE-RECAPTURE

2522 A standard situation in capture-recapture models is when a covariate which is
 2523 thought to influence encounter probability is measured for each individual. These
 2524 are often called “individual covariate models” but, in keeping with the classical
 2525 nomenclature on closed population models, Kéry and Schaub (2012) referred to
 2526 this class of models as “model M_x ” (the x here being an explicit covariate). As
 2527 with other closed population models, we begin with the basic binomial observation
 2528 model:

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

2529 To model the covariate, we use a logit model for encounter probability of the form:

2530

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

2531 where x_i is the covariate value for individual i and the parameters $\boldsymbol{\alpha} = (\alpha_0, \alpha_1)$
 2532 are the regression coefficients. Classical examples of covariates influencing detec-
 2533 tion probability are type of animal (juvenile/adult or male/female), a continuous
 2534 covariate such as body mass, or a discrete covariate such as group or cluster size.
 2535 For example, in models of aerial survey data, it is natural to model the detection
 2536 probability of a group as a function of the observation-level individual covariate,
 2537 “group size” (Royle, 2008; Langtimm et al., 2011).

2538 Model M_x is similar in structure to model M_h , except that the individual effects
 2539 are *observed* for the n individuals that appear in the sample. These models are
 2540 important here because spatial capture-recapture models can be described precisely
 2541 as a form of model M_x , where the covariate describes *where* the individual is located
 2542 in relation to the trapping array. Specifically, SCR models *are* individual covariate
 2543 models, but where the individual covariate is only observed imperfectly (or partially
 2544 observed) for each captured individual. Unlike model M_h , in SCR models (and
 2545 model M_x) we do have some direct information about the latent variable, which
 2546 comes from the spatial locations/distribution of individual recaptures.

2547 Traditionally, estimation of N in model M_x is achieved using methods based on
 2548 ideas of unequal probability sampling (i.e., Horvitz-Thompson estimation⁵; Huggins
 2549 (1989), Alho (1990) and Borchers et al. (2002)). An estimator of N is

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

2550 where \tilde{p}_i is the probability that individual i appeared in the sample. This quantity
 2551 is $\tilde{p}_i = \Pr(y_i > 0)$ and, in closed population capture-recapture models, it can be
 2552 computed as:

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

2553 where p_i is a function of parameters α_0 and α_1 according to Eq. 3.5.1. In practice,
 2554 parameters are estimated from the conditional-likelihood of the observed encounter
 2555 histories which is, for observation y_i ,

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

2556 This derives from a straightforward application of the law of total probability.
 2557 Conceptually, we partition $\Pr(y)$ according to $\Pr(y) = \Pr(y|y > 0)\Pr(y > 0) +$
 2558 $\Pr(y|y = 0)\Pr(y = 0)$. For any positive value of y the 2nd term is necessarily

⁵For a quick summary of the idea see:

http://en.wikipedia.org/wiki/Horvitz-Thompson_estimator

2559 0, and so we rearrange to obtain $\Pr(y|y > 0) = \Pr(y)/\Pr(y > 0)$ which, in the
 2560 specific case where $\Pr(y)$ is the binomial probability mass function (pmf) produces
 2561 Eq. 3.5.2.

2562 Here we take a formal model-based approach to Bayesian analysis of such models
 2563 based on the joint likelihood using data augmentation (Royle, 2009). Classical
 2564 likelihood analysis of the so-called “full likelihood” is covered by Borchers et al.
 2565 (2002). For Bayesian analysis of model M_x , because the individual covariate is
 2566 unobserved for the $n_0 = N - n$ uncaptured individuals, we require a model to
 2567 describe variation in x among individuals, essentially allowing the sample to be
 2568 extrapolated to the population. For example, if we have a continuous trait measured
 2569 on each individual, then we might assume that x has a normal distribution:

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

2570 Data augmentation can be applied directly to this class of models. In particular,
 2571 reformulation of the model under DA yields a basic zero-inflated binomial model of
 2572 the following form, for each $i = 1, 2, \dots, M$:

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

2573 Fully spatial capture-recapture models use this formulation with a latent covariate
 2574 that is directly related to the individual detection probability (see next section).
 2575 As with the previous models, implementation is trivial in the **BUGS** language.
 2576 The **BUGS** specification is very similar to that for model M_h , but we require the
 2577 distribution of the covariate to be specified, along with priors for the parameters of
 2578 that distribution.

2579 3.5.1 Example: Location of capture as a covariate

2580 Here we consider a special type of model M_x that is especially relevant to spatial
 2581 capture-recapture. Intuitively, some measure of distance from home range center
 2582 to traps for an individual should be a reasonable covariate to explain heterogeneity
 2583 in encounter probability, i.e., individuals with more exposure to traps should have
 2584 higher encounter probabilities and vice versa. So we can imagine *estimating* such a
 2585 quantity, say average distance from home range center to “the trap array”, and then
 2586 using it as an individual covariate in capture-recapture models. A version of this
 2587 idea was put forth by Boulanger and McLellan (2001) (see also Ivan (2012)), but
 2588 using the Huggins-Alho estimator and with covariate “distance from home range
 2589 center to edge” of the trapping array, where the home range center is estimated by
 2590 the average capture location. This is intuitively appealing because we can imagine,

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

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

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

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

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

3.5.2 Fort Drum bear study

We have to do a little bit of data processing to fit this individual covariate model to the Fort Drum data. We need to compute the individual covariate \mathbf{x}_i (distance from the centroid of the trapping array) using the R function `spiderplot` provided in `scrbook`. This function also produces the keen plot shown in Fig. 3.5 which we call

2629 a “spider plot”. The **R** commands for obtaining the individual covariate “distance
 2630 from trap centroid” (the variable `xcent` returned by `spiderplot`) and making the
 2631 spider plot are as follows:

```
2632 > library(scrbook)
2633 > data(beardata)
2634 > toad <- spiderplot(beardata$bearArray,beardata$trapmat)
2635 > xcent <- toad$xcent
```

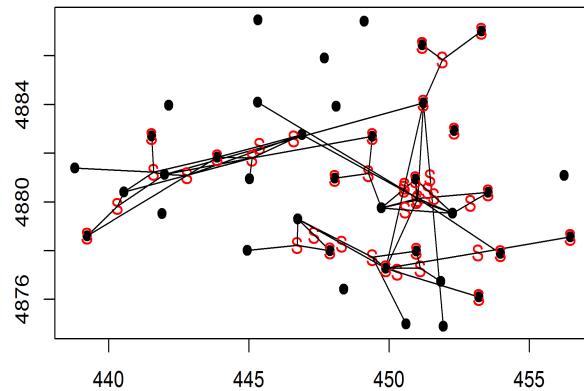


Figure 3.5. Spider plot of the Fort Drum study data. The black dots represent the 47 trap locations with the “S” symbols being the average capture location of each bear. i.e., its estimated home range center. All traps in which a bear was captured are connected to its estimated home range center with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

2636 For the analysis of these data using the individual covariate “distance from
 2637 centroid” we used $x_i \sim \text{Uniform}(0, B)$ with $B = 11.5 \text{ km}^2$, which is about the
 2638 distance from the array center to the furthest trap. Once we choose a value for B ,
 2639 the direct implication is that the population size parameter, N , applies to the area
 2640 within 11.5 units of the trap centroid. Therefore, the model associates a precise
 2641 area within which the population of N individuals resides. We will see shortly
 2642 that N does, in fact, scale with our choice of B to reflect the changing area over
 2643 which the N individuals of the model reside. The **BUGS** model specification and
 2644 **R** commands to package the data and fit the model are as follows:

```
2645 cat("
```

```

2646 model{
2647   p0 ~ dunif(0,1)                      # prior distributions
2648   alpha0 <- log(p0/(1-p0))
2649   psi ~ dunif(0,1)
2650   beta ~ dnorm(0,.01)
2651
2652 for(i in 1:(nind+nz)){
2653   xcent[i] ~ dunif(0,B)
2654   z[i] ~ dbern(psi)                     # DA variables
2655   lp[i] <- alpha0 + beta*xcent[i] # individual effect
2656   logit(p[i]) <- lp[i]
2657   mu[i] <- z[i]*p[i]
2658   y[i] ~ dbin(mu[i],K)                 # observation model
2659 }
2660
2661 N <- sum(z[1:(nind+nz)])
2662 }
2663 ",file="modelMcov.txt")
2664
2665 data2 <- list(y=y,nz=nz, nind=nind, K=K, xcent=xcent,B=11.5)
2666 params2 <- c('p0','psi','N','beta')
2667 inits <- function() {list(z=zst, psi=psi, p0=runif(1), beta=rnorm(1) ) }
2668 fit2 <- bugs(data2, inits, params2, model.file="modelMcov.txt",
2669                   n.chains=3, n.iter=11000, n.burnin=1000, n.thin=1)

```

2669 This produces the posterior summary statistics in Table 3.6.

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

parameter	Mean	SD	2.5%	50 %	97.5%	Rhat	n.eff
p_0	0.54	0.07	0.40	0.54	0.67	1	1100
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

2670 We note that the estimated N is much lower than obtained by model M_h but
2671 there is a good explanation for this which we discuss in the next section. That issue
2672 notwithstanding, it is worth pondering how this model could be an improvement
2673 (conceptually or technically) over some other model/estimator including M_0 and
2674 M_h considered previously. Well, for one, we have accounted formally for hetero-
2675 geneity due to spatial location of individuals relative to exposure to the trap array,
2676 characterized by the centroid of the array. Moreover, we have done so using a model
2677 that is based on an explicit mechanism, as opposed to a phenomenological one such

2678 as model M_h . In addition, and importantly, using our new model, *the estimated*
 2679 *N applies to an explicit area which is defined by our prescribed value of B*. That
 2680 is, this area is a fixed component of the model and the parameter N therefore has
 2681 explicit spatial context, as the number of individuals with home range centers less
 2682 than B from the centroid of the trap array. As such, the implied “effective area”
 2683 of the trap array for a given B is a precisely defined quantity—it is that of a circle
 2684 with radius B .

2685 **3.5.3 Extension of the model**

2686 The model developed in the previous section is not a very good model for one
 2687 important reason: Imposing a uniform prior distribution on x implies that density
 2688 is *not constant* over space. In particular, this model implies that density *decreases*
 2689 as we move away from the centroid of the trap array. That is, $x_i \sim \text{Uniform}(0, B)$
 2690 implies constant N in each distance band from the centroid but obviously the *area*
 2691 of each distance band is increasing. This is one reason we have a lower estimate of
 2692 density than that obtained previously from model M_h (Sec. 3.4.2) and also why, if
 2693 we were to increase B , we would see density continue to decrease.

2694 Fortunately, we are not restricted to use of this specific distribution for the
 2695 individual covariate. Clearly, it is a bad choice and, therefore, we should think
 2696 about whether we can choose a better distribution for B —one that doesn’t imply a
 2697 decreasing density as distance from the centroid increases. Conceptually, what we
 2698 want to do is impose a prior on distance from the centroid, x , such that abundance
 2699 should be proportional to the amount of area in each successive distance band as
 2700 you move farther away from the centroid, so that density is *constant*. In fact, theory
 2701 exists which tells us we should choose $[x] = 2x/B^2$. This can be derived by noting
 2702 that $F(x) = \Pr(X < x) = (\pi x^2)/(\pi * B^2)$. Then, $f(x) = dF/dx = 2 * x/(B^2)$.
 2703 This is a sort of triangular distribution in density induced because the incremental
 2704 area in each additional distance band increases linearly with radius (i.e., distance
 2705 from centroid). This can be verified empirically as follows:

```
2706 > u <- runif(10000,-1,1)
2707 > v <- runif(10000,-1,1)
2708 > d <- sqrt(u*u+v*v)
2709 > hist(d[d<1])
2710 > hist(d[d<1],100)
2711 > hist(d[d<1],100,probability=TRUE)
2712 > abline(0,2)
```

2713 It would be useful if we could describe this distribution directly in **BUGS** but
 2714 there is not a built-in way to do so. However, we can implement a discrete version
 2715 of the pdf⁶. To do this, we break B into L distance classes of width δ , with
 2716 probabilities proportional to $2 * x$. In particular, if we denote the cut-points by

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.

2717 $g_1 = 0, g_2, \dots, g_{L+1} = B$ and the interval midpoints are $m_i = g_{i+1} - \delta$. Then the
 2718 interval probabilities are, approximately⁷, $p_i = \delta(2m_i/B^2)$, which we can compute
 2719 once and then pass them to **BUGS** as data. The **R** commands for doing all of this
 2720 (noting that we have already loaded and processed the Fort Drum bear data) are
 2721 given in the following **R/BUGS** script:

```

2722 > delta <- .2
2723 > xbin <- xcent%/%delta + 1                      # Put x in bins
2724 > midpts <- seq(delta,Dmax,delta)
2725 > xprobs <- delta*(2*midpts/(B*B))
2726 > xprobs <- xprobs/sum(xprobs)

2727 > cat("
2728 model{
2729   p0 ~ dunif(0,1)                                # Prior distributions
2730   alpha0 <- log(p0/(1-p0))
2731   psi ~ dunif(0,1)
2732   beta ~ dnorm(0,.01)

2733   for(i in 1:(nind+nz)){
2734     xbin[i] ~ dcat(xprobs[])
2735     z[i] ~ dbern(psi)                            # DA variables
2736     lp[i] <- alpha0 + beta*xbin[i]*delta        # Individual covariate model
2737     logit(p[i]) <- lp[i]
2738     mu[i] <- z[i]*p[i]
2739     y[i] ~ dbin(mu[i],K)                         # Observation model
2740   }
2741
2742   N <- sum(z[1:(nind+nz)])                      # N is derived
2743 }
2744 ",file="modelMcov.txt")

```

2747 In the model description, the variable x (observed distance from centroid of
 2748 the trap array) has been rounded or binned (placed into a distance bin) so that
 2749 the discrete version of the pdf of x can be used, as described previously. The new
 2750 variable labeled **xbin** is then the *integer category* in units of δ from 0. Thus, to
 2751 convert back to distance in the expression for $lp[i]$, **xbin[i]** has to be multiplied
 2752 by δ . To fit the model, keeping in mind that the data objects required below have
 2753 been defined in previous analyses of this chapter, we do this:

```

2754 > data2 <- list(y=y, nz=nz, nind=nind, K=K, xbin=xbin, xprobs=xprobs,
2755   delta=delta)
2756 > params2 <- c('p0','psi','N','beta')
2757 > inits <- function() {list(z=z, psi=psi, p0=runif(1),beta=rnorm(1) ) }
2758 > fit <- bugs(data2, inits, params2, model.file="modelMcov.txt",
2759   working.directory=getwd(), debug=FALSE, n.chains=3,

```

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

2760 n.iter=11000, n.burnin=1000, n.thin=2)

2761 By specification of B , this model induces a clear definition of area in which the
 2762 population of N individuals reside. The parameter N of the model is the population
 2763 size that applies to the particular value of B and, as such, we will see that N scales
 2764 with our choice of B . This might be disconcerting to some—we can get whatever
 2765 value of N we want by changing B ! However, it is intuitively reasonable that, as
 2766 we increase the area under consideration, there should be more individuals in it.
 2767 Fortunately, we find empirically, that while N is highly sensitive to the prescribed
 2768 value of B , density appears invariant to B as long as B is sufficiently large. We
 2769 fit the model for a set of values of B from $B = 12$ (restricting values of x to be in
 2770 close proximity to the trap array) on up to 20. The results are given in Table 3.7.

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

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

2771 We see that the posterior mean and SD of density (individuals per square km)
 2772 appear insensitive to choice of B once we reach about $B = 17$ or so. The estimated
 2773 density of 0.25 per km^2 is actually quite a bit lower than we reported using model
 2774 M_h for which no relevant “area” quantity is explicit in the model (and so we had
 2775 to make it up). Using MLEs of N in conjunction with buffer strips (see Tab. 1.1)
 2776 our estimates were in the range of $0.32 - 0.43$ and see Sec. 3.4 above. On the
 2777 other hand our estimate of $\hat{D} = 0.25$ here (based on the posterior mean) is higher
 2778 than that reported from model M_0 using the buffered area ($\hat{D} = 0.18$). There is
 2779 no basis really for comparing or contrasting these various estimates. In particular,
 2780 application of models M_0 and M_h are distinctly *not* spatially explicit models—the
 2781 area within which the population resides is not defined under either model. There is
 2782 therefore no reason at all to think that the estimates produced under either closed
 2783 population model, based on a buffered “trap area”, are justifiable by any theory.
 2784 In fact, we would get exactly the same estimate of N no matter what we declare
 2785 the area to be. On the other hand, the individual covariate model uses an explicit
 2786 model for “distance from centroid” that is a reasonable and standard null model—
 2787 it posits, in the absence of direct information, that individual home range centers
 2788 are randomly distributed in space and that probability of detection depends on the
 2789 distance between home range center and the centroid of the trap array. Under this

2790 definition of the system, we see that density is invariant to the choice of area, which
 2791 seems like a desirable feature.

2792 **3.5.4 Invariance of density to B**

2793 Under model M_x , and also under models that we consider in later chapters, a
 2794 general property of the estimators is that while N increases with the prescribed
 2795 area of the model (defined by B in this model), we expect that density estimators
 2796 should be invariant to this area. In the model used above, we note that $\text{Area}(B) =$
 2797 πB^2 and $\mathbb{E}(N(B)) = \lambda \text{Area}(B)$ and thus $\mathbb{E}(\text{Density}(B)) = \lambda$, i.e., constant. This
 2798 should be interpreted as the *prior* density. Absent data, then realizations under the
 2799 model will have density λ regardless of what B is prescribed to be. As we verified
 2800 empirically above, posterior summaries of density are also invariant to B as long
 2801 as the prescribed area is sufficiently large.

2802 **3.5.5 Toward fully spatial capture-recapture models**

2803 While the use of an individual covariate model resolves two important problems
 2804 inherent in almost all capture-recapture studies (induced heterogeneity and absence
 2805 of a precise relationship between N and area), is not ideal for all purposes because
 2806 it does not make full use of the spatial information in the data set, i.e., the trap
 2807 locations and the locations of each individual encounter, so that we cannot use
 2808 this model to model trap-specific effects (e.g., trap effort or type). Moreover, we
 2809 applied this model for “data” being the average observed encounter location, and
 2810 equated that summary to the home range center s_i . Intuitively, taking the average
 2811 encounter location as an estimate of home range center makes sense but more so
 2812 when the trapping grid is dense and expansive relative to typical home range sizes
 2813 which might not be reasonable in practice. Moreover, this approach also ignored
 2814 the variable precision with which each s_i is estimated. Finally, it ignores that
 2815 estimates of s_i around the “edge” (however we define that) are biased because the
 2816 observations are truncated—we can only observe locations interior to the array.

2817 However, there is hope to extend this model in order to resolve these remaining
 2818 deficiencies. In the next chapter we provide a further extension of this individual
 2819 covariate model that definitively resolves the *ad hoc* nature of the approach we took
 2820 here. In that chapter we build a model in which s_i are regarded as latent variables
 2821 and the observation locations (i.e., trap specific encounters) are linked to those
 2822 latent variables with an explicit model. We note that the model fitted previously
 2823 could be adapted easily to deal with s_i as a latent variable, simply by adding a
 2824 prior distribution for s_i . This is actually easier, and less ad hoc in a number of
 2825 respects, and you should try it out.

3.6 DISTANCE SAMPLING: A PRIMITIVE SCR MODEL

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

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

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

for $\alpha_1 > 0$, where x_i denotes the distance at which the i th individual is detected relative to some reference location where perfect detectability ($p = 1$) is assumed. This encounter probability model is more often written with $\alpha_1 = 1/2\sigma^2$. If $K > 1$ then an intercept in this model, say α_0 , is identifiable and such models are usually called "capture-recapture distance sampling" (Alpízar-Jara and Pollock, 1996; Borchers et al., 1998).

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

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

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

2863 distance sampling model in the **BUGS** language is shown in Panel 3.2, taken from
 2864 Royle and Dorazio (2008).

```

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

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

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

```

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

2865 As with the individual covariate model in the previous section, the distance
 2866 sampling model can be equivalently specified by putting a prior distribution on
 2867 individual *location* instead of distance between individual and observation point
 2868 (or transect). Thus we can write the general distance sampling model as

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

2869 along with

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

2870 where \mathbf{x}_0 is a fixed point (or line) and \mathbf{u}_i is the individual's location, which is
 2871 observed for the sample of n individuals. In practice it is easier to record distance
 2872 instead of location. Basic math can be used to argue that if individuals have a
 2873 uniform distribution in space, then the distribution of Euclidean distance is also
 2874 uniform. In particular, if a transect of length L is used and x is distance to the
 2875 transect then $F(x) = \Pr(X \leq x) = L * x / L * B = x / B$ and $f(x) = dF/dx = (1/B)$.
 2876 For measurements of radial distance, we provided the analogous argument in the
 2877 previous section.

2878 The preceding paragraph makes it clear that distance sampling is a special case
 2879 of spatial capture-recapture models, such as those derived from model M_x of the
 2880 previous section, where the encounter probability is related directly to *distance*,
 2881 which is a reduced information summary of *location*, \mathbf{u} . Some intermediate forms
 2882 of SCR/DS models can be described (Royle et al., 2011a). In the context of our
 2883 general characterization of SCR models (Chapt. 2.6), we suggested that every SCR

2884 model can be described, conceptually, by a hierarchical model of the form:

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

2885 Distance sampling ignores the part of the model pertaining to \mathbf{s} , and deals only
 2886 with the model components for the observed data \mathbf{u} ⁸. Thus, we are left with a
 2887 hierarchical model of the form

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

2888 In contrast, as we will see in the next chapters, many SCR models (Chapt. 4)
 2889 ignore \mathbf{u} and condition on \mathbf{s} , which is not observed:

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

2890 Since $[\mathbf{u}]$ and $[\mathbf{s}]$ are both assumed to be uniformly distributed, these are equivalent
 2891 models! The main differences have to do with interpretation of model components
 2892 and whether or not the latent variables are observable (in distance sampling they
 2893 are).

2894 So why bother with SCR models when distance sampling yields density esti-
 2895 mates and accounts for spatial heterogeneity in detection? For one, imagine trying
 2896 to collect distance sampling data on species such as jaguars or tigers! Clearly,
 2897 distance sampling requires that one can collect large quantities of distance data,
 2898 which is not always possible. For tigers, it is much easier, efficient, and safer to
 2899 employ camera traps or track plates and then apply SCR models. Furthermore, as
 2900 we will see in Chapt. 14, SCR models can make use of distance data, allowing us
 2901 to study distribution, movement, and density. Thus, SCR models are more general
 2902 and versatile than distance sampling models (which clearly are a special case), and
 2903 can accommodate data from virtually all animal survey designs.

2904 3.6.1 Example: Sonoran desert tortoise study

2905 We illustrate the application of distance sampling models using data on the Sonoran
 2906 desert tortoise (*Gopherus agassizii*), shown in Fig. 3.6, collected along transects
 2907 in southern Arizona (see Zylstra et al. (2010) for details). The data are from 120
 2908 square transects having four 250-m sides, although we ignore this detail in our
 2909 analysis here and regard them as 1 km transects, and we pooled the detection
 2910 data from all 120 transects. The histogram of encounter distances from the 65
 2911 encountered individuals is shown in Fig. 3.7

2912 Commands for reading in and organizing the data for analysis using **Win-**
2913 BUGS are given in the help file `?tortoise` provided with the `scrbook` package.
 2914 To compute density, the total sampled area of the transects `striparea` is input
 2915 as data, and computed as: 120 (transects) multiplied by the length (1000 m) and
 2916 half-width ($B = 40$ m), then multiplied by 2, and divided by 10000 to convert to

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



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

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

Posterior summaries for the tortoise data are given in Tab. 3.8. Estimated density (posterior mean) is 0.54 individuals per ha and the estimated scale parameter of the distance function (posterior mean) is $\sigma = 9.12$ meters. The R-hat statistics of around 1.02 suggest that slightly longer MCMC simulations might be called for. The posterior mass of the data augmentation parameter ψ is located away from the upper bound $\psi = 1$ and so the degree of data augmentation appears sufficient.

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

parameter	Mean	SD	2.5%	50 %	97.5%	Rhat	n.eff
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
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
ψ	0.61	0.07	0.49	0.61	0.75	1.02	96

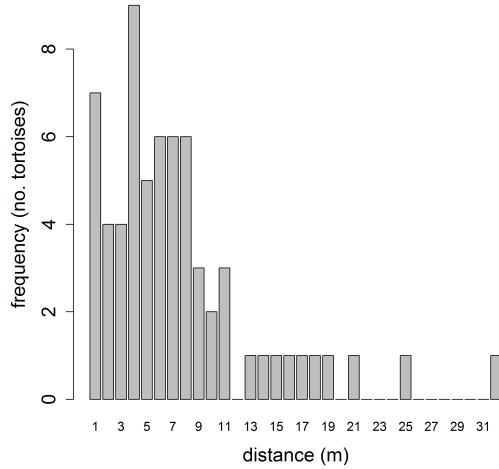


Figure 3.7. Distance histogram of $n = 65$ Sonoran desert tortoise detections from a total of 120 km of survey transect.

3.7 SUMMARY AND OUTLOOK

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

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

2945 using distance to the centroid of the trapping array as a covariate. We analyzed
2946 the *full likelihood* using data augmentation, and placed a prior distribution on the
2947 individual covariate which was derived from an assumption that individual loca-
2948 tions are, a priori, uniformly distributed in space. This assumption provides for
2949 invariance of the density estimator to the choice of population size area (induced
2950 by maximum distance from the centroid of the trap array). The model addressed
2951 some important problems in the use of closed population models: it allows for het-
2952 erogeneity in encounter probability due to the spatial juxtaposition of individuals
2953 with the array of traps, and it also provides a direct estimate of density because
2954 area is a feature of the model (via the prior on the individual covariate). The model
2955 is still not completely general, however, because it does not make full use of the
2956 spatial encounter histories, which provide direct information about the locations
2957 and density of individuals.

2958 A specific individual covariate model that is in widespread use is classical dis-
2959 tance sampling. The model underlying distance sampling is precisely a special
2960 kind of SCR model—but one without replicate samples. Understanding distance
2961 sampling and individual covariate models more broadly provides a solid basis for
2962 understanding and analyzing spatial capture-recapture models. In fact if, instead
2963 of placing an explicit model on *distance* in the classical distance sampling model,
2964 we were to place the prior distribution on *location*, \mathbf{s} , of each individual, then the
2965 form of the distance sampling model more closely resembles the SCR model we
2966 introduce in the next chapter.

2967

Part II

2968

2969

Basic SCR Models

2970
2971

2972

4

2973

FULLY SPATIAL CAPTURE-RECAPTURE MODELS

2974 In the previous chapter, we discussed models that could be viewed as primitive
2975 spatial capture-recapture models. We looked at a basic distance sampling model,
2976 and we also considered a classical individual covariate modeling approach in which
2977 we defined a covariate to be the distance from the (estimated) home range center to
2978 the center of the trap array. The individual covariate model that we conjured up was
2979 “spatial” in the sense that it included some characterization of where individuals
2980 live but, on the other hand, only a primitive or no characterization of trap location.
2981 That said, there is only a small step from this model to spatial capture-recapture
2982 models that we consider in this chapter, which fully recognize the spatial attribution
2983 of both individual animals *and* the locations of encounter devices.

2984 Capture-recapture models must accommodate the spatial organization of indi-
2985 viduals and the encounter devices because the encounter process occurs at the level
2986 of individual traps. Failure to consider the trap-specific data is one of the key defi-
2987 ciencies with classical ad-hoc approaches which aggregate encounter information to
2988 the resolution of the entire trap array. We have previously addressed some problems
2989 that this causes including induced heterogeneity in encounter probability, imprecise
2990 notation of “sample area” and not being able to accommodate trap-specific effects
2991 or trap-specific missing values. In this chapter we resolve these issues by developing
2992 our first fully spatial capture-recapture model. This model is not too different from
2993 that considered in Sec. 3.5 but, instead of defining the individual covariate to be
2994 distance to the centroid of the array we define J individual covariates - the distance
2995 to *each* trap. And, instead of using estimates of individual locations \mathbf{s} , we consider
2996 a fully hierarchical model in which we regard \mathbf{s} as a latent variable and impose a
2997 prior distribution on it.

2998 In this chapter we investigate the basic spatial capture-recapture model, which
 2999 we refer to as “model SCR0”, and address some important considerations related
 3000 to its analysis in **BUGS**. We demonstrate how to summarize posterior output for
 3001 the purposes of producing density maps or spatial predictions of density. The key
 3002 aspect of the SCR models considered in this chapter is the formulation of a model
 3003 for encounter probability that is a function of distance between individual home
 3004 range center and trap locations. We also discuss how encounter probability models
 3005 are related to explicit models of space usage or “home range area.” Understanding
 3006 this allows us to compute, for example, the area used by an individual during some
 3007 prescribed time. While it is intuitive that SCR models should be related to some
 3008 model of space usage, this has not been discussed much in the literature (but see
 3009 Royle et al. (2012b) which we address further in Chapt. 12).

4.1 SAMPLING DESIGN AND DATA STRUCTURE

3010 In our development here, we will assume a standard sampling design in which
 3011 an array of J traps is operated for K sample occasions (say, nights) producing
 3012 encounters of n individuals. Because sampling occurs by traps and also over time,
 3013 the most general data structure yields temporally *and* spatially indexed encounter
 3014 histories for *each individual*. Thus a typical data set will include an encounter
 3015 history *matrix* for each individual indicating which trap the individual was captured,
 3016 during each sample occasion. For example, suppose we sample at 4 traps over 3
 3017 nights. A plausible data set for a single individual captured one time in trap 1 on
 3018 the first night and one time in trap 3 on the 3rd night is:

```
3019     night1 night2 night3
3020 trap1    1    0    0
3021 trap2    0    0    0
3022 trap3    0    0    1
3023 trap4    0    0    0
```

3024 This data structure would be obtained for *each* of the $i = 1, 2, \dots, n$ captured
 3025 individuals.

3026 We develop models in this chapter for passive detection devices such as “hair
 3027 snares” or other DNA sampling methods (Kéry et al., 2010; Gardner et al., 2010b)
 3028 and related types of sampling devices in which (i) devices (“traps”) may capture any
 3029 number of individuals (i.e., they don’t fill up); (ii) an individual may be captured
 3030 in more than one trap during each occasion but (iii) individuals can be encountered
 3031 at most 1 time by each trap during any occasion. Hair snares for sampling DNA
 3032 from bears and other species function according to these rules. An individual bear
 3033 wandering about its territory might come into contact with > 1 devices; a device
 3034 may encounter multiple bears; however, in practice, it will often not be possible to
 3035 attribute multiple visits of the same individual during a single occasion (e.g., night)
 3036 to distinct encounter events. Thus, an individual may be captured at most 1 time

Table 4.1. Hypothetical spatial capture-recapture data set showing 6 individuals captured in 4 traps. Each entry is the number of captures out of $K = 3$ nights of sampling.

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

in each trap during any occasion. While this model, which we refer to as SCR0, is most directly relevant to hair snares and other DNA sampling methods for which multiple detections of an individual are not distinguishable, we will also make use of the model for data that arise from camera-trapping studies. In practice, with camera trapping, individuals might be photographed several times in a night but it is common to distill such data into a single binary encounter event for reasons discussed later in Chapt. 8.

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

4.2 THE BINOMIAL OBSERVATION MODEL

We begin by considering the simple model in which there are no time-varying covariates that influence encounter, there are no explicit individual-specific covariates, and there are no covariates that influence density. In this case, we can aggregate the binary encounters over the K sample occasions and record the total number of encounters out of K . We will denote these individual- and trap-specific encounter frequencies by y_{ij} for $i = 1, 2, \dots, n$ captured individuals and $j = 1, 2, \dots, J$ traps. For example, suppose we observe 6 individuals in sampling at 4 traps over 3 nights of sampling then a plausible data set is the 6×4 matrix of encounters (out of 3 sampling occasions) shown in Table 4.1. We assume that y_{ij} are mutually independent outcomes of a binomial random variable which we express as:

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

This is the basic model underlying standard closed population models (Chapt. 3) except that, in the present case, the encounter frequencies are individual- *and* trap-specific, and encounter probability p_{ij} depends on both individual *and* trap.

As we did in Sec. 3.5, we will make explicit the notion that p_{ij} is defined conditional on *where* individual i lives. Naturally, we think about defining an individual

3065 home range and then relating p_{ij} explicitly to a summary of its location relative to
 3066 each trap. For example, the centroid of the individuals home range, or its center of
 3067 activity (Efford, 2004; Borchers and Efford, 2008; Royle and Young, 2008). In what
 3068 follows, we define \mathbf{s}_i , a two-dimensional spatial coordinate, to be the home range
 3069 or activity center of individual i . Then, the SCR model postulates that encounter
 3070 probability, p_{ij} , is a decreasing function of distance between \mathbf{s}_i and the location of
 3071 trap j , \mathbf{x}_j (also a two-dimensional spatial coordinate). A standard model for mod-
 3072 eling binomial counts is the logistic regression, where we model the dependence of
 3073 p_{ij} on distance according to:

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

3074 where, here, $\|\mathbf{x}_j - \mathbf{s}_i\|$ is the distance between \mathbf{s}_i and \mathbf{x}_j . We sometimes write
 3075 $\|\mathbf{x}_j - \mathbf{s}_i\| = \text{dist}(\mathbf{x}_j, \mathbf{s}_i) = d_{ij}$. Alternatively, a popular model is

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

3076 which is similar to the “half-normal” model in distance sampling, except with an
 3077 intercept $p_0 \leq 1$ which can be estimated in SCR studies. Because it is the kernel of a
 3078 bivariate normal, or Gaussian, probability density function for the random variable
 3079 “individual location” we will refer to it as the “(bivariate) normal” or “Gaussian”
 3080 model although the distance sampling term “half-normal” is widely used. In the
 3081 context of 2-dimensional space, the model is clearly interpretable as a primitive
 3082 model of movement outcomes or space usage (we discuss this in Sec. 4.4).

3083 There are a large number of standard detection models commonly used (see
 3084 Chapt. 6). All other standard models that relate encounter probability to \mathbf{s} will
 3085 also have a parameter that multiplies distance in some non-linear function. To be
 3086 consistent with parameter naming across models, we will sometimes parameterize
 3087 any encounter probability model so that the coefficient on distance (or distance
 3088 squared) is α_1 . So, for the Gaussian model, $\alpha_1 = 1/(2\sigma^2)$. A characteristic of
 3089 the common parametric forms is they are monotone decreasing with distance, but
 3090 vary in their characteristic behavior as they approach distance = 0. We show the
 3091 standard Gaussian, Gaussian hazard, negative exponential and logistic models in
 3092 Fig. 4.1. The negative exponential model has $p_{ij} = p_0 \exp(-\alpha_1 \|\mathbf{x}_j - \mathbf{s}_i\|)$ and
 3093 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
 3094 Gaussian kernel. Whatever model we choose for encounter probability, we should
 3095 always keep in mind that the activity center for individual i , \mathbf{s}_i , is an unobserved
 3096 random variable. To be precise about this in the model, we should express the
 3097 observation model as

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

3098 but sometimes, for notational simplicity, we abbreviate this by omitting some of
 3099 the arguments to p .

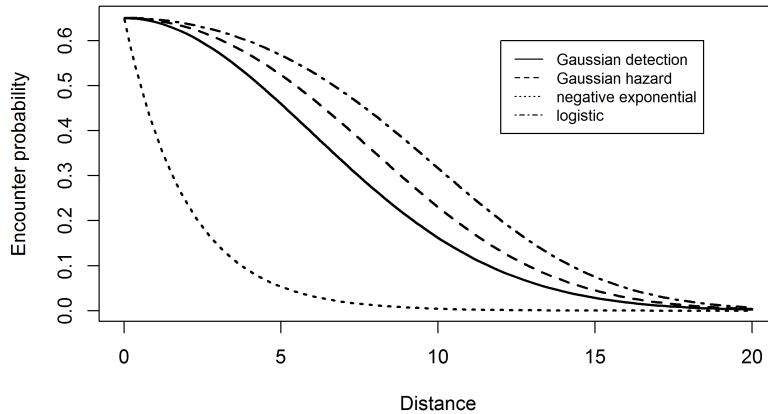


Figure 4.1. Some common encounter probability models showing the characteristic monotone decrease of encounter probability with distance between activity center and trap location.

3100 4.2.1 Definition of home range center

3101 We define an individual's home range as *the area used by an organism during some*
 3102 *time period* which has a clear meaning for most species regardless of their biology.
 3103 We therefore define the home range center (or activity center) to be the center of
 3104 the space that individual was occupying (or using) during the period in which traps
 3105 were active. Thinking about it in that way, it could even be observable (almost) as
 3106 the centroid of a very large number of radio fixes over the course of a survey period
 3107 or a season. Thus, this practical version of a home range center in terms of space
 3108 usage is a well-defined construct regardless of whether one thinks the home range
 3109 itself is a meaningful concept. We use the terms home range center and activity
 3110 center interchangeably, and we recognize that this is a transient thing which applies
 3111 only to a well-defined period of study.

3112 4.2.2 Distance as a latent variable

3113 If we knew precisely every s_i in the population (and population size N), then the
 3114 model specified by Eqs. 4.2.1 and 4.2.2 would be just an ordinary logistic regression-
 3115 type of a model (with covariate d_{ij}) which we learned how to fit using **WinBUGS**
 3116 previously (Chapt. ??). However, the activity centers are unobservable even in the
 3117 best possible circumstances. In that case, d_{ij} is an unobserved variable, analogous

3118 to the situation in classical random effects models. We need to therefore extend the
 3119 model to accommodate these random variables with an additional model compo-
 3120 nent – the random effects distribution. The customary assumption is the so-called
 3121 “uniformity assumption,” which is to assume that the \mathbf{s}_i are uniformly distributed
 3122 over space (the obvious next question: “which space?” is addressed below). This
 3123 uniformity assumption amounts to a uniform prior distribution on \mathbf{s}_i , i.e., the pdf
 3124 of \mathbf{s}_i is constant, which we may express

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

3125 As it turns out, this assumption is usually not precise enough to fit SCR models
 3126 in practice for reasons we discuss shortly. We will give another way to represent
 3127 this prior distribution that is more concrete, but depends on specifying the “state-
 3128 space” of the random variable \mathbf{s}_i . The term state-space is a technical way of saying
 3129 “the space of all possible outcomes” of the random variable.

4.3 THE BINOMIAL POINT PROCESS MODEL

3130 In the SCR model, the individual activity centers are unobserved and thus we treat
 3131 them as random effects. Specifically, the collection of individual activity centers
 3132 $\mathbf{s}_1, \dots, \mathbf{s}_N$ represents a realization of a *binomial point process* (Illian et al., 2008, p.
 3133 61). The binomial point process (BPP) is analogous to a Poisson point process in
 3134 the sense that it represents a “random scatter” of points in space – except that the
 3135 total number of points is *fixed*, whereas, in a Poisson point process, it is random
 3136 (having a Poisson distribution). As an example, we show in Fig. 4.2 locations of
 3137 20 individual activity centers (black dots) in relation to a grid of 25 traps. For
 3138 a Poisson point process the number of such points in the prescribed state-space
 3139 would be random whereas often we will simulate fixed numbers of points, e.g., for
 3140 evaluating the performance of procedures, e.g., how well does our estimator perform
 3141 when $N = 50$?

3142 It is natural to consider a binomial point process in the context of capture-
 3143 recapture models because it preserves N in the model and thus preserves the linkage
 3144 directly with closed population models. In fact, under the binomial point process
 3145 model, model M_0 and other closed models are simple limiting cases of SCR models,
 3146 i.e., they arise as the coefficient on distance (α_1 above) tends to 0.

3147 While we often will express SCR models “conditional-on- N ”, it will sometimes
 3148 be convenient to impose specific prior distributions on N . By assuming N has a
 3149 binomial distribution, we can make use of data augmentation, our preferred tool, for
 3150 Bayesian analysis of the models as in Chapt. 3, thus yielding a methodologically
 3151 coherent approach to analyzing the different classes of models. We might also
 3152 assume that N has a Poisson distribution in some cases (see Chapt. 13). Of
 3153 course, the two assumptions are closely related in the usual limiting sense.

3154 One consequence of having fixed N in the BPP model is that the model is
 3155 not strictly a model of “complete spatial randomness”. This is because, if one

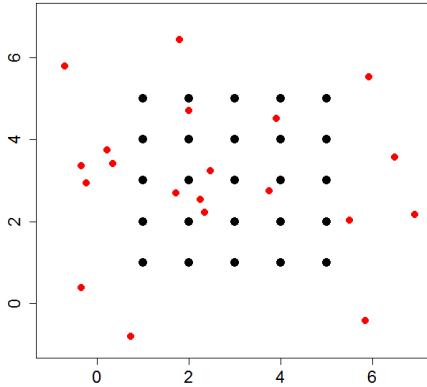


Figure 4.2. Realization (small dots) of a binomial point process with $N = 20$. The large dots represent trap locations.

3156 forms counts $n(A_1), \dots, n(A_k)$ in any set of disjoint regions of the state-space, say
 3157 A_1, \dots, A_k , then these counts are *not* independent. In fact, they have a multinomial
 3158 distribution (see Illian et al., 2008, p. 61). Thus, the BPP model introduces a slight
 3159 bit of dependence in the distribution of points. However, in most situations this will
 3160 have no practical effect on any inference or analysis and, as a practical matter, we
 3161 will usually regard the BPP model as one of spatial independence among individual
 3162 activity centers because each activity center is distributed independently of each
 3163 other activity center. Despite this independence we see in Fig. 4.2 that *realiza-*
 3164 *tions* of randomly distributed points will typically exhibit distinct non-uniformity.
 3165 Thus, independent, uniformly distributed points will almost never appear regu-
 3166 larly, uniformly or systematically distributed. For this reason, the basic binomial
 3167 (or Poisson) point process models are enormously useful in practical settings since
 3168 they allow for a range of distribution patterns without violating the assumption of
 3169 spatial randomness. More relevant for SCR models is that we actually have a little
 3170 bit of data for some individuals and thus the resulting posterior point pattern can
 3171 deviate strongly from uniformity, a point we come back to repeatedly in this book.
 3172 The uniformity hypothesis is only a *prior* distribution which is directly affected by
 3173 the quantity and quality of the observed data, to produce a posterior distribution
 3174 which may appear distinctly non-uniform. In addition, we can build more flexible
 3175 models for the point process, which we take up in Chapt. 10.

3176 **4.3.1 The state-space of the point process**

3177 Shortly we will focus on Bayesian analysis of model SCR0 with N known so that we
 3178 can gain some basic experience with important elements of the model, and its anal-
 3179 ysis. To do this, we note that the individual activity centers $\mathbf{s}_i, \dots, \mathbf{s}_N$ are unknown
 3180 quantities and we will need to be able to simulate each \mathbf{s}_i in the population from
 3181 the posterior distribution. In order to simulate the \mathbf{s}_i , it is necessary to describe
 3182 precisely the region over which they are distributed. This is the quantity referred
 3183 to above as the state-space, which is sometimes called the *observation window* in
 3184 the point process literature. We denote the state-space henceforth (throughout this
 3185 book) by \mathcal{S} , which is a region or a set of points comprising the potential values (the
 3186 support) of the random variable \mathbf{s} . Thus, an equivalent explicit statement of the
 3187 “uniformity assumption” is

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

3188 where \mathcal{S} is a precisely defined region. e.g., in Fig. 4.2, \mathcal{S} is the square defined
 3189 by $[-1, 7] \times [-1, 7]$. Thus each of the $N = 20$ points were generated by randomly
 3190 selecting each coordinate on the line $[-1, 7]$. When points are distributed uniformly
 3191 over some region, the point process is usually called a *homogeneous point process*.

3192 **Prescribing the state-space**

3193 Evidently, to define the model, we need to define the state-space, \mathcal{S} . How can we
 3194 possibly do this objectively? Prescribing any particular \mathcal{S} seems like the equivalent
 3195 of specifying a “buffer” which we have criticized as being ad hoc. How is it, then,
 3196 that the choice of a state-space is *not* ad hoc? As we observed in Chapt. 3, it is
 3197 true that N increases with \mathcal{S} , but only at the same rate as the area of \mathcal{S} increases
 3198 under the prior assumption of constant density. As a result, we say that density is
 3199 invariant to \mathcal{S} as long as \mathcal{S} is sufficiently large. Thus, while choice of \mathcal{S} is (or can
 3200 be) essentially arbitrary, once \mathcal{S} is chosen, it defines the population being exposed
 3201 to sampling, which scales appropriately with the size of the state-space.

3202 For our simulated system developed previously in this chapter, we defined the
 3203 state-space to be a square within which our trap array was centered. For many
 3204 practical situations this might be an acceptable approach to defining the state-
 3205 space, i.e., just a rectangle around the trap array. Although defining the state-space
 3206 to be a regular polygon has computational advantages (e.g., we can implement this
 3207 more efficiently in **BUGS** and cannot for irregular polygons), a regular polygon
 3208 induces an apparent problem of admitting into the state-space regions that are
 3209 distinctly non-habitat (e.g., oceans, large lakes, ice fields, etc.). It is difficult to
 3210 describe complex regions in mathematical terms that can be used in **BUGS**. As
 3211 an alternative, we can provide a representation of the state-space as a discrete set
 3212 of points which the **R** package **secr** (Efford, 2011) permits (**secr** uses the term
 3213 “mask” for what we call the state-space). Defining the state-space by a discrete set
 3214 of points is handy because it allows specific points to be deleted or not, depending
 3215 on whether they represent available or suitable habitat (see Sec. 4.10). We can

3216 also define the state-space as an arbitrary collection of polygons stored as a GIS
 3217 shapefile which can be analyzed easily by MCMC in **R** (see Sec. 16.5), but not so
 3218 easily in the **BUGS** engines. In Sec. 4.10, we provide an analysis of the wolverine
 3219 camera trapping data, in which we define the state-space to be a regular continuous
 3220 polygon (a rectangle).

3221 **Invariance to the state-space**

3222 We will assert for all models we consider in this book that density is invariant to
 3223 the size and extent of \mathcal{S} , if \mathcal{S} is sufficiently large, and as long as our model relating
 3224 p_{ij} to \mathbf{s}_i is a decreasing function of distance. We can prove this easily by drawing
 3225 an analogy with a 1-d case involving distance sampling. Let y_j be the number of
 3226 individuals captured in some interval $[d_{j-1}, d_j]$, and define $d_J = B$ for some large
 3227 value of B . The observations from a survey are y_1, \dots, y_J and the likelihood is a
 3228 multinomial likelihood, so the log-likelihood is of the form

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

3229 where π_j is the probability of detecting an individual in distance class j , which
 3230 depends on parameters of the detection function (the manner of which is not rel-
 3231 evant for the present discussion). Choosing B sufficiently large guarantees that
 3232 $\mathbb{E}(y_J) = 0$ and therefore the observed frequency in the “last cell” contributes noth-
 3233 ing to the likelihood, in regular situations in which the detection function decays
 3234 monotonically with distance and prior density is constant. We can think of B as
 3235 being related to the state-space in an SCR model, as the width of a rectangular
 3236 state-space with area $B \times L$, L being the length of the transect. Thus, if we choose
 3237 B large enough, then we ensure that the expected trap-frequencies beyond B will
 3238 be 0, and thus contribute nothing to the likelihood.

3239 Sometimes our estimate of density can be affected by choosing \mathcal{S} too small.
 3240 However, this might be sensible if \mathcal{S} is naturally well-defined. As we discussed in
 3241 Chapt. 1, \mathcal{S} is part of the model, and thus it is sensible that estimates of density
 3242 might be sensitive to its definition in problems where it is natural to restrict \mathcal{S} . One
 3243 could imagine, however, in specific cases, e.g., a small population with well-defined
 3244 habitat preferences, that a problem could arise because changing the state-space
 3245 based on differing opinions, and GIS layers, might have substantial affects on the
 3246 density estimate. But this is a real biological problem, and a natural consequence
 3247 of the spatial formalization of capture-recapture models – a feature, not a bug
 3248 or some statistical artifact – and it should be resolved with better information,
 3249 research, and thinking. For situations where there is not a natural choice of \mathcal{S} ,
 3250 we should default to choosing \mathcal{S} to be very large in order to achieve invariance or,
 3251 otherwise, evaluate sensitivity of density estimates by trying a couple of different
 3252 choices of \mathcal{S} . This is a standard “sensitivity to prior” argument that Bayesians
 3253 always have to be conscious of. We demonstrate this in our analysis of Sec. 4.9

3254 below. As an additional practical consideration, we note that the area of the state-
 3255 space \mathcal{S} affects data augmentation. If you increase the size of \mathcal{S} , then there are
 3256 more individuals to account for and therefore the size of the augmented data set
 3257 M must increase. This has computational implications.

3258 **4.3.2 Connection to model M_h and distance sampling**

3259 SCR models are closely related to “model M_h ” and also distance sampling. In
 3260 SCR models, heterogeneity in encounter probability is induced by both the effect
 3261 of distance in the model for detection probability and also from specification of
 3262 the state-space. Hence, the state-space is an explicit element of the model. To
 3263 understand this, suppose activity centers have the uniform distribution:

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

3264 and encounter probability is a function of \mathbf{s} , denoted by $p(\mathbf{s}) = p(y = 1|\mathbf{s})$. For
 3265 example, under Eq. 4.2.2 we have that

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

3266 and we can work out, either analytically or empirically, what is the implied distri-
 3267 bution of p for a population of individuals. Fig. 4.3 shows a histogram of p for a
 3268 hypothetical population of 100000 individuals on a state-space enclosing our 5×5
 3269 trap array above, under the logistic model for distance given by Eq. 4.2.2 with
 3270 buffers of 0.2, 0.5 and 1.0. We see the mass shifts to the left as the buffer increases,
 3271 implying more individuals with lower encounter probabilities, as their home range
 3272 centers increase in distance from the trap array.

3273 Another way to understand this is by representing \mathcal{S} as a set of discrete points
 3274 on a grid. In the coarsest possible case where \mathcal{S} is a single arbitrary point, then
 3275 every individual has exactly the same p . As we increase the number of points in \mathcal{S} ,
 3276 more distinct values of p are possible. Indeed, when \mathcal{S} is characterized by discrete
 3277 points, then SCR models are precisely a type of finite-mixture model (Norris and
 3278 Pollock, 1996; Pledger, 2004), except, in the case of SCR models, we have some
 3279 information about which group an individual belongs to (i.e., where their activity
 3280 center is), as a result of which traps it is captured in.

3281 It is also worth re-emphasizing that the basic SCR encounter model is a binomial
 3282 encounter model in which distance is a covariate. As such, it is strikingly similar
 3283 to classical distance sampling models (Buckland et al., 2001). Both have distance
 3284 as a covariate but, in classical distance sampling problems, the focus is on the
 3285 distance between the observer and the animal at an instant in time, not the distance
 3286 between a trap and an animal’s home range center. As a practical matter, in
 3287 distance sampling, “distance” is *observed* for those individuals that appear in the
 3288 sample. Conversely, in SCR problems, it is only imperfectly observed (we have
 3289 partial information in the form of trap observations). Clearly, it is preferable to

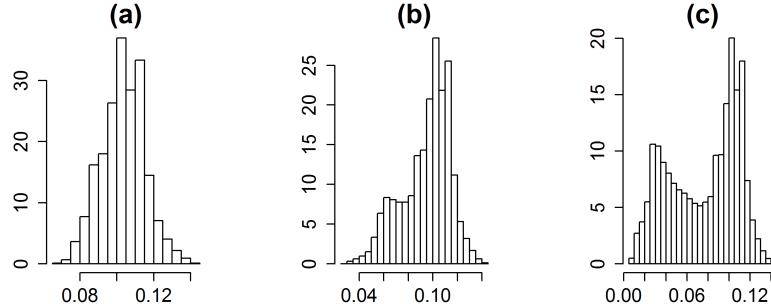


Figure 4.3. Implied distribution of p_i for a population of individuals as a function of the size of the state-space buffer around the trap array. The state-space buffer is 0.2, 0.5 and 1.0 for panels (a), (b), (c), respectively. In each case, the trap array is fixed and centered within a square state-space.

3290 observe distance if possible, but distance sampling requires field methods that are
 3291 not practical in many situations, e.g. when studying carnivores such as bears or
 3292 large cats. Furthermore, SCR models allow us to relax many of the assumptions
 3293 made in classical distance sampling, such as perfect detection at distance zero, and
 3294 SCR models allow for estimates of quantities other than density, such as home range
 3295 size, and space usage (see Chaps. 11 and 12).

4.4 THE IMPLIED MODEL OF SPACE USAGE

3296 We developed the basic SCR model in terms of a latent variable, s , the home range
 3297 center or activity center. Surely the encounter probability model, which relates
 3298 encounter of individuals in specific traps to s must somehow imply a certain model
 3299 for home range geometry and size. Here we explore the nature of that relationship
 3300 and we argue that any given detection model implies a model of space usage – i.e.,
 3301 the amount and extent of area used some prescribed percentage of the time. So we
 3302 might say, for example, 95% of animal movements are within some distance from an
 3303 individual's activity center. While we have used the term “home range” or similar,
 3304 what we really mean to imply is something that would be more clearly identified as
 3305 resource selection or space usage (the latter term meaning resource selection, when
 3306 the resource is only homogeneous space).

3307 Intuitively, the detection function of SCR models is related to space usage by
 3308 individuals. Indeed, it is natural to interpret the detection model as the composite
 3309 of two processes: movement of an individual about its home range i.e., how it uses
 3310 space within its home range (“space usage”), and detection *conditional on use* in
 3311 the vicinity of a trapping device. It is natural to decompose encounter probability

3312 according to:

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

3313 In practice it might make sense to think about the first component,
 3314 i.e., $\Pr(\text{encounter}|\text{usage of } \mathbf{x}, \mathbf{s})$ as being a constant (e.g., if traps are located within
 3315 arbitrarily small grid cells) and then, in that case, the encounter probability model
 3316 is directly proportional to this model for individual movements about their home
 3317 range center determining the use frequency of each \mathbf{x} . This is a sensible heuristic
 3318 model for what ecologists would call a central place forager although, as we have
 3319 stated previously, it may be meaningful as a description of transient space usage as
 3320 well (that is, the space usage during the period of sampling).

3321 To motivate a specific model for space usage, imagine the area we are interested
 3322 in consists of some large number of small pixels (i.e. we're looking at a discrete
 3323 representation of space), and that we have some kind of perfect observation device
 3324 (e.g., continuous telemetry) so that we observe every time an individual moves into
 3325 a pixel. After a long period of time, we observe an enormous sample size of \mathbf{x}
 3326 values. We tally those up into each pixel, producing the frequency $m(\mathbf{x}, \mathbf{s})$, which
 3327 is something like the "true" usage of pixel \mathbf{x} by individual with activity center \mathbf{s} . So,
 3328 then, the usage model should be regarded as a probability mass function for these
 3329 counts and, naturally, we regard the counts $m(\mathbf{x}, \mathbf{s})$ as a multinomial observation
 3330 with probabilities $\pi(\mathbf{x}|\mathbf{s})$, and prescribe a suitable model for $\pi(\mathbf{x}|\mathbf{s})$ that describes
 3331 how use events should accumulate in space. A natural null model for $\pi(\mathbf{x}|\mathbf{s})$ has a
 3332 decreasing probability of use as \mathbf{x} gets far away from \mathbf{s} ; i.e., animals spend more
 3333 time close to their activity centers than far away. We can regard points used by
 3334 the individual with activity center \mathbf{s} as the realization of a point process with
 3335 conditional intensity:

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

3336 where $k(\mathbf{x}, \mathbf{s})$ is any positive function. In continuous space, the equivalent repre-
 3337 sentation would be:

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

3338 Clearly the space used by an individual will be proportional to whatever kernel,
 3339 $k(\mathbf{x}, \mathbf{s})$, we plug-in here. If we use a negative exponential function, then this pro-
 3340 duces a standard resource selection function (RSF) model (e.g., Manly et al., 2002,
 3341 Chapt. 8). But, here we use a Gaussian kernel, i.e.,

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

3342 so that contours of the probability of space usage resemble a bivariate normal or
 3343 Gaussian probability distribution function.

3344 To apply this model of space-usage to SCR problems we allow for imperfect
 3345 detection by introducing a non-uniform "thinning rate" of the true counts $m(\mathbf{x}, \mathbf{s})$.

3346 This yields, precisely, our Gaussian encounter probability model where the thinning
 3347 rate is our baseline encounter probability p_0 for each pixel where we place a trap,
 3348 and $p = 0$ in each pixel where we don't place a trap.

3349 The main take-away point here is that underlying most SCR models is some kind
 3350 of model of space-usage, implied by the specific choice of $k(\mathbf{x}, \mathbf{s})$. Whether or not
 3351 we have perfect sampling devices, the function we use in the encounter probability
 3352 model equates to some conditional distribution of points, a utilization distribution,
 3353 as in Eq. 4.4.1, from which we can compute effective home range area, i.e., the area
 3354 that contains some percent of the mass of a probability distribution proportional
 3355 to $k(\mathbf{x}, \mathbf{s})$; e.g., 95% of all space used by an individual with activity center \mathbf{s} .

3356 4.4.1 Bivariate normal case

3357 One encounter model that allows direct analytic computation of home range area
 3358 is the Gaussian encounter probability model

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

3359 For this model, encounter probability is proportional to the kernel of a bivariate
 3360 normal (Gaussian) pdf and so the natural interpretation is that in which movement
 3361 outcomes (or successive locations of an individual) are draws from a bivariate nor-
 3362 mal distribution with standard deviation σ . We say that use of this model implies
 3363 a bivariate normal model of space usage. Under this model we can compute pre-
 3364 cisely the effective home range area. In particular, if use outcomes are bivariate
 3365 normal, then $\|\mathbf{x} - \mathbf{s}\|^2$ has a chi-square distribution with 2 d.f. and the quantity
 3366 $B(\alpha)$ that encloses $(1 - \alpha)\%$ of all realized distances i.e., $\Pr(d \leq B(\alpha)) = 1 - \alpha$,
 3367 is $B(\alpha) = \sigma * \sqrt{q(\alpha, 2)}$ where $q(\alpha, 2)$ is the 0.05 chi-square critical value on 2 df.
 3368 For example, to compute $q(.05, 2)$ in R we execute the command `qchisq(.95, 2)`
 3369 which is $q(2, \alpha) = 5.99$. Then, for $\sigma = 1$, $B(\alpha) = 1 * \sqrt{5.99} = 2.447$. Therefore
 3370 95% of the points used will be within 2.447 (standard deviation) units of the home
 3371 range center. So, in practice, we can estimate σ by fitting the bivariate normal
 3372 encounter probability model to some SCR data, and then use the estimated σ to
 3373 compute the “95% radius”, say $r_{.95} = \sigma\sqrt{5.99}$, and convert this to the 95% use area
 3374 – the area around \mathbf{s} which contains 95% of the movement outcomes – according to
 3375 $A_{.95} = \pi r_{.95}^2$.

3376 An alternative bivariate normal model is the bivariate normal hazard rate model:

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

3378 We use λ_0 here because this parameter, the baseline encounter *rate*, can be > 1 .
 3379 This arises by assuming the latent “use frequency” $m(\mathbf{x}, \mathbf{s})$ is a Poisson random
 3380 variable with intensity $\lambda_0 k(\mathbf{x}, \mathbf{s})$. The model is distinct from our Gaussian encounter
 3381 model $p(\mathbf{x}, \mathbf{s}) = p_0 k(\mathbf{x}, \mathbf{s})$ used previously, although we find that they produce

3382 similar results in terms of estimates of density or 95% use area, as long as baseline
 3383 encounter probability is low. We discuss these two formulations of the bivariate
 3384 normal model further in Chapt. 8.

3385 **4.4.2 Empirical analysis**

3386 For any encounter model we can compute space usage quantiles empirically by
 3387 taking a fine grid of points and either simulating movement outcomes with proba-
 3388 bilities proportional to $p(\mathbf{x}, \mathbf{s})$ and accumulating area around \mathbf{s} , or else we can do
 3389 this precisely by varying $B(\alpha)$ to find that value within which 95% of all move-
 3390 ments are concentrated, i.e., the set of all \mathbf{x} such that $\|\mathbf{x} - \mathbf{s}\| \leq B(q)$. Under any
 3391 detection model, movement outcomes will occur in proportion to $p(\mathbf{x}, \mathbf{s})$, as long as
 3392 the probability of encounter is constant, *conditional on use*, and so we can define
 3393 our space usage distribution according to:

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

3394 Given the probabilities $\pi(\mathbf{x}, \mathbf{s})$ for all \mathbf{x} we can find the value of $B(q)$, for any q ,
 3395 such that

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

3396 (here, we use \ni to mean “such that”). We have a function called `hra` in the
 3397 `scrbook` package that computes the home range area for any encounter model and
 3398 prescribed parameter values. The help file for `hra` has an example of simulating
 3399 some data. The following commands illustrate this calculation for two different
 3400 bivariate normal models of space usage:

```
3401 ##
3402 ## Define encounter probability model as R function
3403 ##
3404 > pGauss2 <- function(parms,Dmat){
3405   a0 <- parms[1]
3406   sigma <- parms[2]
3407   lp <- parms[1] -(1/(2*parms[2]*parms[2]))*Dmat*Dmat
3408   p <- 1-exp(-exp(lp))
3409   p
3410 }
3411
3412 > pGauss1 <- function(parms,Dmat){
3413   a0 <- parms[1]
3414   sigma <- parms[2]
3415   p <- plogis(parms[1])*exp( -(1/(2*parms[2]*parms[2]))*Dmat*Dmat )
3416   p
3417 }
```

```

3418
3419 ## Execute hra with sigma = .3993
3420 ## Execute hra with sigma = .3993
3421 ##
3422 > hra(pGauss1,parms=c(-2,.3993),plot=FALSE,xlim=c(0,6),ylim=c(0,6),
3423   ng=500,tol=.0005)
3424
3425 [1] 0.9784019
3426 radius to achieve 95% of area: 0.9784019
3427 home range area: 3.007353
3428 [1] 3.007353
3429
3430
3431 ## Analytic solution:
3432 ## true sigma that produces area of 3
3433 > sqrt(3/pi)/sqrt(5.99)
3434 [1] 0.3992751

```

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

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

```

3450 > hra(pGauss1,parms=c(-2,.3993),plot=FALSE,xlim,ylim,ng=500,
3451   target.area=3.00735,tol=.0005)
3452
3453 Value of parm[2] to achieve 95% home range area of 3.00735: 0.3993674

```

3454 4.4.3 Relevance of understanding space usage

3455 One important reason that we need to be able to deduce “home range area” from
3456 a detection model is so that we can compare different models with respect to a
3457 common biological currency. Many encounter probability models have some “scale
3458 parameter”, which we might call σ no matter the model, but this relates to 95%
3459 area in a different manner under each model. Therefore, we want to be able to

3460 convert different models to the same currency. Another reason to understand the
3461 relationship between models of encounter probability and space usage is that it
3462 opens the door to combining traditional resource selection data from telemetry
3463 with spatial capture-recapture data. In Chapt. 12 we consider this problem, for
3464 the case in which a sample of individuals produces encounter history data suitable
3465 for SCR models and, in addition, we have telemetry relocations on a sample of
3466 individuals. This is achieved by regarding the two sources of data as resulting from
3467 the same underlying process of space usage but telemetry data produce “perfect”
3468 observations, like always-on camera traps blanketing a landscape. We use this idea
3469 to model the effect of a measured covariate at each pixel, say $C(\mathbf{x})$, on home range
3470 size and geometry and, hence, the probability of encounter in traps.

3471 **4.4.4 Contamination due to behavioral response**

3472 Interpretation of encounter probability models as models of animal home range and
3473 space usage can be complicated by a number of factors, including whether traps are
3474 baited or not. In the case of baited traps, this might lead to a behavioral response
3475 (Sec. 6.2.3) which could affect animal space usage. For example, if traps attract
3476 animals from a long distance, it could make typical home ranges appear larger than
3477 normal. More likely, in our view, it wouldn’t change the typical size of a range but
3478 would change how individuals use their range e.g., by moving from baited trap to
3479 baited trap, so that observed movement distances of individuals are typically larger
3480 than normal.

3481 In other cases, the reliance on Euclidean distance in models for encounter prob-
3482 ability might be unrealistic, and can lead to biased estimates of density (Royle
3483 et al., 2012a). For example, animals might concentrate their movements along
3484 trails, roads, or other landscape features. In this case, models that accommodate
3485 other distance metrics can be considered. We present models based on least-cost
3486 path in Chapt. 11.

4.5 SIMULATING SCR DATA

3487 It is always useful to simulate data because it allows you to understand the system
3488 that you’re modeling and also calibrate your understanding with specific values of
3489 the model parameters. That is, you can simulate data using different parameter
3490 values until you obtain data that “look right” based on your knowledge of the spe-
3491 cific situation that you’re interested in. Here we provide a simple script to illustrate
3492 how to simulate spatial encounter history data. In this exercise we simulate data
3493 for 100 individuals and a 25 trap array laid out in a 5×5 grid of unit spacing.
3494 The specific encounter model is the Gaussian model given above and we used this
3495 code to simulate data used in subsequent analyses. The 100 activity centers were
3496 simulated on a state-space defined by a 8×8 square within which the trap array

3497 was centered (thus the trap array is buffered by 2 units). Therefore, the density of
 3498 individuals in this system is fixed at 100/64.

```

3499 > set.seed(2013)
3500 # Create 5 x 5 grid of trap locations with unit spacing
3501 > traplocs <- cbind(sort(rep(1:5,5)),rep(1:5,5))
3502 > ntraps <- nrow(traplocs)
3503 # Compute distance matrix:
3504 > Dmat <- e2dist(traplocs,traplocs)
3505
3506
3507 # Define state-space of point process. (i.e., where animals live).
3508 # "buffer" just adds a fixed buffer to the outer extent of the traps.
3509 #
3510 > buffer <- 2
3511 > xlim <- c(min(traplocs[,1] - buffer),max(traplocs[,1] + buffer))
3512 > ylim <- c(min(traplocs[,2] - buffer),max(traplocs[,2] + buffer))
3513
3514 > N <- 100    # population size
3515 > K <- 20    # number nights of effort
3516
3517 > sx <- runif(N,xlim[1],xlim[2])    # simulate activity centers
3518 > sy <- runif(N,ylim[1],ylim[2])
3519 > S <- cbind(sx,sy)
3520 # Compute distance matrix:
3521 > D <- e2dist(S,traplocs) # distance of each individual from each trap
3522
3523 > alpha0 <- -2.5      # define parameters of encounter probability
3524 > sigma <- 0.5        # scale parameter of half-normal
3525 > alpha1 <- 1/(2*sigma*sigma) # convert to coefficient on distance
3526
3527 # Compute Probability of encounter:
3528 #
3529 > probcap <- plogis(-2.5)*exp( - alpha1*D*D)
3530
3531 # Generate the encounters of every individual in every trap
3532 > Y <- matrix(NA,nrow=N,ncol=ntraps)
3533 > for(i in 1:nrow(Y)){
3534   Y[i,] <- rbinom(ntraps,K,probcap[i,])
3535 }
```

3536 We remind the reader that, in presenting **R** or other code snippets throughout
 3537 the book, we will deviate from our standard variable expressions for some quantities.
 3538 In particular, we sometimes substitute words for integer variable designations: `nind`
 3539 (for n), `ntraps` (for J), and `nocc` (for K). In our opinion this leaves less to be
 3540 inferred by the reader in trying to understand code snippets.

3541 Subsequently we will generate data using this code packaged in an **R** function
 3542 called **simSCR0** in the package **scrbook** which takes a number of arguments includ-
 3543 ing **discard0** which, if TRUE, will return only the encounter histories for captured in-
 3544 dividuals. A second argument is **array3d** which, if TRUE, returns the 3-dimensional
 3545 encounter history array instead of the aggregated **nind** \times **ntraps** encounter frequen-
 3546 cies (see below). Finally we provide a random number seed, **rnd** = 2013 to ensure
 3547 repeatability of the analysis here. We obtain a data set as above using the following
 3548 command:

```
3549 > data <- simSCR0(discard0=TRUE, array3d=FALSE, rnd=2013)
```

3550 The **R** object **data** is a list, so let's take a look at what's in the list and then harvest
 3551 some of its elements for further analysis below.

```
3552 > names(data)
3553 [1] "Y"      "traplocs" "xlim"     "ylim"     "N"       "alpha0"   "beta"
3554 [8] "sigma"   "K"
3555
3556 ## Grab encounter histories from simulated data list
3557 > Y <- data$Y
3558 ## Grab the trap locations
3559 > traplocs <- data$traplocs
```

3560 4.5.1 Formatting and manipulating real data sets

3561 Conventional capture-recapture data are easily stored and manipulated as a 2-
 3562 dimensional array, an **nind** \times **K** (individuals by sample occasions) matrix, which is
 3563 maximally informative for any conventional capture-recapture model, but not for
 3564 spatial capture-recapture models. For SCR models we must preserve the spatial
 3565 information in the encounter history information. We will routinely analyze data
 3566 from 3 standard formats:

- 3567 (1) The basic 2-dimensional data format, which is an **nind** \times **ntraps** encounter
 3568 frequency matrix such as that simulated previously. These are the total number
 3569 of encounters in each trap, summed over the **K** sample occasions.
- 3570 (2) The maximally informative 3-dimensional array, for which we establish here
 3571 the convention that it has dimensions **nind** \times **ntraps** \times **K**.
- 3572 (3) We use a compact format – the “encounter data file” – which we describe below
 3573 in Sec. 4.9.

3574 To simulate data in the most informative format - the “3-d array” - we can use the
 3575 **R** commands given previously but replace the last 4 lines with the following:

```
3576 > Y <- array(NA,dim=c(N,ntraps,K))
3577
3578 > for(i in 1:nrow(Y)){
3579   for(j in 1:ntraps){
```

```

3580     Y[i,j,1:K] <- rbinom(K,1,probcap[i,j])
3581   }
3582 }

```

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

```

3590 # Sum over the "replicates" dimension (3rd margin of the array)
3591 > Y2d <- apply(Y,c(1,2),sum)
3592
3593 # Compute how many times each individual was captured
3594 > ncaps <- apply(Y2d,1,sum)
3595
3596 # Keep those individuals that were captured
3597 > Y <- Y[ncaps>0,,]

```

4.6 FITTING MODEL SCR0 IN BUGS

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

```

3606 > data <- simSCR0(discard0=FALSE,rnd=2013)
3607 > y <- data$Y
3608 > traplocs <- data$traplocs
3609
3610 # In this case nind=N because we're doing the known-N problem
3611 #
3612 > nind <- nrow(y)
3613 > X <- data$traplocs
3614 > J <- nrow(X) # number of traps
3615 > K <- data$K
3616 > xlim <- data$xlim
3617 > ylim <- data$ylim

```

3618 Note that we specify `discard0 = FALSE` so that we have a “complete” data set,
 3619 i.e., one with the all-zero encounter histories corresponding to uncaptured individ-
 3620 uals. Now, within an **R** session, we can create the **BUGS** model file and fit the
 3621 model using the following commands.

```
3622 cat("
3623   model{
3624     alpha0 ~ dnorm(0,.1)
3625     logit(p0) <- alpha0
3626     alpha1 ~ dnorm(0,.1)
3627     sigma <- sqrt(1/(2*alpha1))
3628     for(i in 1:N){    # note N here -- N is KNOWN in this example
3629       s[i,1] ~ dunif(xlim[1],xlim[2])
3630       s[i,2] ~ dunif(ylim[1],ylim[2])
3631       for(j in 1:J){
3632         d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
3633         y[i,j] ~ dbin(p[i,j],K)
3634         p[i,j] <- p0*exp(- alpha1*d[i,j]*d[i,j])
3635       }
3636     }
3637   }
3638 ",file = "SCR0a.txt")
```

3639 This model describes the Gaussian encounter probability model, but it would be
 3640 trivial to modify that to various others including the logistic described above. One
 3641 consequence of using the half-normal is that we have to constrain the encounter
 3642 probability to be in $[0, 1]$ which we do here by defining `alpha0` to be the logit of the
 3643 intercept parameter `p0`. Note that the distance covariate is computed within the
 3644 **BUGS** model specification given the matrix of trap locations, `X`, which is provided
 3645 to **WinBUGS** as data.

3646 Next we do a number of organizational activities including bundling the data for
 3647 **WinBUGS**, defining some initial values, the parameters to monitor and some basic
 3648 MCMC settings. We choose initial values for the activity centers `s` by generating
 3649 uniform random numbers in the state-space but, for the observed individuals, we
 3650 replace those values by each individual’s mean trap coordinate for all encounters

```
3651 ### Starting values for activity centers, s
3652 > sst <- cbind(runif(nind,xlim[1],xlim[2]),runif(nind,ylim[1],ylim[2]))
3653 > for(i in 1:nind){
3654   if(sum(y[i,])==0) next
3655   sst[i,1] <- mean( X[y[i,>0,1] )
3656   sst[i,2] <- mean( X[y[i,>0,2] )
3657 }
3658 > data <- list (y=y, X=X, K=K, N=nind, J=J, xlim=xlim, ylim=ylim)
3659 > inits <- function(){
3660   list (alpha0=rnorm(1,-4,.4), alpha1=runif(1,1,2), s=sst)
```

```

3662   }
3663
3664 > library(R2WinBUGS)
3665 > parameters <- c("alpha0","alpha1","sigma")
3666 > out <- bugs (data, inits, parameters, "SCR0a.txt", n.thin=1, n.chains=3,
3667           n.burnin=1000,n.iter=2000,debug=TRUE,working.dir=getwd())

```

3668 There is little to say about the preceding operations other than to suggest that
 3669 you might explore the output and investigate additional analyses by running the
 3670 `simSCR0` script provided in the **R** package `scrbook`.

3671 For purposes here, we ran 1000 burn-in and 1000 post-burn-in iterations, and
 3672 3 chains, to obtain 3000 posterior samples. Because we know N for this particular
 3673 data set we only have 2 parameters of the detection model to summarize (`alpha0`
 3674 and `alpha1`), along with the derived parameter σ , the scale parameter of the Gaus-
 3675 sian kernel, i.e., $\sigma = \sqrt{1/(2\alpha_1)}$. When the object `out` is produced we print a
 3676 summary of the results as follows:

```

3677 > print(out,digits=2)
3678 Inference for Bugs model at "SCR0a.txt", fit using WinBUGS,
3679   3 chains, each with 2000 iterations (first 1000 discarded)
3680   n.sims = 3000 iterations saved
3681     mean    sd   2.5%   25%   50%   75% 97.5% Rhat n.eff
3682 alpha0   -2.50  0.22  -2.95  -2.65  -2.48  -2.34  -2.09 1.01  190
3683 alpha1    2.44  0.42   1.64   2.15   2.44   2.72   3.30 1.00  530
3684 sigma     0.46  0.04   0.39   0.43   0.45   0.48   0.55 1.00  530
3685 deviance 292.80 21.16 255.60 277.50 291.90 306.00 339.30 1.01  380
3686
3687
3688 [...some output deleted...]
3689

```

3690 We know the data were generated with `alpha0 = -2.5` and `alpha1 = 2`. The
 3691 estimates look reasonably close to those data-generating values and we probably feel
 3692 pretty good about the performance of the Bayesian analysis and MCMC algorithm
 3693 that **WinBUGS** cooked-up based on our sample size of 1 data set. It is worth
 3694 noting that the `Rhat` statistics indicate reasonable convergence but, as a practical
 3695 matter, we might choose to run the MCMC algorithm for additional time to bring
 3696 these closer to 1.0 and to increase the effective posterior sample size (`n.eff`). Other
 3697 summary output includes “deviance” and related things including the deviance
 3698 information criterion (DIC). We discuss general issues of convergence and other
 3699 MCMC considerations in Chapt. 16, and DIC and model selection in Chapt. 7.

4.7 UNKNOWN N

3700 In all real applications N is unknown. We handled this important issue in Chapt.
 3701 3 using the method of data augmentation (DA) which we apply here to achieve

3702 a realistic analysis of model SCR0. As with the basic closed population models
 3703 considered previously, we formulate the problem by augmenting our observed data
 3704 set with a number of “all-zero” encounter histories - what we referred to in Chapt. 3
 3705 as potential individuals. If n is the number of observed individuals, then let $M - n$
 3706 be the number of potential individuals in the data set. For the 2-dimensional
 3707 y_{ij} data structure (n individual \times J traps encounter frequencies) we simply add
 3708 additional rows of all-zero observations to that data set. Because such “individuals”
 3709 are unobserved, they therefore necessarily have $y_{ij} = 0$ for all j . A data set, say
 3710 with 4 traps and 6 individuals, augmented with 4 pseudo-individuals therefore might
 3711 look like this:

```
3712      trap1 trap2 trap3 trap4
3713  [1,]    1    0    0    0
3714  [2,]    0    2    0    0
3715  [3,]    0    0    0    1
3716  [4,]    0    1    0    0
3717  [5,]    0    0    1    1
3718  [6,]    1    0    1    0
3719  [7,]    0    0    0    0
3720  [8,]    0    0    0    0
3721  [9,]    0    0    0    0
3722  [10,]   0    0    0    0
```

3723 We typically have more than 4 traps and, if we’re fortunate, many more individuals
 3724 in our data set.

3725 For the augmented data set, we introduce a set of binary latent variables (the
 3726 data augmentation variables), z_i , and the model is extended to describe $\Pr(z_i = 1)$
 3727 which is, in the context of this problem, the probability that an individual in the
 3728 augmented data set is a member of the population of size N that was exposed to
 3729 sampling. In other words, if $z_i = 1$ for one of the all-zero encounter histories, this is
 3730 implied to be a sampling zero whereas observations for which $z_i = 0$ are “structural
 3731 zeros” under the model. Under DA, we also express the binomial observation model
 3732 *conditional on z_i* as follows:

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

3733 where we see that the binomial probability evaluates to 0 if $z_i = 0$ (so y_{ij} is a fixed
 3734 0 in that case) and evaluates to p_{ij} if $z_i = 1$.

3735 How big does the augmented data set have to be? We discussed this issue
 3736 in Chapt. 3 where we noted that the size of the data set is equivalent to the
 3737 upper limit of a uniform prior distribution on N . Practically speaking, it should
 3738 be sufficiently large so that the posterior distribution for N is not truncated. On
 3739 the other hand, if it is too large then unnecessary calculations are being done. An
 3740 approach to choosing M by trial-and-error is indicated. Do a short MCMC run
 3741 and then consider whether you need to increase M . See Chapt. 16 for an example
 3742 of this. Kéry and Schaub (2012, Chapt. 6) provide an assessment of choosing M

3743 in closed population models. The useful thing about DA is that it removes N as
 3744 an explicit parameter of the model. Instead, N is a derived parameter, computed
 3745 by $N = \sum_{i=1}^M z_i$. Similarly, *density*, D , is also a derived parameter computed as
 3746 $D = N/\text{area}(\mathcal{S})$.

3747 4.7.1 Analysis using data augmentation in WinBUGS

3748 We provide a complete **R** script for simulating and organizing a data set, and
 3749 analyzing the data in **WinBUGS**. As before we begin by obtaining a data set
 3750 using our **simSCR0** function and then harvesting the required data objects from the
 3751 resulting data list. Note that we use the **discard0=TRUE** option this time so that
 3752 we get a “real looking” data set with no all-zero encounter histories:

```
3753 ##  

3754 ## Simulate the data and extract the required objects  

3755 ##  

3756 > data <- simSCR0(discard0=TRUE,rnd=2013)  

3757 > y <- data$Y  

3758 > nind <- nrow(y)  

3759 > X <- data$traplocs  

3760 > K <- data$K  

3761 > J <- nrow(X)  

3762 > xlim <- data$xlim  

3763 > ylim <- data$ylim
```

3764 After harvesting the data we augment the data matrix y with $M - n$ all-zero
 3765 encounter histories, and create starting values for the variables z_i and also the
 3766 activity centers s_i of which, for each, we require M values. One thing to take care
 3767 of in using the **BUGS** engines is the starting values for the activity centers. It is
 3768 usually helpful to start the s_i for each observed individual at or near the trap(s) it
 3769 was captured. All of this happens as follows:

```
3770 ## Data augmentation  

3771 > M <- 200  

3772 > y <- rbind(y,matrix(0,nrow=M-nind,ncol=ncol(y)))  

3773 > z <- c(rep(1,nind),rep(0,M-nind))  

3774  

3775 ## Starting values for s  

3776 > sst <- cbind(runif(M,xlim[1],xlim[2]),runif(M,ylim[1],ylim[2]))  

3777 > for(i in 1:nind){  

3778   sst[i,1] <- mean( X[y[i,]>0,1] )  

3779   sst[i,2] <- mean( X[y[i,]>0,2] )  

3780 }
```

3781 Next, we write out the **BUGS** model specification and save it to an external
 3782 file called **SCR0b.txt**. The model specification now includes M encounter histories
 3783 including the augmented potential individuals, the data augmentation parameters
 3784 z_i , and the data augmentation parameter ψ :

```

3785 > cat("
3786 model{
3787   alpha0 ~ dnorm(0,.1)
3788   logit(p0) <- alpha0
3789   alphai ~ dnorm(0,.1)
3790   sigma <- sqrt(1/(2*alphai))
3791   psi ~ dunif(0,1)
3792
3793   for(i in 1:M){
3794     z[i] ~ dbern(psi)
3795     s[i,1] ~ dunif(xlim[1],xlim[2])
3796     s[i,2] ~ dunif(ylim[1],ylim[2])
3797     for(j in 1:J){
3798       d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
3799       y[i,j] ~ dbin(p[i,j],K)
3800       p[i,j] <- z[i]*p0*exp(- alphai*d[i,j]*d[i,j])
3801     }
3802   }
3803   N <- sum(z[])
3804   D <- N/64
3805 }
3806 ",file = "SCR0b.txt")

```

3807 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:

```

3810 > data <- list (y=y, X=X, K=K, M=M, J=J, xlim=xlim, ylim=ylim)
3811 > inits <- function(){
3812   list (alpha0=rnorm(1,-4,.4), alpha1=runif(1,1,2), s=sst, z=z)
3813 }
3814
3815 > library(R2WinBUGS)
3816 > parameters <- c("alpha0","alpha1","sigma","N","D")
3817 > out <- bugs (data, inits, parameters, "SCR0b.txt", n.thin=1,n.chains=3,
3818   n.burnin=1000,n.iter=2000,debug=TRUE,working.dir=getwd())

```

3819 Note the differences in this new **WinBUGS** model with that appearing in the
3820 known- N version – there are not many! The loop over individuals goes up to M
3821 now, and there is a model component for the DA variables z . We are also computing
3822 some derived parameters: population size $N(\mathcal{S})$ is computed by summing up all of
3823 the data augmentation variables z_i (as we've done previously in Chapt. 3) and
3824 density, D , is also a derived parameter, being a function of N . The input data has
3825 changed slightly too, as the augmented data set has more rows to include excess
3826 all-zero encounter histories. Previously we knew that $N = 100$ but in this analysis
3827 we pretend not to know N , but think that $N = 200$ is a good upper bound. This

3828 analysis can be run directly using the `SCR0bayes` function once the `scrbook` package
 3829 is loaded, by issuing the following commands:

```
3830 > library(scrbook)
3831 > data <- simSCR0(discard0=TRUE, rnd=2013)
3832 > out1 <- SCR0bayes(data, M=200, engine="winbugs", ni=2000, nb=1000)
```

3833 Summarizing the output from **WinBUGS** produces:

```
3834 > print(out1,digits=2)
3835 Inference for Bugs model at "SCR0b.txt", fit using WinBUGS,
3836 3 chains, each with 2000 iterations (first 1000 discarded)
3837 n.sims = 3000 iterations saved
3838      mean   sd  2.5%   25%   50%   75% 97.5% Rhat n.eff
3839 alpha0  -2.57 0.23 -3.04 -2.72 -2.56 -2.41 -2.15 1.01  320
3840 alpha1   2.46 0.42  1.63  2.16  2.46  2.73  3.33 1.02  120
3841 sigma    0.46 0.04  0.39  0.43  0.45  0.48  0.55 1.02  120
3842 N       113.62 15.73 86.00 102.00 113.00 124.00 147.00 1.01  260
3843 D        1.78 0.25  1.34  1.59  1.77  1.94  2.30 1.01  260
3844 deviance 302.60 23.67 261.19 285.47 301.50 317.90 354.91 1.00  1400
3845
3846 [...some output deleted...]
3847
```

3848 The `Rhat` statistic (discussed in Secs. ?? and 16.4.5) for this analysis indicates
 3849 satisfactory convergence. We see that the estimated parameters (α_0 and α_1) are
 3850 comparable to the previous results obtained for the known- N case, and also not
 3851 too different from the data-generating values. The posterior of N overlaps the
 3852 data-generating value substantially.

3853 Use of other BUGS engines: JAGS

3854 There are two other popular **BUGS** engines in widespread use: **OpenBUGS**
 3855 (Thomas et al., 2006) and **JAGS** (Plummer, 2003). Both of these are easily called
 3856 from **R**. **OpenBUGS** can be used instead of **WinBUGS** by changing the package
 3857 option in the `bugs` call to `package='OpenBUGS'`. **JAGS** can be called using
 3858 the function `jags()` in package `R2jags` which has nearly the same arguments as
 3859 `bugs()`. Or, it can be executed from the **R** package `rjags` (Plummer, 2011) which
 3860 has a slightly different implementation that we demonstrate here as we reanalyze
 3861 the simulated data set in the previous section (note: the same **R** commands are used
 3862 to generate the data and package the data, inits and parameters to monitor). The
 3863 function `jags.model` is used to initialize the model and run the MCMC algorithm
 3864 for an adaptive period during which tuning of the MCMC algorithm might take
 3865 place. These samples cannot be used for inference. Then the Markov chains are
 3866 updated using `coda.samples()` to obtain posterior samples for analysis, as follows:

```
3867 > jinit <- jags.model("SCR0b.txt", data=data, inits=inits,
3868                      n.chains=3, n.adapt=1000)
3869 > jout <- coda.samples(jinit, parameters, n.iter=1000, thin=1)
```

Table 4.2. Posterior mean of model parameters for 4 different models fitted to a single simulated data set, and the effective home range area under each detection model.

	Gaussian	Cloglog	Exponential	Logit
α_0	-2.57	-2.60	-1.51	-0.47
α_1	2.46	2.56	3.59	3.86
N	113.62	114.16	119.69	118.29
D	1.78	1.78	1.87	1.85
hra	3.85	3.78	5.51	2.64

3870 These commands can be executed using the function `SCR0bayes` provided with
 3871 the **R** package `scrbook`. Hobbs (2011) provides a good introduction to ecological
 3872 modeling with **JAGS** which we recommend.

3873 **4.7.2 Implied home range area**

3874 Here we apply the method described in Sec. 4.4 to compute the effective home
 3875 range area under different encounter probability models fit to simulated data. We
 3876 simulated a data set from the Gaussian kernel model as in Sec. 4.7 and then we
 3877 fitted 4 models to it: (1) the true data-generating Gaussian encounter probability
 3878 model; (2) the “hazard” or complementary log-log link model (Eq. 4.4.2); (3) the
 3879 negative exponential model and (4) the logit model (Eq. 4.2.2). We modified the
 3880 function `SCR0bayes` for this purpose which you should be able to do with little
 3881 difficulty. We fit each model to the same simulated data set using **WinBUGS**,
 3882 based only on 1000 post-burn-in samples and 3 chains, which produced the posterior
 3883 summaries given in Table 4.2. The main thing we see is that, while the implied home
 3884 range area can vary substantially, there are smaller differences in the estimated N
 3885 and hence D .

3886 **4.7.3 Realized and expected density**

3887 In Bayesian analysis of the SCR model, we estimate a parameter N which is the
 3888 size of the population for the prescribed state-space (presumably the state-space is
 3889 defined so as to be relevant to where our traps were located, so N can be thought
 3890 of as the size of the sampled population). In the context of Efford and Fewster
 3891 (2012) this is the *realized* population size. Conversely, sometimes we see estimates
 3892 of *expected* population size reported, which are estimates of $\mathbb{E}(N)$, the expected
 3893 size of some hypothetical, unspecified population. Usually the distinction between
 3894 realized and expected population size is not made in SCR models, because almost
 3895 everyone only cares about actual populations – and their realized population size.

3896 If you do likelihood analysis of SCR models, then the distinction between re-
 3897 alized and expected is often discussed by whether the estimator is “conditional on
 3898 N ” (realized) or not (expected). The naming arises because in obtaining the MLE

3899 of N , its properties are evaluated *conditional* on N – in particular, if the estimator
 3900 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
 3901 does not conform to any concept or quantity that is relevant to Bayesian inference.
 3902 If we care about N for the population that we sampled it is understood to be a
 3903 realization of a random variable, but the relevance of “conditional on N ” is hard to
 3904 see. Bayesian analysis will provide a prediction of N that is based on the posterior
 3905 $[N|y, \theta]$ – which is certainly *not* conditional on N .

3906 There is a third type of inference objective that is relevant in practice and that
 3907 is prediction of N for a population that was not sampled – i.e., a “new” population.
 3908 To elaborate on this, consider a situation in which we are concerned about the tiger
 3909 population in 2 distinct reserves in India. We do a camera trapping study on one of
 3910 the reserves to estimate N_1 and we think the reserves are similar and homogeneous
 3911 so we’re willing to apply a density estimate based on N_1 to the 2nd reserve. For
 3912 the 2nd reserve, do we want a prediction of the realized population size, N_2 , or do
 3913 we want an estimates of its expected value? We believe the former is the proper
 3914 quantity for inference about the population size in the 2nd reserve. An estimate of
 3915 N_2 should include the uncertainty with which the mean is estimated (from reserve
 3916 1) and it should also include “process variation” for making the prediction of the
 3917 latent variable N_2 .

3918 As a practical matter, to do a Bayesian analysis of this you could just define the
 3919 state-space to be the union of the two state-spaces, increase M so that the posterior
 3920 of the total population size is not truncated, and then have MCMC generate a
 3921 posterior sample of individuals on the joint state-space. You can tally-up the ones
 3922 that are on S_2 as an estimate of N_2 . Alternatively, we can define $\mu = \psi M / A_1$
 3923 and then simulate posterior samples of $N_s \sim \text{Binomial}(M, \mu A_2 / M)$ for the new
 3924 state-space area, A_2 .

3925 To carry out a classical likelihood analysis of this 2nd type of problem, what
 3926 should we do? The argument for making a prediction of a new value of N would
 3927 go something like this: If you obtain an MLE of N , say \hat{N} , then the inference
 3928 procedure tells us the variance of this *conditional* on N . i.e., $\text{Var}(\hat{N}|N)$. This is
 3929 fine, if we care about the specific value of N that generated our data set. However,
 3930 if we don’t care about the specific one in question then we want to “uncondition”
 3931 on N to introduce a new variance component. Law of total variance says:

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

3932 If \hat{N} is unbiased then we say the unconditional variance is

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

3933 The first part is estimation error and the 2nd component is the “process variance.”
 3934 If you do Bayesian analysis, then you don’t have to worry too much about how to
 3935 compute variances properly. You decide if you care about N , or its expected value,

3936 or predictions of some “new” N , and you tabulate the correct posterior distribution
 3937 from your MCMC output.

3938 The considerations for estimating density are the same. Density can be N/A
 3939 where N is the realized population, which we understand it to be unless we put an
 3940 expectation operator around the N like $\mathbb{E}(N)/A$. Classically, density is thought of
 3941 as being defined as the expected value of N but this might not always be meaningful
 3942 because the context of whether we mean realized density, of an actual population,
 3943 or expected density for some hypothetical unspecified population, should matter.
 3944 The formula for obtaining “expected density” is slightly different depending on
 3945 whether we assume N has a Poisson distribution or whether we assume a binomial
 3946 distribution (under data augmentation). In the latter case ψ is related to the point
 3947 process intensity (see Chapt. 10) in the sense that, under the binomial prior:

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

3948 so, what we think of as “density”, D , is $D = M\psi/A$. Under the Poisson point
 3949 process model we have:

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

3950 In summary, there are 3 basic inference problems that relate to estimating pop-
 3951 ulation size (or density):

- 3952 (1) What is the value of N for some population that was sampled. This is what
 3953 Efford and Fewster call “realized N ” In general, we want the uncertainty to reflect
 3954 having to estimate n_0 , the part of the population not seen.
- 3955 (2) We need to estimate N for some population that we didn’t sample but it is
 3956 “similar” to the population that we have information on. In this case, we have to
 3957 account for both variation in having to estimate parameters of the distribution of
 3958 N and we have to account for process variation in N (i.e., due to the stochastic
 3959 model of N).
- 3960 (3) In some extremely limited cases we might care about estimating the expected
 3961 value of N , $\mathbb{E}(N)$. This is only useful as a hypothetical statement that we might
 3962 use, e.g., if we were to establish a new million ha refuge somewhere, then we
 3963 might say its expected population size is 200 tigers.

4.8 THE CORE SCR ASSUMPTIONS

3964 It’s always a good idea to sit down and reflect on the meaning of any particular
 3965 model, its various assumptions, and what they mean in a specific context. From
 3966 the statistician’s point of view, the basic assumption, the omnibus assumption, as
 3967 in all of statistics, and for every statistical model, is that “the model is correctly
 3968 specified”. So, naturally, that precludes everything that isn’t explicitly addressed
 3969 by the model. To point this out to someone seems to cause a lot of anxiety, so we
 3970 enumerate here what we think are the most important statistical assumptions of
 3971 the basic SCR0 model:

- 3972 • **Demographic closure.** The model does not allow for demographic processes.
3973 There is no recruitment or entry into the sampled population. There is no mor-
3974 tality or exit from the sampled population.
- 3975 • **Geographic closure.** We assume no permanent emigration or immigration
3976 from the state-space. However, we allow for “temporary” movements around
3977 the state-space and variable exposure to encounter as a result. The whole point
3978 of SCR models is to accommodate this dynamic. In ordinary capture-recapture
3979 models we have to assume geographic closure to interpret N in a meaningful way.
- 3980 • **Activity centers are randomly distributed.** That is, uniformity and inde-
3981 pendence of the underlying point process $\mathbf{s}_1, \dots, \mathbf{s}_N$ (see next section).
- 3982 • **Detection is a function of distance.** A detection model that describes how
3983 encounter probability declines as a function of distance from an individual’s home
3984 range center.
- 3985 • **Independence of encounters** among individuals. Encounter of any individual
3986 is independent of encounter of each other individual.
- 3987 • **Independence of encounters** of the same individual. Encounter of an individ-
3988 ual in any trap is independent of its encounter in any other trap, and subsequent
3989 sample occasion.

3990 It’s easy to get worried and question the whole SCR enterprise just on the grounds
3991 that these assumptions combine to form such a simplistic model, one that surely
3992 can’t describe the complexity of real populations. On this sentiment, a few points
3993 are worth making. First, you don’t have inherently fewer assumptions by using an
3994 ordinary capture-recapture model but, rather, the SCR model relaxes a number of
3995 important assumptions compared to the non-spatial counterpart. For one, here,
3996 we’re not assuming that p is constant for all individuals but rather that p varies
3997 substantially as a matter of the spatial juxtaposition of individuals with traps. So
3998 maybe the manner in which p varies isn’t quite right, but that’s not an argument
3999 that supports doing less modeling. Fundamentally a distance-based model for p has
4000 some basic biological justification in virtually every capture-recapture study. Sec-
4001 ondly, for some of these core assumptions such as uniformity, and independence of
4002 individuals and of encounters, we expect a fair amount of robustness to departures.
4003 They function primarily to allow us to build a model and an estimation scheme and
4004 we don’t usually think they represent real populations (of course, no model does!).
4005 Third, we can extend these assumptions in many different ways and we do that
4006 to varying extents in this book, and more work remains to be done in this regard.
4007 Forth, we can also evaluate the reasonableness of the assumptions formally in some
4008 cases using standard methods of assessing model fit (Chapt. 7).

4009 Finally, we return back to our sentiment about the omnibus assumptions which
4010 is that the model is properly specified. This precludes *everything* that isn’t in
4011 the model. Sometimes you see in capture-recapture literature statements like “we
4012 assume no marks are lost”, “marks are correctly identified” and similar things. We
4013 might as well also assume that, a shopping mall is not built, or a meteor does not

4014 crash down into our study area, the sun does not go super-nova, and so forth. Our
4015 point is that we should separate statistical assumptions about model parameters or
4016 aspects of the probability model from what are essentially logistical or operational
4017 assumptions about how we interpret our data, or based on our ability to conduct
4018 the study. It is pointless to enumerate all of the possible explanations for apparent
4019 *departures*, because there are an infinity of such cases.

4.9 WOLVERINE CAMERA TRAPPING STUDY

4020 We provide an illustration of some of the concepts we've introduced previously
4021 in this chapter by analyzing data from a camera trapping data from a study of
4022 wolverines *Gulo gulo* (Magoun et al., 2011; Royle et al., 2011b). The study took
4023 place in SE Alaska (Fig. 4.4) where 37 cameras were operational for variable periods
4024 of time (min = 5 days, max = 108 days, median = 45 days). A consequence of this
4025 is that the number of sampling occasions, K , is variable for each camera. Thus,
4026 we must provide a vector of sample sizes as data to **BUGS** and modify the model
4027 specification in Sec. 4.7 accordingly.

4028 4.9.1 Practical data organization

4029 To carry out an analysis of these data, we require the matrix of trap coordinates
4030 and the encounter history data. We usually store data in 2 distinct data files which
4031 contain all the information needed for an analysis. These files are

- 4032 • The encounter data file (EDF) containing a record of which traps and when each
4033 individual encounter occurred.
- 4034 • The trap deployment file (TDF) which contains the coordinates of each trap,
4035 along with information indicating which sample occasions each trap was operat-
4036 ing.

4037 **Encounter Data File (EDF)** – We store the encounter data in the an efficient
4038 file format which is easily manipulated in **R** and easy to create in Excel and other
4039 spreadsheets which are widely used for data management. The file structure is a
4040 simple matrix with 4 columns, those being: (1) **session ID**: the trap *session* which
4041 usually corresponds to a year or a primary period in the context of a Robust Design
4042 situation, but it could also correspond to a distinct spatial unit (see Sec. 5.5.4 and
4043 Chapt. 13). For a single-year study (as considered here) this should be an integer
4044 that is the same for all records; (2) **individual ID**: the individual identity, being an
4045 integer from 1 to n (repeated for multiple captures of the same individual) indicating
4046 which individual the record (row) of the matrix belongs to; (3) **occasion ID**: The
4047 integer sample occasion which generated the record, and (4) **trap ID**: the trap
4048 identity, an integer from 1 to J , the number of traps. The structure of the EDF
4049 is the same as used in the **secr** package (Efford, 2011) and similar to that used
4050 in the **SPACECAP** (Gopalaswamy et al., 2012a), and **SCRbayes** (Russell et al.,



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

4051 2012) packages, both of which have a 3-column format (`trapID`, `indID`, `sampID`).
 4052 We note that the naming of the columns is irrelevant as far as anything we do in
 4053 this book, although `secr` and other software may have requirements on variable
 4054 naming.

4055 To illustrate this format, the wolverine data are available in the package `scrbook`
 4056 by typing:

4057

```
> data(wolverine)
```

4058 which contains a list having elements `wcaps` (the EDF) and `wtraps` (the TDF). We
 4059 see that `wcaps` has 115 rows, each representing a unique encounter event including
 4060 the trap identity, the individual identity and the sample occasion index (`sample`).
 4061 The first 5 rows of `wcaps` are:

```

4062 > wolverine$wcaps[1:5,]
4063   year individual day trap
4064 [1,]    1          2 127    1
4065 [2,]    1          2 128    1
4066 [3,]    1          2 129    1
4067 [4,]    1         18 130    1
4068 [5,]    1          3 106    2

```

4069 The 1st column here, labeled `year`, is an integer indicating the year or session
4070 of the encounter. All these data come from a single year (2008) and so `year` is set
4071 to 1. Variable `individual` is an integer identity of each individual captured, `day`
4072 is the sample occasion of capture (in this case, the sample occasions correspond
4073 to days), and `trap` is the integer trap identity. The variable `trapid` will have to
4074 correspond to the row of a matrix containing the trap coordinates - in this case the
4075 TDF file `wtraps` which we describe further below.

4076 Note that the information provided in this encounter data file `wcaps` does not
4077 represent a completely informative summary of the data. For example, if no indi-
4078 viduals were captured in a certain trap or during a certain period, then this compact
4079 data format will have no record. Thus we will need to know J , the number of traps,
4080 and K , the number of sample occasions when reformatting this SCR data format
4081 into a 2-d encounter frequency matrix or 3-d array. In addition, the encounter data
4082 file does not provide information about which periods each trap was operated. This
4083 additional information is also necessary as the trap-specific sample sizes must be
4084 passed to **BUGS** as data. We provide this information along with trap coordinates,
4085 in the “trap deployment file” (TDF) which is described below.

4086 For our purposes, we need to convert the `wcaps` file into the $n \times J$ array of
4087 binomial encounter frequencies, although more general models might require an
4088 encounter-history formulation of the model which requires a full 3-d array. To
4089 obtain our encounter frequency matrix, we do this the hard way by first converting
4090 the encounter data file into a 3-d array and then summarize to trap totals. We have
4091 a handy function `SCR23darray` which takes the compact encounter data file, and
4092 converts it to a 3-d array, and then we use the **R** function `apply` to summarize over
4093 the sample occasion dimension (by convention here, this is the 2nd dimension). To
4094 apply this to the wolverine data in order to compute the 3-d array we do this:

```

4095 > y3d <- SCR23darray(wolverine$wcaps,wolverine$wtraps)
4096 > y <- apply(y3d,c(1,2),sum)

```

4097 See the help file for more information on `SCR23darray`. The 3-d array is necessary to
4098 fit certain types of models (e.g., behavioral response) and this is why we sometimes
4099 will require this maximally informative 3-d data format but, here, we analyze the
4100 summarized data.

4101 **Trap Deployment File (TDF)** – The other important information needed to
4102 fit SCR models is the “trap deployment file” (TDF) which provides additional
4103 information not contained in the encounter data file. The traps file has $K + 3$

4104 columns. The first column is assumed to be a trap identifier, columns 2 and 3
 4105 are the easting and northing coordinates (assumed to be in a Euclidean coordinate
 4106 system), and columns 4 to $K + 3$ are binary indicators of whether each trap was
 4107 operational during each sample occasion. The first 10 rows (out of 37) and 10
 4108 columns (out of 167) of the trap deployment file for the wolverine data are shown
 4109 as follows:

```
4110 > wolverine$wtraps[1:10,1:10]
4111
4112   Easting Northing 1 2 3 4 5 6 7 8
4113 1 632538 6316012 0 0 0 0 0 0 0 0
4114 2 634822 6316568 1 1 1 1 1 1 1 1
4115 3 638455 6309781 0 0 0 0 0 0 0 0
4116 4 634649 6320016 0 0 0 0 0 0 0 0
4117 5 637738 6313994 0 0 0 0 0 0 0 0
4118 6 625278 6318386 0 0 0 0 0 0 0 0
4119 7 631690 6325157 0 0 0 0 0 0 0 0
4120 8 632631 6316609 0 0 0 0 0 0 0 0
4121 9 631374 6331273 0 0 0 0 0 0 0 0
4122 10 634068 6328575 0 0 0 0 0 0 0 0
```

4123 This tells us that trap 2 was operated during occasions (days) 1-7 but the
 4124 other traps were not operational during those periods. It is extremely important
 4125 to recognize that each trap was operated for a variable period of time and thus
 4126 the binomial “sample size” is different for each, and this needs to be accounted for
 4127 in the **BUGS** model specification. To compute the vector of sample sizes K , and
 4128 extract the trap locations, we do this:

```
4129 > traps <- wolverine$wtraps
4130 > traplocs <- traps[,1:2]
4131 > K <- apply(traps[,3:ncol(traps)],1,sum)
```

4132 This results in a matrix `traplocs` which contains the coordinates of each trap and
 4133 a vector K containing the number of days that each trap was operational. We now
 4134 have all the information required to fit a basic SCR model in **BUGS**.

4135 Summarizing the data for the wolverine study, we see that 21 unique individuals
 4136 were captured a total of 115 times. Most individuals were captured 1-6 times, with
 4137 4, 1, 4, 3, 1, and 2 individuals captured 1-6 times, respectively. In addition, 1
 4138 individual was captured each 8 and 14 times and 2 individuals each were captured
 4139 10 and 13 times. The number of unique traps that captured a particular individual
 4140 ranged from 1-6, with 5, 10, 3, 1, 1, and 1 individual captured in each of 1 to
 4141 6 different traps, respectively, for a total of 50 unique wolverine-trap encounters.
 4142 These numbers might be hard to get your mind around whereas some tabular
 4143 summary is often more convenient. For that it seems natural to tabulate individuals
 4144 by trap and total encounter frequencies. The spatial information in SCR data is
 4145 based on multi-trap captures, and so, it is informative to understand how many

4146 unique traps each individual is captured in, and the total number of encounters.
4147 For the wolverine data, we reproduce Table 1 from Royle et al. (2011b) as Table
4148 4.3.

Table 4.3. Individual frequencies of capture for wolverines captured in camera traps in South-east Alaska in 2008. Rows index unique traps of capture for each individual and columns represent total number of captures (e.g., we captured 4 individuals 1 time, necessarily in only 1 trap; we captured 3 individuals 3 times but in 2 different traps).

No. of traps	No. of captures									
	1	2	3	4	5	6	8	10	13	14
1	4	1	0	0	0	0	0	0	0	0
2	0	0	3	2	0	2	1	2	0	0
3	0	0	1	1	0	0	0	0	0	1
4	0	0	0	0	0	0	0	0	1	0
5	0	0	0	0	1	0	0	0	0	0
6	0	0	0	0	0	0	0	0	1	0

4149 4.9.2 Fitting the model in WinBUGS

4150 Here we fit the simplest SCR model with the Gaussian encounter probability model,
4151 although we revisit these data and fit additional models in later chapters. Model
4152 SCR0 is summarized by the following 4 elements:

- 4153 (1) $y_{ij} | \mathbf{s}_i \sim \text{Binomial}(K, z_i p_{ij})$
- 4154 (2) $p_{ij} = p_0 \exp(-\alpha_1 ||\mathbf{x}_j - \mathbf{s}_i||^2)$
- 4155 (3) $\mathbf{s}_i \sim \text{Uniform}(\mathcal{S})$
- 4156 (4) $z_i \sim \text{Bernoulli}(\psi)$

4157 We assume customary flat priors on the structural (hyper-) parameters of the model,
4158 $\alpha_0 = \text{logit}(p_0)$, α_1 and ψ .

4159 It remains to define the state-space \mathcal{S} . For this, we nested the trap array
4160 (Fig. 4.4) in a rectangular state-space extending 20 km beyond the traps in each
4161 cardinal direction. We scaled the coordinate system so that a unit distance was
4162 equal to 10 km, producing a rectangular state-space of dimension 9.88×10.5 units
4163 ($\text{area} = 10374 \text{ km}^2$) within which the trap array was nested. As a general rule, we
4164 recommend scaling the state-space so that it is defined near the origin $(x, y) = (0, 0)$.
4165 While the scaling of the coordinate system is theoretically irrelevant, a poorly
4166 scaled coordinate system can produce Markov chains that mix poorly. The buffer
4167 of the state space should be large enough so that individuals beyond the state-
4168 space boundary are not likely to be encountered (Sec. 4.3.1). To evaluate this, we
4169 fit models for various choices of a rectangular state-space based on buffers from 1.0
4170 to 5.0 units (10 km to 50 km). In the R package `scrbook` we provide a function
4171 `wolvSCR0` which will fit model SCR0. For example, to fit the model in **WinBUGS**
4172 using data augmentation with $M = 300$ potential individuals, using 3 Markov

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

Buffer	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

4173 chains each of 12000 total iterations, discarding the first 2000 as burn-in, we execute
 4174 the following **R** commands:

```
4175 > library(scrbook)
4176 > data(wolverine)
4177 > traps <- wolverine$wtraps
4178 > y3d <- SCR23darray(wolverine$wcaps,wolverine$wtraps)
4179 > wolv <- wolvSCR0(y3d,traps,nb=2000,ni=12000,buffer=1,M=300)
```

4180 The argument **buffer** determines the buffer size of the state-space in the scaled
 4181 units (i.e., 10 km). Note that this analysis takes between 1-2 hours on many
 4182 machines (in 2013) so we recommend testing it with lower values of M and fewer
 4183 iterations. The posterior summaries are shown in Table 4.9.2.

4184 4.9.3 Summary of the wolverine analysis

4185 We see that the estimated density is roughly consistent as we increase the state-
 4186 space buffer from 15 to 55 km. We do note that the data augmentation parameter
 4187 ψ (and, correspondingly, N) increase with the size of the state space in accordance
 4188 with the deterministic relationship $N = D * A$. However, density is more or less
 4189 constant as we increase the size of the state-space beyond a certain point. For the
 4190 10 km state-space buffer, we see a slight effect on the posterior distribution of D
 4191 because the state-space is not sufficiently large. The full results from the analysis
 4192 based on 20 km state-space buffer are given in Table 4.5.

4193 Our point estimate of wolverine density from this study, using the posterior
 4194 mean from the state-space based on the 20 km buffer, is approximately 5.77 indi-
 4195 viduals/1000 km² with a 95% posterior interval [3.86, 8.29]. Density is estimated

Table 4.5. Posterior summaries of SCR model parameters for the wolverine camera trapping data from SE Alaska. The model was run with the trap array centered in a state-space with a 20 km rectangular buffer.

parameter	mean	SD	2.5%	25%	50%	75%	97.5%	Rhat
ψ	0.20	0.05	0.12	0.17	0.20	0.23	0.30	1
α_1	1.26	0.21	0.87	1.11	1.25	1.40	1.71	1
σ	0.64	0.06	0.54	0.60	0.63	0.67	0.76	1
p_0	0.06	0.01	0.04	0.05	0.06	0.06	0.08	1
N	59.84	11.89	40.00	51.00	59.00	67.00	86.00	1
D	5.77	1.15	3.86	4.92	5.69	6.46	8.29	1

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

It is worth thinking about this model, and these estimates, computed under a rectangular state space roughly centered over the trapping array (Fig. 4.4). 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. 16). 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. 4.10.

4.9.4 Wolverine space usage

The parameter α_1 is related to the home range radius (Sec. 4.4). 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 $\pi \times 15.66^2 = 770.4$ km². Using our handy function **hra** we do this:

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

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

Royle et al. (2011b) reported estimates for σ in the range $6.3 - 9.8 \text{ km}$ 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 biological interpretations of home range area based on σ , although SCR models can be extended to handle more general, non-Euclidean, patterns of space usage. See Chaps. 11 and 12.

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

4.10 USING A DISCRETE HABITAT MASK

The SCR model developed previously in this chapter assumes that individual activity centers are distributed uniformly over the prescribed state-space. Clearly this will not always be a reasonable assumption. In Chapt. 10, we develop models that allow explicitly for non-uniformity of the activity centers by modeling covariate effects on density. A simplistic method of affecting the distribution of activity centers, which we address here, is to modify the shape and organization of the state-space explicitly. For example, we might be able to classify the state-space into distinct blocks of habitat and non-habitat. In that case we can remove the non-habitat from the state-space and assume uniformity of the activity centers over the remaining portions judged to be suitable habitat. There are several ways to approach this: We can use a grid of points to represent the state-space, i.e., by the set of coordinates $\mathbf{s}_1, \dots, \mathbf{s}_G$, and assign equal probabilities to each possible value. Alternatively, we can retain the continuous formulation of the state-space but attempt to describe constraints analytically, or we can use polygon clipping methods to enforce

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.

4259 constraints on the state-space in the MCMC analysis. We focus here on the formulation
 4260 of the basic SCR model in terms of a discrete state-space but in Chapt. 16 we
 4261 demonstrate the latter approach based on using polygon operations to define an irregular
 4262 state-space. Use of a discrete state-space can be computationally expensive
 4263 in **WinBUGS**. That said, it isn't too difficult to perform the MCMC calculations
 4264 in **R** (discussed in Chapt. 16). The **R** package **SPACECAP** (Gopalaswamy et al.,
 4265 2012a) arose from the **R** implementation of the SCR model in Royle et al. (2009a).

4266 While clipping out non-habitat seems like a good idea, we think investigators
 4267 should go about this very cautiously. We might prefer to do it when non-habitat
 4268 represents a clear-cut restriction on the state-space such as a reserve boundary or
 4269 a lake, ocean or river. But, having the capability to do this also causes people
 4270 to start defining "habitat" vs. "non-habitat" based on their understanding of the
 4271 system whereas it can't be known whether the animal being studied has the same
 4272 understanding. Moreover, differentiating the landscape by habitat or habitat quality
 4273 must affect the geometry and morphology of home ranges (see Chapt. 12) much
 4274 more so than the plausible locations of activity centers. That is, a home range
 4275 centroid could, in actual fact, occur in a shopping mall parking lot if there is pretty
 4276 good habitat around the shopping mall, so there is probably no sense preclude it
 4277 as the location for an activity center. It would generally be better to include some
 4278 definition of habitat quality in the model for the detection probability (Royle et al.,
 4279 2012a) which we address in Chaps. 11 and 12.

4280 **4.10.1 Evaluation of coarseness of habitat mask**

4281 The coarseness of the state-space should not really have much of an effect on es-
 4282 timates if the grain is sufficiently fine relative to typical animal home range sizes.
 4283 Why is this? We have two analogies that can help us understand. First is the
 4284 relationship to model M_h . As noted in Sec. 4.3.2 above, we can think about SCR
 4285 models as a type of finite mixture (Norris and Pollock, 1996; Pledger, 2004) where
 4286 we are fortunate to be able to obtain direct information about which group indi-
 4287 viduals belong to (group being location of activity center). In the standard finite
 4288 mixture models we typically find that a small number of groups (e.g., 2 or 3 at
 4289 the most) can explain high levels of heterogeneity and are adequate for most data
 4290 sets of small to moderate sample sizes. We therefore expect a similar effect in SCR
 4291 models when we discretize the state-space. We can also think about discretizing
 4292 the state-space as being related to numerical integration where we find (see Chapt.
 4293 5) that we don't need a very fine grid of support points to evaluate the integral to
 4294 a reasonable level of accuracy. We demonstrate this here by reanalyzing simulated
 4295 data using a state-space defined by a different number of support points. We pro-
 4296 vide an **R** script called **SCR0bayesDss** in the **R** package **scrbook**. We note that for
 4297 this comparison we generated the actual activity centers as a continuous random
 4298 variable and thus the discrete state-space is, strictly speaking, an approximation
 4299 to truth. That said, we regard all state-space specifications as approximations to

Table 4.6. Comparison of the effect of state-space grid coarseness on estimates of N for a simulated data set. Posterior summaries and run time are given. Results obtained using **WinBUGS** run from R2WinBUGS.

grid	Mean	SD	NaiveSE	Time-seriesSE	runtime (sec)
6	111.6699	16.61414	0.1516657	0.682008	2274
9	114.2294	17.99109	0.1642355	0.833291	4300
12	115.9806	17.3843	0.1586964	0.762756	7100
15	115.379	17.93721	0.1637436	0.832483	13010

4300 truth in the sense that they represent a component of the SCR model.

4301 As with our **R** function **SCR0bayes**, the modification **SCR0bayesDss** will use
 4302 either **WinBUGS** or **JAGS**. In addition, it requires a grid resolution argument
 4303 (**ng**) which is the dimension of 1 side of a square state-space. To execute this
 4304 function we do, for example:

```
4305 > library(scrbook)
4306 > data <- simSCR0(discard0=TRUE,rnd=2013)      # generate data set
4307
4308 # run with JAGS
4309 > out1 <- SCR0bayesDss(data,ng=8,M=200,engine="jags",ni=2000,nb=1000)
4310
4311 # run with WinBUGS
4312 > out2 <- SCR0bayesDss(data,ng=8,M=200,engine="winbugs",ni=2000,nb=1000)
```

4313 We fit this model to the same simulated data set for 6×6 , 9×9 , 12×12 , 15×15
 4314 state-space grids. For **WinBUGS**, we used 3 chains of 5000 total length with 1000
 4315 burn-in, which yields 12000 total posterior samples. Summary results are shown
 4316 in Table 4.6. The results are broadly consistent except for the 6×6 case. We see
 4317 that the run time increases with the size of the state-space grid (not unexpected),
 4318 such that we imagine it would be impractical to run models with more than a
 4319 few hundred state-space grid points. We found (not shown here) that the runtime
 4320 of **JAGS** is much faster and, furthermore, relatively *constant* as we increase the
 4321 grid size. We suspect that **WinBUGS** is evaluating the full-conditional for each
 4322 activity center at all G possible values whereas it may be that **JAGS** is evaluating
 4323 the full-conditional only at a subset of values or perhaps using previous calculations
 4324 more effectively. While this might suggest that one should always use **JAGS** for
 4325 this analysis, we found in our analysis of the wolverine (next section) that **JAGS**
 4326 could be extremely sensitive to starting values, producing MCMC algorithms that
 4327 often simply do not work for some problems, so be careful when using **JAGS**.
 4328 To improve its performance, always start the latent activity centers at values near
 4329 where individuals were captured. The performance of either should improve if we
 4330 compute the full distance matrix outside of **BUGS** and pass it as data, although
 4331 we haven't fully evaluated this approach.

4.10.2 Analysis of the wolverine camera trapping data

We reanalyzed the wolverine data using discrete state-space grids with points spaced by 2, 4 and 8 km (see Fig. 4.5). These were constructed from a 40 km buffered state-space, and deleting the points over water (see Royle et al., 2011b). Our interest in doing this was to evaluate the relative influence of grid resolution on estimated density because the coarser grids will be more efficient from a computational standpoint and so we would prefer to use them, but only if there is no strong influence on estimated density. The posterior summaries for the 3 habitat grids are given in Table 4.7. We see that the density estimates are quite a bit larger than obtained in our analysis (Table 4.9.2) based on a rectangular, continuous state-space. We also see that there are slight differences depending on the resolution of the state-space grid. Interestingly, the effectiveness of the MCMC algorithms, as measured by effective sample size (*n.eff*) is pretty remarkably different. Furthermore, the finest grid resolution (2 km spacing) took about 6 days to run and thus it would not be practical for large problems or with many models.

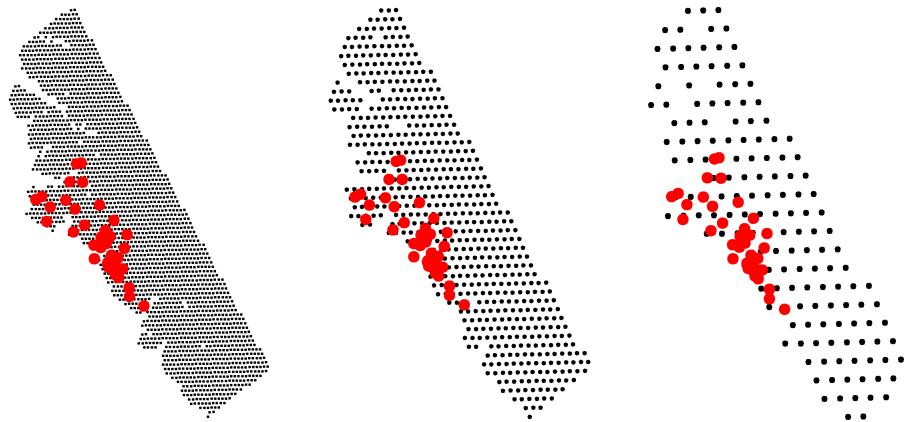


Figure 4.5. Three habitat mask grids used in the comparison of the effect of pixel size on the estimated density surface of wolverines. The 3 cases are 2 (left), 4 (center) and 8 (right) km spacing of state-space points, extending 40 km from the vicinity of the trap array.

4.11 SUMMARIZING DENSITY AND ACTIVITY CENTER LOCATIONS

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

Table 4.7. Posterior summaries for the wolverine camera trapping data, using model SCR0, with a Gaussian hazard encounter probability model, and a discrete habitat mask of 3 different resolutions: 2, 4 and 8 km. Parameters are 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	
σ	0.62	0.05	0.54	0.59	0.62	0.65	0.73	1.01	160	
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	
ψ	0.43	0.09	0.27	0.37	0.43	0.49	0.63	1.00	560	
N	86.56	16.94	57.00	75.00	85.00	97.00	124.00	1.00	510	
D	8.78	1.72	5.78	7.60	8.62	9.83	12.57	1.00	510	
4 km spacing										
	Mean	SD	2.5%	25%	50%	75%	97.5%	Rhat	n.eff	
σ	0.61	0.04	0.53	0.58	0.61	0.64	0.71	1	1600	
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	
ψ	0.45	0.09	0.28	0.38	0.44	0.50	0.64	1	1300	
N	89.25	17.44	59.00	77.00	88.00	100.00	127.00	1	1100	
D	9.01	1.76	5.96	7.77	8.88	10.10	12.82	1	1100	
8 km spacing										
	Mean	SD	2.5%	25%	50%	75%	97.5%	Rhat	n.eff	
σ	0.68	0.05	0.59	0.64	0.67	0.71	0.77	1.01	220	
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	
ψ	0.42	0.09	0.26	0.36	0.41	0.47	0.61	1.00	940	
N	83.18	16.14	56.00	72.00	82.00	93.00	119.00	1.00	700	
D	8.28	1.61	5.57	7.17	8.16	9.26	11.84	1.00	700	

4349 compute many useful and interesting summaries of the activity centers using output
 4350 from an MCMC simulation, including maps of density (the number of activity
 4351 centers per unit area), estimates of N for any well-defined polygon, or estimates of
 4352 where the activity centers for specific individuals reside. In Bayesian analysis by
 4353 MCMC, obtaining such summaries entails no added calculations, because we need
 4354 only post-process the output for the individual activity centers to obtain the desired
 4355 summaries. We demonstrate that in this section. Note that you have to be sure
 4356 to retain the MCMC history for the s variables and also the data augmentation
 4357 variables z in order to do the following analyses.

4358 4.11.1 Constructing density maps

4359 Because SCR models are spatially-explicit, it is natural to want to summarize the
 4360 results of fitting a model by producing a map of density. Using Bayesian analysis
 4361 by MCMC, it is most easy to make a map of *realized* density. We can do this by

4362 tallying up the number of activity centers \mathbf{s}_i in pixels of arbitrary size and then
 4363 producing a nice multi-color spatial plot of the result. Specifically, let $B(\mathbf{x})$ indicate
 4364 a pixel centered at \mathbf{x} then

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

4365 (here, $I(arg)$ is the indicator function which evaluates to 1 if arg is true, and 0
 4366 otherwise) is the population size of pixel $B(\mathbf{x})$, and $D(\mathbf{x}) = N(\mathbf{x})/\|B(\mathbf{x})\|$ is the
 4367 local density. Note that these $N(\mathbf{x})$ parameter are just “derived parameters” as we
 4368 normally obtain from posterior output using the appropriate Monte Carlo average
 4369 (see Chapt. ??).

4370 One thing to be careful about, in the context of models in which N is unknown,
 4371 is that, for each MCMC iteration m , we only tabulate those activity centers which
 4372 correspond to individuals in the sampled population, i.e., for which the data aug-
 4373 mentation variable $z_i = 1$. In this case, we take all of the output for MCMC
 4374 iterations $m = 1, 2, \dots, \text{niter}$ and compute this summary:

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

4375 Thus, $N(\mathbf{x}, 1), N(\mathbf{x}, 2), \dots$, is the Markov chain for parameter $N(\mathbf{x})$. In what fol-
 4376 lows we will provide a set of **R** commands for doing this calculation and making a
 4377 basic image plot from the MCMC output.

4378 **Step 1:** Define the center points of each pixel $B(\mathbf{x})$, or point at which local density
 4379 will be estimated:

```
4380 > xg <- seq(xlim[1], xlim[2], , 50)
4381 > yg <- seq(ylim[1], ylim[2], , 50)
```

4382 **Step 2:** Extract the MCMC histories for the activity centers and the data aug-
 4383 mentation variables. Note that these are each $N \times \text{niter}$ matrices. Here we do this
 4384 assuming that **WinBUGS** was run producing the **R** object named **out**:

```
4385 > Sxout <- out$sims.list$s[, , 1]
4386 > Syout <- out$sims.list$s[, , 2]
4387 > z <- out$sims.list$z
```

4388 **Step 3:** We associate each coordinate with the proper pixel using the **R** command
 4389 **cut()**. Note that we keep only the activity centers for which $z = 1$ (i.e., individuals
 4390 that belong to the population of size N):

```
4391 > Sxout <- cut(Sxout[z==1], breaks=xg, include.lowest=TRUE)
4392 > Syout <- cut(Syout[z==1], breaks=yg, include.lowest=TRUE)
```

4393 **Step 4:** Use the `table()` command to tally up how many activity centers are in
 4394 each $B(x)$:

4395 > `Dn <- table(Sxout,Syout)`

4396 **Step 5:** Use the `image()` command to display the resulting matrix.

4397 > `image(xg, yg, Dn/nrow(z), col=terrain.colors(10))`

4398 It is worth emphasizing here that density maps will not usually appear uniform
 4399 despite that we have assumed that activity centers are uniformly distributed. This is
 4400 because the observed encounters of individuals provide direct information about the
 4401 location of the $i = 1, 2, \dots, n$ activity centers and thus their “estimated” locations
 4402 will be affected by the observations. In a limiting sense, were we to sample space
 4403 intensely enough, every individual would be captured a number of times and we
 4404 would have considerable information about all N point locations. Consequently,
 4405 the uniform prior would have almost no influence at all on the estimated density
 4406 surface in this limiting situation. Thus, in practice, the influence of the uniformity
 4407 assumption decreases as the fraction of the population encountered, and the total
 4408 number of encounters per individual, increases.

4409 **On the non-intuitiveness of `image()`** – the R function `image()`, invoked for
 4410 a matrix M by `image(M)`, might not be very intuitive to some – it plots $M[1, 1]$ in
 4411 the lower left corner. If you want $M[]$ to be plotted “as you look at it” then $M[1, 1]$
 4412 should be in the upper left corner. We have a function `rot()` which does that. If
 4413 you do `image(rot(M))` then it puts it on the monitor as if it was a map you were
 4414 looking at. You can always specify the x - and y -labels explicitly as we did above.

4415 **Spatial dot plots** – A cruder version of the density map can be made using
 4416 our “spatial dot map” function `spatial.plot` (in `scrbook`). This function requires,
 4417 as input, point locations and the value to be displayed. A simplified version of this
 4418 function is as follows:

```
4419 > spatial.plot <- function(x,y){
  4420   nc <- as.numeric(cut(y,20))
  4421   plot(x,pch=" ")
  4422   points(x,pch=20,col=topo.colors(20)[nc],cex=2)
  4423   image.scale(y,col=topo.colors(20))
  4424 }
  4425 #
  4426 # To execute the function do this:
  4427 #
  4428 > spatial.plot(cbind(xg,yg), Dn/nrow(z))
```

4429 **4.11.2 Example: Wolverine density map**

4430 We return to the wolverine study which took place in 2008 in SE Alaska (Fig. 4.4)
 4431 and we produce a density map of wolverines from that analysis. We include the

4432 function SCRdensity which requires a specific data structure as shown below. In
 4433 particular, we have to package up the MCMC history for the activity centers and
 4434 the data augmentation variables z into a list. This also requires that we add those
 4435 variables to the parameters-to-be-monitored list when we pass things to **BUGS**.

4436 We used the posterior output from the wolverine model fitted previously to
 4437 compute a relatively coarse version of a density map, using 100 pixels in a 10×10
 4438 grid (Fig. 4.6 top panel) and using 900 pixels arranged in a 30×30 grid (Fig. 4.6
 4439 lower panel) for a fine-scale map. The **R** commands for producing such a plot (for
 4440 a short MCMC run) are as follows:

```
4441 > library(scrbook)
4442 > data(wolverine)
4443 > traps <- wolverine$wtraps
4444 > y3d <- SCR23darray(wolverine$wcaps,wolverine$wtraps)
4445
4446 # this takes 341 seconds on a standard CPU circa 2011
4447 > out <- wolvSCR0(y3d,traps,nb=1000,ni=2000,buffer=1,M=100,keepz=TRUE)
4448
4449 > Sx <- out$sims.list$s[,,1]
4450 > Sy <- out$sims.list$s[,,2]
4451 > z <- out$sims.list$z
4452 > obj <- list(Sx=Sx,Sy=Sy,z=z)
4453 > tmp <- SCRdensity(obj,nx=10,ny=10,scalein=100,scaleout=100)
```

4454 In these figures density is expressed in units of individuals per 100 km^2 , while the area of
 4455 the pixels is about 103.7 km^2 and 11.5 km^2 , respectively. That calculation is based on:

```
4456 > total.area <- (ylim[2]-ylim[1])*(xlim[2]-xlim[1])*100
4457 > total.area/(10*10)
4458 [1] 103.7427
4459 > total.area/(30*30)
4460 [1] 11.52697
```

4461 A couple of things are worth noting: First is that as we move away from “where the
 4462 data live” – away from the trap array – we see that the density approaches the mean
 4463 density. This is a property of the estimator as long as the detection function decreases
 4464 sufficiently rapidly as a function of distance. Relatedly, it is also a property of statistical
 4465 smoothers such as splines, kernel smoothers, and regression smoothers – predictions tend
 4466 toward the global mean as the influence of data diminishes. Another way to think of it is
 4467 that it is a consequence of the prior, which imposes uniformity, and as you get far away
 4468 from the data, the predictions tend to the expected constant density under the prior.
 4469 Another thing to note about this map is that density is not 0 over water (although the
 4470 coastline is not shown). This might be perplexing to some who are fairly certain that
 4471 wolverines do not like water. However, there is nothing about the model that recognizes
 4472 water from non-water and so the model predicts over water *as if* it were habitat similar to
 4473 that within which the array is nested. But, all of this is OK as far as estimating density
 4474 goes and, furthermore, we can compute valid estimates of N over any well-defined region
 4475 which presumably wouldn’t include water if we so wished. Alternatively, areas covered by
 4476 water could be masked out, which we discuss in the next section.

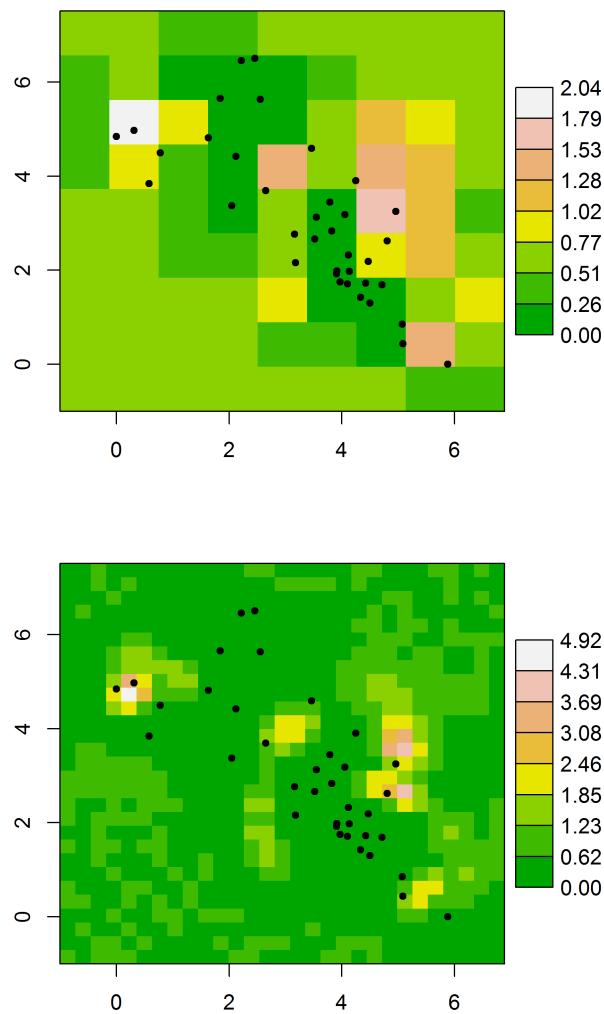


Figure 4.6. Density of wolverines (individuals per 100 km^2) in SE Alaska in 2007 based on model SCR0. Map grid cells are about 103.7 km^2 (top panel) and 11.5 km^2 (bottom panel) in area. Dots are the trap locations.

4.11.3 Predicting where an individual lives

4477 The density maps in the previous section show the expected number of individuals per
 4478 unit area. A closely related problem is that of producing a map of the probable location
 4479 of a specific individual's activity center. For any observed encounter history, we can easily
 4480 generate a posterior distribution of s_i for individual i . In addition, for an individual that
 4481 is *not* captured, we can use the MCMC output to produce a corresponding plot of where
 4482 such an individual might live, say s_{n+1} . Obviously, all such uncaptured individuals (for
 4483 $i = n + 1, \dots, N$) should have the same posterior distribution. To illustrate, we show the
 4484 posterior distribution of s_1 , the activity center for the individual labeled 1 in the data
 4485 set, in Fig. 4.7. This individual was captured a single time at trap 30 which is circled
 4486 in Fig. 4.7. We see that the posterior distribution is affected by traps of capture *and*
 4487 traps of non-capture in fairly intuitive ways. In particular, because there are other traps
 4488 in close proximity to trap 30, in which individual 1 was *not* captured, the model pushes
 4489 its activity center away from the trap array. The help file for **SCRdensity** shows how to
 4490 calculate Fig. 4.7.

4.12 EFFECTIVE SAMPLE AREA

4492 One of the key issues in using ordinary capture recapture models which we've brought up
 4493 over and over again is this issue that the area which is sampled by a trapping array is
 4494 unknown – in other words, the N that is estimated by capture-recapture models does not
 4495 have an explicit region of space associated with it. Classically this has been addressed in
 4496 the ad hoc way of prescribing an area that contains the trap array, usually by adding a
 4497 buffer of some width, which is not estimated as part of the capture-recapture model. In
 4498 SCR models we avoid the problem of not having an explicit linkage between N and “area”,
 4499 by prescribing explicitly the area within which the underlying point process is defined – the
 4500 state-space of the point process. This state-space is *not* the effective sample (or sampled)
 4501 area (ESA) – it is desirable that it be somewhat larger than the ESA, whatever that may
 4502 be, in the sense that individuals at the edge of the state-space have no probability of being
 4503 captured, but as part of the SCR model we don't need to try to estimate or otherwise
 4504 characterize the ESA explicitly.

4505 However, it is possible to provide a characterization of effective sampled area under
 4506 any SCR model. This is directly analogous to the calculation of “effective strip width” in
 4507 distance sampling (Buckland et al., 2001; Borchers et al., 2002). The conceptual definition
 4508 of ESA follows from equating density to “apparent density” – ESA is the magic number
 4509 that satisfies that equivalence:

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

4510 In other words, the ratio of N to the area of the state-space should be equal to the ratio
 4511 of the observed sample size n to this number ESA. Both of these should equal density.
 4512 So, to compute ESA for a model, we substitute $\mathbb{E}(n)$ for n into the above equation, and
 4513 solve for ESA , to get:

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

4514 Our following development assumes that D is constant, but these calculations can be
 4515 generalized to allow for D to vary spatially. Imagine our habitat mask for the wolverine

4516 data, or the bins we just used to produce a density map, then we can write $\mathbb{E}(n)$ according
 4517 to

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

4518 where if we prefer to think of this more conceptually we could replace the summation with
 4519 an integration (which, in practice, we would just replace with a summation, and so we
 4520 just begin there). In this expression note that $\mathbb{E}(N(\mathbf{s}))$ is the expected population size at
 4521 pixel \mathbf{s} which is the density times the area of the pixel, i.e., $\mathbb{E}(N(\mathbf{s})) = D \times a$. Therefore

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

4522 and (plugging this into the expression above for ESA)

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

4523 We see that D cancels and we have $ESA = a \times \sum_s \Pr(\text{encounter}|\mathbf{s})$ So what you have to
 4524 do here is substitute in $\Pr(\text{encounter}|\mathbf{s})$ and just sum them up over all pixels. For the
 4525 Bernoulli model of model SCR0

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

4526 with slight modifications when encounter probability depends on covariates. Thus,

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

4527 Clearly the calculation of ESA is affected by the use of a habitat mask, because the
 4528 summation in Eq. 4.12.1 only occurs over pixels that define the state-space.

4529 For the wolverine camera trapping data, we used the 2×2 km habitat mask and the
 4530 posterior means of p_0 and σ (see Sec. 4.10.2) to compute the probability of encounter for
 4531 each \mathbf{s} of the mask points. The result is shown graphically in Fig. 4.8. The ESA is the
 4532 sum of the values plotted in that figure multiplied by 4, the area of each pixel. For the
 4533 wolverine study, the result is 2507.152 km². We note that the probability of encounter
 4534 declines rapidly to 0 as we move away from the periphery of the camera traps, indicating
 4535 the state-space constructed from a 40 km buffered trap array was indeed sufficient for the
 4536 analysis of these data. An R script for producing this figure is in the `wolvESA` function of
 4537 the `scrbook` package.

4.13 SUMMARY AND OUTLOOK

4538 In this chapter, we introduced the simplest SCR model – “model SCR0” – which is an ordinary
 4539 capture-recapture model like model M_0 , but augmented with a set of latent individual
 4540 effects, \mathbf{s}_i , which relate encounter probability to some sense of individual location using a
 4541 covariate, “distance”, from \mathbf{s}_i to each trap location. Thus, individuals in close proximity
 4542 to a trap will have a higher probability of encounter, and *vice versa*. The explicit modeling
 4543 of individual locations and distance in this fashion resolves classical problems related to

4544 estimating density: unknown sample area, and heterogeneous encounter probability due
4545 to variable exposure to traps.

4546 SCR models are closely related to classical individual covariate models (“model M_x ”,
4547 as introduced in Chapt. 3), but with imperfect information about the individual covari-
4548 ate. Therefore, they are also not too dissimilar from standard GLMMs used throughout
4549 statistics and, as a result, we find that they are easy to analyze using standard MCMC
4550 methods encased in black boxes such as **WinBUGS** or **JAGS**. We will also see that they
4551 are easy to analyze using likelihood methods, which we address in Chapt. 5.

4552 Formal consideration of the collection of individual locations (s_1, \dots, s_N) is funda-
4553 mental to all models considered in this book. In statistical terminology, we think of the
4554 collection of points $\{s_i\}$ as a realization of a point process. Because SCR models formally
4555 link individual encounter history data to an underlying point process, we can obtain for-
4556 mal inferences about the point process. For example, we showed how to produce a density
4557 map (Fig. 4.6), or even a probability map for an individual’s home range center (Fig.
4558 4.7). We can also use SCR models as the basis for doing more traditional point process
4559 analyses, such as testing for “complete spatial randomness” (CSR) (see Chapt. 7), and
4560 computing other point process summaries (Illian et al., 2008).

4561 Part of the promise, and ongoing challenge, of SCR models is to develop models that
4562 reflect interesting biological processes, for example interactions among points or temporal
4563 dynamics in point locations. In this chapter we considered the simplest possible point
4564 process model in which points are independent and uniformly (“randomly”) distributed
4565 over space. Despite the simplicity of this model, it should suffice in many applications of
4566 SCR models, although we do address generalizations in later chapters. Moreover, even
4567 though the *prior* distribution on the point locations is uniform, the realized pattern may
4568 deviate markedly from uniformity as the observed encounter data provide information to
4569 impart deviations from uniformity. Thus, estimated density maps will typically appear
4570 distinctly non-uniform (as we saw in the wolverine example). In applications of the basic
4571 SCR model, we find that this simple *a priori* model can effectively reflect or adapt to
4572 complex realizations of the underlying point process. For example, if individuals are
4573 highly territorial then the data should indicate this in the form of individuals not being
4574 encountered in the same trap – the resulting posterior distribution of point locations should
4575 therefore reflect non-independence. Obviously the complexity of posterior estimates of the
4576 point pattern will depend on the quantity of data, both number of individuals and captures
4577 per individual. Because the point process is such an integral component of SCR models,
4578 the state-space of the point process plays an important role in developing SCR models.
4579 As we emphasized in this chapter, the state-space is part of the model. It can have an
4580 influence on parameter estimates and other inferences, such as model selection (see chapter
4581 7).

4582 One concept we introduced in this chapter, which has not been discussed much in
4583 the literature on SCR models, is the manner in which the encounter probability model
4584 relates to a model of space usage by individuals. The standard SCR models of encounter
4585 probability can all be motivated as simplistic models of space usage and movement, in
4586 which individuals make random use decisions from a probability distribution proportional
4587 to the encounter probability model. This both clarifies the simplicity of the underlying
4588 model of space usage and also suggests a direct extension to produce more realistic models,
4589 which we discuss in Chapt. 12. We consider some other important extensions of the basic

4590 SCR model in later chapters. For example, we consider models that include covariates that
4591 vary by individual, trap, or over time (Chapt. 6), spatial covariates on density (Chapt.
4592 10), open populations (Chapt. 15), and methods for model assessment and selection
4593 (Chapt. 7) among other topics. We also consider technical details of maximum likelihood
4594 (Chapt. 5) and Bayesian (Chapt. 16) estimation, so that the interested reader can develop
4595 or extend methods to suit their own needs.

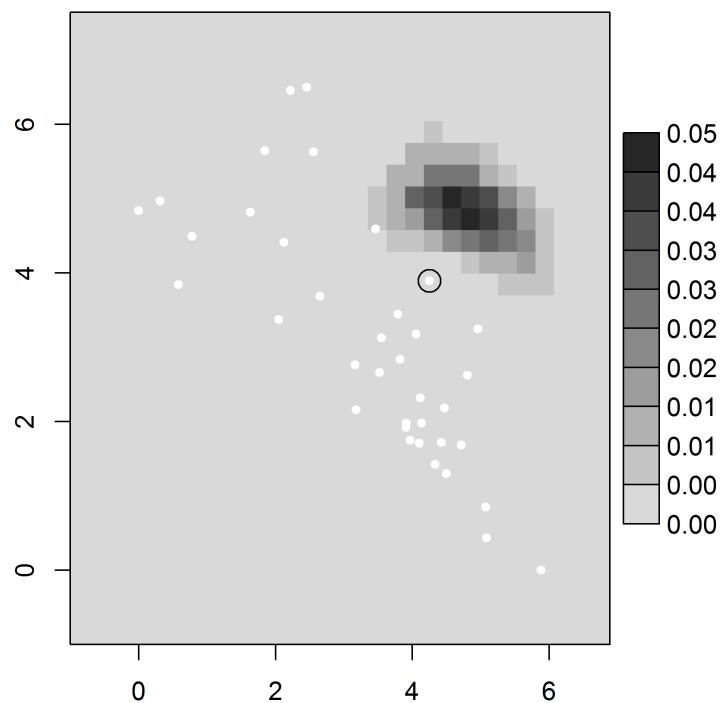


Figure 4.7. Posterior probability distribution of s_1 , the activity center for individual 1 in the wolverine data set. This individual was captured a single time in one trap (trap 30) which is circled. White dots are trap locations.

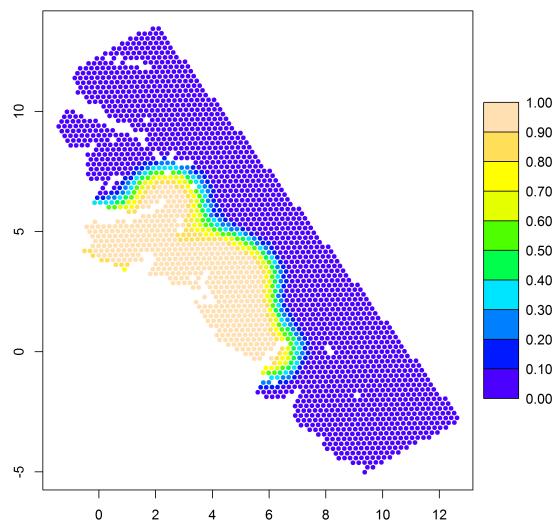


Figure 4.8. Probability of encounter used in computing effective sampled area for the wolverine camera trapping array, using the parameter estimates (posterior means) for the 2×2 km habitat mask.

4596

4597

4598

4599

5

LIKELIHOOD ANALYSIS OF SPATIAL CAPTURE-RECAPTURE MODELS

4600 We have so far mainly focused on Bayesian analysis of spatial capture-recapture models.
4601 And, in the previous chapters we learned how to fit some basic spatial capture-recapture
4602 models using a Bayesian formulation of the models analyzed in **BUGS** engines including
4603 **WinBUGS** and **JAGS**. Despite our focus on Bayesian analysis, it is instructive to de-
4604 velop the basic concepts and ideas behind classical analysis based on likelihood methods
4605 and frequentist inference for SCR models. We recognized earlier (Chapt. 4) that SCR
4606 models are versions of binomial (or other) GLMs, but with random effects (i.e., GLMMs).
4607 Throughout statistics, such models are routinely analyzed by likelihood methods. In par-
4608 ticular, likelihood analysis is based on the integrated or marginal likelihood in which the
4609 random effects are removed, by integration, from the conditional-on-s likelihood (s being
4610 the individual activity center). This has been the approach taken by Borchers and Ef-
4611 ford (2008); Dawson and Efford (2009) and related papers. Therefore, in this chapter, we
4612 provide some conceptual and technical foundation for likelihood-based analysis of spatial
4613 capture-recapture models.

4614 We will show here that it is straightforward to compute the maximum likelihood esti-
4615 mates (MLE) for SCR models by integrated likelihood. We develop the MLE framework
4616 using **R**, and we also provide a basic introduction to the **R** package **secr** (Efford, 2011)
4617 which does likelihood analysis of SCR models (see also the stand-alone program **DEN-**
4618 **SITY** (Efford et al., 2004)). To set the context for likelihood analysis of SCR models,
4619 we first analyze the SCR model when N is known because, in that case, analysis is no
4620 different at all than a standard GLMM. We generalize the model to allow for unknown N
4621 using both conventional ideas based on the “full likelihood” (e.g., Borchers et al., 2002)
4622 and also using a formulation based on data augmentation. We obtain the MLEs for the
4623 SCR model from the wolverine camera trapping study (Magoun et al., 2011) analyzed in
4624 previous chapters to compare/contrast the results.

5.1 MLE WITH KNOWN N

We noted in Chapt. 4 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. 3).

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

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

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

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

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

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

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

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

In most SCR models, $[\mathbf{s}] = 1/A(\mathcal{S})$ where $A(\mathcal{S})$ is the area of the prescribed state-space \mathcal{S} (but see Chapt. 10 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:

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

4654 We emphasize that two independence assumptions are explicit in this development: independence of trap-specific encounters within individuals and also independence among
 4655 individuals. In particular, this would only be valid when individuals are not physically
 4656 restrained or removed upon capture, and when traps do not “fill up.”

4658 The key operation for computing the likelihood is solving a 2-dimensional integration
 4659 problem. There are some general purpose **R** packages that implement a number of multi-
 4660 dimensional integration routines including **adapt** (Genz et al., 2007) and **R2cuba** (Hahn
 4661 et al., 2010). In practice, we won’t rely on these extraneous **R** packages (except see Chapt.
 4662 10 for an application of **R2cuba**) but instead will use perhaps less efficient methods in which
 4663 we replace the integral with a summation over an equal area mesh of points on the state-
 4664 space \mathcal{S} and explicitly evaluate the integrand at each point. We invoke the rectangular
 4665 rule for integration here¹ in which we evaluate the integrand on a regular grid of points
 4666 of equal area and compute the average of the integrand over that grid of points. Let
 4667 $u = 1, 2, \dots, nG$ index a grid of nG points, \mathbf{s}_u , where the area of grid cells is constant, say
 4668 A . In this case, the integrand, i.e., the marginal pmf of \mathbf{y}_i , is approximated by

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

4669 This is a specific case of the general expression that could be used for approximating
 4670 the integral for any arbitrary distribution $[\mathbf{s}]$. The general case is

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

4671 Under the uniformity assumption, $[\mathbf{s}] = 1/A(\mathcal{S})$ and thus the grid-cell area cancels in the
 4672 above expression to yield Eq. 5.1.2. The rectangular rule for integration can be seen as
 4673 an application of the Law of Total Probability for a discrete random variable \mathbf{s} , having
 4674 nG unique values with equal probabilities $1/nG$.

4675 5.1.1 Implementation (simulated data)

4676 Here we will illustrate how to carry out this integration and optimization based on the
 4677 integrated likelihood using simulated data (i.e., see Sec. 4.5). Using **simSCR0** we simulate
 4678 data for 100 individuals and an array of 25 traps laid out in a 5×5 grid of traps having unit
 4679 spacing. The specific encounter model is the Gaussian model. The 100 activity centers
 4680 were simulated on a state-space defined by an 8×8 square within which the trap array was
 4681 centered (thus the trap array is buffered by 2 units). Therefore, the density of individuals
 4682 in this system is fixed at 100/64. In the following set of **R** commands we generate the
 4683 data and then harvest the required data objects:

```
4684 ## simulate a complete data set (perfect detection)
4685 > data <- simSCR0(discard0=FALSE, rnd=2013)
4686     ## extract the objects that we need for analysis
4687 > y <- data$Y
```

¹e.g., http://en.wikipedia.org/wiki/Rectangle_method

```

4688 > traplocs <- data$traplocs
4689 > nind <- nrow(y) ## in this case nind=N
4690 > J <- nrow(traplocs)
4691 > K <- data$K
4692 > xlim <- data$xlim
4693 > ylim <- data$ylim

```

4694 Now, we need to define the integration grid, say **G**, which we do with the following set of
 4695 **R** commands (here, **delta** is the grid spacing):

```

4696 > delta <- .2
4697 > xg <- seq(xlim[1]+delta/2,xlim[2]-delta/2,by=delta)
4698 > yg <- seq(ylim[1]+delta/2,ylim[2]-delta/2,by=delta)
4699 > npix <- length(xg)           # valid for square state-space only
4700 > G <- cbind(rep(xg,npix),sort(rep(yg,npix)))
4701 > nG <- nrow(G)

```

4702 In this case, the integration grid is set up as a grid with spacing $\delta = 0.2$ which produces,
 4703 for our example, a 40×40 grid of points for evaluating the integrand if the state-space
 4704 buffer is set at 2. We note that the integration grid is set-up here to correspond exactly
 4705 to the state-space used in simulating the data. However, in practice, we wouldn't know
 4706 this, and our estimate of N (for the unknown case, see below) would be sensitive to choice
 4707 of the extent of the integration grid. As we've discussed previously, density, which is N
 4708 standardized by the area of the state-space, will not be so sensitive in most cases.

4709 We are now ready to compute the conditional-on-s likelihood and carry out the
 4710 marginalization described by Eq. 5.1.2. We need to do this by defining an **R** function
 4711 that computes the likelihood for the integration grid, as a function of the data objects
 4712 **y** and **traplocs** which were created above. However, it is a bit untidy to store the grid
 4713 information in your workspace, and define the likelihood function in a way that depends
 4714 on these things that exist in your workspace. Therefore, we build the **R** function so that
 4715 it computes the integration grid *within* the function, thereby avoiding potential problems
 4716 if our trapping grid locations change, or if we want to modify the state-space buffer easily.
 4717 We therefore define the function, called **intlik1**, to which we pass the data objects and
 4718 other information necessary to compute the marginal likelihood. This function is available
 4719 in the **scrbook** package (use **?intlik1** at the **R** prompt). The code is reproduced here:

```

4720 intlik1 <- function(parm,y=y,X=traplocs, delta=.2, ssbuffer=2){
4721
4722   Xl <- min(X[,1]) - ssbuffer ## These lines of code are setting up the
4723   Xu <- max(X[,1]) + ssbuffer ## support for the integration which is
4724   Yu <- max(X[,2]) + ssbuffer ## the same as the state-space of "s"
4725   Yl <- min(X[,2]) - ssbuffer
4726   xg <- seq(Xl+delta/2,Xu-delta/2,,length=npix)
4727   yg <- seq(Yl+delta/2,Yu-delta/2,,length=npix)
4728   npix<- length(xg)
4729
4730   G <- cbind(rep(xg,npix),sort(rep(yg,npix)))

```

```

4731   nG <- nrow(G)
4732   D <- e2dist(X,G)
4733
4734   alpha0 <- parm[1]
4735   alpha1 <- exp(parm[2]) # alpha1 restricted to be positive here
4736
4737   probcap <- plogis(alpha0)*exp(-alpha1*D*D)
4738   Pm <- matrix(NA,nrow=nrow(probcap),ncol=ncol(probcap))
4739           # Frequency of all-zero encounter histories
4740   n0 <- sum(apply(y,1,sum)==0)
4741           # Encounter histories with at least 1 detection
4742   ymat <- y[apply(y,1,sum)>0,]
4743   ymat <- rbind(ymat,rep(0,ncol(ymat)))
4744   lik.marg <- rep(NA,nrow(ymat))
4745
4746   for(i in 1:nrow(ymat)){
4747       ## Next line: log conditional likelihood for ALL possible values of s
4748       Pm[1:length(Pm)] <- dbinom(rep(ymat[i,],nG),K,probcap[1:length(Pm)],
4749           log=TRUE)
4750       ## Next line: sum the log conditional likelihoods, exp() result
4751       ## same as taking the product
4752       lik.cond <- exp(colSums(Pm))
4753       ## Take the average value == computing marginal
4754       lik.marg[i] <- sum(lik.cond*(1/nG))
4755   }
4756   ## n0 = number of all-0 encounter histories
4757   nv <- c(rep(1,length(lik.marg)-1),n0)
4758   return( -1*(sum(nv*log(lik.marg)) ) )
4759 }
```

4760 We emphasize that this function (and subsequent) are not meant to be general-purpose
4761 routines for solving all of your SCR problems but, rather, they are meant for illustrative
4762 purposes – so you can see how the integrated likelihood is constructed and how we connect
4763 it to data and other information that is needed.

4764 The function `intlik1` accepts as input the encounter history matrix, `y`, the trap locations,
4765 `X`, and the state-space buffer. This allows us to vary the state-space buffer and easily
4766 evaluate the sensitivity of the MLE to the size of the state-space. Note that we have a
4767 peculiar handling of the encounter history matrix `y`. In particular, we remove the all-zero
4768 encounter histories from the matrix and tack-on a single all-zero encounter history as the
4769 last row which then gets weighted by the number of such encounter histories (`n0`). This is
4770 a bit long-winded and strictly unnecessary when N is known, but we did it this way be-
4771 cause the extension to the unknown- N case is now transparent (as we demonstrate in the
4772 following section). The matrix `Pm` holds the log-likelihood contributions of each encounter
4773 frequency for each possible state-space location of the individual. The log contribu-
4774 tions are summed up and the result exponentiated on the next line, producing `lik.cond`, the
4775 conditional-on- s likelihood (Eq. 5.1.1 above). The marginal likelihood (`lik.marg`) sums
4776 up the conditional elements weighted by the probabilities [s] (Eq. 5.1.2 above).

4777 This is a fairly primitive function which doesn't allow much flexibility in the data
 4778 structure. For example, it assumes that K , the number of replicates, is constant for each
 4779 trap. Further, it assumes that the state-space is a square. We generalize this to some
 4780 extent later in this chapter.

4781 Here is the **R** command for maximizing the likelihood using **nlm** (the function **optim**
 4782 could also be used) and saving the results into an object called **frog**. The output is a list
 4783 of the following structure and these specific estimates are produced using the simulated
 4784 data set:

```
4785 # should take 15-30 seconds
4786
4787 > starts <- c(-2,2)
4788 > frog <- nlm(intlik1,starts,y=y,X=traplocs,delta=.1,ssbuffer=2,hessian=TRUE)
4789 > frog
4790
4791 $minimum
4792 [1] 297.1896
4793
4794 $estimate
4795 [1] -2.504824  2.373343
4796
4797 $gradient
4798 [1] -2.069654e-05  1.968754e-05
4799
4800 $hessian
4801      [,1]      [,2]
4802 [1,]  48.67898 -19.25750
4803 [2,] -19.25750  13.34114
4804
4805 $code
4806 [1] 1
4807
4808 $iterations
4809 [1] 11
```

4810 Details about this output can be found on the help page for **nlm**. We note briefly that
 4811 **frog\$minimum** is the negative log-likelihood value at the MLEs, which are stored in the
 4812 **frog\$estimate** component of the list. The order of the parameters is as they are defined
 4813 in the likelihood function so, in this case, the first element (value = -2.504824) is the
 4814 logit transform of p_0 and the second element (value = 2.373343) is the value of α_1 the
 4815 “coefficient” on distance-squared. The Hessian is the observed Fisher information matrix,
 4816 which can be inverted to obtain the variance-covariance matrix using the command:

```
4817 > solve(frog$hessian)
```

4818 It is worth drawing attention to the fact that the estimates are slightly different than
 4819 the Bayesian estimates reported previously in Sec. 4.6. There are several reasons for this.
 4820 First Bayesian inference is based on the posterior distribution and it is not generally the

4821 case that the MLE should correspond to any particular value of the posterior distribution.
 4822 If the prior distributions in a Bayesian analysis are uniform, then the (multivariate) mode
 4823 of the posterior is the MLE, but note Bayesians almost always report posterior *means*
 4824 and so there will typically be a discrepancy there. Secondly, we have implemented an
 4825 approximation to the integral here and there might be a slight bit of error induced by
 4826 that. We will evaluate that shortly. Third, the Bayesian analysis by MCMC is itself
 4827 subject to some amount of Monte Carlo error which the analyst should always be aware of
 4828 in practical situations. All of these different explanations are likely responsible for some
 4829 of the discrepancy. Accounting for these, we see general consistency between the two
 4830 estimates.

4831 In summary, for the basic SCR model, computing the integrated likelihood is a simple
 4832 task when N is known. Even for N unknown it is not too difficult, and we will do that
 4833 shortly. However, if you can solve the known- N problem then you should be able to do a
 4834 real analysis, for example by considering different values of N and computing the results
 4835 for each value and then making a plot of the log-likelihood or AIC and choosing the value
 4836 of N that produces the best log-likelihood or AIC. As a homework problem we suggest
 4837 that you can take the code given above and try to estimate N without modifying the
 4838 code by just repeatedly applying it for different values of N in attempt to deduce the best
 4839 value. We will formalize the unknown- N problem next.

5.2 MLE WHEN N IS UNKNOWN

4840 Here we build on the previous introduction to integrated likelihood but we consider now
 4841 the case in which N is unknown. We will see that adapting the analysis based on the
 4842 known- N model is straightforward for the more general problem. The main distinction is
 4843 that we don't observe the all-zero encounter history so we have to make sure we compute
 4844 the probability for that encounter history, which we do by tacking a row of zeros onto the
 4845 encounter history matrix. In addition, we include the number of such all-zero encounter
 4846 histories (that is, the number of individuals *not* encountered) as an unknown parameter of
 4847 the model. Call that unknown quantity n_0 , so that $N = n_0 + n$ where n is the number of
 4848 unique individuals encountered. We will usually parameterize the likelihood in terms of n_0
 4849 because optimization over a parameter space in which $\log(n_0)$ is unconstrained is preferred
 4850 to a parameter space in which N must be constrained $N \geq n$. With n_0 unknown, we have
 4851 to be sure to include a combinatorial term to account for the fact that, of the n observed
 4852 individuals, there are $\binom{N}{n}$ ways to realize a sample of size n . The combinatorial term
 4853 involves the unknown n_0 and thus it must be included in the likelihood. In evaluating the
 4854 log-likelihood, we have to compute terms such as the log-factorial, $\log(N!) = \log((n_0+n)!)$.
 4855 We do this in **R** by making use of the log-gamma function (`lgamma`) and the identity

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

4856 Therefore, to compute the likelihood, we require the following 3 components: (1) The
 4857 marginal probability of each \mathbf{y}_i as before,

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

4858 (2) We compute the probability of an all-0 encounter history:

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

4859 (3) The combinatorial term: $\binom{N}{n}$. Then, the marginal likelihood has this form:

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

4860 This is discussed in Borchers and Efford (2008, p. 379) as the conditional-on- N form of the
4861 likelihood – we also call it the “binomial form” of the likelihood because of its appearance.

4862 Operationally, things proceed much as before: We compute the marginal probability
4863 of each observed \mathbf{y}_i , i.e., by removing the latent \mathbf{s}_i by integration. In addition, we com-
4864 pute the marginal probability of the “all-zero” encounter history \mathbf{y}_{n+1} , and make sure to
4865 weight it n_0 times. We accomplish this by “padding” the data set with a single encounter
4866 history having $y_{n+1,j} = 0$ for all traps $j = 1, 2, \dots, J$. Then we be sure to include the
4867 combinatorial term in the likelihood or log-likelihood computation. We demonstrate this
4868 shortly. To analyze a specific case, we’ll simulate our fake data set (simulated using the
4869 parameters given above). To set some things up in our workspace we do this:

```
4870 ## Obtain a simulated data set
4871 > data <- simSCRO(discard0=TRUE, rnd=2013)
4872
4873 ## Extract the items we need for analysis
4874 > y <- data$Y
4875 > nind <- nrow(y)
4876 > traplocs <- data$traplocs
4877 > J <- nrow(traplocs)
4878 > K <- data$K
```

4879 Recall that these data are simulated by default with $N = 100$, on an 8×8 unit state-
4880 space representing the trap locations buffered by 2 units, although you can modify the
4881 simulation script easily.

4882 As before, the likelihood is defined in the **R** workspace as an **R** function, **intlik2**,
4883 which takes an argument being the unknown parameters of the model and additional
4884 arguments as prescribed. In particular, we provide the encounter history matrix **y**, the
4885 trap locations **traplocs**, the spacing of the integration grid (argument **delta**) and the
4886 state-space buffer. Here is the new likelihood function:

```
4887 intlik2 <- function(parm,y=y,X=traplocs,delta=.3,ssbuffer=2){
4888
4889   Xl <- min(X[,1]) - ssbuffer
4890   Xu <- max(X[,1]) + ssbuffer
4891   Yu <- max(X[,2]) + ssbuffer
4892   Yl <- min(X[,2]) - ssbuffer
4893
4894   xg <- seq(Xl+delta/2,Xu-delta/2,delta)
```

```

4895   yg <- seq(Yl+delta/2,Yu-delta/2,delta)
4896   npix.x <- length(xg)
4897   npix.y <- plength(yg)
4898   area <- (Xu-Xl)*(Yu-Yl)/((npix.x)*(npix.y))
4899   G <- cbind(rep(xg,npix.y),sort(rep(yg,npix.x)))
4900   nG <- nrow(G)
4901   D <- e2dist(X,G)
4902   # extract the parameters from the input vector
4903   alpha0 <- parm[1]
4904   alpha1 <- exp(parm[2])
4905   n0 <- exp(parm[3]) # note parm[3] lives on the real line
4906   probcap <- plogis(alpha0)*exp(-alpha1*D*D)
4907   Pm <- matrix(NA,nrow=nrow(probcap),ncol=ncol(probcap))
4908   ymat <- rbind(y,rep(0,ncol(y)))
4909
4910   lik.marg <- rep(NA,nrow(ymat))
4911   for(i in 1:nrow(ymat)){
4912     Pm[1:length(Pm)] <- (dbinom(rep(ymat[i,],nG),K,probcap[1:length(Pm)],
4913                                     log=TRUE))
4914     lik.cond <- exp(colSums(Pm))
4915     lik.marg[i] <- sum(lik.cond*(1/nG) )
4916   }
4917   nv <- c(rep(1,length(lik.marg)-1),n0)
4918   ## part1 here is the combinatorial term.
4919   ## math: log(factorial(N)) = lgamma(N+1)
4920   part1 <- lgamma(nrow(y)+n0+1) - lgamma(n0+1)
4921   part2 <- sum(nv*log(lik.marg))
4922   return( -1*(part1+ part2) )
4923 }
```

4924 To execute this function for the data that we created with `simSCR0`, we execute the
 4925 following command (saving the result in our friend `frog`). This results in the usual output,
 4926 including the parameter estimates, the gradient, and the numerical Hessian which is useful
 4927 for obtaining asymptotic standard errors (see below):

```

4928 > starts <- c(-2.5,0,4)
4929 > frog <- nlm(intlik2,starts,hessian=TRUE,y=y,X=traplocs,delta=.2,ssbuffer=2)
4930
4931 Warning message:
4932 In nlm(intlik2, starts, hessian = TRUE, y = y, X = traplocs, delta = 0.2, :
4933 NA/Inf replaced by maximum positive value
4934
4935 > frog
4936 $minimum
4937 [1] 113.5004
4938
4939 $estimate
```

```

4940 [1] -2.538333 0.902807 4.232810
4941
4942 [... additional output deleted ...]

4943 Executing nlm here usually produces one or more R warnings due to numerical calculations
4944 happening on extremely small or large numbers (calculation of  $p$  near the edge of the
4945 state-space), and they also happen if a poor parameterization is used which produces
4946 evaluations of the objective function beyond the boundary of the parameter space (e.g.,
4947  $n_0 < 0$ ). Such numerical warnings can often be minimized or avoided altogether by picking
4948 judicious starting values of parameters or properly transforming or scaling the parameters
4949 but, in general, they can be ignored. You will see from the nlm output that the algorithm
4950 performed satisfactory in minimizing the objective function. The estimate of population
4951 size,  $\hat{N}$ , for the state-space (using the default state-space buffer) is

4952 > Nhat <- nrow(y) + exp(4.2328) #### This is n + MLE of n0
4953 > Nhat
4954 [1] 110.9099

4955 Which differs from the data-generating value ( $N = 100$ ), as we might expect for a single
4956 realization. We usually will present an estimate of uncertainty associated with this MLE
4957 which we can obtain by inverting the Hessian. Note that  $\text{Var}(\hat{N}) = n + \text{Var}(\hat{n}_0)$ . Since
4958 we have parameterized the model in terms of  $\log(n_0)$  we use the delta method2 described
4959 in Williams et al. (2002, Appendix F4) (see also Ver Hoef, 2012) to obtain the variance
4960 on the scale of  $n_0$  as follows:

4961 > (exp(4.2328)^2)*solve(frog$hessian)[3,3]
4962 [1] 260.2033
4963
4964 > sqrt(260)
4965 [1] 16.12452

4966 Therefore, the asymptotic “Wald-type” confidence interval for  $N$  is  $110.91 \pm 1.96 \times 16.125 =$ 
4967  $(79.305, 142.515)$ . To report this in terms of density, we scale appropriately by the area
4968 of the prescribed state-space which is 64 units of area (i.e., an  $8 \times 8$  square). Our MLE
4969 of  $D$  is  $\hat{D} = 110.91/64 = 1.733$  individuals per square unit. To get the standard error
4970 for  $\hat{D}$  we need to divide the SE for  $\hat{N}$  by the area of the state-space, and so  $\text{SE}(\hat{D}) =$ 
4971  $(1/64) * 16.12452 = 0.252$ .

```

4972 5.2.1 Integrated likelihood under data augmentation

4973 The likelihood analysis developed in the previous sections is based on the likelihood in
4974 which N (or n_0) is an explicit parameter. This is usually called the “full likelihood” or
4975 sometimes “unconditional likelihood” (Borchers et al., 2002) because it is the likelihood
4976 for all individuals in the population, not just those which have been captured, i.e., not that
4977 which is *conditional on capture*. It is also possible to express an alternative unconditional

² We found a good set of notes on the delta approximation on Dr. David Patterson’s ST549 notes: <http://www.math.umt.edu/patterson/549/Delta.pdf>

4978 likelihood using data augmentation, replacing the parameter N with ψ (e.g., see Sec. 7.1.6
 4979 Royle and Dorazio, 2008, for an example). We don't go into detail here, but we note that
 4980 the likelihood under data augmentation is a zero-inflated binomial mixture – precisely an
 4981 occupancy type model (Royle, 2006). Thus, while it is possible to carry out likelihood
 4982 analysis of models under data augmentation, we primarily advocate data augmentation
 4983 for Bayesian analysis.

4984 5.2.2 Extensions

4985 We have only considered basic SCR models with no additional covariates. However,
 4986 in practice, we are interested in covariate effects including “behavioral response”, sex-
 4987 specificity of parameters, and potentially others. Some of these can be added directly to
 4988 the likelihood if the covariate is fixed and known for all individuals captured or not. An
 4989 example is a behavioral response, which amounts to having a covariate $x_{ik} = 1$ if individual
 4990 i was captured prior to occasion k and $x_{ik} = 0$ otherwise. For uncaptured individuals,
 4991 $x_{ik} = 0$ for all k . Royle et al. (2011b) called this a global behavioral response because the
 4992 covariate is defined for all traps, no matter the trap in which an individual was captured.
 4993 We could also define a *local* behavioral response which occurs at the level of the trap, i.e.,
 4994 $x_{ijk} = 1$ if individual i was captured in trap j prior to occasion k , etc... Trap-specific
 4995 covariates such as trap type or status, or time-specific covariates such as date, are eas-
 4996 ily accommodated as well. As an example, Kéry et al. (2010) develop a model for the
 4997 European wildcat *Felis silvestris* in which traps are either baited or not (a trap-specific
 4998 covariate with only 2 values), and also encounter probability varies over time in the form
 4999 of a quadratic seasonal response. We consider models with behavioral response or fixed
 5000 covariates in Chapt. 6. The integrated likelihood routines we provided above can be
 5001 modified directly for such cases, which we leave to the interested reader to investigate.

5002 Sex-specificity is more difficult to deal with since sex is not known for uncaptured
 5003 individuals (and sometimes not even for all captured individuals). To analyze such models,
 5004 we do Bayesian analysis of the joint likelihood using data augmentation (Gardner et al.,
 5005 2010b; Russell et al., 2012), discussed further in Chapt. 6. For such covariates (i.e., that
 5006 are not fixed and known for all individuals), it is somewhat more challenging to do MLE
 5007 based on the joint likelihood as we have developed above. Instead it is more conventional
 5008 to use what is colloquially referred to as the “Huggins-Alho” type model which is one of
 5009 the approaches taken in the software package **secr** (Efford, 2011). We introduce the **secr**
 5010 package in Sec. 5.5 below.

5.3 CLASSICAL MODEL SELECTION AND ASSESSMENT

5011 In most analyses, one is interested in choosing from among various potential models, or
 5012 ranking models, or something else to do with assessing the relative merits of a set of
 5013 models. A good thing about classical analysis based on likelihood is we can apply Akaike
 5014 Information Criterion (AIC) methods (Burnham and Anderson, 2002) without difficulty.
 5015 AIC is convenient for assessing the relative merits of these different models although if
 5016 there are only a few models it is not objectionable to use hypothesis tests or confidence
 5017 intervals to determine importance of effects. A second model selection context has to
 5018 do with choosing among various detection models, although, as a general rule, we don't

5019 recommend this application of model selection. This is because there is hardly ever (if at
 5020 all) a rational subject-matter based reason motivating specific distance functions. As a
 5021 result, we believe that doing too much model selection will invariably lead to over-fitting
 5022 and thus over-statement of precision. This is the main reason that we haven't loaded you
 5023 down with a basket of models for detection probability so far, although we discuss many
 5024 possibilities in Chapt. 6.

5025 **Goodness-of-fit or model-checking** – For many standard capture-recapture mod-
 5026 els, it is possible to identify goodness-of-fit statistics based on the multinomial likelihood,
 5027 (Cooch and White, 2006, Chapt. 5), and evaluate model adequacy using formal statistical
 5028 tests. Similar strategies can be applied to SCR models using expected cell-frequencies
 5029 based on the marginal distribution of the observations. Also, because computing MLEs
 5030 is somewhat more efficient in many cases compared to Bayesian analysis, it is sometimes
 5031 feasible to use bootstrap methods. At the present time, we don't know of any applica-
 5032 tions of goodness-of-fit testing for SCR models based on likelihood inference, although we
 5033 discuss the use of Bayesian p-values for assessing model fit in Chapt. 7. An important
 5034 practical problem in trying to evaluate goodness-of-fit is that, in realistic sample sizes, fit
 5035 tests often lack the power to detect departures from the model under consideration and
 5036 so they may not be generally useful in practice.

5.4 LIKELIHOOD ANALYSIS OF THE WOLVERINE CAMERA TRAPPING DATA

5037 Here we compute the MLEs for the wolverine data using an expanded version of the
 5038 function we developed in the previous section. To accommodate that each trap might
 5039 be operational a variable number of nights, we provided an additional argument to the
 5040 likelihood function (allowing for a vector $\mathbf{K} = (K_1, \dots, K_J)$), which requires also a modifi-
 5041 cation to the construction of the likelihood. In addition, we accommodate the state-space
 5042 is a general rectangle, and we included a line in the code to compute the state-space
 5043 area which we apply below for computing density. The more general function (`intlik3`)
 5044 is given in the **R** package `scrbook`. Incidentally, this function also returns the area of
 5045 the state-space for a given set of parameter values, as an attribute to the function value,
 5046 which will be used in converting \hat{N} to \hat{D} . To use this function to obtain the MLEs for the
 5047 wolverine camera trap study, we execute the following commands (note: these are in the
 5048 help file and will execute if you type `example(intlik3)`):

```
5049 > library(scrbook)
5050 > data(wolverine)
5051
5052 > traps <- wolverine$wtraps
5053 > traplocs <- traps[,2:3]/10000
5054 > K.wolv <- apply(traps[,4:ncol(traps)],1,sum)
5055
5056 > y3d <- SCR23darray(wolverine$wcaps,traps)
5057 > y2d <- apply(y3d,c(1,2),sum)
5058
5059 > starts <- c(-1.5,0,3)
5060
```

```

5061 > wolv <- nlm(intlik3,starts,hessian=TRUE,y=y2d,K=K.wolv,X=traplocs,
5062           delta=.2,ssbuffer=2)
5063
5064 > wolv
5065 $minimum
5066 [1] 220.4313
5067
5068 $estimate
5069 [1] -2.8176120 0.2269395 3.5836875
5070
5071 [.... output deleted ....]

```

5072 Of course we're interested in obtaining an estimate of population size for the prescribed
 5073 state-space, or density, and associated measures of uncertainty which we do using the delta
 5074 method (Williams et al., 2002, Appendix F4). To do all of that we need to manipulate the
 5075 output of `nlm` since we have our estimate in terms of $\log(n_0)$. We execute the following
 5076 commands:

```

5077 > wolv <- nlm(intlik3,starts,hessian=TRUE,y=y2d,K=K.wolv,X=traplocs,delta=.2,
5078           ssbuffer=2)
5079 > Nhat <- nrow(y2d)+exp(wolv$estimate[3])
5080 > area <- attr(intlik3(starts,y=y2d,K=K.wolv,X=traplocs,delta=.2,ssbuffer=2),
5081           "SSarea")
5082 > Dhat <- Nhat/area
5083
5084 > Dhat
5085 [1] 0.5494947
5086
5087 > SE <- (1/area)*exp(wolv$estimate[3])*sqrt(solve(wolv$hessian)[3,3])
5088
5089 > SE
5090 [1] 0.1087073

```

5091 Our estimate of density is 0.55 individuals per “standardized unit” which is 100 km^2 ,
 5092 because we divided UTM coordinates by 10000. So this is about 5.5 individuals per 1000
 5093 km^2 , with a SE of around 1.09 individuals. This compares closely with 5.77 reported in
 5094 Sec. 4.9 based on Bayesian analysis of the model.

5095 5.4.1 Sensitivity to integration grid and state-space buffer

5096 The effect of approximating the integral by a discrete mesh of points is that it induces
 5097 some numerical error in evaluation of the integral and, further, that error increases as the
 5098 coarseness of the mesh increases. To evaluate the effect (or sensitivity) of the integration
 5099 grid spacing, we obtained the MLEs for a state-space buffer of 2 (standardized units) and
 5100 for integration grid with spacing $\delta = .3, .2, .1, .05$. The MLEs for these 4 cases including
 5101 the relative runtime are given in Table 5.1. We see the results change only slightly as the
 5102 integration grid changes. Conversely, the runtime on the platform of the day for the 4 cases

5103 increases rapidly. These runtimes could be regarded in relative terms, across platforms,
 5104 for gaging the decrease in speed as the fineness of the integration grid increases.

Table 5.1. Runtime and MLEs for different integration grid resolutions for the wolverine camera trapping data.

δ	Estimates			
	runtime (sec)	$\hat{\alpha}_0$	$\hat{\alpha}_1$	$\log(n_0)$
0.30	9.9	-2.819786	1.258468	3.569731
0.20	32.3	-2.817610	1.254757	3.583690
0.10	115.1	-2.817570	1.255112	3.599040
0.05	407.3	-2.817559	1.255281	3.607158

5105 We studied the effect of the state-space buffer on the MLEs, using a fixed $\delta = .2$ for
 5106 all analyses. We used state-space buffers of 1 to 4 units stepped by .5. As we can see
 5107 (Table 5.2), the estimates of D stabilize rapidly and the incremental difference is within
 5108 the numerical error associated with approximating the integral.

Table 5.2. Results of the effect of the state-space buffer on the MLE. Given here are the state-space buffer, area of the state-space (area), the MLE of N (\hat{N}) for the prescribed state-space and the corresponding MLE of density (\hat{D}).

Buffer	Area	\hat{N}	\hat{D}
1.0	66.98212	37.73338	0.5633352
1.5	84.36242	46.21008	0.5477567
2.0	103.74272	57.00617	0.5494956
2.5	125.12302	69.03616	0.5517463
3.0	148.50332	82.17550	0.5533580
3.5	173.88362	96.44018	0.5546249
4.0	201.26392	111.83524	0.5556646

5109 5.4.2 Using a habitat mask (Restricted state-space)

5110 In Sec. 4.10 we used a discrete representation of the state-space in order to have control
 5111 over its extent and shape. This makes it easy to do things like clip out non-habitat, or
 5112 create a *habitat mask* which defines suitable habitat. Clearly that formulation of the model
 5113 is relevant to the calculation of the marginal likelihood in the sense that the discrete state-
 5114 space is equivalent to the integration grid. Thus, for example, we could easily compute
 5115 the MLE of parameters under some model with a restricted state-space merely by creating
 5116 the required state-space at whatever grid resolution is desired, and then inputting that
 5117 state-space into the likelihood function above, instead of computing it within the function.
 5118 We can easily create an explicit state-space grid for integration from arbitrary polygons or
 5119 GIS shapefiles which we demonstrate here. Our approach is to create the integration grid
 5120 (or state-space grid) outside of the likelihood evaluation, and then determine which points
 5121 of the grid lie in the polygon defined by the shapefile using functions in the **R** packages **sp**
 5122 and **maptools**. For each point in the state-space grid (object **G** in the code below which is

5123 assumed to exist), we determine whether it is inside the polygon³, identifying such points
 5124 with a value of `mask=1` and `mask=0` for points that are *not* in the polygon. We load the
 5125 shapefile which originates by an application of the `readShapeSpatial` function. We have
 5126 saved the result into an **R** data object called `SSp` which is in the `scrbook` package. Here
 5127 are the **R** commands for doing this (see the helpfile `?intlik4`):

```
5128 > library(maptools)
5129 > library(sp)
5130 > library(scrbook)
5131
5132 ##### If we have the .shp file in place, we would use this command:
5133 ##### SSp <- readShapeSpatial('Sim_Polygon.shp')
5134 ##### The object SSp is in data(fakeshapefile)
5135 > data(fakeshapefile)
5136 > Pcoord <- SpatialPoints(G)
5137 > PinPoly <- over(Pcoord,SSp)  ### determine if each point is in polygon
5138 > mask <- as.numeric(!is.na(PinPoly[,1])) ## convert to binary 0/1
5139 > G <- G[mask==1,]
```

5140 We created the function `intlik4` which accepts the integration grid as an explicit argument,
 5141 and this function is also available in the package `scrbook`.

5142 We apply this modification to the wolverine camera trapping study. Royle et al.
 5143 (2011b) created 2, 4 and 8 km state-space grids so as to remove “non-habitat” (mostly
 5144 ocean, bays, and large lakes). We previously analyzed the model using **JAGS** and **Win-**
BUGS in Chapt. 4. To set up the wolverine data and fit the model using maximum
 5145 likelihood we execute the following commands:

```
5147 > library(scrbook)
5148 > data(wolverine)
5149
5150 > traps <- wolverine$wtraps
5151 > traplocs <- traps[,2:3]/10000
5152 > K.wolv <- apply(traps[,4:ncol(traps)],1,sum)
5153
5154 > y3d <- SCR23darray(wolverine$wcaps,traps)
5155 > y2d <- apply(y3d,c(1,2),sum)
5156 > G <- wolverine$grid2/10000
5157
5158 > starts <- c(-1.5,0,3)
5159 > wolv <- nlm(intlik4, starts, y=y2d, K=K.wolv, X=traplocs, G=G)
```

³We perform this check using the `over` function. This function takes as its second argument (among others) an object of the class “`SpatialPolygons`” or “`SpatialPolygonsDataFrame`”, which can hold additional information for each polygon, and the output value of the function differs slightly for these two classes: if using a “`SpatialPolygons`” object, the function returns a vector of length equal to the number of points (e.g., in the example above), but if using a “`SpatialPolygonsDataFrame`” it returns a data frame (e.g., see Sec. 16.5 in Chapt. 16). If you use the `over` function, make sure you know the class of your second argument so that when processing the function output you index it correctly.

Table 5.3. MLEs for the wolverine camera trapping data using 2, 4 and 8 km state-space grids.

grid	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

```

5160
5161 > wolv
5162
5163 $minimum
5164 [1] 225.8355
5165
5166 $estimate
5167 [1] -2.9955424 0.2350885 4.1104757
5168
5169 [... some output deleted ...]

```

5170 Next we convert the parameter estimates to estimates of total population size for the
 5171 prescribed state-space, and then obtain an estimate of density (per 1000 km²) using the
 5172 area computed as the number of pixels in the state-space grid, G, multiplied by the area
 5173 per grid cell. In the present case (the calculation above) we used a state-space grid with 2
 5174 km × 2 km pixels. Finally, we compute a standard errors using the delta approximation:

```

5175 > area <- nrow(G)*4
5176 # Nhat = n (observed) + MLE of n0 (not observed)
5177 > Nhat <- 21 + exp(wolv$estimate[3])
5178 > SE <- exp(wolv$estimate[3])*sqrt(solve(wolv$hessian)[3,3])
5179 > D <- (Nhat/(nrow(G)*area))*1000
5180 > SE.D <- (SE/(nrow(G)*area))*1000

```

5181 We did this for each the 2 km, 4 km and 8 km state-space grids which produced the
 5182 estimates summarized in Table 5.3. These estimates compare with the 8.6 (2 km grid)
 5183 and 8.2 (8 km grid) reported in Royle et al. (2011b) based on a clipped state-space as
 5184 described in Sec. 4.10.

5.5 DENSITY AND THE R PACKAGE SECR

5185 **DENSITY** is a software program developed by Efford (2004) for fitting spatial capture-
 5186 recapture models based mostly on classical maximum likelihood estimation and related
 5187 inference methods. Efford (2011) has also released an **R** package called **secr**, that contains
 5188 much of the functionality of **DENSITY** but also incorporates new models and features.
 5189 Here, we briefly introduce the **secr** package which we prefer to use over **DENSITY**, be-
 5190 cause it allows us to remain in the **R** environment for data processing and summarization.
 5191 We provide a brief introduction to **secr** and some of its capabilities here, and we also use

5192 it for doing some analysis in other parts of this book. We believe that **secr** will be sufficient
 5193 for many (if not most) of the SCR problems that one might encounter. It provides
 5194 a flexible analysis platform, with a large number of summary features, and “publication
 5195 ready” output. Its user-interface is clean and intuitive to **R** users, and it has been stable,
 5196 efficient and reliable in the (fairly extensive) evaluations that we have done.

5197 To install and run models in **secr**, you must download the package and load it in **R**.

```
5198 > install.packages("secr")
5199 > library(secr)
```

5200 **secr** allows the user to simulate data and fit a suite of models with various detection functions
 5201 and covariate responses. It also contains a number of helpful constructor functions
 5202 for creating objects of the proper class that are recognized by other **secr** functions. We
 5203 provide a brief overview of the capabilities here, but the **secr** help manual can be accessed
 5204 with the command:

```
5205 > RShowDoc("secr-manual", package = "secr")
```

5206 We note that **secr** has many capabilities that we will not cover or do so only sparingly.
 5207 We encourage you to read through the manual, the extensive documentation, and the
 5208 vignettes, in order to get a better understanding of what the package is capable of. We
 5209 also cover certain capabilities of **secr** in other chapters.

5210 The main model-fitting function in **secr** is called **secr.fit**, which makes use of the
 5211 standard **R** model specification framework with tildes. As an example, the equivalent of
 5212 the basic model SCR0 is fitted as follows:

```
5213 > secr.fit(capturedata, model = list(D ~ 1, g0 ~ 1, sigma ~ 1),
5214   buffer = 20000)
```

5215 where **capturedata** is the object created by **secr** containing the encounter history data
 5216 and the trap information, and the model expression $g0 \sim 1$ indicates the intercept-only (i.e.,
 5217 constant) model. Note that we use p_0 for the baseline encounter probability parameter,
 5218 which is g_0 in **secr** notation. A number of possible models for encounter probability can
 5219 be fitted including both pre-defined variables (e.g., **t** and **b** corresponding to “time” and
 5220 “behavior”), and user-defined covariates of several kinds. For example, to include a global
 5221 behavioral response, this would be written as $g0 \sim b$. The discussion of this (global versus
 5222 local trap-specific behavioral response) and other covariates is developed more in Chapt.
 5223 6. We can also model covariates on density in **secr**, which we discuss in Chapt. 10. It
 5224 is important to note that **secr** requires the buffer distance to be defined in meters and
 5225 density will be returned as number of animals per hectare. Thus to make comparisons
 5226 between **secr** and output from other programs, we will often have to convert the density
 5227 to the same units.

5228 Before we can fit the models, the data must first be packaged properly for **secr**.
 5229 We require data files that contain two types of information: trap layout (location and
 5230 identification information for each trap), which is equivalent to the trap deployment file
 5231 (TDF) described in Sec. 4.9 and the capture data file containing sampling *session*, animal
 5232 identification, trap occasion, and trap location, equivalent in information content to the
 5233 encounter data file (EDF). Sample session can be thought of as primary period identifier

5234 in a robust design like framework – it could represent a yearly sample or multiple sample
 5235 periods within a year, each of them producing data on a closed population. We discuss
 5236 “multi-session” models in more detail below, in Sec. 5.5.4 and Chapt. 13.

5237 There are three important constructor functions that help package-up your data for
 5238 use in **secr**: **read.traps**, **make.capthist** and **read.mask**. We provide a brief description
 5239 of each here, but apply them to our wolverine camera trapping data in the next section:

5240 (1) **read.traps**: This function points to an external file *or* **R** data object containing the
 5241 trap coordinates, and other information, and also requires specification of the type of
 5242 encounter devices (described in the next section). A typical application of this function
 5243 looks like the following, invoking the **data=** option when there is an existing **R** object
 5244 containing the trap information:

```
5245 > trapfile <- read.traps(data=traps, detector="proximity")
```

5246 (2) **make.capthist**: This function takes the EDF and combines it with trap information,
 5247 and the number of sampling occasions. A typical application looks like this:

```
5248 > capturedata <- make.capthist(enc.data, trapfile, fmt="trapID",  

  5249 noccasions=165)
```

5250 See **?make.capthist** for definition of distinct file formats. Specifying **fmt = trapID** is
 5251 equivalent to our EDF format.

5252 (3) **read.mask**: If there is a habitat mask available (as described in sec. 5.4.2), then this
 5253 function will organize it so that **secr.fit** knows what to do with it. The function
 5254 accepts either an external file name (see **?read.mask** for details of the structure) or a
 5255 $nG \times 2$ **R** object, say **mask.coords**, containing the coordinates of the mask. A typical
 5256 application looks like the following:

```
5257 > grid <- read.mask(data=mask.coords)
```

5258 These constructor functions produce output that can then be used in the fitting of models
 5259 using **secr.fit**.

5260 5.5.1 Encounter device types and detection models

5261 The **secr** package requires that you specify the type of encounter device. Instead of
 5262 describing models by their statistical distribution (Bernoulli, Poisson, etc..), **secr** uses
 5263 certain operational classifications of detector types including ‘proximity’, ‘multi’, ‘single’,
 5264 ‘polygon’ and ‘signal’. For camera trapping/hair snares we might consider ‘proximity’
 5265 detectors or ‘count’ detectors. The ‘proximity’ detector type allows, at most, one detection
 5266 of each individual at a particular detector on any occasion (i.e., it is equivalent to what
 5267 we call the Bernoulli or binomial encounter process model, or model SCR0). The ‘count’
 5268 detector designation allows repeat encounters of each individual at a particular detector
 5269 on any occasion. There are other detector types that one can select such as: ‘polygon’
 5270 detector type which allows for a trap to be a sampled polygon (Royle and Young, 2008)
 5271 which we discuss further in Chapt. 14, and ‘signal’ detector which allows for traps that
 5272 have a strength indicator, e.g., acoustic arrays (Dawson and Efford, 2009). The detector
 5273 types ‘single’ and ‘multi’ refer to traps that retain individuals, thus precluding the ability
 5274 for animals to be captured in other traps during the sampling occasion. The ‘single’ type

5275 indicates trap that can only catch one animal at a time (single-catch traps), while 'multi'
 5276 indicates traps that may catch more than one animal at a time (multi-catch). These are
 5277 both variations of the multinomial encounter models described in Chapt. 8.

5278 As with all SCR models, **secr** fits an encounter probability model ("detection function"
 5279 in **secr** terminology relating the probability of encounter to the distance of a
 5280 detector from an individual activity center. **secr** allows the user to specify one of
 5281 a variety of detection functions including the commonly used half-normal ("Gaus-
 5282 sian"), hazard rate ("Gaussian hazard"), and (negative) exponential models. There
 5283 are 12 different functions as of version 2.3.1 (see Table 6.1 in Chapt. 6), but some
 5284 are only available for simulating data. The different detection functions are defined
 5285 in the **secr** manual and can be found by calling the help function for the detection
 5286 function:

5287 > ?detectfn

5288 Most of the detection functions available in **secr** contain some kind of a scale
 5289 parameter which is usually labeled σ . The units of this parameter default to meters
 5290 in the **secr** output. We caution that the meaning of this parameter depends on
 5291 the specific detection model being used, and it should not be directly compared as
 5292 a measure of home-range size across models. Instead, as we noted in Sec. 4.4 most
 5293 encounter probability models imply a model of space-usage and fitted encounter
 5294 models should be converted to a common currency such as "area used."

5295 5.5.2 Analysis using the **secr** package

5296 To demonstrate the use of the **secr** package, we will show how to do the same
 5297 analysis on the wolverine study as shown in Sec. 4.9. To use the **secr** package, the
 5298 data need to be formatted in a similar but slightly different manner than we use in
WinBUGS.

5299 For example, in Sec. 4.9 we introduced a standard data format for the encounter
 5300 data file (EDF) and trap deployment file (TDF). The EDF shares the same format
 5301 as that used by the **secr** package with 1 row for every encounter observation and
 5302 4 columns representing trap session ('Session'), individual identity ('ID'), sample
 5303 occasion ('Occasion'), and trap identity ('trapID'). For a standard closed population
 5304 study that takes place during a single season, the 'Session' column in our case is
 5305 all 1's, to indicate a single primary sampling occasion. In addition to providing the
 5306 encounter data file (EDF), we must tell **secr** information about the traps, which
 5307 is formated as a matrix with column labels 'trapID', 'x' and 'y', the last two being
 5308 the coordinates of each trap, with additional columns representing the operational
 5309 state of each trap during each occasion (1=operational, 0=not).

5310 We demonstrate these differences now by walking through an analysis of the
 5311 wolverine camera trapping data using **secr**. To read in the trap locations and other
 5312 related information, we make use of the constructor function **read.traps** which also
 5313 requires that we specify the detector type. The detector type is important because

5315 it will determine the likelihood that **secr** will use to fit the model. Here, we have
 5316 selected “proximity” which corresponds to the Bernoulli encounter model in which
 5317 individuals are captured at most once in each trap during each sampling occasion:

```
5318 > library(secr)
5319 > library(scrbook)
5320 > data(wolverine)
5321
5322 > traps <- as.matrix(wolverine$wtraps)
5323 > dimnames(traps) <- list(NULL,c("trapID","x","y",paste("day",1:165,sep="")))
5324 > traps1 <- as.data.frame(traps[,1:3])
5325 > trapfile1 <- read.traps(data=traps1,detector="proximity")
```

5326 Here we note that trap coordinates are extracted from the wolverine data but
 5327 we do *not* scale them. This is because **secr** defaults to coordinate scaling of meters
 5328 which is the extant scaling of the wolverine trap coordinates. Note that we add
 5329 a ‘trapID’ column to the trap coordinates and provide appropriate column labels
 5330 to the ‘traps’ matrix. An important aspect of the wolverine study is that while
 5331 the camera traps were operated over a 165 day period, each trap was operational
 5332 during only a portion of that period. We need to provide the trap operation infor-
 5333 mation which is contained in the columns to the right of the trap coordinates in
 5334 our standard trap deployment file (TDF). Unfortunately, this is less easy to do in
 5335 **secr**⁴, which requires an external file with a single long string of 1’s and 0’s indi-
 5336 cating the days in which each trap was operational (1) or not (0). The **read.traps**
 5337 function will not allow for this information on trap operation if the data exists as
 5338 an **R** object – instead, we can create this external file and then read it back in with
 5339 **read.traps** using these commands:

```
5340 > hold <- rep(NA,nrow(traps))
5341 > for(i in 1:nrow(traps)){
5342 >   hold[i] <- paste(traps[i,4:ncol(traps)],collapse="")
5343 > }
5344 > traps1 <- cbind(traps[,1:3],"usage"=hold)
5345
5346 > write.table(traps1, "traps.txt", row.names=FALSE, col.names=FALSE)
5347 > trapfile2 <- read.traps("traps.txt",detector="proximity")
```

5348 These operations can be accomplished using the function **scr2secr** which is pro-
 5349 vided in the **R** package **scrbook**.

5350 After reading in the trap data, we now need to create the encounter matrix or
 5351 array using the **make.capthist** command, where we provide the capture histories
 5352 in EDF format, which is the existing format of the data input file **wcaps**. In
 5353 creating the capture history, we provide also the trapfile created previously, the
 5354 format (e.g., here EDF format is **fmt= ‘trapID’**), and finally, we provide the
 5355 number of occasions.

⁴as of v. 2.3.1

```

5356 #
5357 # Grab the encounter data file and format it:
5358 #
5359 wolv.dat <- wolverine$wcaps
5360 dimnames(wolv.dat) <- list(NULL,c("Session","ID","Occasion","trapID"))
5361 wolv.dat <- as.data.frame(wolv.dat)
5362 wolvcapt2 <- make.captist(wolv.dat,trapfile2,fmt="trapID",noccasions=165)

```

5363 We also set up a habitat mask using the 2×2 km grid which we used previously
 5364 in the analysis of the wolverine data and then pass the relevant objects to `secr.fit`
 5365 as follows:

```

5366 #
5367 # Grab the habitat mask (2 x 2 km) and format it:
5368 #
5369 gr2 <- (as.matrix(wolverine$grid2))
5370 dimnames(gr2) <- list(NULL,c("x","y"))
5371 gr2 <- read.mask(data=gr2)
5372 #
5373 # To fit the model we use secr.fit:
5374 #
5375 wolv.secr2 <- secr.fit(wolvcapt2,model=list(D ~ 1, g0 ~ 1, sigma ~ 1),
5376                           buffer=20000,mask=gr2)

```

5377 We are using the “proximity detector” model (SCR0), so we do not need to
 5378 make any specifications in the command line because we have specified the detec-
 5379 tor type using the constructor function `read.traps`, except to provide the buffer
 5380 size (in meters). To specify different models, you can change the default model
 5381 $D \sim 1$, $g_0 \sim 1$, $\sigma \sim 1$. We provide all of these commands and additional analyses
 5382 in the `scrbook` package with the function called `secr_wolverine`. Printing the
 5383 output object produces the following (slightly edited):

```

5384 > wolv.secr2
5385
5386 secr 2.3.1, 15:52:45 29 Aug 2012
5387
5388 Detector type      proximity
5389 Detector number     37
5390 Average spacing     4415.693 m
5391 x-range             593498 652294 m
5392 y-range             6296796 6361803 m
5393 N animals           : 21
5394 N detections         : 115
5395 N occasions          : 165
5396 Mask area            : 987828.1 ha
5397
5398 Model                 : D ~ 1 g0 ~ 1 sigma ~ 1

```

```

5399 Fixed (real)      : none
5400 Detection fn     : halfnormal
5401 Distribution      : poisson
5402 N parameters     : 3
5403 Log likelihood    : -602.9207
5404 AIC               : 1211.841
5405 AICc              : 1213.253
5406
5407 Beta parameters (coefficients)
5408          beta   SE.beta      lcl      ucl
5409 D      -9.390124 0.22636698 -9.833795 -8.946452
5410 g0     -2.995611 0.16891982 -3.326688 -2.664535
5411 sigma  8.745547 0.07664648  8.595323  8.895772
5412
5413 Variance-covariance matrix of beta parameters
5414          D       g0       sigma
5415 D      0.0512420110 -0.0004113326 -0.003945371
5416 g0     -0.0004113326  0.0285339045 -0.006269477
5417 sigma -0.0039453711 -0.0062694767  0.005874683
5418
5419 Fitted (real) parameters evaluated at base levels of covariates
5420      link   estimate  SE.estimate      lcl      ucl
5421 D      log 8.354513e-05 1.915674e-05 5.360894e-05 1.301982e-04
5422 g0     logit 4.762453e-02 7.661601e-03 3.466689e-02 6.509881e-02
5423 sigma  log 6.282651e+03 4.822512e+02 5.406315e+03 7.301037e+03

```

5424 The object returned by `secr.fit` provides extensive default output when printed.
5425 Much of this is basic descriptive information about the model, the traps, or the
5426 encounter data. We focus here on the parameter estimates. Under the fitted
5427 (real) parameters, we find D , the density, given in units of individuals/hectare
5428 (1 hectare = 10000 m^2). To convert this into individuals/1000 km 2 , we multiply
5429 by 100000, thus our density estimate is 8.35 individuals/1000 km 2 . The parameter
5430 σ is given in units of meters, and so this corresponds to 6.283 km. Both of these
5431 estimates are very similar to those obtained in our likelihood analysis summarized
5432 in Table 5.3 which, for the 2×2 km grid, we obtained $\hat{D} = 8.31$ with a SE of
5433 $100000 \times 1.915674e - 05 = 1.9156$ and, accounting for the scale difference (1 unit
5434 = 10000 m in the previous analysis), $\hat{\sigma} = \sqrt{1/(2\hat{\alpha}_1)} * 10000 = 6.289$ km. The
5435 difference in the MLE between Table 5.3 and those produced by `secr` could be due
5436 to subtle differences in internal tuning of optimization algorithms, starting values
5437 or other numerical settings. In addition, the likelihood is based on a Poisson prior
5438 for N (see the next section). On the other hand, the SE is slightly larger based on
5439 `secr` which is due to a subtle difference in the interpretation of D under the `secr`
5440 model (See below).

5441 **5.5.3 Likelihood analysis in the `secr` package**

5442 The `secr` package does likelihood analysis of SCR models for most classes of models
 5443 as developed by Borchers and Efford (2008). Their formulation deviates slightly
 5444 from the binomial form we presented in Sec. 5.2 above (though Borchers and
 5445 Efford (2008) also mention the binomial form). Specifically, the likelihood that
 5446 `secr` implements is that based on removing N from the likelihood by integrating
 5447 the binomial likelihood (Eq. 5.2.1 above) over a Poisson prior for N – what we will
 5448 call the *Poisson-integrated likelihood* as opposed to the conditional-on- N (*binomial-*
 5449 *form*) considered previously.

5450 To develop the Poisson-integrated likelihood we compute the marginal proba-
 5451 bility of each \mathbf{y}_i and the probability of an all-0 encounter history, π_0 , as before, to
 5452 arrive at the marginal likelihood in the binomial-form:

$$\mathcal{L}(\boldsymbol{\alpha}, n_0 | \mathbf{y}) = \frac{N!}{n! n_0!} \left\{ \prod_i [\mathbf{y}_i | \boldsymbol{\alpha}] \right\} \pi_0^{n_0}$$

5453 Now, what Borchers and Efford (2008) do is assume that $N \sim \text{Poisson}(\Lambda)$ and they
 5454 do a further level of marginalization over this prior distribution:

$$\sum_{n_0=0}^{\infty} \frac{N!}{n! n_0!} \left\{ \prod_i [\mathbf{y}_i | \boldsymbol{\alpha}] \right\} \pi_0^{n_0} \frac{\exp(-\Lambda) \Lambda^{n_0}}{N!}$$

5455 In Chapt. 10 we write $\Lambda = \mu ||\mathcal{S}||$ where $||\mathcal{S}||$ is the area of the state-space, and μ is
 5456 the density (“intensity”) of the point process. Carrying out the summation above
 5457 produces exactly this marginal likelihood:

$$\mathcal{L}_2(\boldsymbol{\alpha}, \Lambda | \mathbf{y}) = \left\{ \prod_i [\mathbf{y}_i | \boldsymbol{\alpha}] \right\} \Lambda^n \exp(-\Lambda(1 - \pi_0))$$

5458 which is Eq. 2 of Borchers and Efford (2008) except for notational differences. It
 5459 also resembles the binomial-form of the likelihood in Eq. 5.2.1 with $\Lambda^n \exp(-\Lambda \pi_0)$
 5460 replacing the combinatorial term and the $\pi_0^{n_0}$ term. We emphasize there are two
 5461 marginalizations going on here: (1) the integration to remove the latent variables
 5462 \mathbf{s} ; and, (2) summation to remove the parameter N . We provide a function for
 5463 computing this in the `scrbook` package called `intlik3Poisson`. The help file for
 5464 that function shows how to conduct a small simulation study to compare the MLE
 5465 under the Poisson-integrated likelihood with that from the binomial form.

5466 The essential distinction between our MLE and Borchers and Efford as imple-
 5467 mented in `secr` is whether you keep N in the model or remove it by integration over
 5468 a Poisson prior. If you have prescribed a state-space explicitly with a sufficiently
 5469 large buffer, then we imagine there should be hardly any difference at all between
 5470 the MLEs obtained by either the Poisson-integrated likelihood or the binomial-form

of the likelihood which retains N . There is a subtle distinction in the sense that under the binomial form, we estimate the realized population size N for the state-space whereas, for the Poisson-integrated form we estimate the *prior* expected value which would apply to a hypothetical new study of a similar population (see Sec. 4.7.3).

Both models (likelihoods) assume s is uniformly distributed over space, but for the binomial model we make no additional assumption about N whereas we assume N is Poisson using the formulation in **secr** from (Borchers and Efford, 2008). Using data augmentation we could do a similar kind of integration but integrate N over a binomial (M, ψ) prior – which we referred to as the binomial-integrated likelihood in Sec. 3.2.4. So obviously the two approaches (data augmentation and Poisson-integrated likelihood) are approximately the same as M gets large. However, doing a Bayesian analysis by MCMC, we obtain an estimate of both N , the *realized population size*, and the parameter controlling its expected value ψ which are, in fact, both identifiable from the data even using likelihood analysis (Royle et al., 2007). That said we can integrate N out completely and just estimate ψ as we noted in Sec. 5.2.1 above.

5.5.4 Multi-session models in **secr**

In practice we will often deal with SCR data that have some meaningful stratification or group structure. For example, we might conduct mist-netting of birds on K consecutive days, repeated, say, T times during a year, or perhaps over T years. Or we might collect data from R distinct trapping grids. In these cases, we have T or R groups which we might reasonably regard as being samples of independent populations. While the groups might be distinct sites, year, or periods within years, they could also be other biological groups such as sex or age. Conveniently, **secr** fits a specific model for stratified populations – referred to as *multi-session* models. These models build on the Poisson assumption which underlies the integrated likelihood used in **secr** (as described in the previous section). To understand the technical framework, let N_g be the population size of group g and *assume*

$$N_g \sim \text{Poisson}(\Lambda_g).$$

Naturally, we model group-specific covariates on Λ_g :

$$\log(\Lambda_g) = \beta_0 + \beta_1 z_g$$

where z_g is some group-specific covariate such as a categorical index to the group, or a trend variable, or a spatial covariate, such as treatment effect or habitat structure, if the groups represent spatial units. Under this model, we can marginalize *all* N_g parameters out of the likelihood to concentrate the likelihood on the parameters β_0 and β_1 precisely as discussed in the previous section. This Poisson hierarchical model is the basis of the multi-session models in **secr**.

5507 To implement a multi-session model (or stratified population model) in **secr**, we
 5508 provide the relevant stratification information in the ‘Session’ variable of the input
 5509 encounter data file (EDF). If ‘Session’ has multiple values then a “multi-session”
 5510 object is created by default and session-specific variables can be described in the
 5511 model. For example, if the session has 2 values for males and females then we have
 5512 sex-specific densities , and baseline encounter probability p_0 (g_0 in **secr**) by just
 5513 doing this (see Chapt. 7 for the **R** code to set this up):

```
5514 > out <- secr.fit(capdata, model=list(D ~ session, g0 ~ session, sigma~ 1),  

  5515   buffer=20000)
```

5516 More detailed analysis is given in Sec. 7.1 where we fit a number of different
 5517 models and apply methods of model selection to obtain model-averaged estimates
 5518 of density.

5519 We can also easily implement stratified population models in the various **BUGS**
 5520 engines using data augmentation (Converse and Royle, 2012; Royle and Converse,
 5521 in review) which we discuss, with examples, in Chapt. 13.

5522 **5.5.5 Some additional capabilities of secr**

5523 The **secr** package has capabilities to do a complete analysis of SCR data sets,
 5524 including model fitting, selection, and many summary analyses. In the previous
 5525 sections, we’ve given a basic overview, and we do more in later chapters of this
 5526 book. Here we mention a few of these other capabilities that you should know about
 5527 as you use **secr**. Of course, you should skim through the associated documentation
 5528 (**?secr**) to see more of what’s available.

5529 **Alternative observation models**

5530 **secr** fits a wide range of alternative observation models besides the Bernoulli en-
 5531 counter model, including multinomial encounter models for “multi-catch” and “single
 5532 catch” traps, models for sound attenuation from acoustic detection devices, and
 5533 many others. We discuss many of these other methods in Chapt. 8 and elsewhere
 5534 in the book.

5535 **Summary statistics**

5536 **secr** provides a useful default summary of the data, but it also has summary
 5537 statistics about animal movement including mean-maximum distance moved (the
 5538 function **MMDM**). For example, see the help page **?MMDM** which lists a number of other
 5539 summary functions which take a **capthist** object:

```
5540 > moves(capthist)  

  5541 > dbar(capthist)  

  5542 > RPSV(capthist)  

  5543 > MMDM(capthist, min.recapt = 1, full = FALSE)  

  5544 > ARL(capthist, min.recapt = 1, plt = FALSE, full = FALSE)
```

5545 The function `moves` returns the observed distances moved, `dbar` returns the average distance moved, `RPSV` produces a measure of dispersion about the home-range center, and `ARL` gives the *Asymptotic Range Length* which is the asymptote of an exponential model fit to the observed range length vs. the number of detections of each individual (Jett and Nichols, 1987).

5550 **State-space buffer**

5551 `secr` will produce a warning if the state-space buffer is chosen too small. For
5552 example, in fitting the wolverine data as in Sec. 5.5.2 but with a 1000 m buffer,
5553 and we see the following warning message:

```
5554 Warning message:  
5555 In secr.fit(wolvcapt2, model=list(D ~ 1, g0 ~ 1, sigma ~ 1), buffer=1000):  
5556   predicted relative bias exceeds 0.01 with buffer = 1000
```

5557 This should cause you to contemplate modifying the state-space buffer if that is a
5558 reasonable thing to do in the specific application.

5559 **Model selection and averaging**

5560 `secr` does likelihood ratio tests to compare nested models using the function `LR.test`.
5561 You can create model selection tables based on AIC or AICc, using the function `AIC`,
5562 and obtain model-averaged parameter estimates using the function `model.average`
5563 (See Chapt. 7 for examples).

5564 **Population closure test**

5565 `secr` has a population closure test with the function `closure.test` which implements
5566 the tests of Stanley and Burnham (1999) or Otis et al. (1978). The function
5567 is used like this:

5568 `closure.test(object, SB = FALSE)`. Here `object` is a capthist object and `SB` is
5569 a logical variable that, if TRUE, produces the Stanley and Burnham (1999) test.

5570 **Density mapping and effective sample area**

5571 `secr` produces likelihood versions of the various summaries of posterior density
5572 and effective sample area that we discussed in Chapt. 4. For example, while `secr`
5573 reports estimates of the expected value of N or density directly in the summary
5574 output from fitting a model, you can use the function `region.N` to produce estimates
5575 of N for any given region. In addition, `secr` has functions for creating maps
5576 of detection contours for individuals traps, or for the entire trap array. See the
5577 function `pdot.contour`, and also `fxi.contour` for computing the 2-dimensional
5578 pdf of the locations of one or more individual activity centers (as in Sec. 4.11.3).
5579 In the context of likelihood analysis, estimation of a random effect s is based on a
5580 plug-in application of Bayes' Rule. When s has a uniform distribution, and we use
5581 a discrete evaluation of the integral, it can be computed simply by renormalizing
5582 the likelihood:

$$[s|y, \theta] = \frac{[y|s, \theta]}{\sum_s [y|s, \theta]}.$$

5583 Any of the `intlik` functions given previously in this chapter can be easily modified
5584 to return the posterior distribution of `s` for any, or all, individuals, or an individual
5585 that is not encountered.

5586 Effective sample area (see Sec. 4.12) can be calculated in `secr` using the func-
5587 tions `esa` and `esa.plot`.

5588 Covariate models

5589 `secr` has many capabilities for modeling covariates. It has a number of built-in
5590 models that allow certain covariates on encounter probability, which we cover to a
5591 large extent in Chapt. 6, and also see Chapt. 7 for more examples. `secr` also allows
5592 covariates to be built into the density model (see Chapt. 10). It has some built
5593 in response surface models, allowing for the fitting of linear or quadratic response
5594 surfaces. This is done by modifying the density model in `secr.fit`. For example,
5595 $D \sim 1$ is a constant density surface, and $D \sim x + y$ fits a linear response surface,
5596 etc.. See the manual `secr-densitysurfaces.pdf` for the details.

5597 There are a number of ways to model your own “custom” covariates (as op-
5598 posed to pre-specified models). One way is to use the `addCovariates` function
5599 and supply it a `mask` or `traps` object along with some “spatialdata.” Or, if you
5600 have covariates at each trap location then it will extrapolate to all points on the
5601 habitat mask. There’s also a method by which the user can create a function of
5602 geographic coordinates, `userDfn`, which seems to provide additional flexibility, al-
5603 though we haven’t used this method. There is a handy function `predictDsurface`
5604 for producing density maps under the specified model for density.

5.6 SUMMARY AND OUTLOOK

5605 In this chapter, we discussed basic concepts related to classical analysis of SCR
5606 models based on likelihood methods. Analysis is based on the so-called integrated
5607 or marginal likelihood in which the individual activity centers (random effects) are
5608 removed from the conditional-on-`s` likelihood by integration. We showed how to
5609 construct the integrated likelihood and fit some simple models in the **R** pro-
5610 gramming language. In addition, likelihood analysis for some broad classes of SCR
5611 models can be accomplished using the **R** library `secr` (Efford, 2011) which we pro-
5612 vided a brief introduction to. In later chapters we provide more detailed analyses
5613 of SCR data using likelihood methods and the `secr` package.

5614 Why or why not use likelihood inference exclusively? For certain specific models,
5615 it is may be more computationally efficient to produce MLEs (for an example see
5616 Chapt. 11). And, likelihood analysis makes it easy to do model-selection by AIC
5617 and compute standard errors or confidence intervals. However, **BUGS** is extremely
5618 flexible in terms of describing models and we can devise models in the **BUGS**
5619 language easily that we cannot fit in `secr`. For example, in Chapt 15 we consider
5620 open population models which are straightforward to develop in **BUGS** but, so
5621 far, there is no available platform for doing MLE of such models. We can also fit

models in **BUGS** that accommodate missing covariates in complete generality (e.g., unobserved sex of individuals), and we can adopt SCR models to include auxiliary data types. For example, we might have camera trapping and genetic data and we can describe the models directly in **BUGS** and fit a joint model (Gopalaswamy et al., 2012b). To do maximum likelihood estimation, we have to write a custom new piece of code for each model⁵ or hope someone has done it for us. You should have some capability to develop your own MLE routines with the tools we provided in this chapter.

⁵Although we may be able to handle multiple survey methods together in **secr** using the multi-session models.

5630
5631

6

5632

MODELING ENCOUNTER PROBABILITY

5633 In previous chapters we showed how to fit basic spatial capture-recapture models
5634 using Bayesian analysis (in **WinBUGS** or **JAGS**; Chapt. 4) or by classical likeli-
5635 hood methods (Chapt. 5 or using **secr**). We covered a suite of possible encounter
5636 models (e.g., the Binomial, Poisson, and Multinomial) for dealing with different
5637 types of sampling. We have not, however, described a general framework for mod-
5638 eling covariates that might influence encounter probability of individuals, traps or
5639 over time. In practice, investigators are invariably concerned with explicit factors
5640 or covariates that might influence variation in parameters. Such covariates include
5641 time (e.g., day of year, or season), behavior (e.g., is there an effect of trapping
5642 on subsequent capture probabilities), sex of the individual, and trap type (e.g.,
5643 various camera types, or different constructions for hair snares). Traditionally, in
5644 the non-spatial capture recapture literature, such models were called “model M_t ”,
5645 “model M_h ”, or “model M_b ”, identifying models that account for variation in de-
5646 tected probability as a function of time, “individual heterogeneity” or “behavior”,
5647 where behavior describes whether or not an individual had been previously cap-
5648 tured. In SCR models, more complex covariate models are possible because we
5649 might also have trap-specific covariates, or covariates that vary spatially over the
5650 landscape, and because we generally have more than one parameter describing the
5651 detection function: Most detection functions include a baseline encounter rate (λ_0)
5652 or probability (p_0) parameter, and a shape parameter (σ), which takes on differ-
5653 ent interpretations depending on the specific encounter probability function under
5654 consideration.

5655 In this chapter, we generalize the basic SCR model to accommodate both alter-
5656 native detection functions as well as many different kinds of covariates. We focus on
5657 the binomial observation model used throughout Chaps. 4 and 5 and the Gaussian
5658 encounter model (also called the “half-normal” model in the distance sampling liter-

ature), but the extension to other observation models is straightforward (and other encounter probability models with different functions of distance are considered in Sec. 6.1). Specifically, we consider three distinct types of covariates – those which are fixed, partially observed or completely unobserved (latent). Fixed covariates are those that are fully observed; for example, the date of all sampling occasions. Partially observed covariates are those which are not known for all observations; for example, the sex of an individual cannot always be determined from photos taken during camera trapping. Even if we are able to observe the sex of all individuals sampled, we cannot know it for those individuals never observed during the study. And finally, unobserved covariates are those which we cannot observe at all, for example, the home range size of individuals, or unstructured random “individual effects”.

We will see that models containing these different types of covariates are relatively easy to describe in **WinBUGS** or **JAGS**, and therefore to analyze using Bayesian analysis of the joint likelihood based on data augmentation thus providing a coherent and flexible framework for inference for all classes of SCR models. Throughout the chapter, we will continue to develop the analysis of the black bear study introduced in Chapt. 3, using the software **JAGS**. We also consider the likelihood analysis of many of these models; to do so, we will demonstrate the use of the **R** package **secr** and how to do model comparison with AIC (Sec. 6.4 at the end of the chapter). There are other types of covariates that we do *not* cover in this chapter; for example, covariates that vary across the landscape might affect density, and we consider these covariates in Chapt. 10. Alternatively, these landscape covariates might affect the way individuals use space. There are probably very few circumstances under which animals use all space uniformly and we develop more realistic models of encounter probability in which covariates affect space usage in Chapt. 11.

6.1 ENCOUNTER PROBABILITY MODELS

In Chapt. 4, we developed a basic spatial capture recapture model using a standard detection probability function based on the kernel of a normal (Gaussian) probability distribution:

$$p_{ij} = p_0 \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

where $||\mathbf{x}_j - \mathbf{s}_i||$ is the distance between \mathbf{x}_j and \mathbf{s}_i and $\alpha_1 = 1/(2 * \sigma^2)$. We argued (see Sec. 4.4) that this model corresponds to an explicit model of space usage – namely, that individual locations are draws from a bivariate normal distribution. We also mentioned that other detection models are possible, including a logit model of the form:

$$\text{logit}(p_{ij}) = \alpha_0 + \alpha_1 ||\mathbf{x}_j - \mathbf{s}_i||. \quad (6.1.1)$$

5694 However, there's nothing preventing us from constructing a myriad of other models
 5695 for detection probability as a function of distance. The most commonly used detec-
 5696 tion probability models are also those used in the distance sampling literature: the
 5697 half-normal (Gaussian), the hazard, and the negative exponential. The negative
 5698 exponential model is:

$$p_{ij} = p_0 * \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||)$$

5699 where we define $\alpha_1 = 1/\sigma$. We could use the general power model (Russell et al.,
 5700 2012):

$$p_{ij} = p_0 * \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^\theta)$$

5701 of which the Gaussian and exponential models are special cases. Another model
 5702 that could be considered is the Gaussian hazard rate model (Hayes and Buckland,
 5703 1983):

$$p_{ij} = 1 - \exp(-\lambda_0 * \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2))$$

5704 which was previously discussed in Sec. .

5705 In each of the cases, the relationship of α_1 to σ varies and must be properly
 5706 specified. The **R** package **secr** allows the user to access 12 different detection
 5707 models, of which some are only used for simulating data (see Table 6.1). These
 5708 detection functions can also be implemented in **R**, **WinBUGS**, **JAGS** etc..

5709 Insofar as all these encounter probability models are symmetric and stationary,
 5710 they are pretty crude descriptions of space usage by real animals. This is not to
 5711 say they are inadequate descriptions of the data and, as we discuss in Chaps. 12
 5712 and 11, we can use them as the basis for producing more realistic models of space
 5713 usage.

5714 By changing the encounter probability model and the specification of α_1 , we
 5715 can basically create any function of distance for the data. It is important to note
 5716 that σ is not comparable under these different encounter probability models and
 5717 should not be regarded as "home range radius" in general. While there is generally
 5718 a relationship between σ and home range size, that relationship varies depending
 5719 on the model under consideration. We demonstrate how to fit different encounter
 5720 probability models in the Bayesian framework here, and then provide a section on
 5721 the likelihood analysis (in **secr**) in a separate section below.

5722 **6.1.1 Bayesian analysis with bear.JAGS**

5723 To demonstrate how to incorporate various types of covariates into models for
 5724 encounter probability using **JAGS**, we return to the data collected during the Fort
 5725 Drum bear study. This data set was first introduced in Chapt. 3, but, to refresh
 5726 your memory, there were 38 baited hair snares that were operated between June
 5727 and July 2006. The snares were checked each week for a total for $K = 8$ sample

Table 6.1. Basic encounter probability models (“distance functions”) available in `secr`. (Table taken from the `secr` help files). Notation deviates from that used in the text. In this table g_0 is the baseline encounter rate or probability parameter used in `secr` but this is equivalent to our p_0 or 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	$g(d) = 1 - F[\{c - (\beta_0 + \beta_1(d-1) - 10 * \log_{10}(d^2))\}/S]$

5728 occasions and $n = 47$ individual bears were encountered at least once. The data
 5729 are provided in the **R** package `scrbook` and an **R** function called `bear.JAGS` allows
 5730 the user to easily pick which model to analyze. The function `bear.JAGS` will set
 5731 up the data, write the model, define the MCMC specifications (e.g., initial values,
 5732 etc.) and, finally, run the selected model in **JAGS**. In addition to choosing which
 5733 model to run, the user can also specify the number of chains, iterations and length
 5734 of the burn-in phase. Calling the function will provide all the code to implement
 5735 the models independently as well. In the following sections we will present the
 5736 model code and output for the most commonly employed models; for all analyses
 5737 we ran 3 chains with a burn-in of 500 iterations and 20000 saved iterations.

5738 6.1.2 Bayesian analysis of encounter probability models

5739 In Panel 6.1, we present the basic SCR model and show how to specify the negative
 5740 exponential encounter probability model. To call each of these from the function
 5741 `bear.JAGS` set `model='SCR0'` or `model='SCRexp'` in the function call, respectively.
 5742 To reduce repetition of the R coding, we include the basic code here and then only
 5743 show modifications when necessary throughout the chapter. All of the R coding can
 5744 be found within the `bear.JAGS` function as well. To begin, the required R libraries
 5745 are installed and then we attached the Ft. Drum bear data set. The bear data set

5746 includes a 3-d data array (called `bearArray` in our code), with dimensions `nind` ×
 5747 `ntraps` × `nreps` representing the capture histories of `nind` captured individuals
 5748 at `ntraps` trap locations. In the Bayesian analysis, data augmentation is used to
 5749 estimate N and therefore the `bearArray` data must be augmented with $M - nind$
 5750 all zero encounter histories. In models without time dependence, the augmented
 5751 `bearArray` (called `Yaug` in the code) will be reduced to a 2 dimensional array
 5752 (denoted `y` in the code) that has dimensions `M` × `ntraps`.

```

5753 > library(rjags) #load the necessary libraries
5754 > library(scrbook)
5755
5756 > data(beardata) #attach the bear data for Ft. Drum
5757 > ymat <- beardata$bearArray
5758 > trapmat <- beardata$trapmat
5759 > nind <- dim(beardata$bearArray)[1]
5760 > K <- dim(beardata$bearArray)[3]
5761 > ntraps <- dim(beardata$bearArray)[2]
5762 > M <- 650
5763 > nz <- M-nind
5764
5765 #create augmented array
5766 > Yaug <- array(0, dim=c(M,ntraps,K))
5767 > Yaug[1:nind,,] <- ymat
5768 > y <- apply(Yaug,1:2, sum)
```

5769 The function `bear.JAGS` also establishes the upper and lower limits on the
 5770 state space by centering the trap array coordinates (which are imported with the
 5771 `beardata` and saved in the code above as `trapmat`) and then buffering by 20km.

5772 Applying the SCR model with Gaussian encounter probability model provides
 5773 an estimate (posterior mean) of $D = 0.167$ bears per km^2 and with the negative
 5774 exponential encounter probability model the posterior mean is virtually the same
 5775 $D = 0.167$. In distance sampling, the use of different encounter probability
 5776 models often results in very different estimates of density (especially when using the
 5777 negative exponential model). There are two main reasons why the different models
 5778 may have less of an impact on the density estimates under the SCR models.
 5779 First, we can estimate the baseline encounter probability parameter (p_0). In most
 5780 distance sampling models, detection at distance 0 is set to 1. In Table 6.2, the
 5781 posterior mean of p_0 is 0.11 under the Gaussian model and 0.34 under the negative
 5782 exponential model. The larger baseline encounter probability under the negative
 5783 exponential model reduces the impact of the having “no shoulder”. Secondly, the
 5784 detection probability function here is governing ‘movement’ of individuals (which
 5785 we have more information on than in distance sampling), not the whole detection
 5786 process, so the shape of the detection probability function does not impact the
 5787 density estimation as much.

```
model{
  alpha0 ~ dnorm(0,.1)                                # Prior distributions
  logit(p0) <- alpha0
  alpha1 <- 1/(2*sigma*sigma)
  sigma ~ dunif(0, 15)
  psi ~ dunif(0,1)

  for(i in 1:M){
    z[i] ~ dbern(psi)
    s[i,1] ~ dunif(xlim[1],xlim[2])
    s[i,2] ~ dunif(ylim[1],ylim[2])
    for(j in 1:J){
      d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
      y[i,j] ~ dbin(p[i,j],K)
      p[i,j] <- z[i]*p0*exp(- alpha1*d[i,j]*d[i,j]) # Gaussian model
      #p[i,j] <- z[i]*p0*exp(- alpha1*d[i,j])          # exponential model
    }
  }
  N <- sum(z[])
  D <- N/area
}
```

Panel 6.1: **JAGS** model specification for a basic SCR model with Gaussian encounter probability function and the alternative exponential encounter probability function.

5788 In all analyses it is important to check that the size of the augmented data set
 5789 (M) is sufficiently large and does not impact the estimate of N . Here, the 97.5%
 5790 percentile for N is 628 (Table 6.2), thus not reaching our $M = 650$ value. We could
 5791 also increase M and compare the posterior of N under the different scenarios as
 5792 another check that the data augmentation is sufficient.

Table 6.2. Posterior summaries of SCR model parameters having different encounter probability models, for the Fort Drum black bear data.

Parameter	Mean	SD	2.5	97.5
Gaussian				
D	0.17	0.022	0.122	0.207
N	500.63	66.652	371.000	628.000
p_0	0.11	0.014	0.081	0.135
ψ	0.77	0.104	0.566	0.966
σ	1.99	0.131	1.762	2.275
Exponential				
D	0.17	0.022	0.130	0.210
N	512.06	65.771	382.000	634.000
p_0	0.34	0.056	0.246	0.465
ψ	0.79	0.102	0.584	0.974
σ	1.12	0.095	0.951	1.323

5793 A very important consideration when using different detection probability functions
 5794 is the interpretation of σ . The estimate of σ under the negative exponential
 5795 model is 1.12, which is distinct from our estimate of σ under the Gaussian model,
 5796 $\sigma = 1.996$. The interpretation of σ in the two models is really quite distinct. In
 5797 the normal model it can be interpreted as the standard deviation of a bivariate
 5798 normal movement model whereas the manner in which σ relates to “area used” for
 5799 the negative exponential model has nothing to do with a bivariate normal model of
 5800 movement. This highlights that it is important for the user to know what detection
 5801 probability function is used and what the interpretation of σ might be in relation
 5802 to the home range size. This relationship was discussed in Sec. 4.4.

5803 We now move onto incorporating covariates into the model using the **JAGS**
 5804 language. For this part, we will stick with the Gaussian encounter probability
 5805 model shown in the Panel 6.1 above.

6.2 MODELING COVARIATE EFFECTS

5806 The basic strategy for modeling covariate effects is to include them on the baseline
 5807 encounter rate or probability parameter, p_0 (or λ_0), or the scale parameter of the
 5808 encounter model, σ , or in some cases, both parameters.

5809 Broadly speaking, we recognize (here) 3 types of covariates. Fixed covariates
 5810 that are fully observable and might vary by trap alone (e.g., type of trap, baited
 5811 or not, disturbance regime, even habitat), sample occasion (e.g., day of season or

weather conditions), or both (e.g., behavior, weather - if over a large region). Another class of covariates are those which vary at the level of the individual (and possibly also over time). As a technical matter, and as noted before, these are different from fixed covariates because we cannot see all of the individuals and the covariates are almost always incompletely observed (if at all). The lone exception is the behavioral response to capture which is known for all individuals, captured or not (an animal never captured/observed has never been captured before). We noted many times before that space itself (i.e., the activity centers) is a type of individual covariate and this notion actually helped us derive the fully spatial capture-recapture model from the traditional, non-spatial model (Chapt. 3). We do not get to observe the activity center for any individuals, but for individuals that are encountered we get to observe some information about it in the form of which traps the individual was encountered in. And finally, we have completely unobserved covariates such as heterogeneity in home range size. We consider heterogeneity in a separate section below since alone there are a suite of models for describing latent heterogeneity.

Table 6.3. Examples of different types of covariates in SCR models.

Covariate type	Examples
individual	sex, age, home range
trap	baited/not, habitat (see also Chapter 12)
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 6.3 shows examples of covariates by type – trap, individual, and time – and also gives examples of some combined types. These

5841 are the types of covariates we will specifically address in this chapter demonstrating
 5842 how to analyze the different types in the following sections.

5843 **6.2.1 Date and time**

5844 Often, researchers are interested in modeling the effect of date or chronological
 5845 time on encounter probability. For example, in a long term hair snare study, we
 5846 may expect that seasonal shedding (Wegan et al., 2012) will influence encounter
 5847 probabilities directly. Or, we may expect behaviors such as denning, mating, etc.,
 5848 to influence the encounter of certain species at certain times of year (Kéry et al.,
 5849 2011). There are two common ways to incorporate date or time information into
 5850 a model for encounter probability. For cases with a small number of sampling
 5851 occasions we can fit a time-specific intercept (analogous to “model M_t ” in classical
 5852 capture-recapture (Otis et al., 1978)). In this model, there are K sampling occasion-
 5853 specific parameters to reflect potential variation in sampling effort or other factors
 5854 that might vary across samples. Alternatively, we can model parametric functions
 5855 of date or time such as polynomial or sinusoidal functions.

5856 In the first case, we allow each sampling occasion, k , to have its own baseline
 5857 encounter probability, e.g.,

$$\text{logit}(p_{0,k}) = \alpha_{0,k}$$

5858 so that

$$p_{ijk} = p_{0,k} \exp(-\alpha_1 * \|\mathbf{x}_j - \mathbf{s}_i\|^2).$$

5859 This description of the model includes k occasion-specific baseline encounter proba-
 5860 bilities. Thus, if there were 4 sampling occasions, then there are 4 different baseline
 5861 encounter probabilities. We imagine that complete time-specificity of p_0 (i.e., one
 5862 distinct value for each sample occasion) would be most useful in situations where
 5863 there are just a few sampling occasions (if there are many, this formulation will dra-
 5864 matically increase the number of parameters to be estimated) or we do not expect
 5865 systematic patterns over time (e.g., explainable by a polynomial function).

5866 To implement this in **JAGS**, α_0 has to be estimated for each time period k
 5867 either using an index vector or dummy variables (as described in Chapt. 2 and Sec.
 5868 3.3) and this can be done by only changing only a few lines in Panel 6.1:

```
5869 alpha0[k] ~ dnorm(0,.1)
5870 logit(p0[k]) <- alpha0[k]
5871 .....
5872 .....
5873 y[i,j,k] ~ dbin(p[i,j,k],K)
5874 p[i,j,k] <- z[i]*p0[k]*exp(- alpha1*d[i,j]*d[i,j])
```

5875 Since the model contains a parameter for each time period, the encounter his-
 5876 tories must be time-dependent. Thus, a 3-d data array (called **bearArray** in our
 5877 code), with dimensions **nind** × **ntraps** × **nreps** is required (recall that we use the 3-d

augmented array called `Yaug` with dimensions `M × ntraps × nreps` for the Bayesian analysis). In addition to using the 3-d data array, the initial values must be updated so that there are K values generated for α_0 . And finally, this means that another nested for loop is needed in the code to account for the K sample occasions. A side note: the computation time will increase quite a bit (this model for the bear data may take up to 15 hours or more on your machine to obtain a sufficient posterior sample).

Running this model with the function `bear.JAGS` by setting `model=SCRt`, returns estimates of density similar to those from the model without covariates (see Table 6.4), but now we have a characterization of variation in encounter probability over time. Encounter probability seems to increase for the first few time periods before stabilizing around 0.14, dropping off again at the end of the study. The differences in encounter probability from the first time periods to the others might actually be due to something like a behavioral response (see below) or possibly seasonal differences in the efficiency of the sampling technique. Researchers have found that hair snares are more effective at different times of the year (even within season) due to shedding (Wegan et al., 2012). In this particular example, our density estimates are similar to the base model, likely because the differences in encounter probability between occasion were not that large. In a longer term study or in one with greater variation in the encounter probability, the implication of such differences might have a bigger impact on the estimates of density and σ .

Table 6.4. Posterior summaries of parameter estimates from a SCR model with time-dependent baseline encounter probability for the Ft. Drum black bear data set.

Parameter	Mean	SD	2.5	97.5
D	0.17	0.02	0.13	0.21
N	509.24	66.13	381	632
$p_0(t = 1)$	0.06	0.02	0.03	0.10
$p_0(t = 2)$	0.05	0.02	0.02	0.09
$p_0(t = 3)$	0.15	0.03	0.09	0.22
$p_0(t = 4)$	0.14	0.03	0.09	0.21
$p_0(t = 5)$	0.15	0.03	0.09	0.22
$p_0(t = 6)$	0.12	0.03	0.07	0.19
$p_0(t = 7)$	0.15	0.03	0.09	0.22
$p_0(t = 8)$	0.08	0.02	0.04	0.13
ψ	0.78	0.10	0.58	0.97
σ	1.96	0.12	1.73	2.22

The occasion specific intercepts (baseline encounter probability) model might not be the most appropriate for all scenarios (and could require the estimation of many parameters if we had many sampling occasions, take the wolverine example from Chapt. 4.9 where there were 165 daily sampling occasions). Particularly in such a case, variation in the encounter process over time is to be expected. For example, if a camera trap study is conducted for an entire year, it is expected that

5905 there would be behavioral patterns in individuals due to mating or denning. Instead
 5906 of fitting a model with K baseline encounter probabilities, we can include date as
 5907 a linear (or quadratic, ...) effect. An example can be found in Kéry et al. (2011)
 5908 who incorporated a day-of-year covariate, both as a linear and a quadratic effect,
 5909 into their SCR model of European wildcats; the data had been collected over a
 5910 year-long period and cat behavior was expected to vary seasonally thus influencing
 5911 the probability of encounter. In these cases, we would specifically incorporate day
 5912 of year (variable “Date”) as a numeric covariate as:

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * \text{Date}_k \\ p_{ijk} = p_{0,ijk} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

5913 or a quadratic effect of day-of-year:

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * \text{Date}_k + \alpha_3 * \text{Date}_k^2 \\ p_{ijk} = p_{0,ijk} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

5914 where the variable Date is an integer coding of day-of-year, indexed to some arbitrary
 5915 start point in time.

5916 6.2.2 Trap-specific covariates

5917 In some studies it makes sense to model encounter probability as a function of local
 5918 or trap-specific covariates. These can be one of two types: genuine trap covariates
 5919 that describe the trap or encounter site, such as whether a trap is baited or not,
 5920 or how many traps were set at a sampling location, or what kind of bait was used,
 5921 etc., or local covariates that describe the likelihood that an animal would use the
 5922 habitat in the vicinity of the trap (see Chapt. 12 for more on this situation).
 5923 We imagine that these covariates, of either type, should affect baseline encounter
 5924 probability. For example, Sollmann et al. (2011) found a large difference in the
 5925 encounter probability of jaguars due to traps being located on roads, which the
 5926 animals were using to travel along, as opposed to traps placed off of roads. In this
 5927 case, the trap type is a binary variable – on/off road, (another binary variable could
 5928 be baited/non-baited). We can write this as:

$$\text{logit}(p_{0,j}) = \alpha_{0,type_j} \\ p_{ijk} = p_{0,j} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2).$$

5929 Here, we use an index variable, “type”, an integer value for the trap-specific co-
 5930 variate. Thus for our example of on/off road, we would have $type_j = 1$ if trap
 5931 j is on a road and $type_j = 2$ otherwise, and we would estimate two separate α_0
 5932 parameters – one for on-road and one for off-road cameras. This general set up also
 5933 allows for more than 2 categories, say if 4 different camera models were used in a

study, we would use a set of 3 binary dummy variables to allow for estimation of the different encounter rates (i.e., the intercept). To express the model in terms of dummy variables using the 2-category example above, we would specify our “type” vector as $Type_j = 0$ if trap j is on a road and $Type_j = 1$ otherwise, and write our model as

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * Type_j$$

Now, α_0 is the baseline encounter probability (on the logit scale) for traps on a road ($Type_j = 0$) and α_2 is the effect on baseline encounter probability of a trap being of Type = 1. While these models are equivalent, and should yield identical results, sometimes one parameterization might work better than the other in **WinBUGS** or **JAGS** (Kéry, 2010).

6.2.3 Behavior or trap response by individual

One of the most basic of encounter models is that which accommodates a change in encounter probability as a result of initial encounter. This is colloquially referred to as “trap happiness” or “trap shyness”, or in other words, a behavioral response of individuals to being captured (Otis et al., 1978). If a trap is baited with a food source, an individual might come back for more. On the other hand, if being captured is traumatic then an individual might learn to avoid traps. Both of these types of responses can occur in most species depending on the type of encounter mechanisms being employed. Moreover, behavioral response can be either global (Gardner et al., 2010b) or local (Royle et al., 2011b). The local response is a trap-specific response while a global response suggests that initial capture provides a net increase or decrease in subsequent probabilities of capture (across all traps). A behavioral response does not need to be enduring (i.e., persist for the entire study after the individual has been captured/observed for the first time) but can also be ephemeral, if, for example, an animal only avoids a trap on the occasion immediately after it was captured (Yang and Chao, 2005; Royle, 2008). While we will focus the examples in this chapter on enduring behavioral effects, extending such a model to the case of an ephemeral response should not pose any difficulties.

To describe these behavioral models we need to create a binary matrix that indicates if an individual has been captured previously. For the global behavioral response, define the $n \times K$ matrix, \mathbf{C} , where $C_{ik} = 1$ if individual i was captured at least once prior to session k , otherwise $C_{ik} = 0$.

$$\begin{aligned}\text{logit}(p_{0,ik}) &= \alpha_0 + \alpha_2 * C_{ik} \\ p_{ijk} &= p_{0,ik} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2)\end{aligned}$$

For the local behavioral response, which is trap specific, we create an array, C_{ijk} , that indicates if an individual i has been previously captured in trap j at time k . We then include this in the model in the exact same form as above (with the sole

5969 difference that both C and p are now also indexed by k):

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * C_{i,j,k}$$

$$p_{ijk} = p_{0,ijk} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

5970 Since the behavioral response is occasion specific, to implement either the local
 5971 or global response model in **JAGS**, we will have to use the 3-d array of the
 5972 augmented capture histories ($M \times ntraps \times nreps$) as we did for the time-varying
 5973 encounter probability model above. The code must loop over each sampling occa-
 5974 sion, but otherwise, the model varies only a little from the basic SCR model shown
 5975 in Panel 6.1. Here is the specification of the the occasion specific (k) loop:

```
5976 for(k in 1:K){
  5977   logit(p0[i,j,k]) <- alpha0 + alpha2*C[i,j,k]
  5978   y[i,j,k] ~ dbin(p[i,j,k],1)
  5979   p[i,j,k] <- z[i]*p0[i,j,k]*exp(- alpha1*d[i,j]*d[i,j])
  5980 }
```

5981 Despite the minor changes to the **BUGS** code, this model can require quite a bit
 5982 of time and computational effort to carry out the behavior response models. Imple-
 5983 menting the behavioral models with the function **bear.JAGS** by setting **model=SCRb**
 5984 or **model=SCRb** for the local or global model respectively, returns the results, shown
 5985 in Table 6.5. There is a strong global behavior response suggested by the posterior
 5986 mean of $\alpha_2 = 0.90$. The estimate of N and subsequently D are larger than under
 5987 the model wihout a behavioral response, here we estimate $N = 577.56$ and in the
 5988 SCR0 model, we estimated $N = 500$. This makes sense given the large estimate of
 5989 α_2 , which suggests that bears are trap happy. In situations where animals are trap
 5990 happy, the model tends to over estimate encounter probability (i.e., the bears that
 5991 are never observed have a lower encounter probability than those that have been
 5992 captured in the study) and thereby reduce the estimate of N . We do not include
 5993 the results here, but the estimates were similar under the local behavioral response
 5994 model.

Table 6.5. Posterior summaries of parameter estimates from the SCR model with a global behavioral response of encounter for the Fort Drum black bear data set.

Parameter	Mean	SD	2.5%	97.5%
D	0.19	0.02	0.15	0.21
N	577.56	54.30	452	648
α_0	-2.81	0.24	-2.91	-2.36
α_2	0.90	0.23	0.45	1.35
ψ	0.88	0.08	0.69	0.99
σ	2.00	0.13	1.77	2.28

5995 6.2.4 Individual covariates

5996 Individual covariates are those which are measured (or measurable) on individuals,
 5997 so we get to observe them only for the captured individuals. Sex is a simple
 5998 example of an individual covariate, but one of the most commonly used in capture-
 5999 recapture studies. The sex of an individual can influence many aspects of its ecology
 6000 and behavior, including, for example, its home range size, frequency of movement,
 6001 and seasonal behavior. This is common in studies of carnivores where females
 6002 often have smaller home ranges than males (Gardner et al., 2010b; Sollmann et al.,
 6003 2011). Additionally, we may find differences in the baseline encounter probability
 6004 between males and females because females may move around less frequently, or
 6005 possibly because they are less likely to use landscape structures that researchers
 6006 may target with sampling devices in order to increase sample size, such as roads
 6007 (e.g. Salom-Pérez et al., 2007). Therefore, we can imagine that sex may impact
 6008 both the baseline encounter probability α_0 and the typical home range size, so that
 6009 α_1 might also be sex-specific also. The fully sex-specific model is:

$$\text{logit}(p_{0,i}) = \alpha_{0,\text{sex}_i}$$

$$p_{ijk} = p_{0,i} \exp(-\alpha_{1,\text{sex}_i} * \|\mathbf{x}_j - \mathbf{s}_i\|^2)$$

6010 where sex_i is a vector indicating the sex of each individual (1 = male, 2 = female).
 6011 While we might know the sex of all individuals observed in the study, we will
 6012 never know the sex of individuals that are not observed, resulting in missing values
 6013 (Gardner et al., 2010b). It is also possible that we may not be able to determine the
 6014 sex of individuals that are observed during the study. For example photographic
 6015 captures do not necessarily result in pictures that allow the sex to be absolutely
 6016 determined, thus sometimes resulting in missing values of this covariate for animals
 6017 captured in the study. We deal with this slightly differently depending on the
 6018 inference framework that we adopt (Bayesian or likelihood). Here we demonstrate
 6019 the Bayesian implementation and we discuss the likelihood approach using **secr** in
 6020 detail below in Sec. 6.4.2. Before proceeding with that, we note that it would be
 6021 possible also to model covariates directly on the parameter σ (or its logarithm),
 6022 e.g., $\log(\sigma_i) = \theta_1 + \theta_2 \text{sex}_i$ (see Sec. 7.1). One or the other (or perhaps *some* other)
 6023 parameterization may yield a better performing MCMC algorithm or provide a
 6024 more natural or preferred interpretation. In the context of Bayesian analysis, given
 6025 that priors are not invariant to transformation of the parameters, this may be a
 6026 consideration in choosing the particular parameterization.

6027 Specifying a fully sex-specific model for **JAGS** is similar to the time-specific
 6028 model shown above. We need to use an index or dummy variable to let α_0 and/or
 6029 α_1 be defined separately for males and females. The main difference in this specification
 6030 is that we do not observe sex for the augmented individuals. Therefore,
 6031 we have missing observations of the covariate for those individuals. As a result, sex
 6032 is regarded as a random variable and so the missing values can be estimated along
 6033 with the other structural parameters of the model.

Because we are regarding sex as a random variable, we have to specify a distribution for it. With only two possible outcomes, it is natural to suppose that $\text{Sex}_i \sim \text{Bernoulli}(\psi_{\text{sex}})$ where the parameter ψ_{sex} is the sex ratio of the population. We assume our default non-informative prior for this parameter: $\psi_{\text{sex}} \sim \text{Uniform}(0, 1)$. The model specification in Panel 6.2 demonstrates how to incorporate a partially observed covariate (i.e., “sex”). It is important to note that in the previous equation, sex_i is a vector with two categories indicating the sex of each individual (e.g., 1 = male, 2 = female). This corresponds directly to having a binary indicator of sex (e.g., $\text{Sex}_i = 1$ if individual i is female, and 0 otherwise). In the Bayesian formulation of the model, we use both the binary indicator (`Sex`) and a categorical indicator (`Sex2 = Sex + 1`). The former (termed `Sex` in Panel 6.2) allows us to specify the Bernoulli distribution for the random variable, and the latter (termed `Sex2`) allows us to use the dummy or indicator variable specification in the model.

In both **JAGS** or **BUGS** missing data are indicated by `NA` in the data objects passed to the program through the `bugs` or `jags` functions in **R**. To set up the data, we need to create a vector of length M with the first n elements being 0 if individual i is a female, or 1 if i is a male (for the Fort Drum black bear data the function `bear.JAGS` extracts this information automatically from the `beardata` object), and the subsequent $M - n$ elements being `NA`. It is generally a good idea to provide starting values for the missing data, but we cannot provide starting values for observed data; in this case where one vector (or other object) contains both observed and missing data, initial values for the observed data have to be specified as `NA`. The code snippet below shows you how to set up the data including the `Sex` vector and the initial values function (the remainder of the code is identical to what we've shown before).

```
6059 > sex <- beardata$sex #the sex data for captured individual
6060 > Sex <- c(sex-1, rep(NA, nz)) #sex enters as 1/2, this recodes it to 0/1
6061                               #so we can use Bernoulli distribution
6062
6063 > data <- list(y=y, Sex=Sex, M=M, K=K, J=ntraps, xlim=xlim, ylim=ylim, area=areaX)
6064 > params <- c('psi', 'p0', 'N', 'D', 'sigma', 'psi.sex')
6065 > inits <- function() { list(z=c(rep(1,nind), rbinom(nz,1,0.5)), psi=runif(1),
6066   s=cbind(runif(M, xlim[1],xlim[2]), runif(M,ylim[1],ylim[2])), 
6067   psi.sex=runif(1), Sex=c(rep(NA, nind), rbinom(nz,1,0.5)),
6068   sigma=runif(2,2,3), alpha0=runif(2)) }
```

6069 The **BUGS** model specification is shown in Panel ??.

6070 Our estimate of density under the fully sex-specific model is still very similar
6071 to the previous models (Table 6.6), and while the baseline detection was not very
6072 different between males and females, we can see that they had very different σ
6073 estimates (note that the BCIs do not overlap). As usual, you can reproduce this
6074 analysis by calling the function `bear.JAGS` and set `model='SCRsex'`.

```

model{

psi ~ dunif(0,1)                                # Prior distributions
psi.sex ~ dunif(0,1)
for(t in 1:2){
  alpha0[t] ~ dnorm(0,.1)
  logit(p0[t]) <- alpha0[t]
  alphai[t] <- 1/(2*sigma[t]*sigma[t])
  sigma[t] ~ dunif(0, 15)
}

for(i in 1:M){
  z[i] ~ dbern(psi)
  Sex[i] ~ dbern(psi.sex)                      # Sex is binary
  Sex2[i] <- Sex[i] + 1                         # Convert to categorical
  s[i,1] ~ dunif(xlim[1],xlim[2])
  s[i,2] ~ dunif(ylim[1],ylim[2])

  for(j in 1:J){
    d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
    y[i,j] ~ dbin(p[i,j],K)
    p[i,j] <- z[i]*p0[Sex2[i]]*exp(-alphai[Sex2[i]]*d[i,j]*d[i,j])
  }
}
N <- sum(z[])
D <- N/area
}

```

Panel 6.2: **JAGS** model specification for an SCR model with sex-specific encounter probability parameters.

Table 6.6. Posterior summaries of parameter estimates from sex-specific SCR models for the Fort Drum black bear data set.

Parameter	Mean	SD	2.5	97.5
D	0.168	0.022	0.12	0.21
N	509.982	66.355	376	631
$p_{0,female}$	0.136	0.025	0.09	0.19
$p_{0,male}$	0.092	0.017	0.06	0.13
ψ_{sex}	0.310	0.068	0.19	0.45
ψ	0.784	0.103	0.58	0.97
σ_{female}	1.542	0.132	1.31	1.83
σ_{male}	2.682	0.389	2.09	3.62

6.3 INDIVIDUAL HETEROGENEITY

6075 Here we consider SCR models with individual heterogeneity. Capture-recapture
 6076 models with individual heterogeneity in detection probability, so-called model M_h ,
 6077 have a long history in classical capture recapture models and they have special
 6078 relevance to SCR (Sec. 3.4). While the advent of SCR models may appear to
 6079 have rendered the use of classical model M_h obsolete (because the heterogeneity is
 6080 being accounted for explicitly) we may still wish to consider heterogeneity models
 6081 for other biological reasons. It is reasonable to expect in real populations that
 6082 there exists heterogeneity in home range size and so we think that α_1 could exhibit
 6083 heterogeneity among individuals. As we noted previously, it may be advantageous
 6084 or desirable in some cases to model heterogeneity directly in terms of the scale
 6085 parameter of the distance function σ or some other transformation of the “distance
 6086 coefficient”, perhaps even 95% home range area.

6087 In this section, we describe a class of spatial capture-recapture models to allow
 6088 for individual heterogeneity in encounter probability. In particular, one class of
 6089 models we propose explicitly admits individual heterogeneity in home range *size*.
 6090 In addition, we consider a standard representation for heterogeneity in which an
 6091 additive individual-specific random effect is included in the linear predictor for
 6092 baseline encounter probability.

6.3.1 Models of heterogeneity

6093 An obvious extension to the SCR model is to include an additive individual effect,
 6094 analogous to classical “model M_h ”. We’ll call this model “SCR+Mh”:

$$\text{logit}(p_{0,i}) = \alpha_0 + \eta_i$$

$$p_{ijk} = p_{0,i} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

6095 where η_i is an individual random effect having distribution $[\eta|\sigma_p]$. A popular class
 6096 of models arises by assuming $\eta_i \sim \text{Normal}(0, \sigma_p^2)$ (Coull and Agresti, 1999; Dorazio
 6097 and Royle, 2003). We show how to implement this specific SCR + Mh model in

6099 Panel 6.3, although many other random effects distributions are possible. A popular
 6100 one is the finite-mixture of point masses (Norris and Pollock, 1996; Pledger, 2004)
 6101 which we demonstrate how to fit using `secr` in Sec. 6.4.3.

```
model{

  alpha0 ~ dnorm(0,.1)                                # Prior distributions
  alpha1 <- 1/(2*sigma*sigma)
  sigma ~ dunif(0, 15)
  psi ~ dunif(0,1)
  tau_p ~ dgamma(.001,.001)

  for(i in 1:M){
    eta[i] ~ dnorm(0, tau_p)                         # Individual level variables
    z[i] ~ dbern(psi)
    s[i,1] ~ dunif(xlim[1],xlim[2])
    s[i,2] ~ dunif(ylim[1],ylim[2])

    for(j in 1:J){                                    # The "likelihood" etc..
      d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
      y[i,j] ~ dbin(p[i,j],K)
      logit(p0[i,j]) <- alpha0 + eta[i]
      p[i,j] <- z[i]*p0[i,j]*exp(-alpha1*d[i,j]*d[i,j])
    }
  }
  N <- sum(z[])                                     # N, D are derived
  D <- N/area
}
```

Panel 6.3: **JAGS** model specification for the SCR + Mh model with Gaussian encounter probability model and additive normal random effect.

6102 6.3.2 Heterogeneity induced by variation in home range size

6103 An alternative heterogeneity model, one that has more of a direct biological moti-
 6104 vation and interpretation, describes heterogeneity in home range size among indi-
 6105 viduals. To model heterogeneity in home range area, we can assume a distribution
 6106 for a transformation of the scale parameter of the encounter probability model such
 6107 as σ^2 , or $\log(\sigma^2)$, etc.. We call this “model SCR + Ah” (Ah here for area-induced
 6108 heterogeneity).

6109 Consider the following log-normal model for individual scale parameter of the
 6110 Gaussian encounter probability model, σ_i^2 :

$$\log(\sigma_i^2) \sim \text{Normal}(\mu_{hra}, \tau_{hra}^2)$$

6111 then the 95% home range area has a scaled log-normal distribution with mean

$$6\pi \exp(\mu_{hra} + \tau_{hra}^2/2).$$

6112 The variance is slightly more complicated, but you can look-up the variance of a
 6113 log-normal distribution and combine it with the 95% home range area calculation
 6114 in Sec. 4.4 to work out the implied variance of home range area under this model.
 6115 We show two examples of the implied *population* distribution of home range area
 6116 under this log-normal model that implies a mean home range area of about 6.9 area
 6117 units (Figure 6.1). The left panel shows a standard deviation in home range area
 6118 of 2.88 units and the right panel shows a standard deviation in home range area of
 6119 0.70 units. The two cases were generated by tweaking the μ_{hra} and τ_{hra}^2 parameters
 6120 of the log-normal distribution to achieve a constant expected value of home range
 6121 area, but modify the standard deviation.

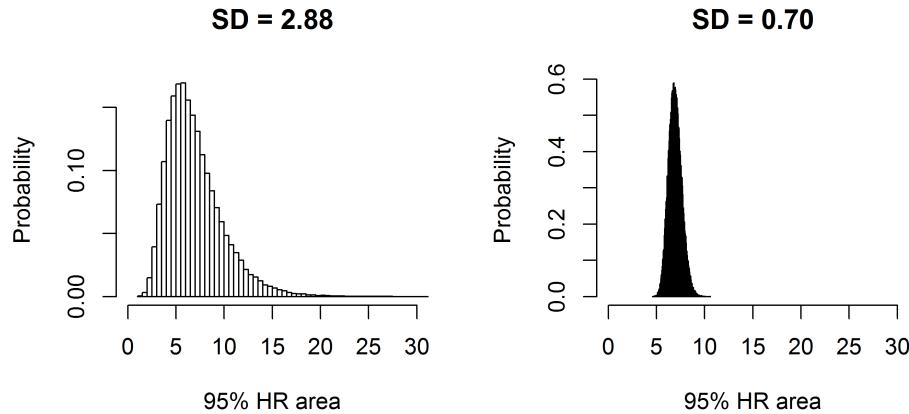


Figure 6.1. Population distribution of home range area for a model in which $\log(\sigma^2)$ has a normal distribution with mean μ_{hra} and variance τ_{hra}^2 . The parameters were chosen to yield a constant expected value of about 6.9 units of area, but to produce two different levels of heterogeneity: A population standard deviation of 2.88 units (left panel) and 0.70 units (right panel).

6.4 LIKELIHOOD ANALYSIS IN SECR

Previously, in Chapt. 5, we introduced the **R** package **secr** and described the likelihood based inference approach taken by that package (see Sec. 5.5.3). Here we discuss how to implement some standard covariate models in **secr** and provide an example of model selection using AIC. As we saw in Chapt. 5, **secr** uses the standard **R** model specification syntax, defining the dependent and independent variable relationship using tildes (e.g., $y \sim x$). Thus, in **secr** we might have $g_0 \sim \text{behavior}$ or $\sigma \sim \text{time}$; when left unspecified or set to 1 (e.g., $g_0 \sim 1$), this will default to a model with no covariates (i.e., constant parameter values). A number of default model formulas for the baseline and scale parameter of the encounter probability model are available in **secr**. Additionally, **secr** allows us to specify covariates on density (we cover this in Chapt. 10), which are set for example as $D \sim \text{habitat}$.

To demonstrate models with various types of covariates using **secr**, we continue using the Fort Drum black bear data. We include in the **scrbook** package a function called **secr.bear** that will format the data (see Chapt. 5 for the **secretdat** format) and then fit and compare 8 models (details shown in Panel 6.4). We have described all of these models in the previous sections, so we only briefly comment here on how to fit certain models in **secr** and compare them using AIC, and give a few helpful notes.

6.4.1 Notes for fitting standard models

In the **secr** package, the encounter probability model is called the “detection function” and it is specified by changing the “**detectfn**” option (an integer code) within the **secr.fit** command. Table 6.1 shows the possible encounter probability models that **secr** allows; the default is that based on the kernel of a bivariate normal probability distribution function (hence we call this the Gaussian model, but it is referred to as “half-normal” in **secr**) and the (negative) exponential is **detectfn** = 2. See model 2 in Panel 6.4 for how to fit the exponential model to the Fort Drum bear data set.

The **secr** package easily fits a range of SCR equivalents of standard capture-recapture models. The package has pre-defined versions of the classic model M_t where each occasion has its own encounter probability, as well as a linear trend in baseline encounter probability over occasions (in a spatial modeling framework σ could also be an occasion specific parameter, but having encounter probability change with time seems like the more common case). For the classical time-effects type of model with K distinct parameters **secr** uses ‘t’ to denote this in the model specification formula (see model 3 in panel 6.4); whereas, for a linear trend over occasions **secr** uses ‘T’.

The global trap response model (what we called model M_B), or a local trap-specific behavioral response (model M_b) can be fitted in **secr** using formulae with

6161 “b” for the global response model and “bk” for the local trap response model (see
 6162 models 4 and 5 in Panel 6.4; note that to fit the trap specific behavioral response
 6163 model you need version 2.3.1 or newer of **secr**).

1. null model with a bivariate normal encounter probability model
`bear_0=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ 1, sigma ~ 1))`
2. null model with an exponential encounter probability model
`bear_0exp=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ 1, sigma ~ 1), detectfn=2)`
3. model with fixed time effects
`bear_t=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ t, sigma ~ 1))`
4. global behavioral model
`bear_B=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ b, sigma ~ 1))`
5. trap specific behavioral response
`bear_b=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ bk, sigma ~ 1))`
6. global behavior model with fixed time effects
`bear_bt=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ b+t, sigma ~ 1))`
7. sex-specific model
`bear_sex=secr.fit(bear.cap, model=list(D ~ session, g0 ~ session, sigma ~ session))`
8. heterogeneity model
`bear_h2=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ h2, sigma ~ h2))`

Panel 6.4: Models called from **secr.bear** function. All models use **buffer = 20000**

6164 6.4.2 Sex effects

6165 Incorporating sex effects into models with **secr** can be done a few different ways,
 6166 but there are not pre-defined models for this. A limitation of fitting models with sex
 6167 effects in **secr** is that it does not accommodate missing values of the sex variable.
 6168 Thus, in all cases, individuals that are of unknown sex must be removed from the
 6169 data set (recall that in a Bayesian framework we can keep these individuals in the
 6170 data set by specifying a distribution for the individual covariate “sex”). In **secr**,
 6171 the easiest way to include sex effects is to code sex as a “session” variable using the

6172 multi-session models (see Sec. 5.5.4 for a description of the multi-session models),
 6173 providing two sessions, one representing males and one for females (see model 7 in
 6174 Panel 6.4). This method provides two separate density estimates, which can then
 6175 be combined into a total density.

6176 **6.4.3 Individual heterogeneity**

6177 To incorporate heterogeneity, **secr** fits a set of finite mixture models (Norris and
 6178 Pollock, 1996; Pledger, 2004). These are expensive in terms of parameters but
 6179 they have been widely adopted because they are easy to analyze using likelihood
 6180 methods, as the marginal distribution of the data is just a sum of a small number
 6181 of components. Using **secr**, individual heterogeneity can be incorporated into the
 6182 encounter probability model using default models for either a 2- or 3-component
 6183 finite mixture model using the “**h2**” or “**h3**” model terms. The 2-part mixture is
 6184 shown in model 8 of panel 6.4 and the 3-part mixture can easily be fit by substituting
 6185 **h3** for **h2**. The finite-mixture model can be fit in **JAGS** or **BUGS**, but we only
 6186 showed the SCR + Mh logit-normal mixture in the version above (see Sec. 6.3.1).

6187 **6.4.4 Model selection in secr using AIC**

6188 One practical advantage to using the **secr** package, or likelihood inference in
 6189 general, is the convenience of automatic model selection using AIC (Burnham and
 6190 Anderson, 2002). The **secr** package has a number of convenient functions for
 6191 computing AIC and producing model selection tables, or doing model-averaging
 6192 (as described in Chapt. 7). Running the function **secr.bear**, which calls all of
 6193 the models we have described, will return, in addition to all model results, an AIC
 6194 table with all of the summarized results including the AIC values, delta AIC, and
 6195 model weights (see Table 6.7 or reproduce results in R using **out<- secr.bear();**
out\$AIC.tab).

6196 It is important to note that AIC is not comparable between a multi-session
 6197 model and a model that is not a multi-session model. Therefore, to compare the
 6198 sex-specific model (which uses “sessions”) with all the other models including the
 6199 null, time, and behavioral models, we coded the data set as a multi-session design
 6200 when first loading it to **secr**. This results in all the model outputs listing sepa-
 6201 rate parameter estimates for each session, even the null model with no covariates;
 6202 however, the estimates are the same for both “sessions” in all but the sex-specific
 6203 model.

6204 The results from this AIC analysis are straightforward to interpret; the model
 6205 with a local trap response of encounter probability, “**bk**”, has a model weight of 1
 6206 and thus, according to AIC, 100% support. The 2-part finite mixture model for g_0
 6207 and σ has the second lowest AIC, but considering the large dAICc compared to the
 6208 local trap response model we would probably not consider it any further.

Table 6.7. Log-likelihood, AIC, deltaAIC and AIC weight for several models run in `secr` for the Fort Drum black bear data set.

model	logLik	AIC	AICc	dAICc	AICwt
bear.b	-641.7215	1291.443	1292.395	0.000	1
bear.h2	-653.8382	1319.676	1321.776	29.381	0
bear.0exp	-663.9152	1333.830	1334.389	41.994	0
bear.B	-677.6175	1363.235	1364.187	71.792	0
bear.bt	-668.3044	1358.609	1366.152	73.757	0
bear.sex	-677.7151	1367.430	1369.530	77.135	0
bear.t	-674.4134	1368.827	1374.938	82.543	0
bear.0	-686.2455	1378.491	1379.049	86.654	0

6.5 SUMMARY AND OUTLOOK

There are endless covariates and encounter probability models that can be defined and our goal in this chapter was to introduce basic types of covariate models and demonstrate how to implement them in **BUGS** and `secr`. Essentially, SCR's are GLMMs and therefore we develop covariate models in much the same way, using a suitable transformation (link function) of the parameter(s). In SCR models, we typically have 2 parameters of the encounter probability model for which we might specify covariate models – the baseline encounter probability (or rate) parameter, and a scale parameter that is related in many cases to the home range size of the species. A few examples of different covariate models are given in Table 6.3. We can also consider covariates by their classification as fixed, partially observed, or unobserved (see Table 6.8). This classification of covariate types can be important because the MLE and Bayesian approaches to dealing with partially and unobserved covariates is often different. This was seen above in how the covariate `Sex` was handled in the two frameworks.

Table 6.8. Examples of different covariate classifications.

Covariate class	Examples
Fixed	baited, weather, habitat
Partially observed	sex, age,
Unobserved	home range size, ind. effects

While the move to spatially explicit models in capture-recapture studies has largely rendered the basic CR models (Otis et al., 1978) obsolete, we continue to find this classification useful for categorizing the *spatial* extensions of these standard CR models. The extended models include the standard M_0 , M_t , M_b , and M_h , but also new models that allow for trap-specific information such as "baited/not-baited" or "on/off road". In addition, in Chaps. 11, 12 and 10, we explore additional models for explaining variation in encounter probability and density based on spatial covariates that describe variation in landscape or habitat conditions.

6232
6233

7

6234

MODEL SELECTION AND ASSESSMENT

6235 Our purpose in life is to analyze models. By that, we mean one or more of the
6236 following basic 4 tasks: (1) estimate parameters, (2) make predictions of unobserved
6237 random variables, (3) evaluate the relative merits of different models or choosing a
6238 best model (model selection), and (4) checking whether a specific model appears to
6239 provide a reasonable description of the data or not (model checking, assessment, or
6240 “goodness-of-fit”). In previous chapters we addressed the problems of estimation
6241 of model parameters, and also making predictions of latent variables, s or z , or
6242 functions of these variables such as density or population size. In this chapter, we
6243 focus on the last two of these basic inference tasks: model selection (which model
6244 or models should be favored), and model assessment (do the data appear to be
6245 consistent with a particular model).

6246 In this chapter we review basic strategies of model selection using both likelihood
6247 methods (as implemented in the `secr` package) and Bayesian analysis. Specifically,
6248 we review a number of standard methods of model selection that apply to “variable
6249 selection” problems, when our set of models consists of distinct covariate effects
6250 and they represent constraints of some larger model. For classical analysis based on
6251 likelihood, model selection by Akaike Information Criterion (AIC) is the standard
6252 approach (Burnham and Anderson, 2002). For Bayesian analysis we rely on a
6253 number of different methods. We demonstrate the use of the deviance information
6254 criterion (DIC) (Spiegelhalter et al., 2002) for variable selection problems although
6255 it has deficiencies when applied to hierarchical models in some cases (Millar, 2009).
6256 We use the Kuo and Mallick indicator variable selection approach (Kuo and Mallick,
6257 1998) which produces direct statements of posterior model probabilities which we
6258 think are the most useful, and leads directly to model-averaged estimates of density.
6259 There is a good review paper recently by O’Hara and Sillanpää (2009) that discusses
6260 these and many other related ideas for variable selection. In addition to O’Hara

6261 and Sillanpää (2009) we also recommend Link and Barker (2010, Chapt. 7) for
 6262 general information on model selection and assessment.

6263 To check model adequacy in a Bayesian framework, or whether a specific model
 6264 provides a satisfactory description of our data set, we rely exclusively on the
 6265 Bayesian p-value framework (Gelman et al., 1996). For assessing fit of SCR mod-
 6266 els, part of the challenge is coming up with good measures of model fit, and there
 6267 does not appear much definitive guidance in the literature on this point. Following
 6268 Royle et al. (2011a), we break the problem up into 2 components which we attack
 6269 separately: (1) Conditional on the underlying point process, does the encounter
 6270 model fit? (2) Do the uniformity and independence assumptions appear adequate
 6271 for the point process model of activity centers? The latter component of model fit
 6272 has a considerable precedence in the ecological literature as it is analogous to the
 6273 classical problem of testing “complete spatial randomness” (Cressie, 1991; Illian
 6274 et al., 2008).

6275 We apply some of these methods to the wolverine camera trapping data first
 6276 introduced in Chapt. 4 to investigate sex specificity of model parameters and
 6277 whether there is a behavioral response to encounter. We note that individuals are
 6278 drawn to the camera trap devices by bait and therefore it stands to reason that
 6279 once an individual discovers a trap, it might be more likely to return subsequently,
 6280 a response termed “trap happiness”. We evaluate whether certain models for en-
 6281 counter probability appear to be adequate descriptions of the data, and we evaluate
 6282 the uniformity assumption for the underlying point process.

7.1 MODEL SELECTION BY AIC

6283 Using classical analysis based on likelihood, model selection is easily accomplished
 6284 using AIC (Burnham and Anderson, 2002) which we demonstrate below. The
 6285 AIC of a model is simply twice the negative log-likelihood evaluated at the MLE,
 6286 penalized by the number of parameters (np) in the model:

$$\text{AIC} = -2\log L(\hat{\theta}|\mathbf{y}) + 2np$$

6287 Models with small values of AIC are preferred. It is common to use a modified
 6288 (“corrected”) AIC referred to as AIC_c for small sample sizes which is

$$\text{AIC}_c = -2\log L(\hat{\theta}|\mathbf{y}) + \frac{2np(np+1)}{n-np-1}$$

6289 where n is the sample size. Two important problems with the use of AIC and
 6290 AIC_c are that they don’t apply directly to hierarchical models that contain random
 6291 effects, unless they are computed directly from the marginal likelihood (for SCR
 6292 models we can do this, see Chapt. 5). Moreover, it is not clear what should be the
 6293 effective sample size n in calculation of AIC_c , as there can be covariates that affect

6294 individuals, that vary over time, or space. We do not offer strict guidelines as to
 6295 when to use a small sample size adjustment.

6296 The **R** package **secr** computes and outputs AIC automatically for each model
 6297 fitted and it provides some capabilities for producing a model selection table (func-
 6298 tion **AIC**) and also doing model-averaging (function **model.average**), which we
 6299 recommend for obtaining estimates of density from multiple models.

6300 7.1.1 AIC analysis of the wolverine data

6301 We provide an example of model selection for the wolverine camera trapping data
 6302 using **secr**. We consider a model set with distinct models to accommodate various
 6303 types of sex specificity of model parameters:

- 6304 Model 0: model SCR0 with constant density and constant encounter model pa-
 6305 rameters;
- 6306 Model 1: model SCR0 with constant parameter values for both male and female
 6307 wolverines but with sex-specific density only;
- 6308 Model 2: Sex-specific density, sex-specific p_0 but constant σ ;
- 6309 Model 3: Sex-specific density, sex-specific σ but constant p_0 ;
- 6310 Model 4: Sex-specific density, sex-specific p_0 and sex-specific σ .

6311 To model sex-specific abundance (density), we use the multi-session models
 6312 provided by **secr** (introduced in Sec. 5.5.4), which allow one to model session-
 6313 specific effects on density, baseline encounter probability, p_0 (labeled g_0 in **secr**),
 6314 and also the scale parameter σ of the encounter probability model. Using this
 6315 formulation, we define the “Session” variable to be a *categorical* sex code having
 6316 value 1 or 2 (demonstrated below) and thus *session*-specific parameters represent
 6317 sex-specific parameters. For example, if we model session-specific density, D , then
 6318 this corresponds to Model 1 in our list above. We note that “Model 0” in our list
 6319 corresponds to a model where all of the encounter histories have the same session
 6320 ID. This model is one of constant density, which implies that the population sex
 6321 ratio is fixed at 0.5, i.e., $\psi_{\text{sex}} = 0.5$.

6322 Although **secr** also uses the logit/log linear predictors as the default for mod-
 6323 eling covariates on baseline encounter probability and the scale parameter, respec-
 6324 tively, **secr** does something different with the multi-session models. It reports
 6325 estimates in a *session mean* parameterization (equivalent to, in **BUGS**, using an
 6326 index variable instead of a set of dummy variables), and not the *session effect* (i.e.,
 6327 deviation from the intercept) which arises from the use of dummy variables. We
 6328 show this **BUGS** model description in Sec. 7.2.2.

6329 To fit these models using **secr**, we load the wolverine data and do a slight bit
 6330 of formatting to prepare the data objects for analysis by **secr**. The key differ-
 6331 ence from our analysis in Chapt. 5 is, here, we use the wolverine sex information
 6332 (**wolverine\$wsex**) which is a binary 0/1 variable (1=male) and we add 1 so that
 6333 we can define a categorical “Session” variable (having values 1 or 2). We also have

6334 a function `scr2secr` which converts a standard trap-deployment file (TDF) matrix
 6335 into a `secr` object of class “traps.” The **R** commands are as follows (contained in
 6336 the help file `?secr_wolverine`):

```

6337
6338 > library(secr)
6339 > library(scrbook)
6340 > data(wolverine)
6341 > traps <- as.matrix(wolverine$wtraps)
6342
6343 ## Name variables as required by secr
6344 > dimnames(traps) <- list(NULL,c("trapID","x","y",paste("day",1:165,sep="")))
6345 ## Convert trap information to a secr "traps" object
6346 > trapfile <- scr2secr(scrtraps=traps,type="proximity")
6347
6348 ## Grab the wolverine state-space grid (2km here)
6349 > gr <- as.matrix(wolverine$grid2)
6350 > dimnames(gr) <- list(NULL,c("x","y"))
6351 > gr2 <- read.mask(data=gr)
6352
6353 ## Grab the encounter data, and re-name variables
6354 > wolv.dat <- wolverine$wcaps
6355 > dimnames(wolv.dat) <- list(NULL,c("Session","ID","Occasion","trapID"))
6356
6357 ## Convert binary 0/1 sex variable to categorical 1/2 for "session"
6358 > wolv.dat[,1] <- wolverine$wsex[wolv.dat[,2]]+1
6359 > wolv.dat <- as.data.frame(wolv.dat)
6360
6361 ## Convert to capthist object
6362 > wolvcapt <- make.capthist(wolv.dat,trapfile,fmt="trapID",noccasions=165)
```

6363 Once the data have been prepared in this way, we use the `secr` model fitting
 6364 function `secr.fit` to fit the different models, and then the function `AIC` to pack-
 6365 age the models together and summarize them in the form of an AIC table, with
 6366 rows of the table ordered from best to worst. The function `model.average` per-
 6367 forms AIC-based model-averaging of the parameters specified by the `realnames`
 6368 variable (below this is demonstrated for the parameter density, D). Because this
 6369 function defaults to averaging by AIC_c , we slightly modified this function (called
 6370 `model.average2`) to do model averaging by either AIC or AIC_c as specified by the
 6371 user. The model fitting commands look like this (for Model 0 and Model 1):

```

6372 > model0 <- secr.fit(wolvcapt, model=list(D~1, g0~1, sigma~1),
6373                         buffer=20000)
6374 > model1 <- secr.fit(wolvcapt, model=list(D~session, g0~1, sigma~1),
6375                         buffer=20000)
```

6376 Next we use the function `AIC`, passing the fit objects from all 5 models, and
 6377 that produces the following output (abbreviated horizontally to fit on the page):

```

6378 > AIC (model0,model1,model2,model3,model4)
6379   model      ... npar logLik    AIC    AICc dAICc  AICwt
6380 model0 D~1 g0~1 sigma~1 ... 3 -627.2603 1260.521 1261.932 0.000 0.5831
6381 model2   ..     ... 5 -624.9051 1259.810 1263.810 1.878 0.2280
6382 model1   ..     ... 4 -627.2365 1262.473 1264.973 3.041 0.1275
6383 model4   ..     ... 6 -624.6632 1261.326 1267.326 5.394 0.0393
6384 model3   ..     ... 5 -627.2358 1264.472 1268.472 6.540 0.0222

```

6385 Model averaging the results is done as follows:

```

6386 > model.average (model0,model1,model2,model3,model4,realnames="D")
6387   estimate SE.estimate      lcl      ucl
6388 session=1 2.707190e-05 7.913577e-06 1.544474e-05 4.745224e-05
6389 session=2 2.927423e-05 8.270402e-06 1.700631e-05 5.039193e-05

```

6390 As usual, estimates and standard errors of the individual model parameters can
 6391 be obtained from the **secr.fit** summary output of any of the **modelX** objects shown
 6392 above. The default output of estimated density is in individuals per ha, so we have
 6393 to scale this up to something more reasonable. To get into units of per 1000 km²,
 6394 we need to first multiply by 100 to get to units of km² and then multiply by 1000.
 6395 This produces an estimated density of about 2.71 for **session=1** (females) and 2.93
 6396 for **session=2** (males). We can use the generic **R** function **predict** applied to the
 6397 **secr.fit** output to obtain specific information about the MLEs on the natural
 6398 scale.

6399 We don't necessarily agree with the use of AIC_c here and think its better to
 6400 use AIC, in general. This is because, as noted previously, it is not clear what the
 6401 effective sample size is for most capture-recapture problems. While we have 21
 6402 individuals in the data set, most of the model structure has to do with encounter
 6403 probability samples and for that there are hundreds of observations. We do note
 6404 that the AIC and AIC_c results are not entirely consistent. By looking at the best
 6405 model by AIC (Table 7.1), we find that the model with sex specific density and
 6406 sex-specific baseline encounter probability, p_0 , is preferred (Model 2). This is just
 6407 slightly better than the null model (Model 0) with no sex effects at all and hence
 6408 an implied fixed sex ratio of $\psi_{sex} = 0.50$.

6409 We fit the same models but now using a modified state-space which excludes
 6410 the ocean (this is a habitat mask in **secr**). Results are shown in Table 7.1 along
 6411 with the previous models without a mask. We see AIC values are smaller for the
 6412 model without the mask. It is probably acceptable to compare these different fits
 6413 (with and without habitat mask) by AIC because we recognize the mask as having
 6414 the effect of modifying the random effects distribution (i.e., of the activity centers,
 6415 **s**) and the results should be sensitive to choice of the distribution for **s**. That said,
 6416 we tend to prefer the mask model because it makes sense to exclude the areas of
 6417 open water from the state-space of **s**. For females the model-averaged density estimate
 6418 is 3.88 individuals per 1000 km² and for males the model-averaged density estimate
 6419 is 4.46 individuals per 1000 km² as we see here:

Table 7.1. Model selection results for the wolverine models of sex specificity, with/without habitat mask. Fitting was done using `secr` with a half-normal (Gaussian) encounter probability model. Models are ordered by *AIC*. Density, *D*, is reported in units of individuals per 1000 km². Model abbreviations indicate which parameters are sex-specific in order $D/p_0/\sigma$.

NO HABITAT MASK									
model	npar	AIC	AICc	D	Female		Male		
					p_0	σ	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	AIC	AICc	D	Female		Male		
					p_0	σ	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

```

6420 > model.average (model0b,model1b,model2b,model3b,model4b,realnames="D")
6421
6422      estimate   SE.estimate        lcl        ucl
6423 session=1 3.876615e-05 1.189102e-05 2.153795e-05 6.977518e-05
6424 session=2 4.459658e-05 1.323696e-05 2.523280e-05 7.882022e-05

```

6425 This is quite a bit higher than that based on the rectangular state-space (i.e., not
6426 specifying a habitat mask). This is not surprising given that **the state-space is**
6427 **part of the model** and the specific state-space modification we made here, which
6428 reduces the area from the rectangular state-space, should be extremely important
6429 from a biological standpoint (i.e., wolverines are not actively using open ocean).

7.2 BAYESIAN MODEL SELECTION

6430 Model selection is somewhat less straightforward as a Bayesian, and there is no
6431 canned all-purpose method like AIC. As such we recommend a pragmatic approach,
6432 in general, for all problems, based on a number of basic considerations:

- 6433 (1) For a small number of fixed effects we think it is reasonable to adopt a con-
6434 ventional “hypothesis testing” approach – i.e., if the posterior for a parameter
6435 overlaps zero substantially, then it is probably reasonable to discard that effect
6436 from the model.
- 6437 (2) Calculation of posterior model probabilities: In some cases we can implement
6438 methods which allow calculation of posterior model probabilities. One such idea

6439 is the indicator variable selection method from Kuo and Mallick (1998). For this,
 6440 we introduce a latent variable $w \sim \text{Bern}(.5)$ and expand the model to include the
 6441 variable w as follows:

$$\text{logit}(p_{ijk}) = \alpha_0 + w * \alpha_1 * C_{ijk}.$$

6442 The importance of the covariate C is then measured by the posterior probability
 6443 that $w = 1$.

6444 (3) The Deviance Information Criterion (DIC): Bayesian model selection is now
 6445 routinely carried out using DIC ((Spiegelhalter et al., 2002)), although its effec-
 6446 tiveness in hierarchical models depends very much on the manner in which it is
 6447 constructed (Millar, 2009). We recommend using it if it leads to sensible results,
 6448 but we think it should be calibrated to the extent possible for specific classes of
 6449 models. This has not yet been done in the literature for SCR models, to our
 6450 knowledge.

6451 (4) Logical argument: For something like sex specificity of certain parameters,
 6452 it seems to make sense to leave an extra parameter in the model no matter
 6453 what because, biologically, we might expect a difference (e.g., home range size).
 6454 In some cases failure to apply logical argument leads to meaningless tests of
 6455 gratuitous hypotheses (Johnson, 1999).

6456 In all modeling activities, as in life itself, the use of logical argument should not be
 6457 under-utilized.

6458 7.2.1 Model selection by DIC

6459 The availability of AIC makes the use of likelihood methods convenient for problems
 6460 where likelihood estimation is achievable. For Bayesian analysis, DIC seemed like
 6461 a general-purpose equivalent, at least for a brief period of time after its invention.
 6462 However, there seem to be many variations of DIC, and a consistent version is not
 6463 always reported across computing platforms. Even statisticians don't have general
 6464 agreement on practical issues related to the use of DIC (Millar, 2009). Despite
 6465 this, it is still widely reported. We think DIC is probably reasonable for certain
 6466 classes of models that contain only fixed effects, or for which the latent variable
 6467 structure is the same across models so that only the fixed effects are varied (this
 6468 covers many SCR model selection problems). However, it would be useful to see
 6469 some calibration of DIC for some standardized model selection problems.

6470 Model deviance is defined as negative twice the log-likelihood; i.e., for a given
 6471 model with parameters θ : $\text{Dev}(\theta) = -2 * \text{logL}(\theta|\mathbf{y})$. The DIC is defined as the
 6472 posterior mean of the deviance, $\overline{\text{Dev}}(\theta)$, plus a measure of model complexity, p_D :

$$\text{DIC} = \overline{\text{Dev}}(\theta) + p_D$$

6473 The standard definition of p_D is

$$p_D = \overline{\text{Dev}}(\theta) - \text{Dev}(\bar{\theta})$$

6474 where the 2nd term is the deviance evaluated at the posterior mean of the model
 6475 parameter(s), $\bar{\theta}$. The p_D here is interpreted as the effective number of parameters in
 6476 the model. Gelman et al. (2004) suggest a different version of p_D based on one-half
 6477 the posterior variance of the deviance:

$$p_V = \text{Var}(\text{Dev}(\theta)|\mathbf{y})/2.$$

6478 This is what is produced from **WinBUGS** and **JAGS** if they are run from **R2WinBUGS**
 6479 or **R2jags**, respectively. It is less easy to get DIC summaries from **rjags**, so we
 6480 used **R2jags** in our analyses below.

6481 7.2.2 DIC analysis of the wolverine data

6482 We repeated the analysis of the wolverine models with sex specificity, but this time
 6483 doing a Bayesian analysis paralleling the likelihood analysis we did above in **secr**,
 6484 using the logit/log parameterization of the model parameters. To do so in **BUGS**,
 6485 we used dummy variables. Thus, we can express models allowing for sex specificity
 6486 using a dummy variable **Sex** and new parameters $(\alpha_{sex}, \beta_{sex})$ which represent the
 6487 effect of **Sex** at level 1:

$$\text{logit}(p_{0,i}) = \alpha_0 + \alpha_{sex}\mathbf{Sex}_i$$

6488 and

$$\log(\sigma_i) = \log(\sigma_0) + \beta_{sex}\mathbf{Sex}_i.$$

6489 In these expressions, the sex variable \mathbf{Sex}_i is a binary variable where $\mathbf{Sex}_i = 0$
 6490 corresponds to female, and $\mathbf{Sex}_i = 1$ corresponds to male.

6491 Unlike the multi-session model in **secr**, we carry out the analysis of the sex-
 6492 specific model here by putting all of the data into a single data set, and explicitly
 6493 accounting for the covariate 'sex' in the model by assigning it a Bernoulli prior
 6494 distribution with ψ_{sex} being the proportion of males in the population. In this
 6495 case, we produce "Model 0" above, the model with no sex effect on density, by
 6496 setting the population proportion of males at one-half: $\psi_{sex} = 0.5$ (see also Sec.
 6497 6.2.4). As usual, handling of missing values of the sex variable is done seamlessly
 6498 which might be a practical advantage of Bayesian analysis in situations where sex
 6499 is difficult to record in the field which may lead to individuals of unknown sex (i.e.,
 6500 missing values).

6501 The **BUGS** model specification for the most complex model, Model 4, is shown
 6502 in Panel 7.1. This model has sex-specific intercept, scale parameter, σ , and density.
 6503 We provide an **R** script named **wolvSCROms** in the **scrbook** package which will fit
 6504 each model. The function uses **JAGS** by default for the fitting, using the **R2jags**
 6505 package. The kernel of this function is the model specification in Panel 7.1, which
 6506 gets modified depending on the model we wish to fit using a command line option
 6507 **model**. For example, **model = 1** fits the model with constant parameter values for
 6508 males and females, but sex-specific population sizes (**model = 0** constrains the male
 6509 probability parameter, ψ_{sex} , to be 0.5). The **R** function fits each of the 5 models

```

alpha.sex ~ dunif(-3,3)                      ## Prior distributions
beta.sex ~ dunif(-3,3)
sigma0 ~ dunif(0,50)
alpha0 ~ dnorm(0,.1)
psi ~ dunif(0,1)                             ## Data augmentation parameter
psi.sex ~ dunif(0,1)                          ## Probability of 'male'

for(i in 1:M){
  wsex[i] ~ dbern(psi.sex)                  ## Latent sex state (male = 1)
  z[i] ~ dbern(psi)                         ## DA variables
  s[i,1] ~ dunif(Xl,Xu)
  s[i,2] ~ dunif(Yl,Yu)
  logit(p0[i]) <- alpha0 + alpha.sex*wsex[i]
  log(sigma.vec[i]) <- log(sigma0) + beta.sex*wsex[i]
  alpha1[i] <- 1/(2*sigma.vec[i]*sigma.vec[i])
  for(j in 1:ntraps){
    mu[i,j] <- z[i]*p[i,j]
    y[i,j] ~ dbin(mu[i,j],K[j])
    dd[i,j] <- pow(s[i,1] - traplocs[j,1],2) + pow(s[i,2] - traplocs[j,2],2)
    p[i,j] <- p0[i]*exp( - alpha1[i]*dd[i,j] )
  }
}

```

Panel 7.1: Part of the **BUGS** specification for a complete sex specificity of model parameters. This is a simplified version of the model contained in the `wolvSCR0ms` script, because it does not contain the on/off switches for creating the various sub-models.

6510 using a binary indicator variable to turn ‘on’ or ‘off’ each effect. Here is how we
 6511 obtain the MCMC output for each of the 5 models:

```

6512 > wolv0 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=0)
6513 > wolv1 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=1)
6514 > wolv2 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=2)
6515 > wolv3 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=3)
6516 > wolv4 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=4)

```

6517 We fitted the 5 models to the wolverine data and summarize the DIC computa-
 6518 tion results in Table 7.2. The model rank has model 0, model 2, model 1, model 4,
 6519 model 3. Interestingly, this is the same order as the models based on AIC_c which
 6520 we found above (see Table 7.1). The posterior mean and SD of model parameters
 6521 under the 5 models are given in Table 7.3.

Table 7.2. DIC results for the 5 models of sex specificity fitted to the wolverine camera trapping data, using the function `wolvSCR0ms`. Results are based on 3 chains of length 61000 yielding 180000 posterior samples.

	meandev	pd	DIC	rank
Model 0	441.01	77.09	518.10	1
Model 1	441.78	77.504	519.28	3
Model 2	440.12	78.440	518.56	2
Model 3	443.31	79.478	522.79	5
Model 4	441.24	80.078	521.32	4

Table 7.3. Posterior summaries of model parameters for models with varying sex specificity of model parameters. Model 0 = no sex specificity, model 4 = fully sex-specific (see text). Models are based on the Gaussian encounter probability model, each with 21000 iterations, 1000 burn-in, 3 chains for a total of 60000 posterior samples.

Parameter	model 0		model 1		model 2		model 3		model 4	
	Mean	SD								
N	60.02	11.91	60.24	11.93	59.37	11.97	59.67	11.97	58.77	11.75
D	5.79	1.15	5.81	1.15	5.72	1.15	5.75	1.15	5.66	1.13
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	0.30	0.07	0.30	0.07	0.30	0.07	0.30	0.07	0.30	0.07
ψ_{sex}	0.50	0.29	0.52	0.10	0.56	0.10	0.52	0.11	0.54	0.11
deviance	441.01	12.42	441.78	12.45	440.12	12.53	443.31	12.61	441.24	12.66
	pD = 77.1		pD = 77.5		pD = 78.4		pD = 79.5		pD = 80.1	
	DIC = 518.1		DIC = 519.3		DIC = 518.6		DIC = 522.8		DIC = 521.3	

7.2.3 Bayesian model averaging with indicator variables

A convenient way to deal with model selection and averaging problems in Bayesian analysis by MCMC is to use the method of model indicator variables (Kuo and Mallick, 1998). Using this approach, we expand the model to include a set of pre-specified models as specific reductions of a larger model. This has been demonstrated in some specific capture-recapture models in Royle and Dorazio (2008, Sec. 3.4.3), and Royle (2009) and in the context of SCR by Tobler et al. (2012). A useful aspect of this method is that model-averaged parameters are produced by default. We emphasize the need to be careful of reporting model-averaged parameters that don't have a common interpretation in 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

6541 the latent indicator variables, say w_m , for variable m in the model, such that

$$w_m = \begin{cases} 1 & \text{linear predictor includes covariate } m \\ 0 & \text{linear predictor does not include covariate } m \end{cases}$$

6542 We assume that the indicator variables w_m are mutually independent with

$$w_m \sim \text{Bernoulli}(0.5)$$

6543 for each variable $m = 1, 2, \dots$, in the model. For example, with 2 variables, the
6544 expanded model has the linear predictor:

$$\text{logit}(p_{ijk}) = \alpha_0 + \alpha_1 w_1 C_{1,i} + \alpha_2 w_2 C_{2,ijk}$$

6545 where, let's suppose, $C_{1,i}$ is an individual covariate such as sex, and $C_{2,ijk}$ is a
6546 behavioral response covariate which is individual-, trap-, and occasion-specific. We
6547 can assume a parallel model specification on the parameter σ which is liable to vary
6548 by individual level covariates such as sex:

$$\log(\sigma_i) = \beta_0 + \beta_1 w_3 C_{1,i}.$$

6549 Using this indicator variable formulation of the model selection problem we
6550 can characterize unique models by the sequence of w variables. In this case, each
6551 unique sequence (w_1, w_2, w_3) represents a model, and we can tabulate the posterior
6552 frequencies of each model by post-processing the MCMC histories of (w_1, w_2, w_3) ,
6553 as we demonstrate shortly. This method then evaluates all possible combinations
6554 of covariates or 2^m models.

6555 Conceptually, analysis of this expanded model within the data augmentation
6556 framework does not pose any additional difficulty. One broader, technical consider-
6557 ation is that posterior model probabilities are well known to be sensitive to priors
6558 on parameters (Aitkin, 1991; Link and Barker, 2006). See also Royle and Dorazio
6559 (2008, Sec. 3.4.3) and Link and Barker (2010, Sec. 7.2.5). What might normally
6560 be viewed as vague or non-informative priors, are not usually innocuous or unin-
6561 formative when evaluating posterior model probabilities. The use of AIC seems
6562 to avoid this problem largely by imposing a specific and perhaps undesirable prior
6563 that is a function of the sample size (Kadane and Lazar, 2004). One solution is to
6564 compute posterior model probabilities under a model in which the prior for param-
6565 eters is fixed at the posterior distribution under the full model (Aitkin, 1991). At
6566 a minimum, one should evaluate the sensitivity of posterior model probabilities to
6567 different prior specifications.

6568 Analysis of the wolverine data

6569 The R script `wolvSCR0ms` in the package `scrbook` provides the model indicator
6570 variable implementation for the fully sex-specific SCR model. It is run by setting
6571 `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:

```
> mod <- wolv5$BUGSoutput$sims.list$mod
> mod <- paste(mod[,1],mod[,2],mod[,3],sep="")
>
> table(mod)
mod
 000   001   010   011   100   101   110   111
17181  4935  1057   296 25211  8337  2275    708
6593
6594 > round( table(mod)/length(mod) , 3)
mod
 000   001   010   011   100   101   110   111
6596  0.286  0.082  0.018  0.005  0.420  0.139  0.038  0.012
6597
```

This results in a comparison of all 8 possible models (based on $m = 3$ covariates) instead of just the 5 models we originally proposed. We see that the best model is that labeled 100 which, according to our construction above, has `mod[1]=1`, `mod[2]=0` and `mod[3]=0`. This is the model having sex-specific baseline encounter probability p_0 , and $\psi_{\text{sex}} = 0.5$. This model has posterior model probability 0.420. The model with no sex specificity at all (the model with label 000) has posterior probability 0.286 and the remaining posterior mass is distributed over the other six models. We could arrive at a qualitatively similar conclusion using a more ad hoc approach based on looking at the posterior mass for each parameter under the full model (model 4; see Table 7.3, in part). Considering the sex-specific intercept, it appears to be very important as its posterior mass is mostly away from 0. On the other hand, the coefficient on log-sigma is concentrated around 0, and the estimated ψ_{sex} (probability that an individual is a male) is 0.54 with a large posterior standard

deviation. We might therefore be inclined to discard the sex effect on $\log(\sigma)$ based on classical thinking-like-a-hypothesis-testing-person and settle for the model with a sex-specific intercept in the encounter probability model. This is consistent with our indicator variable approach which found that model (1,0,0) has posterior probability of 0.420. Looking at the posteriors for each parameter to thin the model down is consistent with these results. We can obtain model-averaged estimates from the indicator variable approach, which produces direct model-averaged estimates of N and D :

```
6619   mu.vect sd.vect    2.5%    25%    50%    75%  97.5% Rhat n.eff
6620 D      5.695  1.133  3.759  4.916  5.591  6.362  8.193 1.002 3600
6621 N     59.077 11.758 39.000 51.000 58.000 66.000 85.000 1.002 3600
```

We obtain a model-averaged estimate (posterior mean) for density of $D = 5.695$ which is hardly any different from our model specific estimates (Table 7.3) and, in particular, from model 2 which has only a sex-specific intercept.

6625 7.2.4 Choosing among detection functions

6626 Another approach to implementing model indicator variables is to introduce a categorical “model identity” variable which is itself a parameter of the model. Using
 6627 this approach, then each distinct model is associated with a unique set of covariates
 6628 or other set of model features. This is convenient especially when we cannot spec-
 6629 ify the linear predictor as some general model that reduces to various alternative
 6630 sub-models simply by switching binary variables on or off. In the context of SCR
 6631 models, choosing among different encounter probability models would be an exam-
 6632 ple. For this case we do something like this `mod ~ dcat(probs[])` where `probs` is
 6633 a vector with elements $1/(\#models)$, and the encounter probability matrix is filled
 6634 in depending on the value of `mod`. In particular, instead of a 2-dimensional array
 6635 `p[i,j]`, we build `p[i,j,m]` for each of $m = 1, 2, \dots, M$ models. An example with
 6636 3 distinct models is:

```
6638 mod ~ dcat(probs[])
6639 ##
6640 ## Using a double loop construction fill-in p[,] for each model:
6641 ##
6642 p[i,j,1] <- p0[1]*exp( - alpha1[1]*dist2[i,j] )
6643 p[i,j,2] <- 1-exp(-p0[2]*exp( - alpha1[2]*dist2[i,j] ) )
6644 logit(p[i,j,3]) <- p0[3] - alpha1[3]*dist2[i,j]
6645
6646 mu[i,j] <- z[i]*p[i,j,mod]
6647 y[i,j] ~ dbin(mu[i,j],K[j])
```

6648 As before the posterior probabilities can be highly sensitive to priors on the
 6649 different model parameters and sometimes mixing is really poor and, in general,
 6650 we’ve experienced mixed success trying to carry out model selection using this
 6651 construction. We do provide a template **R/JAGS** script (`wolvSCR0ms2`) in the

6652 **scrbook** package which has an example of choosing among 3 different encounter
 6653 probability models: The Gaussian encounter probability, Gaussian hazard, and
 6654 logistic model with the square of distance (defined in Sec. 6.1). The key things
 6655 to note are that there are 3 intercepts and 3 different ‘`alpha1`’ parameters (the
 6656 coefficient on distance). The parameters should not be regarded as equivalent across
 6657 the models, so it is important to have them separately defined (and estimated) for
 6658 each model. In our analysis we used a vague normal prior (precision = 0.1) for
 6659 the intercept parameter (either log or logit-scale of baseline encounter probability
 6660 p_0) and a `Uniform(0,5)` prior for one-half the inverse of the coefficient on distance-
 6661 squared. In the **BUGS** model specification the priors look like this:

```
6662 for(i in 1:3){  

  6663   alpha0[i] ~ dnorm(0,.1)  

  6664   sigma[i] ~ dunif(0,5)  

  6665   alpha1[i] <- 1/(2*sigma[i]*sigma[i])  

  6666 }
```

6667 Then, we create a probability of encounter for each individual, trap *and* model
 6668 so that the holder object “`p`” in the model description is a 3-dimensional array
 6669 (sometimes this would have to be a 4 or 5-d array in more complex models with
 6670 time effects, etc..), so that construction of the encounter probability models look
 6671 like this:

```
6672 p[i,j,1] <- p0[1]*exp( - alpha1[1]*dist2[i,j] )  

  6673 p[i,j,2] <- 1-exp(-p0[2]*exp( - alpha1[2]*dist2[i,j] ) )  

  6674 logit(p[i,j,3]) <- p0[3] - alpha1[3]*dist2[i,j]
```

6675 where

```
6676 logit(p0[1]) <- alpha0[1]  

  6677 log(p0[2]) <- alpha0[2]  

  6678 p0[3] <- alpha0[3]
```

6679 You can experiment with the `wolvSCR0ms2` script to investigate the importance of
 6680 different models of encounter probability and whether they have an affect on the
 6681 inferences.

7.3 EVALUATING GOODNESS-OF-FIT

6682 In practical settings, we estimate parameters of a desirable model, or maybe fit a
 6683 bunch of models and report estimates from all of them or a model-averaged sum-
 6684 mary of density. An important question is: Is our model worth anything? In other
 6685 words, does the model appear to be an adequate description of our data? For-
 6686 mal assessment of model adequacy or goodness-of-fit is a challenging problem and
 6687 there are no all-purpose algorithms for doing this in either frequentist or Bayesian

6688 paradigms. Moreover, there are some philosophical challenges to evaluating model
6689 fit, such as, if we do model averaging then should all of the models have to fit? Or
6690 should the averaged model have to fit? What if none of the models fit? We don't
6691 know the answers to these questions and we won't try to answer them. Instead,
6692 we will provide what guidance we can on taking the first steps to evaluating fit, of
6693 a single model, as if it were a cherished family heirloom of great importance. We
6694 suggest that if you have a model that you really like, a single model, then it is a
6695 sensible thing to check that the model is a good fit to your data. If it is not, we
6696 do not imagine that the model is useless but just that some thought should be put
6697 into why the model doesn't fit so that, perhaps, some remediation might happen
6698 as future data are collected. After all, you may have spent 2, 3 or many more years
6699 of your life collecting that data set, perhaps thousands of hours, and therefore it
6700 seems a reasonable proposition to expect to do some estimation and analysis of the
6701 model regardless of model fit. You can still learn something from a model that does
6702 not pass some technical litmus test of model fit.

6703 Conceptually, we can think of evaluation of model fit as follows: if we simulate
6704 data under the model in question, do the simulated realizations resemble the data
6705 set that we actually have? For either Bayesian or classical inference, the basic
6706 strategy to assessing model fit is to come up with a fit statistic that depends on the
6707 parameters and the data set, which we denote by $T(\mathbf{y}, \boldsymbol{\theta})$, and then we compute
6708 this for the observed data set, and compare its value to that computed for perfect
6709 data sets simulated under the correct model. In the case of classical inference, we
6710 will often rely on the standard practice of parametric bootstrapping (Dixon, 2002),
6711 where we simulate data sets conditional on the MLE $\hat{\boldsymbol{\theta}}$ and compare realizations
6712 with what we've observed. The **R** package **unmarked** (Fiske and Chandler, 2011)
6713 contains generic bootstrapping methods for certain hierarchical models, including
6714 distance sampling (e.g., see Sillett et al., 2012, for an application). In simple cases,
6715 using classical inference methods, it is sometimes possible to identify a test statistic
6716 of theoretical merit, perhaps with a known asymptotic distribution. For examples
6717 from capture-recapture see Burnham et al. (1987), Lebreton et al. (1992), and
6718 Chapt. 5 of Cooch and White (2006). For Bayesian analysis we use the Bayesian
6719 p-value method (Gelman et al., 1996) (we introduced the Bayesian p-value in sec.
6720 ??). Using this approach, data sets are simulated based on a posterior sample of the
6721 model parameters $\boldsymbol{\theta}$ and some fit statistic for the simulated data sets, usually based
6722 on the discrepancy of the observed data from its expected values, is compared to
6723 that for the actual data. In most cases, whether Bayesian or frequentist, the main
6724 idea for assessing model fit is the same: We compare data sets from the model we're
6725 interested in with the data set we have in hand. If they appear to be consistent
6726 with one another, then our faith in the model increases, at least to some extent,
6727 and we say "the model fits."

6728 To date, we are unaware of any goodness-of-fit applications based on likelihood
6729 analysis of SCR models. For Bayesian analysis of SCR models, there has not been

6730 a definitive or general proposal for a fit statistic or even a class of fit statistics,
 6731 although a few specialized implementations of Bayesian p-values have been pro-
 6732 vided (Royle, 2009; Gardner et al., 2010a; Royle et al., 2011a; Gopalaswamy et al.,
 6733 2012a,b; Russell et al., 2012). While we universally adopt the Bayesian p-value
 6734 approach, and suggest some fit statistics in the following text, we caution that
 6735 there is no general expectation to support how well they should do. As such, one
 6736 might consider doing some kind of custom evaluation or calibration when using
 6737 such methods, if the power of the test (ability to reject under specific departures
 6738 from the model) is of paramount interest. We note that this uncertain power or
 6739 performance of the Bayesian p-value is not a weakness of the Bayesian approach
 6740 because the same issue applies in using bootstrap approaches applied to classical
 6741 analysis of models, if we were to devise such methods.

7.4 THE TWO COMPONENTS OF MODEL FIT

6742 For most SCR models, there are at least two distinct components of model fit,
 6743 and we propose to evaluate these two distinct components individually. First, we
 6744 can ask, are the data consistent with the *observation* model, conditional on the
 6745 underlying point process? We can evaluate this based on the encounter frequencies
 6746 of individuals *conditional* on (posterior samples of) the underlying point process
 6747 s_1, \dots, s_N . We discuss some potential fit statistics for addressing this in the next
 6748 section. Second, we can evaluate whether the data appear consistent with the *state*
 6749 process model (i.e., the “uniformity” assumption of the point process). For the
 6750 simple model of independence and uniformity, this is similar to the assumption of
 6751 *complete spatial randomness* (CSR) which we consider in Sec. 7.4.1 below. Actually,
 6752 this is not strictly the assumption of CSR because of the binomial assumption on
 6753 N under data augmentation, so we instead use the term *spatial randomness*.

6754 7.4.1 Testing uniformity or spatial randomness

6755 Historically, especially in ecology, there has been an extraordinary amount of inter-
 6756 est in whether a realization of a point process indicates “complete spatial random-
 6757 ness,” i.e., that the points are distributed uniformly and independently in space.
 6758 Two good references for such things are Cressie (1991, Ch. 8) and Illian et al.
 6759 (2008)¹. In the context of animal capture-recapture studies, the spatial random-
 6760 ness hypothesis is manifestly false, purely on biological grounds. Typically individ-
 6761 uals will be clustered, or more regular (for territorial species), than expected under
 6762 spatial randomness and heterogeneous habitat will generate the appearance of clus-
 6763 tering even if individuals are distributed independently of one another. While we
 6764 recommend modeling spatial structure explicitly when possible (Chaps. 10, 11,

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.

6765 12), the uniformity assumption may be an adequate description of data sets in
 6766 some situations. Further, we find that it is generally flexible enough to reflect non-
 6767 uniform patterns in the data, because we do observe some direct information about
 6768 some of the point locations.

6769 The basic technical framework for evaluating the spatial randomness hypothesis
 6770 is based on counts of activity centers in cells or bins. For that we use any standard
 6771 goodness-of-fit test statistic, based on gridding (i.e., binning) the state-space of the
 6772 point process into $g = 1, 2, \dots, G$ cells or bins, and we tabulate $N_g \equiv N(\mathbf{x}_g)$ the
 6773 number of activity centers in bin g , centered at coordinate \mathbf{x}_g . Specifically, let
 6774 $B(\mathbf{x})$ indicate a bin centered at coordinate \mathbf{x} , then² $N(\mathbf{x}) = \sum_{i=1}^N I(\mathbf{s}_i \in B(\mathbf{x}))$ is
 6775 the population size of bin $B(\mathbf{x})$. In Sec. 4.11.1, we used the summaries $N(\mathbf{x})$ for
 6776 producing density maps from MCMC output. Here, we use them for constructing
 6777 a fit statistic. We have used the Freeman-Tukey statistic of this form:

$$T(\mathbf{N}, \theta) = \sum_g (\sqrt{N_g} - \sqrt{\mathbb{E}(N_g)})^2$$

6778 where $\mathbb{E}(N_g)$ is estimated by the mean bin count. An alternative conventional
 6779 assessment of fit is based on the following statistic: Conditional on N , the total
 6780 number of activity centers in the state-space \mathcal{S} , the bin counts N_g should have a
 6781 binomial distribution. It will usually suffice to approximate the binomial cell counts
 6782 by Poisson cell counts, in which case we can use the classical “index-of-dispersion”
 6783 test (Illian et al., 2008, p. 87), based on the variance-to-mean ratio:

$$I = (G - 1) * s^2 / \bar{N}$$

6784 where s^2 is the sample variance of the bin counts and \bar{N} is the sample mean. When
 6785 the point process realization is *observed*, as in classical point pattern modeling
 6786 (but not in SCR), this statistic has approximately a Chi-square distribution on
 6787 $(G - 1)$ degrees-of-freedom under the spatial randomness hypothesis. If $s^2 / \bar{N} > 1$,
 6788 clustering is suggested whereas, $s^2 / \bar{N} < 1$ suggests the point process is too regular.

6789 Whatever statistic we choose as our basis for assessing spatial randomness, *the*
 6790 important technical issue is that we don’t observe the point process and so the
 6791 standard statistics for evaluating spatial randomness cannot be computed directly.
 6792 However, using Bayesian analysis, we do have a posterior sample of the underlying
 6793 point process and so we suggest computing the posterior distribution of any
 6794 statistic in a Bayesian p-value framework. For a given posterior draw of all model
 6795 parameters, N is known, based on the value of the data augmentation variables z_i ,
 6796 and so we can obtain a posterior sample of $N(\mathbf{x})$ by taking all of the output for
 6797 MCMC iterations $m = 1, 2, \dots$, and doing this:

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

² $I(arg)$ is the indicator function which evaluates to 1 if arg is true, otherwise 0

6798 Thus, $N(\mathbf{x})^{(1)}, N(\mathbf{x})^{(2)}, \dots$, is the Markov chain for the derived parameter $N(\mathbf{x})$.

6799 In addition to computing the bin counts for each iteration of the MCMC algorithm,
 6800 at the same time we generate a realization of the activity centers \mathbf{s}_i under the
 6801 spatial randomness model, and we obtain bin counts for these “new” data, $\tilde{N}(\mathbf{x})$.
 6802 For each of the posterior samples – that of the real data, and that of the posterior
 6803 simulated data, we compute the fit-statistic. The fit statistic based on the actual
 6804 data is:

$$T(\mathbf{N}, \theta) = \sum_x (\sqrt{N(\mathbf{x})} - \sqrt{\bar{N}(\mathbf{x})})^2$$

6805 whereas the fit statistic based on a simulated realization of points under the spatial
 6806 randomness hypothesis is:

$$T(\tilde{\mathbf{N}}, \theta) = \sum_x (\sqrt{\tilde{N}(\mathbf{x})} - \sqrt{\bar{N}(\mathbf{x})})^2$$

6807 And we compute the Bayesian p-value by tallying up the proportion of times that
 6808 $T(\tilde{\mathbf{N}}, \theta)$ is larger than $T(\mathbf{N}, \theta)$, as an estimate of: $p = \Pr(T(\tilde{\mathbf{N}}, \theta) > T(\mathbf{N}, \theta))$. The
 6809 R function `SCRgof` in our package `scrbook` will do this, given the output from
 6810 **JAGS** (see below).

6811 Sensitivity to bin size

6812 Evaluating fit based on bin counts in point process models are sensitive to the
 6813 number of bins (Illian et al., 2008, p. 87-88). This is related to the classical problem
 6814 of fit testing for binary regression because in a point process model, as the number of
 6815 grid cells gets small, the grid cell counts go to 0 or 1 and standard fit statistics (e.g.,
 6816 based on deviance or Pearson residuals) are known not to be very useful. There is
 6817 some good discussion of this in McCullagh and Nelder (1989, Sec. 4.4.5). What it
 6818 boils down to is, using the example of the Pearson residual statistic considered by
 6819 McCullagh and Nelder (1989), the fit statistic is exactly a deterministic function of
 6820 the sample size only, which clearly should not be regarded as useful for model fit.
 6821 This is why, in order to do a check of model fit when you have a binary response, one
 6822 must always aggregate the data in some fashion. In the context of testing spatial
 6823 randomness, computing the test statistic we described above has us chop up the
 6824 region \mathcal{S} into bins, and tally up N_g , the frequency of activity centers in each bin g .
 6825 Suppose that we choose the bin size to be extremely small such that $\mathbb{E}(N_g)$ tends
 6826 to N/G (N being the number of activity centers). Further, N_g tends to a binary
 6827 outcome. Therefore the fit statistic has N components that have value $N_g = 1$,
 6828 and it has $G - N$ components that have value $N_g = 0$. Therefore, the fit statistic
 6829 resembles:

$$T(\mathbf{N}, \theta) = \sum_{g \ni N_g=1}^N (1 - \sqrt{N/G})^2 + \sum_{g \ni N_g=0}^{G-N} (N/G)^2 = N(1 + (G - N)/G)$$

(here \ni means “such that”). If G is huge relative to N , then we see that this tends to about $2*N$, which does not provide any meaningful assessment of model fit. So if you look at this in the limit in which the bin counts become binary, the fit statistic loses all its variability to the specific model used and is just a deterministic function of N . As a practical matter, it probably makes sense to restrict the number of bins to *fewer* than the number of observed individuals in the sample size. In typical SCR applications this will therefore result, usually, in very large (and few) bins, and presumably not much power.

There are some extensions that help resolve the issue of sensitivity to bin size. We can construct fit statistics based not just on quadrat counts but also the neighboring quadrat counts – this is the Greig-Smith method (Greig-Smith, 1964). In addition, there are a myriad of “distance methods” for evaluating point process models, and we believe that many of these can (and will) be adapted to SCR models. Again the main feature is that the point process on which inference is focused is completely latent in SCR models – so this makes the fit assessment slightly different than in classical point processes. That said, the methods should be adaptable, e.g., in a Bayesian p-value kind of way.

Sensitivity to state-space extent

An issue that we have not investigated is that any model assessment that applies to a *latent* point process is probably sensitive to the size of the state-space. As the size of the state-space increases then the cell counts (far away from the data) *are* independent binomial counts with constant density, and so we can overwhelm the fit statistic with extraneous “data” simulated from the posterior, which is equal to the prior as we move away from the data, and therefore uninformed by the observed data that live in the vicinity of the trap array. Therefore we recommend computing these goodness-of-fit statistics in the vicinity of the trap array only. Perhaps, as an ad hoc rule-of-thumb, less than the average trap spacing from the rectangle enclosing the trap array. For example, if the average trap spacing is, say, 10 km, then the bins used to obtain the observed and predicted activity centers should not extend any further from the traps than 5 km. This should be a matter of future research.

7.4.2 Assessing fit of the observation model

In evaluating the spatial randomness hypothesis, we could draw on well-established ideas from point process modeling. On the other hand, it is less clear how to approach goodness-of-fit evaluation of the observation model. For most SCR problems, we have a 3-dimensional data array of *binary* observations, y_{ijk} for individual i , trap j and sample occasion k . As discussed in the previous section, we need to construct fit statistics based on observed and expected frequencies that are aggregated in some fashion. In practice, the data will be too sparse to have much power, unless the data are highly aggregated. We recommend focusing on summary

statistics that represent aggregated versions of y_{ijk} over 1 or 2 of the dimensions. We describe 3 such fit statistics below. We recognize that, depending on the model, some information about model fit will be lost by summarizing the data in this way. For example if there is a behavioral response and we aggregate over time to focus on the individual and trap level summaries then some information about lack of fit due to temporal structure in the data is lost.

Fit statistic 1: individual x trap frequencies We summarize the data by individual and trap-specific counts y_{ijk} aggregated over all sample occasions. Using standard “dot notation” to represent summed quantities, we express that as: $y_{ij\cdot} = \sum_{k=1}^K y_{ijk}$. Conditional on \mathbf{s}_i , the expected value under any encounter model is:

$$\mathbb{E}(y_{ij\cdot}) = p_{ij} K$$

(or K_j if the traps are operational for variable periods). If there is time-varying structure to the model, then expected values would have to be computed according to $\mathbb{E}(y_{ij\cdot}) = \sum_k p_{ijk}$. Then we can define a fit statistic from the Freeman-Tukey residuals according to:

$$T_1(\mathbf{y}, \theta) = \sum_i \sum_j (\sqrt{y_{ij\cdot}} - \sqrt{\mathbb{E}(y_{ij\cdot})})^2$$

where we use θ here to represent the collection of all parameters in the model. This is conditional on \mathbf{s} as well as on the data augmentation variables \mathbf{z} . We compute this statistic for *each* iteration of the MCMC algorithm for the observed data set and also for a new data set simulated from the posterior distribution, say $\tilde{\mathbf{y}}$.

We could also use a similar fit statistic derived from summarizing over traps to obtain an $n_{ind} \times K$ matrix of count statistics. We imagine that either summary of the data will probably be too disaggregated (have mostly values of 0) in most practical settings to have much power.

Fit statistic 2: Individual encounter frequencies. SCR models represent a type of model for heterogeneous encounter probability, like model M_h , but with an explicit factor (space) that explains part of the heterogeneity. For model M_h , the individual encounter frequencies are the sufficient statistic for model parameters, and so it makes intuitive sense to provide some kind of omnibus fit assessment of the core heuristic that SCR model is adequately explaining the heterogeneity using a model M_h -like statistic based on individual encounter frequencies. So, we build a fit statistic based on the individual total encounters (Russell et al., 2012), $y_{i..} = \sum_j \sum_k y_{ijk}$. In addition, the expected value is a similar summary over traps and occasions: $\mathbb{E}(y_{i..}) = \sum_j \sum_k p_{ijk}$. Then, we define statistic T_2 according to:

$$T_2(\mathbf{y}, \theta) = \sum_i (\sqrt{y_{i..}} - \sqrt{\mathbb{E}(y_{i..})})^2$$

We imagine this test statistic should provide an omnibus test of extra-binomial variation and should therefore capture some effect of variable exposure to encounter

of individuals, although we have not carried out any evaluations of power under specific alternatives. Obviously, in using this statistic, we lose information on departures from the model that might only be trap- or time-specific.

Fit Statistic 3: Trap frequencies. We construct an analogous statistic based on aggregating over individuals and replicates to form trap encounter frequencies: $y_{j\cdot} = \sum_i \sum_k y_{ijk}$ (Gopalswamy et al., 2012b) and the expected value is a similar summary over individuals and occasions: $\mathbb{E}(y_{j\cdot}) = \sum_i \sum_k p_{ijk}$. Then statistic T_3 is:

$$T_3(\mathbf{y}, \theta) = \sum_j (\sqrt{y_{j\cdot}} - \sqrt{\mathbb{E}(y_{j\cdot})})^2$$

This seems like a sensible fit statistic because we can think of SCR models as spatial models for counts (Chandler and Royle, In press). Therefore, we should seek models that provide good predictions of the observable spatial data, which are the trap totals. In this context, it might even make sense to pursue cross-validation based methods for model selection. Cross-validation is a standard method of evaluating models such as in kriging or spline smoothing, so we could as well develop such ideas based on the trap-specific frequencies.

7.4.3 Does the SCR model fit the wolverine data?

We use the ideas described in the previous section to evaluate goodness-of-fit of the SCR model to the wolverine camera trapping data.

We consider first whether the simple model of spatial randomness of the activity centers is adequate. We think that the encounter model shouldn't have a large effect on whether the spatial randomness assumption is adequate or not, so we fit "Model 0" (in which parameters are *not* sex-specific) using an **R** script provided in the function `wolvSCR0gof` which will default to fitting the model in **JAGS**. This is the same script as `wolvSCR0ms` except that it saves the MCMC output for the activity centers `s` and the data augmentation variables `z`, which are required in order to compute the Bayesian p-value test of spatial randomness.

The MCMC output is processed with the **R** function `SCRgof` which computes the test of spatial randomness based on bin counts, using the Bayesian p-value calculation. The function `SCRgof` requires a few things as inputs: (1) the output from a **BUGS** run (in particular, the activity center coordinates and the data augmentation variables); (2) the number of bins to create for computing spatial frequencies of activity centers; (3) the trap locations and, (4) the buffer around the trap array to use in computing the bin counts. This buffer could be that used in defining the state-space for the model fitting, but we think it should be relatively tighter to the trap array than the state-space used in model-fitting. For the wolverine analysis, where we're using 10-km grid cells (1 unit = 10 km) and a 20 km buffer for model fitting, we'll use a state-space buffer of 0.4 units (4 km) for computing the fit statistic. The **R** code to fit the model and obtain the goodness-of-fit result is as follows:

```

6944 > wolv1 <- wolvSCR0gof(nb=1000,ni=6000,buffer=2,M=200,model=0)
6945
6946 > bugsout <- wolv1$BUGSoutput$sims.list
6947
6948 > traplocs <- wolverine$wtraps[,2:3]
6949 > traplocs[,1] <- traplocs[,1] - min(traplocs[,1])
6950 > traplocs[,2] <- traplocs[,2] - min(traplocs[,2])
6951 > traplocs <- traplocs/10000
6952
6953 > set.seed(2013) # set seed so Bayesian p-value is the same each time
6954
6955 > SCRgof(bugsout,5,5,traplocs=traplocs,buffer=.4)
6956
6957 Cluster index observed: 1.099822
6958 Cluster index simulated: 1.000453
6959 P-value index of dispersion: 0.408
6960 P-value2 freeman-tukey: 0.6842667

```

6961 The output produced by **SCRgof** is the cluster index based on the ratio of the
6962 variance to the mean (see above), which is computed as the posterior mean index
6963 of dispersion for the latent point process, and also the average value for simulated
6964 data. If this value is > 1 then clustering is suggested, which we see a (very) minor
6965 amount of evidence for here. Two Bayesian p-values are produced: the first is
6966 based on the cluster index, and the 2nd is based on the Freeman-Tukey statistic
6967 calculated as described in Sec. 7.4.1. Because our p-values aren't close to 0 or
6968 1, we judge that the model of spatial randomness provides an adequate fit to the
6969 data. You can verify that a similar result is obtained if we use the model with fully
6970 sex-specific parameters (Model 4).

6971 Next, we did a Bayesian p-value analysis of the observation component of the
6972 model, using the 3 fit statistics described in Sec. 7.4.2. These statistics can be
6973 calculated as part of the **BUGS** model specification or by post-processing the
6974 MCMC output returned from a **BUGS** run. The **R** script **wolvSCR0gof** contains
6975 the relevant calculations. For example, to compute fit statistic 1, we have to add
6976 some commands to the **BUGS** model specification such as this (note: this is only
6977 a fraction of the model specification):

```

6978 .....
6979 for(j in 1:ntraps){
6980   mu[i,j] <- w[i]*p[i,j]
6981
6982   y[i,j] ~ dbin(mu[i,j],K[j])
6983   ynew[i,j] ~ dbin(mu[i,j],K[j])
6984
6985   err[i,j] <- pow(pow(y[i,j],.5) - pow(K[j]*mu[i,j],.5),2)
6986   errnew[i,j] <- pow(pow(ynew[i,j],.5) - pow(K[j]*mu[i,j],.5),2)

```

```

6987 }
6988
6989 T1obs <- sum(err[,])
6990 T1new <- sum(errnew[,])
6991 .....
6992 Similar calculations are carried out to obtain the posterior samples of test statistics
6993 2 (individual totals) and 3 (trap totals). For the wolverine data, the Bayesian
6994 p-value calculations produce:
6995 > mean(wolv1$BUGSoutput$sims.list$T1new>wolv1$BUGSoutput$sims.list$T1obs)
6996 [1] 0
6997
6998 > mean(wolv1$BUGSoutput$sims.list$T2new>wolv1$BUGSoutput$sims.list$T2obs)
6999 [1] 0.17
7000
7001 > mean(wolv1$BUGSoutput$sims.list$T3new>wolv1$BUGSoutput$sims.list$T3obs)
7002 [1] 0.02066667

```

7003 Based on statistic T_2 , we might conclude that the model is adequate for explaining individual heterogeneity although the other two statistics suggest a general lack of fit of the observation model. A similar result is obtained using the fully sex-specific model. We note that one individual was captured 8 times in one trap, which is pretty extreme under a model which assumes independent Bernoulli trials. We summarize that the trap-counts simply are not well-explained by this model.

7004 In attempt to resolve this problem, we extended the model to include a local
7005 (trap-specific) behavioral response (following Royle et al. (2011b)) which can be
7006 fitted using the sample **R** script **wolvSCRMb**. To fit a model using **WinBUGS**, and
7007 then compute the Bayesian p-values we do this:

```

7013 > wolv.Mb <- wolvSCRMb(nb=1000,ni=6000,buffer=2,M=200)
7014
7015 > mean(wolv.Mb$sims.list$T1new>wolv.Mb$sims.list$T1obs)
7016 [1] 0.9666667
7017
7018 > mean(wolv.Mb$sims.list$T2new>wolv.Mb$sims.list$T2obs)
7019 [1] 0.3644667
7020
7021 > mean(wolv.Mb$sims.list$T3new>wolv.Mb$sims.list$T3obs)
7022 [1] 0.4990667

```

7023 Given that this model seems to fit better, we might prefer reporting estimates
7024 under this model, which we do in Table 7.4. (the behavioral response parameter is
7025 labeled α_2 in the table). Estimated density is about 1 individual higher per 1000
7026 km^2 compared with the various models that lack a behavioral response. It might be
7027 useful to try these fit assessment exercises using the habitat mask as described in

7028 Sec. 4.10. That takes an extremely long time to run in **BUGS** though, especially
 7029 for the behavioral response model.

Table 7.4. Posterior summary statistics for local (trap-specific) behavioral response model M_b fitted to the wolverine camera trapping data using **WinBUGS**. The parameter 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

7.5 QUANTIFYING LACK-OF-FIT AND REMEDIATION

7030 Molinari-Jobin et al. (2013) used a strategy for assessing model fit in dynamic
 7031 occupancy models (Royle and Kéry, 2007) similar to that which we suggested above.
 7032 They constructed a fit statistic based on aggregating the data over replicate samples
 7033 (k), to obtain the total detections per site i and year j . They used a Bayesian p-
 7034 value analysis based on a Chi-squared test statistic (also see Kéry and Schaub,
 7035 2012, Chapt. 12). Their analysis suggested a model that didn't fit, and, so they
 7036 computed the "lack-of-fit ratio" (see Kéry and Schaub, 2012, Sec. 12.3) – the
 7037 ratio of the fit statistic computed for the actual data to that of the replicate data
 7038 sets. They interpret this analogous to the over-dispersion coefficient in generalized
 7039 linear models (McCullagh and Nelder, 1989), usually called the c-hat statistic in
 7040 capture-recapture literature (see Cooch and White, 2006, Chapt. 5). Molinari-
 7041 Jobin et al. (2013) reported the lack-of-fit ratio for their model to be 1.14 which
 7042 suggests a minor lack-of-fit, compared to perfect data having a value of 1, because
 7043 the posterior standard deviations will be too small by a factor of $\sqrt{1.14} = 1.07$.
 7044 In classical capture-recapture applications of goodness-of-fit assessment, inference
 7045 for non-fitting models is dealt with by inflating the resulting SEs (of the non-
 7046 fitting model), by the square-root of c-hat. We believe that these ideas related to

7047 quantifying lack-of-fit and understanding its effect could also be applied to SCR
7048 models, although we have not yet explored this.

7.6 SUMMARY AND OUTLOOK

7049 In this chapter, we offered some general strategies for model selection and model
7050 checking, or assessment of model fit. We think the strategies we outlined for model
7051 selection are fairly standard and can be effectively applied to many SCR modeling
7052 problems. Some technical issues of Bayesian analysis need to be addressed (in
7053 general) before Bayesian methods are more generally useful and accessible. For
7054 one thing, Bayesian model selection based on the indicator variable approach of
7055 Kuo and Mallick (1998) can be tediously slow even for small data sets, and so
7056 improved computation will improve our ability to do Bayesian model selection in
7057 practical situations. Also, and most importantly, sensitivity to prior distributions
7058 is an important issue. Further research and practice might identify preferred prior
7059 configurations for SCR that provide a good calibration in relevant model selection
7060 problems. Finally, we believe that cross-validation should prove to be a useful
7061 method in model assessment and selection, as SCR models are a form of spatial
7062 model of counts, and so it is natural to pick models that predict the observable
7063 spatial counts (i.e., at trap locations) well.

7064 For Bayesian model assessment, or goodness-of-fit checking, we suggested a
7065 framework based on independent testing of the spatial model of independence and
7066 uniformity, and testing fit of the observation model conditional on the underlying
7067 point process. These ideas are based on mostly *ad hoc* attempts in a number
7068 of published applications (Royle et al., 2009a, 2011a; Gopalaswamy et al., 2012b;
7069 Russell et al., 2012, e.g.). While we think this general strategy should be fruitful,
7070 we know of no studies on the power to detect various model departures, and so
7071 the ideas should be viewed as experimental. We have not discussed assessment
7072 of model fit for SCR models using likelihood methods, although we imagine that
7073 standard bootstrapping ideas should be effective, perhaps based on the fit statistics
7074 (or similar ones) we suggested here for computing Bayesian p-values.

7075 Clearly there is much research to be done on assessment of model fit in SCR
7076 models. For testing the spatial randomness hypothesis, we used a classical ap-
7077 proach based on count frequencies, in which point locations are put into spatial
7078 bins. Other approaches from spatial point process modeling should be pursued in-
7079 cluding nearest-neighbor methods or distance-based methods. In addition, studies
7080 to evaluate the power to detect relevant departures from the standard assumptions,
7081 and the robustness of inferences about N or density, need to be conducted. If the
7082 spatial randomness model appears inadequate, it is possible to fit models that al-
7083 low for a non-uniform distribution of points (see Chapt. 10) and even point process
7084 models that allow for interactions among points (Reich et al., 2012). On the other
7085 hand, we expect that most of these Bayesian p-value tests will have low power in
7086 typical data sets consisting of a few to a few dozen individuals. As such, failure to

7087 detect a lack of fit may not be that meaningful. But, on the other hand, it may not
7088 make a difference in terms of density estimates either. We think inference about
7089 density should be relatively insensitive to departures from spatial randomness, be-
7090 cause we get to observe direct information on some component of the population,
7091 component of density is *observed*. For those activity centers, the assumed model
7092 of the point process should exert little influence on the placement of the activity
7093 centers. Conversely, as is the case with classical closed population models (Otis
7094 et al., 1978; Dorazio and Royle, 2003; Link, 2003), inferences may be somewhat
7095 more sensitive to bad-fitting models for the observation process.

8**ALTERNATIVE OBSERVATION MODELS**

7099 In previous chapters we considered various models of *encounter probability*, both
7100 in terms of parametric functions of distance and also a myriad of covariate models
7101 (Chapt. 6 and elsewhere). However, we have so far only considered a specific
7102 probability model for the observations (we'll call this the "encounter process") –
7103 the Bernoulli encounter process model which, in `secr`, is the *proximity detector*
7104 model. This assumes that individual and trap-specific encounters are independent
7105 Bernoulli trials. Here, we focus on developing additional models for the encounter
7106 process. The encounter process could be thought of as being determined by the
7107 type of device – or the type of "detector" using the terminology of `secr` (Efford,
7108 2011).

7109 In this chapter, we consider alternative observation models that accommodate
7110 observations that are not binary, and do not require independence of the observations.
7111 In particular, we consider models for encounter *frequencies*, and encounter
7112 process models based on the multinomial distribution. For example, if sampling
7113 devices can detect an individual some arbitrary number of times during an interval,
7114 then it is natural to consider observation models for encounter frequencies, such as
7115 the Poisson model. Another type of encounter device is the "multi-catch" device
7116 (Efford et al., 2009a) which is a physical device that can capture and hold an ar-
7117bitrary number of individuals. A typical example is a mist-net for birds (Borchers
7118 and Efford, 2008). It is natural to regard observations from these kinds of studies as
7119 independent multinomial observations. A related type of device that produces *de-
7120 pendent* multinomial observations are the so-called *single-catch* traps (Efford, 2004;
7121 Efford et al., 2009a). The canonical example are small-mammal live traps which
7122 catch and hold a single individual. Competition among individuals for traps in-
7123 duces a complex dependence structure among individual encounters. To date, no
7124 formal inference framework has been devised for this method although it stands to

7125 reason that the independent multinomial model should be a good approximation in
 7126 some situations (Efford et al., 2009a). We analyze a number of examples of these
 7127 different observation models using **JAGS** and also the **R** package **secr** (Efford,
 7128 2011).

8.1 POISSON OBSERVATION MODEL

7129 The models we analyze in Chapt. 4 assumed binary observations – i.e., standard
 7130 encounter history data – so that individuals are captured at most one time in a trap
 7131 on any given sample occasion. This makes sense for many types of DNA sampling
 7132 (e.g., based on hair snares) because distinct visits to sampled locations or devices
 7133 cannot be differentiated. However, for some encounter devices, or methods, the
 7134 potential number of encounters is *not* fixed, and so it is possible to encounter an
 7135 individual some arbitrary number of times during any particular sampling episode.
 7136 That is, we might observe encounter frequencies $y_{ijk} > 1$ for individual i , trap j
 7137 and sampling interval k . As an example, if a camera device is functioning properly
 7138 it may be programmed to take photos every few seconds if triggered. For a second
 7139 example, suppose we are searching a quadrat or length of trail for scat, we may
 7140 find multiple samples from the same individual. Therefore, we seek observation
 7141 models that accommodate such encounter frequency data. In general, any discrete
 7142 probability mass function could be used for this purpose, including the standard
 7143 models for count data used throughout ecology, the Poisson and negative binomial.
 7144 Here we focus on using the Poisson model only although other count frequency
 7145 models are possible for SCR models (Efford et al., 2009b).

7146 Let y_{ijk} be the frequency of encounter for individual i , in trap j , during occasion
 7147 k , then assume:

$$y_{ijk} \sim \text{Poisson}(\lambda_{ij})$$

7148 where the expected encounter frequency λ_{ij} depends on both individual and trap.
 7149 As we did in the binary model of Chapt. 4, we now seek to model the expected
 7150 value of the observation (which was p_{ij} in Chapt 4) as a function of the individual
 7151 activity center \mathbf{s}_i . We propose

$$\lambda_{ij} = \lambda_0 k(\mathbf{x}_j, \mathbf{s}_i)$$

7152 Where $k(\mathbf{x}, \mathbf{s})$ is any positive valued function, such as the negative exponential or
 7153 the bivariate Gaussian kernel, and λ_0 is the baseline encounter rate – the expected
 7154 number of encounters if a trap is placed precisely at an individuals home range
 7155 center (note: in **secr** the notation for this is g_0). Then, $\lambda_0 k(\mathbf{x}_j, \mathbf{s}_i)$ is the expected
 7156 encounter rate in trap \mathbf{x}_j for an individual having activity center \mathbf{s}_i . Note that

$$\log(\lambda_{ij}) = \log(\lambda_0) + \log(k(\mathbf{x}_j, \mathbf{s}_i)).$$

7157 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
 7158 model), then:

$$\log(\lambda_{ij}) = \alpha_0 - \alpha_1 d(\mathbf{x}_j, \mathbf{s}_i)^2 \quad (8.1.1)$$

where $\alpha_1 = 1/(2\sigma^2)$, which is the same linear predictor as we have seen for the Bernoulli model in Chapt. 4. This Poisson SCR model is therefore a type of Poisson generalized linear mixed model (GLMM).

We can accommodate covariates at the level of individual-, trap- or sample occasion by including them on the baseline encounter rate parameter λ_0 . For example, if C_j is some covariate that depends on trap only, then we express the relationship between λ_0 and C_j as:

$$\log(\lambda_{0,ijk}) = \alpha_0 + \alpha_2 C_j$$

and therefore covariates on the logarithm of baseline encounter probability appear also as linear effects on 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. 6 and 7.

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.

8.1.1 Poisson model of space usage

It is natural to interpret the Poisson encounter model as a model of space usage resulting from movement of individuals about their home range (Sec. 4.4). 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:

$$m_{ij} \sim \text{Poisson}(\lambda_0 k(\mathbf{x}_j, \mathbf{s}_i))$$

7192 where λ_0 is related to the baseline movement rate of the animal (how often it
 7193 moves). This model of space usage gives rise to the standard resource selection
 7194 function (RSF) models (see Chapt. 12). But now suppose our samplers are not
 7195 perfect but, rather, record only a fraction of the resulting visits. A sensible model
 7196 is

$$y_{ij}|m_{ij} \sim \text{Binomial}(m_{ij}, p).$$

7197 The marginal distribution of y_{ij} is:

$$y_{ij} \sim \text{Poisson}(p_0 k(\mathbf{x}_j, \mathbf{s}_i)).$$

7198 where p_0 is a composite of the movement rate and conditional detection probability
 7199 p . Therefore, we see that encounters accumulate in proportion to the frequency of
 7200 outcomes of an individual using space (or “selecting resources”).

7201 We introduced an interpretation of SCR models in terms of movement and space
 7202 usage in Sec. 4.4, and it is one of the main underlying concepts of SCR models
 7203 that is not present in ordinary capture-recapture models. As we noted there, the
 7204 underlying model of space usage is only as complex as the encounter probability
 7205 model which has been, so far in this book, only symmetric and stationary (does not
 7206 vary in space). We generalize this model of space usage substantially in Chapt. 12.

7207 8.1.2 Poisson relationship to the Bernoulli model

7208 There is a sense in which the Poisson and Bernoulli models can be viewed as consistent
 7209 with one another. Note that under the Poisson model, the relationship between
 7210 the expected count and the probability of counting “at least 1”, is given by

$$\Pr(y > 0) = 1 - \exp(-\lambda) \quad (8.1.2)$$

7211 where $\mathbb{E}(y) = \lambda$. Therefore, if we equate the event “encountered” with the event
 7212 that the individual was captured at least 1 time under the Poisson model, i.e.,
 7213 $y > 0$, then it would be natural to set $p_{ij} = \Pr(y > 0)$ according to Eq. 8.1.2.
 7214 That is, we can use Eq. 8.1.2 as the model for encounter probability for binary
 7215 observations. This is the “hazard rate” model in distance sampling.

7216 In fact, as λ gets small, the Poisson model is a close approximation to the
 7217 Bernoulli model in the sense that outcomes concentrate on $\{0, 1\}$, i.e., $\Pr(y \in$
 7218 $\{0, 1\}) \rightarrow 1$ as $\lambda \rightarrow 0$. Indeed, under the Poisson model, $\Pr(y > 0) \rightarrow \lambda$ for small
 7219 values of λ . This phenomenon is shown in Fig. 8.1 where the left panel shows
 7220 a plot of $\lambda_{ij} = \lambda_0 k(\mathbf{x}_j, \mathbf{s}_i)$ vs. distance and superimposed on that is a plot of
 7221 $p_{ij} = 1 - \exp(-\lambda_{ij})$ vs. distance, for values $\lambda_0 = 0.1$ and $\sigma = 1$, and the right
 7222 panel shows a plot of $\Pr(y > 0)$ vs. $\mathbb{E}(y)$. We see that the two quantities are
 7223 practically indistinguishable. This is convenient in some cases because the Poisson
 7224 model might be more tractable to fit (or even vice versa). For an example, see
 7225 the models described in Chapt. 17, and we also consider another case in Sec. 8.3
 7226 below. To evaluate the closeness of the approximation, you can use the following
 7227 **R** commands which we used to produce Fig. 8.1:

```

7228 > x <- seq(0.001,5,,200)
7229 > lam0 <- .1
7230 > sigma <- 1
7231 > lam <- lam0*exp(-x*x/(2*sigma*sigma))
7232
7233 > par(mfrow=c(1,2))
7234 > p1 <- 1-exp(-lam)
7235 > plot(x, lam, ylab="E[y] or Pr(y>0)", xlab="distance", type="l", lwd=2)
7236 > lines(x,p1,lwd=2,col="red")
7237 > plot(lam, p1, xlab="E[y]", ylab="Pr(y>0)", type="l", lwd=2)
7238 > abline(0,1,col="red")

```

7239 To summarize, if y is Poisson then, as λ gets small,

$$\begin{aligned} \Pr(y > 0) &\approx \mathbb{E}(y) \\ 1 - \exp(-\lambda_0 k(\mathbf{x}, \mathbf{s})) &\approx \lambda_0 k(\mathbf{x}, \mathbf{s}) \end{aligned} \quad (8.1.3)$$

7240 What all of this suggests it that if we have very few observations > 1 in our SCR
 7241 data set, then we won't lose much information by using the Bernoulli model. On the
 7242 other hand, the Poisson model may have some advantages in terms of analytic or
 7243 numerical tractability in some cases. Further, this approximation explains the close
 7244 correspondence we have found between these two versions of the Gaussian encounter
 7245 probability model (Sec. 4.4). Namely, the Gaussian hazard model and the Gaussian
 7246 encounter probability model are close approximations because $1 - \exp(-\lambda) \approx \lambda$ if
 7247 λ is small.

7248 Even in such cases where the Poisson and Bernoulli models are not quite equiv-
 7249 alent, we might choose to truncate individual encounter frequencies to binary ob-
 7250 servations anyhow (transforming counts to 0/1 is called "quantizing"). We might
 7251 do this intentionally in some cases, such as when the distinct encounter events are
 7252 highly dependent as often happens in camera trap studies when the same individual
 7253 moves back-and-forth in front of a camera during a short period of time. But some-
 7254 times, truncation is a feature of the sampling. For example, in the case of bear hair
 7255 snares, the number of encounters might be well approximated by a Poisson distri-
 7256 bution but we cannot determine unique visits and so only get to observe the binary
 7257 event " $y > 0$ ". In this case, we might choose to model the encounter probability
 7258 for the binary encounter using Eq. 8.1.4. This is equivalent to the complementary
 7259 log-log link model, or the "Gaussian hazard" as we called it in Chapt. 4:

$$\text{cloglog}(p_{ij}) = \log(\lambda_0) + \log(k(\mathbf{x}, \mathbf{s}))$$

7260 where $\text{cloglog}(u) = \log(-\log(1 - u))$.

7261 8.1.3 A cautionary note on modeling encounter frequencies

7262 Other models for counts might be appropriate. For example, ecologists are espe-
 7263 cially fond of negative binomial models for count data (Ver Hoef and Boveng, 2007;

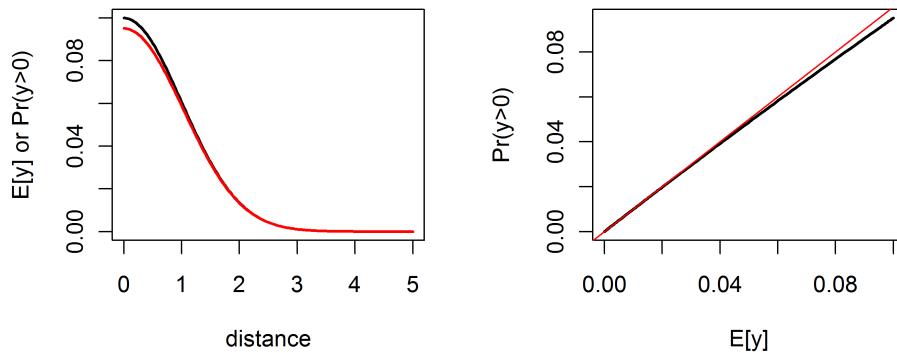


Figure 8.1. Poisson approximation to the binomial. As the Poisson mean approaches 0, then $\Pr(y > 0)$ under the Poisson model approaches λ and therefore $y \sim \text{Poisson}(\lambda)$ is well-approximated by a Bernoulli model with parameter λ .

7264 White and Bennetts, 1996; Kéry et al., 2005) but other models for excess-Poisson
 7265 variation are possible. For example, we might add a normally distributed random
 7266 effect to the linear predictor (Coull and Agresti, 1999).

7267 As a general rule we favor the Bernoulli observation model even if our sampling
 7268 scheme produces encounter frequencies. The main reason is that, with frequency
 7269 data, we are forced to confront a model choice problem (i.e., Poisson, negative
 7270 binomial, log-normal mixture) that is wholly unrelated to the fundamental space
 7271 usage process that underlies the genesis of many types of SCR data. Repeated
 7272 encounters over short time intervals are not likely to be the result of independent
 7273 encounter events. E.g., an individual moving back and forth in front of a camera
 7274 yields a cluster of observations that is not informative about the underlying spatial
 7275 structure of the population. Similarly in scat surveys dogs are used to locate scats
 7276 which are processed in the lab for individuality (Kohn et al., 1999; MacKay et al.,
 7277 2008; Thompson et al., 2012). The process of local scat deposition is not strictly the
 7278 outcome of movement or space usage but rather the outcome of complex behavioral
 7279 considerations as well as dependence in detection of scat by dogs. For example, dogs
 7280 find (or smell) one scat and then are more likely to find one or more nearby ones, if
 7281 present, or they get into a den or latrine area and find many scats. The additional
 7282 assumption required to model variation in observed frequencies (i.e., conditional on
 7283 location) provides relatively no information about space usage and density, and we
 7284 feel that the model selection issue should therefore be avoided.

To elaborate on this, we suppose that an individual with activity center \mathbf{s} visits a particular pixel \mathbf{x} with some probability $p(\mathbf{x}, \mathbf{s})$, and then, once there, deposits a number of scat, or visits a camera some number of times with frequency $y(\mathbf{x}, \mathbf{s}) \geq 0$. We describe the outcome of this movement/usage process with a two-level hierarchical model of the form: $[y|w][w|p(\mathbf{x}, \mathbf{s})]$ where $w(\mathbf{x}, \mathbf{s})$ is a binary variable that indicates whether the individual with activity center \mathbf{s} used pixel \mathbf{x} during some interval, and let $w(\mathbf{x}, \mathbf{s}) \sim \text{Bernoulli}(p(\mathbf{x}, \mathbf{s}))$. If we suppose encounter frequency y is independent of \mathbf{x} and \mathbf{s} conditional on the use variable w , then we see that the model for y (amount of use) does not depend on \mathbf{s} .

8.1.4 Analysis of the Poisson SCR model in BUGS

We consider the simplest possible model here in which we have no covariates that vary over sample occasions $k = 1, 2, \dots, K$ so that we work with the aggregated individual- and trap-specific encounters:

$$y_{ij} = \left(\sum_{k=1}^K y_{ijk} \right) = \text{Poisson}(K\lambda_0 k(\mathbf{x}_j, \mathbf{s}_i))$$

and we consider the bivariate normal form of $k(\mathbf{x}, \mathbf{s})$:

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

so that

$$\log(\lambda_{ij}) = \alpha_0 - \alpha_1 d(\mathbf{x}_j, \mathbf{s}_i)^2$$

where $\alpha_0 = \log(\lambda_0)$ and $\alpha_1 = 1/(2\sigma^2)$.

As usual, we approach Bayesian analysis of these models using data augmentation (Sec. 3.2). Under data augmentation, we introduce a collection of all-zero encounter histories to bring the total size of the data set up to M , and a corresponding set of data augmentation variables $z_i \sim \text{Bern}(\psi)$. Then the observation model is specified conditional on z according to:

$$y_{ij} \sim \text{Poisson}(z_i K \lambda_{ij})$$

which evaluates to a point mass at $y = 0$ if $z = 0$. In other words, the observation model under data augmentation is a zero-inflated Poisson model which is easily analyzed by Bayesian methods, e.g., in one of the **BUGS** dialects or, alternatively, using likelihood methods, which we neglect here although the same principles as in Chapt. 5 apply.

8.1.5 Simulating data and fitting the model

Simulating a sample SCR data set under the Poisson model requires only a couple minor modifications to the procedure we used in Chapt. 4 (see the function

7314 **simSCRO**). In particular, we modify the block of code which defines the model to
 7315 be that of $\mathbb{E}(y)$ and not $\Pr(y = 1)$, and we change the random variable generator
 7316 from `rbinom` to `rpois`:

```

7317 ##  

7318 ## S =activity centers and traplocs defined as in simSCRO()  

7319 ##  

7320 ## Compute distance between activity centers and traps:  

7321 > D <- e2dist(S,traplocs)  

7322  

7323 ## Define parameter values:  

7324 > alpha0 <- -2.5  

7325 > sigma <- 0.5  

7326 > alpha1 <- 1/(2*sigma*sigma)  

7327  

7328 ## Encounter probability model:  

7329 > muy <- exp(alpha0)*exp(-alpha1*D*D)  

7330  

7331 ## Now generate the encounters of every individual in every trap  

7332 > Y <-matrix(NA,nrow=N,ncol=ntraps)  

7333 > for(i in 1:nrow(Y)){  

7334   Y[i,] <- rpois(ntraps,K*muy[i,])  

7335 }

```

7336 We modified our simulation code from Chapt. 4 to simulate Poisson encounter
 7337 frequencies for each trap and then we analyze an ideal data set using **BUGS**. This
 7338 Poisson simulator function `simPoissonSCR` is available in the `scrbook` package (it
 7339 can produce 3-d encounter history data too, although we don't do that here). Here
 7340 is an example of simulating a data set and harvesting the required data objects,
 7341 and doing the data augmentation:

```

7342 ## Simulate data and extract data elements  

7343 ##  

7344 > data <- simPoissonSCR(discard0=TRUE,rnd=2013)  

7345 > y <- data$Y  

7346 > nind <- nrow(y)  

7347 > X <- data$traplocs  

7348 > K <- data$K  

7349 > J <- nrow(X)  

7350 > xlim <- data$xlim  

7351 > ylim <- data$ylim  

7352  

7353 ## Data augmentation  

7354 > M <- 200  

7355 > y <- rbind(y,matrix(0,nrow=M-nind,ncol=ncol(y)))  

7356 > z <- c(rep(1,nind),rep(0,M-nind))

```

7357 The process for fitting the model in **WinBUGS** or **JAGS** is identical to what
 7358 we've done previously in Chapt. 4. In particular, we set up some starting values,

7359 package the data and inits, identify the parameters to be monitored, and then send
 7360 everything off to our MCMC engine. Here it all is for fitting the Poisson observation
 7361 model (these commands are shown in the help file for `simPoissonSCR`):

```

7362 ## Starting values for activity centers
7363 ##
7364 > sst <- X[sample(1:J,M,replace=TRUE),]
7365 > for(i in 1:nind){
7366   if(sum(y[i,])==0) next
7367   sst[i,1] <- mean( X[y[i,]>0,1] )
7368   sst[i,2] <- mean( X[y[i,]>0,2] )
7369 }
7370 ## Dithered a little bit from trap locations
7371 > sst <- sst + runif(nrow(sst)*2,0,1)/8
7372 > data <- list (y=y,X=X,K=K,M=M,J=J,xlim=xlim,ylim=ylim)
7373 > inits <- function(){
7374   list (alpha0=rnorm(1,-2,.4),alpha1=runif(1,1,2),s=sst,z=z,psi=.5)
7375 }
7376 > parameters <- c("alpha0","alpha1","N","D")

```

7377 Next, we write the **BUGS** model to an external file:

```

7378 > cat("
7379 model{
7380   alpha0 ~ dnorm(0,.1)
7381   alpha1 ~ dnorm(0,.1)
7382   psi ~ dunif(0,1)
7383
7384   for(i in 1:M){
7385     z[i] ~ dbern(psi)
7386     s[i,1] ~ dunif(xlim[1],xlim[2])
7387     s[i,2] ~ dunif(ylim[1],ylim[2])
7388     for(j in 1:J){
7389       d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
7390       y[i,j] ~ dpois(lam[i,j])
7391       lam[i,j] <- z[i]*K*exp(alpha0)*exp(- alpha1*d[i,j]*d[i,j])
7392     }
7393   }
7394   N <- sum(z[])
7395   D <- N/64
7396 }
7397 ",file = "SCR-Poisson.txt")

```

7398 To fit the model we execute `bugs` in the usual way:

```

7399 > library(R2WinBUGS)
7400 > out1 <- bugs (data, inits, parameters, "SCR-Poisson.txt", n.thin=1,
7401   n.chains=3,n.burnin=1000,n.iter=2000,working.dir=getwd(),
7402   debug=TRUE)

```

Table 8.1. Results of fitting the SCR model with Poisson encounter frequencies to the wolverine camera trapping data. Posterior summaries were obtained using **WinBUGS** with 3 chains, each with 6000 iterations, discarding the first 1000 as burn-in, to yield a total of 15000 posterior samples.

Parameter	Mean	SD	2.5%	50%	97.5%	Rhat	n.eff
ψ	0.30	0.07	0.19	0.30	0.45	1	650
σ	0.64	0.06	0.54	0.64	0.76	1	730
λ_0	0.06	0.01	0.04	0.06	0.08	1	5000
$\log(p_0)$	-2.89	0.17	-3.22	-2.89	-2.57	1	5000
N	60.12	11.91	40.00	59.00	87.00	1	630
D	5.80	1.15	3.86	5.69	8.39	1	630

7403 Or, using **JAGS** via `rjags` we would do something like this:

```
7404 > library(rjags)
7405 > jm <- jags.model("SCR-Poisson.txt", data=data, inits=inits,
7406   n.chains=3, n.adapt=1000)
7407 > out2 <- coda.samples(jm, parameters, n.iter=1000, thin=1)
```

7408 Summarizing the output from the **WinBUGS** run produces the following:

```
7409 > print(out1,digits=2)
7410 Inference for Bugs model at "SCR-Poisson.txt", fit using WinBUGS,
7411 3 chains, each with 2000 iterations (first 1000 discarded)
7412 n.sims = 3000 iterations saved
7413      mean    sd   2.5%   25%   50%   75% 97.5% Rhat n.eff
7414 alpha0   -2.57  0.19  -2.95 -2.69 -2.57 -2.44 -2.19 1.00 2600
7415 alpha1    2.34  0.36   1.69  2.08  2.32  2.57  3.12 1.00 3000
7416 N       114.13 15.25  87.97 103.00 113.00 124.00 147.00 1.01 370
7417 D        1.78  0.24   1.37  1.61  1.77  1.94  2.30 1.01 370
7418 deviance 329.95 21.92 290.00 314.20 329.50 344.40 375.80 1.00 1700
7419 ...
7420 [..some output deleted..]
7421 ...
```

7422 8.1.6 Analysis of the wolverine study data

7423 We reanalyzed the data from the wolverine camera trapping study that were first
7424 introduced in Sec. 4.9. We modified the **R** script from the function `wolvSCRO` to
7425 fit the Poisson model (see the help file for `wolvSCROpois`). Executing this function
7426 produces the results shown in Table 8.1. The results are almost indistinguishable
7427 from the Bernoulli model fitted previously, where we had a posterior mean for N
7428 of 59.84 and σ was 0.64. You can edit the script `wolvSCROpois` to obtain more
7429 posterior samples, or modify the model in some way.

8.1.7 Count detector models in the secr package

7431 The **R** package **secr** will fit Poisson or negative binomial encounter frequency
 7432 models. The formatting of data and structure of the analysis proceeds in a similar
 7433 fashion to the Bernoulli model described in Sec. 5.5, except that we specify the
 7434 **detector**=‘‘count’’ option when the traps object is created. The set-up proceeds
 7435 as follows:

```
7436 > library(secr)
7437 > library(scrbook)
7438 > data(wolverine)
7439
7440 > traps <- as.matrix(wolverine$wtraps)
7441 > dimnames(traps) <- list(NULL,c("trapID","x","y",paste("day",1:165,sep="")))
7442 > traps1 <- as.data.frame(traps[,1:3])
7443 > trapfile1 <- read.traps(data=traps1,detector="count")
```

7444 You can proceed with analysis of these data and compare/contrast with the
 7445 Bayesian analysis given above, or the results of the Bernoulli model fitted in Chapt.
 7446 5.

8.2 INDEPENDENT MULTINOMIAL OBSERVATIONS

7447 Several types of encounter devices yield multinomial observations in which an in-
 7448 dividual can be caught in a single trap during a particular encounter occasion, but
 7449 traps might catch any number of individuals. Mist netting is the canonical example
 7450 of such a “multi-catch” device (Efford et al., 2009a). Also some kinds of bird or
 7451 mammal cage-traps hold multiple animals, as do pit-fall traps which are commonly
 7452 used for many species of herptiles. Another type of sample method that might be
 7453 viewed (in some cases) as a multi-catch device are area-searches of, for example,
 7454 reptiles where we think of a small polygon as the “trap” – we could get multiple
 7455 individuals (turtles, lizards) in the same plot but not, in the same sample occasion,
 7456 at different plots. The key features of this independent multinomial or multi-catch
 7457 model are: (1) capture of an individual in a trap is *not* independent of its capture
 7458 in other traps, because initial capture precludes capture in any other trap and (2)
 7459 individuals behave independently of one another, so whether a trap captures some
 7460 individual doesn’t have an affect on whether it captures another. A type of model
 7461 in which the 2nd assumption is violated are the “single catch” trap systems which
 7462 we address in Sec. 8.3 below.

7463 In this case we assume the observation y_{ik} for individual i during sample occa-
 7464 sion k is a multinomial observation which consists of a sequence of 0’s and a single
 7465 1 indicating the trap of capture, or “not captured”. For the “not captured” event
 7466 we define an additional outcome, by convention element $J + 1$ of the vector. As
 7467 an example, if we capture an individual in trap 2 during some occasion of a study
 7468 involving $J = 6$ traps. Then, the multinomial observation has length $J + 1 = 7$,

and the observation is $\mathbf{y}_i = (0, 1, 0, 0, 0, 0, 0)$. An individual not captured at all would have the observation vector $(0, 0, 0, 0, 0, 0, 1)$. If we sample for 5 occasions in all and the individual is also caught in trap 4 during occasion 3, but otherwise uncaptured, then the 5 encounter observations for that individual are as follows:

	occassion	trap						"not captured"
		1	2	3	4	5	6	7

1		0	1	0	0	0	0	0
2		0	0	0	0	0	0	1
3		0	0	0	1	0	0	0
4		0	0	0	0	0	0	1
5		0	0	0	0	0	0	1

Statistically we regard the *rows* of this data matrix as *independent* multinomial trials.

Analogous to our previous Bernoulli and Poisson models, we seek to construct the multinomial cell probabilities for each individual, as a function of *where* that individual lives, through its center of activity \mathbf{s} . Thus we suppose that

$$\mathbf{y}_{ik} | \mathbf{s}_i \sim \text{Multinomial}(1, \boldsymbol{\pi}(\mathbf{s}_i)) \quad (8.2.1)$$

where $\boldsymbol{\pi}(\mathbf{s}_i)$ is a vector of length $J + 1$, where $\pi_{i,J+1}$, the last cell, corresponds to the probability of the event “not captured”. Now we have to construct these cell probabilities in some meaningful way that depends on each individual’s \mathbf{s} . We use the standard multinomial logit with distance as a covariate:

$$\pi_{ij} = \frac{\exp(\alpha_0 - \alpha_1 d_{ij})}{1 + \sum_j \exp(\alpha_0 - \alpha_1 d_{ij})}$$

for $j = 1, 2, \dots, J$ and, for $J + 1$, i.e., “not captured”,

$$\pi_{i,(J+1)} = \frac{\exp(0)}{1 + \sum_j \exp(\alpha_0 - \alpha_1 d_{ij})}$$

or, more commonly, we use d_{ij}^2 to correspond to our Gaussian kernel model for encounter probability. Whatever function of distance we use in the construction of multinomial probabilities will have a direct correspondence to the standard encounter probability models we used in the Bernoulli or Poisson models as well (see Sec. 4.4).

It is convenient to express these multinomial models short-hand as follows, e.g., for the Gaussian encounter probability model:

$$\text{mlogit}(\pi_{ij}) = \alpha_0 - \alpha_1 d_{ij}^2$$

7498 In this way we can refer to models with covariates in a more concise way. For
 7499 example, a model with a trap-specific covariate, say C_j , is:

$$\text{mlogit}(\pi_{ij}) = \alpha_0 - \alpha_1 d_{ij}^2 + \alpha_2 C_j$$

7500 or we could include occasion-specific covariates too, such as behavioral response.

7501 A statistically equivalent distribution to the multinomial is the *categorical*
 7502 distribution. If \mathbf{y} is a multinomial trial with probabilities $\boldsymbol{\pi}$ than the *position* of the
 7503 non-zero element of \mathbf{y} is a categorical random variable with probabilities $\boldsymbol{\pi}$. We
 7504 express this for SCR models as

$$\mathbf{y}|\mathbf{s} \sim \text{Categorical}(\boldsymbol{\pi}(\mathbf{s}))$$

7505 In the SCR context, the categorical version of the multinomial trial corresponds to
 7506 the *trap of capture*. Using our example above with 6 traps then we could as well say
 7507 y_{ik} is a categorical random variable with possible outcomes (1, 2, 3, 4, 5, 6, 7) where
 7508 outcome $y = 7$ corresponds to “not captured.” Obviously, how this is organized or
 7509 labeled is completely irrelevant, although it is convenient to use the integers 1 to
 7510 ($J+1$) where $J+1$ is the event not captured. Therefore, for our illustration in the
 7511 previous table, $y_{i1} = 2$, $y_{i2} = 7$, $y_{i3} = 4$ and so on.

7512 For simulating and fitting data in the **BUGS** engines we will typically use the
 7513 categorical representation of the model because it is somewhat more convenient.
 7514 We have found that fitting multinomial models in **WinBUGS** is less efficient than
 7515 **JAGS** (Royle and Converse, in review), which we use in the subsequent examples
 7516 involving multinomial observation models.

7517 8.2.1 Multinomial resource selection models

7518 The multinomial probabilities in Eq. 8.2.2 look similar to the multinomial resource
 7519 selection function (RSF) model for telemetry data (Manly et al., 2002; Lele and
 7520 Keim, 2006). This suggests how we might model landscape or habitat covariates
 7521 using such methods – i.e., by including them as explicit covariates in a larger
 7522 multinomial model for “use” – which, if we take the product of use with encounter,
 7523 produces a model for the observable encounter data. This leads naturally to the
 7524 development of models that integrate RSF data from telemetry studies with SCR
 7525 data (Royle et al., 2012b), which is the topic of Chapt. 12.

7526 8.2.2 Simulating data and analysis using JAGS

7527 We’re going to show the nugget of a simulation function which is used in the function
 7528 **simMnSCR** found in the **R** package **scrbook**. The first lines of the following **R** code
 7529 make use of some things that you need to define, but we omit them here (e.g., **xlim**,
 7530 **ylim** are the boundaries of the state-space, **N** is the population size, etc.):

```

7531 ## 
7532 ## Simulate random activity centers:
7533 ##      (first define N, xlim, ylim, etc..)
7534 ##
7535 > S <- cbind(runif(N,xlim[1],xlim[2]),runif(N,ylim[1],ylim[2]))
7536 
7537 ## Distance from each individual to each trap
7538 > D <- e2dist(S,traplocs)
7539 
7540 ## Set parameter values
7541 > sigma <- 0.5
7542 > alpha0 <- -1
7543 > alpha1 <- -1/(2*sigma*sigma)
7544 
7545 ## make an empty data matrix and fill it up with data
7546 > Ycat <- matrix(NA,nrow=N,ncol=K)
7547 > for(i in 1:N){
7548   for(k in 1:K){
7549     lp <- alpha0 + alpha1*D[i,]*D[i,]
7550     cp <- exp(c(lp,0))
7551     cp <- cp/sum(cp)
7552     Ycat[i,k] <- sample(1:(ntraps+1),1,prob=cp)
7553   }
7554 }
```

7555 We save the data in the matrix `Ycat` to clarify that it is the categorical observation representing “trap of capture”. The matrix `Ycat` here has the maximal dimension N and so, to do an analysis that mimics a real situation, we would have to discard the uncaptured individuals. The function `simMnSCR` in the package `scrbook` will also simulate data that includes a behavioral response which will be the typical situation in small-mammal trapping problems (see Converse and Royle, 2012, for details).

7556 Here we use our function `simMnSCR` to simulate a data set with $K = 7$ occasions.
7557 We'll run the model using JAGS which we have found is much more effective for this
7558 class of models. We get the data set-up for analysis by augmenting the size of
7559 the data set to $M = 200$. In addition we choose starting values for `s` and the
7560 data augmentation variables `z`. For starting values of `s` we cheat a little bit here
7561 and use the true values for the observed individuals and then augment the $M \times 2$
7562 matrix `S` with $M - n$ randomly selected activity centers. Our function `spiderplot`
7563 returns the mean observed location of individuals for use as starting values for the
7564 `nind` encountered individuals. The parameters input to `simMnSCR` are the intercept
7565 α_0 , $\sigma = \sqrt{1/(2\alpha_1)}$ for the Gaussian encounter probability model, and α_2 is the
7566 behavioral response parameter. The data simulation and set-up proceeds as follows:

```

7573 > set.seed(2013)
7574 > parms <- list(N=100,alpha0= -.40, sigma=0.5, alpha2= 0)
7575 > data <- simMnSCR(parms, K=7, ssbuff=2)
```

```

7576 > nind <- nrow(data$Ycat)
7577
7578 > M <- 200
7579 > Ycat <- rbind(data$Ycat,matrix(nrow(data$X)+1,nrow=(M-nind),ncol=data$K))
7580 > Sst <- rbind(data$S,cbind(runif(M-nind,data$xlim[1],data$xlim[2]),
7581                               runif(M-nind,data$ylim[1],data$ylim[2])))
7582 > zst <- c(rep(1,160),rep(0,40))

```

7583 The model specification is not much more complicated than the binomial or
7584 Poisson models given previously. The main consideration is that we define the cell
7585 probabilities for each trap $j = 1, 2, \dots, J$ and then define the last cell probability,
7586 $J + 1$, for “not captured”, to be the complement of the sum of the others. The code
7587 is shown in Panel 8.1. In the last lines of code here we specify N and density, D ,
7588 as derived parameters.

7589 To fit the model, we need to package everything up (inits, parameters, data)
7590 and send it off to **JAGS** to build an MCMC simulator for us (these commands are
7591 executed in the help file for `simMnSCR`). In addition to the usual data objects, we
7592 also pass the limits of the assumed rectangular state-space (`ylim`, `xlim`, both 1×2
7593 vectors) and the scale of the standardized units, called `trap.space` here because
7594 we typically will define the trap coordinates to be an integer grid. If the trap
7595 spacing is 10 m and we want units of density computed in terms of individuals
7596 per meter-squared, then we input `trap.space=10`. The analysis is carried out as
7597 follows:

```

7598 > inits <- function(){ list (z=zst,sigma=runif(1,.5,1) ,S=Sst) }
7599
7600 # Parameters to monitor
7601 > parameters <- c("psi","alpha0","alpha1","sigma","N","D")
7602
7603 # Bundle the data. Note this reuses "data"
7604 > data <- list (X=data$X,K=data$K, trap.space=1,Ycat=Ycat,M=M,
7605                   ntraps=nrow(data$X),ylim=data$ylim,xlim=data$xlim)
7606
7607 > library(R2jags)
7608 > out <- jags (data, inits, parameters, "model.txt", n.thin=1,
7609                   n.chains=3, n.burnin=1000, n.iter=2000)

```

7610 The posterior summaries are provided in the following **R** output (recall that
7611 $N = 100$, $\alpha_0 = -0.40$, and $\sigma = 0.5$):

```

7612 > out
7613 Inference for Bugs model at "model.txt", fit using jags,
7614   3 chains, each with 2000 iterations (first 1000 discarded)
7615   n.sims = 3000 iterations saved
7616     mu.vect sd.vect    2.5%     25%     50%     75%   97.5% Rhat n.eff
7617 D          1.873   0.189   1.531   1.750   1.859   2.000   2.250 1.006  1300
7618 N         119.867  12.107  98.000 112.000 119.000 128.000 144.000 1.006  1300

```

```

model{
psi ~ dunif(0,1)
alpha0 ~ dnorm(0,10)
sigma ~ dunif(0,10)
alpha1 <- 1/(2*sigma*sigma)

for(i in 1:M){
z[i] ~ dbern(psi)
S[i,1] ~ dunif(xlim[1],xlim[2])
S[i,2] ~ dunif(ylim[1],ylim[2])
for(j in 1:ntraps){
#distance from capture to the center of the home range
d[i,j] <- pow(pow(S[i,1]-X[j,1],2) + pow(S[i,2]-X[j,2],2),1)
}
for(k in 1:K){
for(j in 1:ntraps){
lp[i,k,j] <- exp(alpha0 - alpha1*d[i,j])*z[i]
cp[i,k,j] <- lp[i,k,j]/(1+sum(lp[i,k,]))
}
cp[i,k,ntraps+1] <- 1-sum(cp[i,k,1:ntraps]) # last cell = not captured
Ycat[i,k] ~ dcat(cp[i,k,])
}
}

N <- sum(z[1:M])
A <- ((xlim[2]-xlim[1])*trap.space)*((ylim[2]-ylim[1])*trap.space)
D <- N/A
}

```

Panel 8.1: **BUGS** model specification for the independent multinomial observation model. For data simulation and model fitting see the help file `?simMnSCR` in the **R** package `scrbook`.

```

7619 alpha0    -0.435   0.151  -0.738  -0.535  -0.439  -0.331  -0.146  1.004   580
7620 alpha1     2.195   0.286   1.658   2.004   2.180   2.372   2.785  1.003   2400
7621 psi        0.599   0.069   0.465   0.552   0.599   0.645   0.739  1.006   1400
7622 sigma      0.480   0.032   0.424   0.459   0.479   0.500   0.549  1.003   2400
7623 deviance  892.164 21.988  850.922  877.417  891.561  906.246  937.728 1.003   950
7624
7625 [... output deleted ...]

```

7626 8.2.3 Multinomial relationship to the Poisson

7627 The multinomial is related to the Poisson encounter rate model by a conditioning
 7628 argument. Let y_{ij} be the number of encounters for individual i in trap j . If
 7629 $y_{ij} \sim \text{Poisson}(\lambda_{ij})$, then, conditional on the *total* number of captures (i.e., across
 7630 all traps), $y_i = \sum_j y_{ij}$, the trap encounter frequencies are multinomial with prob-
 7631 abilities

$$\pi_{ij} = \frac{\lambda_{ij}}{\sum_j \lambda_{ij}}$$

7632 for $j = 1, 2, \dots, J$. Or equivalently the *trap of capture* is categorical with probabili-
 7633 ties π_{ij} as given above. Under the Gaussian kernel model, these probabilities are:

$$\pi_{ij} = \frac{\exp(-\alpha_1 d(\mathbf{x}_i, \mathbf{s}_i)^2)}{\sum_j \exp(-\alpha_1 d(\mathbf{x}_i, \mathbf{s}_i)^2)} \quad (8.2.2)$$

7635 where, we note, the intercept α_0 has canceled from both the numerator and denom-
 7636 inator. This makes sense because, here, these probabilities describe the trap-specific
 7637 capture probabilities *conditional on capture*. Therefore, the model is not completely
 7638 specified, absent a model for the “overall” probability of encounter or the expected
 7639 frequency of captures, say ϕ_i . Depending on how we specify a model for this quan-
 7640 tity ϕ_i , we can reconcile it directly with the Poisson model. Let y_i be the total
 7641 number of encounters for individual i and suppose y_i has a Poisson distribution
 7642 with mean ϕ_i . Then, marginalizing Eq. 8.2.1 over the Poisson distribution for y_i
 7643 produces the original set of *iid* Poisson frequencies with probabilities:

$$\lambda_{ij} = \phi_i \pi_{ij}$$

7644 for $j = 1, 2, \dots, J$. In particular, if we suppose that $\phi_i = \sum_j \exp(\alpha_0 - \alpha_1 d(\mathbf{x}, \mathbf{s})^2)$
 7645 then the marginal distribution of y_{ij} is Poisson with mean $\exp(\alpha_0 - \alpha_1 d(\mathbf{x}, \mathbf{s})^2)$,
 7646 equivalent to Eq. 8.1.1.

7647 In summary, the Poisson and multinomial models are equivalent in how they
 7648 model the distribution of captures among traps. It stands to reason that, if the
 7649 encounter rate of individuals is low, we could use the Poisson and multinomial
 7650 models interchangeably. In fact, based on our discussion in Sec. 8.1.2 above we
 7651 could use any of the binomial/Poisson/multinomial models with little ill-effect when
 7652 encounter rate is low.

8.2.4 Avian mist-netting example

We analyze data from a mist-netting study of ovenbirds, conducted at the Patuxent Wildlife Research Center, Laurel MD, by D.K. Dawson and M.G. Efford. The data from this study are available in the **secr** package, and have been analyzed previously by Efford et al. (2004), see also Borchers and Efford (2008). Forty-four mist nets spaced 30 m apart on the perimeter of a 600-m x 100-m rectangle were operated on 9 or 10 non-consecutive days in late May and June for 5 years from 2005-2009. The ovenbird data can be loaded as follows:

```
> library(secr)
> data(ovenbird)
```

The data set consists of adult ovenbirds caught during sampling in each of 5 years, 2005-2009. (one ovenbird was killed in 2009, indicated by a negative net number in the encounter data file). As with most mist-netting studies, nets are checked multiple times during a day (e.g., every hour during a morning session). However, for this data set, the within-day recaptures are not included so each bird has at most a single capture per day. Therefore the multinomial model (detector type ‘multi’ in **secr**) is appropriate. Although several individuals were captured in more than one year, this information is not used in the models presently offered in **secr**, but we do make use of it in the development of open models in Chapt. 15.

Multiple sample sessions

Up to this point we have only dealt with a basic closed population sampling situation consisting of repeated sample occasions on a single population of individuals using a single array of traps. In practice, many studies produce repeated samples over longer periods of time over which demographic closure isn’t valid, or at different locations where the populations are completely distinct. We adopt the **secr** terminology of *session* for such replication by groups of time or space, and the models are *multi-session* models, although we think of such models as being relevant to any stratified population (see Chapt. 13). We introduced **secr**’s multi-session models in Sec. 5.5.4. In the case of the ovenbird data, sampling was carried out in multiple years, with a number of sample occasions within each year (9 or 10), a type of data structure commonly referred to as “the robust design” (Pollock, 1982). In this context, it stands to reason that there is recruitment and mortality happening across years. In Chapt. 15 we model these processes explicitly but, here, we provide an analysis of the data that does not require explicit models for recruitment and survival, regarding the yearly populations as independent strata, and fitting a multi-session model.

When the sessions represent explicit time periods, the multi-session model of **secr** can be thought of as a type of open population model. In particular, a special case of open models arises when we assume N_t (time-specific population sizes) are independent from one time period or session to the next – this can be thought of as a “random temporary emigration” model of the Kendall et al. (1997)

variety, and this is the multi-session model implemented in **secr**. In particular, by assuming that N_t is Poisson with mean Λ_t , one can model variation in abundance among sessions based on the Poisson-integrated likelihood in which parameters of Λ_t appear directly in the likelihood as we noted in Sec. 5.5.4. We provide an analysis (below) of the ovenbird data here using the multi-session models in **secr**. We formalize the multi-session model approach from a Bayesian perspective using data augmentation in Chapt. 13 (Converse and Royle, 2012; Royle and Converse, in review).

A 3rd way to develop models for stratified or grouped populations, not based on multi-session models, but that is convenient in **BUGS**, is to regard the data from each session as an independent data set with its own N_t parameter, and do T distinct data augmentations. Because each N_t is regarded as a free parameter, independent of the other parameters, we'll call this the nonparametric multi-session model to distinguish it from the multi-session model which assumes the N_t are related to one another by having been generated from a common Poisson distribution. We can analyze this model in the normal context of data augmentation by augmenting each year separately in the same **BUGS** model specification. This approach avoids making explicit model assumptions about the N_t parameters. This is distinct from the model implemented in **secr** in that **secr** is removing the N_t parameters by integrating the conditional-on- N_t likelihood over the Poisson prior for N_t 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. 13 from a Bayesian standpoint; (3) later, in Chapt. 15, 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

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.

Table 8.2. 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 5000 iterations, discarding the first 1000, for a total of 12000 posterior samples.

parameter	Mean	SD	2.5%	50%	97.5%	Rhat	n.eff
D[1]	1.000	0.202	0.682	0.971	1.446	1.009	250
D[2]	1.041	0.192	0.723	1.033	1.467	1.010	230
D[3]	1.212	0.208	0.868	1.198	1.653	1.004	580
D[4]	0.895	0.173	0.599	0.868	1.281	1.012	190
D[5]	0.760	0.155	0.496	0.744	1.095	1.002	1700
α_0	-3.488	0.159	-3.801	-3.490	-3.171	1.005	450
α_1	0.000	0.000	0.000	0.000	0.000	1.018	120
σ	76.106	6.061	65.620	75.709	89.205	1.018	120
$\psi[1]$	0.484	0.108	0.299	0.475	0.715	1.008	300
$\psi[2]$	0.505	0.103	0.325	0.497	0.725	1.008	270
$\psi[3]$	0.585	0.110	0.388	0.578	0.813	1.004	710
$\psi[4]$	0.434	0.095	0.264	0.428	0.639	1.010	210
$\psi[5]$	0.370	0.087	0.215	0.364	0.556	1.002	2000
N[1]	48.396	9.762	33.000	47.000	70.000	1.009	250
N[2]	50.403	9.290	35.000	50.000	71.000	1.010	230
N[3]	58.687	10.049	42.000	58.000	80.000	1.004	580
N[4]	43.341	8.366	29.000	42.000	62.000	1.012	190
N[5]	36.792	7.502	24.000	36.000	53.000	1.002	1700

single model for all adults, constant 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 8.2). 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 8.2. 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:

```

model{
  alpha0 ~ dnorm(0,.1)
  sigma ~ dunif(0,200)
  alpha1 <- 1/(2*sigma*sigma)

  A <- ((xlim[2]-xlim[1]))*((ylim[2]-ylim[1]))
  for(t in 1:5){
    N[t] <- inprod(z[1:bigM],yrdummy[,t])
    D[t] <- (N[t]/A)*10000 # Put in units of per ha
    psi[t] ~ dunif(0,1)
  }

  for(i in 1:bigM){ # bigM = total size of jointly augmented data set
    z[i] ~ dbern(psi[year[i]])
    S[i,1] ~ dunif(xlim[1],xlim[2])
    S[i,2] ~ dunif(ylim[1],ylim[2])

    for(j in 1:ntraps){ # X = trap locations, S = activity centers
      d2[i,j] <- pow(pow(S[i,1]-X[j,1],2) + pow(S[i,2]-X[j,2],2),1)
    }
    for(k in 1:K){
      Ycat[i,k] ~ dcat(cp[i,k,])
      for(j in 1:ntraps){
        lp[i,k,j] <- exp(alpha0 - alpha1*d2[i,j])*z[i]*(1-dead[i,k])
        cp[i,k,j] <- lp[i,k,j]/(1+sum(lp[i,k,1:ntraps]))
      }
      cp[i,k,ntraps+1] <- 1-sum(cp[i,k,1:ntraps]) # Last cell = not captured
    }
  }
}

```

Panel 8.2: **BUGS** model specification for the non-parametric multi-session model in which each N_t is independent of the other. The implied prior (by data augmentation) is that $N_t \sim \text{Uniform}(0, 100)$. To fit this model to the ovenbird data, see `?SCRovenbird` in the **R** package `scrbook`.

```

7750 ovenbird.model.1 fitted secr model -- null
7751 ovenbird.model.1b fitted secr model -- g0 net shyness
7752 ovenbird.model.1T fitted secr model -- g0 time trend within years
7753 ovenbird.model.h2 fitted secr model -- g0 finite mixture
7754 ovenbird.model.D fitted secr model -- trend in density across years

```

7755 The model fit objects provided in **secr** are based on the use of the habitat
 7756 mask. To make the analyses consistent with our previous analysis in **JAGS**, we
 7757 refit all of the models here without the habitat mask. The re-analysis proceeds as
 7758 follows, changing the “trend in density across years” model to allow for year-specific
 7759 density:

```

7760 ## Fit constant-density model
7761 > ovenbird.model.1 <- secr.fit(ovenCH)
7762 ## Fit net avoidance model
7763 > ovenbird.model.1b <- secr.fit(ovenCH, model = list(g0 ~ b))
7764 ## Fit model with time trend in detection
7765 > ovenbird.model.1T <- secr.fit(ovenCH, model = list(g0 ~ T))
7766 ## Fit model with 2-class mixture for g0
7767 > ovenbird.model.h2 <- secr.fit(ovenCH, model = list(g0 ~ h2))
7768 ## Fit a model with session (year)-specific Density
7769 > ovenbird.model.DT <- secr.fit(ovenCH, model = list(D ~ session))

```

7770 All of these can be fitted easily in **JAGS** but the model we fitted previously
 7771 is roughly equivalent to the last model, **ovenbird.model.DT**, because we allowed
 7772 for year-specific population sizes (and hence density). So, we’ll compare our results
 7773 from **JAGS** to that model. The **secr** output is extensive and so we do not repro-
 7774 duce it completely here. By default, it summarizes the trap information for each
 7775 year, encounter information, and then output for each year. Here is an abbreviated
 7776 version for **ovenbird.model.DT**:

```

7777 > print(ovenbird.model.DT,digits=2)
7778
7779 secr.fit( capthist = ovenCH, model = list(D ~ session), buffer = 300 )
7780 secr 2.3.1, 14:46:52 23 Jan 2013
7781
7782 '$'2005'
7783 Object class      traps
7784 Detector type    multi
7785 Detector number   44
7786 Average spacing   30.27273 m
7787 x-range           -50 49 m
7788 y-range           -285 285 m
7789
7790 [... deleted ...]
7791
7792          2005 2006 2007 2008 2009
7793 Occasions       9   10   10   10   10

```

```

7794 Detections   35   42   52   30   33
7795 Animals      20   22   26   19   16
7796 Detectors    44   44   44   44   44
7797
7798 Model          : D~session g0~1 sigma~1
7799 Fixed (real)   : none
7800 Detection fn   : halfnormal
7801 Distribution    : poisson
7802 N parameters   : 7
7803 Log likelihood  : -1119.845
7804 AIC            : 2253.689
7805 AICc           : 2254.868
7806
7807 [... deleted ...]

7808 To do model selection we use the handy helper-function AIC as follows (output
7809 edited to fit on the page):
7810
7811 AIC (ovenbird.model.1, ovenbird.model.1b, ovenbird.model.1T,
7812     ovenbird.model.h2, ovenbird.model.DT)
7813
7814     model detectfn npar logLik      AIC      AICc      dAICc
7815 ovenbird.model.1T [edited output] 4 -1111.850 2231.700 2232.109 0.000
7816 ovenbird.model.1b      ....      4 -1117.615 2243.229 2243.637 11.528
7817 ovenbird.model.h2      ....      3 -1121.164 2248.327 2248.570 16.461
7818 ovenbird.model.1      ....      5 -1119.762 2249.524 2250.143 18.034
7819 ovenbird.model.DT      ....      7 -1119.845 2253.689 2254.868 22.759

```

7819 We see that our DT model is way down at the bottom of the list. Instead,
7820 the model with a time-trend (within-season) in detection probability is preferred,
7821 followed by a behavioral response. We encourage you to adapt the **JAGS** model
7822 specification for such models which is easily done (see Chapt. 6 for many examples).
7823 We provide the summary results for the model having $D \sim \text{session}$ as follows:

```

7824 > print(ovenbird.model.DT,digits=2)
7825
7826 secr.fit( capthist = ovenCH, model = list(D ~ session), buffer = 300 )
7827 secr 2.3.1, 14:46:52 23 Jan 2013
7828
7829 [...deleted....]
7830
7831 Fitted (real) parameters evaluated at base levels of covariates
7832
7833 session = 2005
7834     link estimate SE.estimate      lcl      ucl
7835 D      log      0.920       0.228  0.571  1.484
7836 g0     logit     0.028       0.004  0.021  0.037
7837 sigma   log     78.566       6.379 67.025 92.095

```

```

7838
7839 session = 2006
7840      link estimate SE.estimate    lcl    ucl
7841 D      log     0.963       0.238  0.598  1.553
7842 g0     logit    0.028       0.004  0.021  0.037
7843 sigma   log    78.566      6.379 67.025 92.095
7844
7845 session = 2007
7846      link estimate SE.estimate    lcl    ucl
7847 D      log     1.139       0.282  0.706  1.836
7848 g0     logit    0.028       0.004  0.021  0.037
7849 sigma   log    78.566      6.379 67.025 92.095
7850
7851 session = 2008
7852      link estimate SE.estimate    lcl    ucl
7853 D      log     0.832       0.206  0.516  1.341
7854 g0     logit    0.028       0.004  0.021  0.037
7855 sigma   log    78.566      6.379 67.025 92.095
7856
7857 session = 2009
7858      link estimate SE.estimate    lcl    ucl
7859 D      log     0.701       0.173  0.435  1.130
7860 g0     logit    0.028       0.004  0.021  0.037
7861 sigma   log    78.566      6.379 67.025 92.095

```

7862 The point estimates (MLEs) of density are uniformly lower than the Bayesian
7863 estimates (posterior means) shown in Table 8.2. We expect some difference in this
7864 direction due to small-sample skew of the posterior. In addition, there may be
7865 slight differences due to the fact that **secr** multi-session model assumes that the
7866 N_t have a Poisson prior, but the implementation in **JAGS** using data augmentation
7867 is based on a binomial prior. The estimated σ is very similar between the **JAGS**
7868 analysis and **secr**.

8.3 SINGLE-CATCH TRAPS

7869 The classical animal trapping experiment is based on a physical trap which captures
7870 a single animal and holds that individual until subsequent molestation by a biol-
7871 ogist. This type of observation model – the “single-catch” trap – was the original
7872 situation considered in the context of spatial capture-recapture by Efford (2004).
7873 Nowadays, capture-recapture data are more often obtained by other methods (DNA
7874 from hair snares, or scat sampling, camera traps etc...) but nevertheless the single-
7875 catch traps are still widely used in small mammal studies (Converse et al., 2006;
7876 Converse and Royle, 2012) and other situations.

7877 The single-catch model is basically a multinomial model but one in which the
7878 number of available traps is reduced as each individual is captured. As such, the

7879 constraints on the joint likelihood for the sample of n encounter histories are very
 7880 complicated. As a result, at the time of this writing, there has not been a formal
 7881 development of either likelihood or Bayesian analysis of this model and applications
 7882 of SCR models to single-catch systems have used the independent multinomial
 7883 model as an approximation (see below).

7884 Nevertheless, we can make some progress to describing the basic observation
 7885 model formally. In particular, if we imagine that all of the individuals captured
 7886 queued up at the beginning of the capture session to draw a number indicating
 7887 their order of capture, then there is a nice conditional structure resulting from a
 7888 “removal process” operating on the traps. The first individual captured has the
 7889 multinomial observation model:

$$\mathbf{y}_1 \sim \text{Multinomial}(\boldsymbol{\pi}_1)$$

7890 whereas the 2nd individual captured also has a multinomial encounter probability
 7891 model but with the trap which captured the first individual removed. We might
 7892 express this as:

$$\mathbf{y}_2 \sim \text{Multinomial}(\boldsymbol{\pi}_2)$$

7893 where

$$\pi_{2j} = \frac{(1 - y_{1j}) * \exp(\alpha_0 - \alpha_1 d_{ij}^2)}{\sum_j (1 - y_{1j}) * \exp(\alpha_0 - \alpha_1 d_{ij}^2)}$$

7894 and so on for $i = 3, 4, \dots, n$. In a certain way, this model is a type of local behav-
 7895 ior model but where the response is to other individuals being captured.
 7896 Evidently, the **order of capture** is relevant to the construction of these multi-
 7897 nomial cell probabilities. More generally, the *time* of capture of an individual in
 7898 any trapping interval will affect the encounter probability of subsequently captured
 7899 individuals, but we think that order of capture might lead to a practical approxi-
 7900 mation to the single-catch process (this is how we simulate the data in our function
 7901 `simScSCR`). In the simulation of single catch data, we randomly ordered the popula-
 7902 tion of individuals for each sample occasion, and then cycled through them, turning
 7903 off each trap if an individual was captured in it.

7904 8.3.1 Inference for single-catch systems

7905 For the single-catch model, we argued that the observations have a multinomial
 7906 type of observation model, but the multinomial observations have a unique condi-
 7907 tional dependence structure among them owing to the “removal” of traps as they
 7908 fill-up with individuals. Thus, competition for single-catch traps renders the inde-
 7909 pendence assumptions for the independent multinomial model invalid. However, as
 7910 Efford et al. (2009a) noted, we expect “bias to be small when trap saturation (the
 7911 proportion of traps occupied) is low. Trap saturation will be higher when popula-
 7912 tion density is high...” relative to trap density, or when net encounter probability

7913 is high. Efford et al. (2009a) did a limited simulation study and found essentially
 7914 no effective bias and concluded that estimators of density from the misspecified
 7915 independent multinomial model are robust to the mild dependence induced when
 7916 trap saturation is low. Naturally then, we expect that the Poisson model could also
 7917 be an effective approximation under the same set of circumstances.

7918 In the **R** package **scrbook** we provide a function for simulating data from a
 7919 single-catch system (function **simScSCR**) and fitting the misspecified model (**example(simScSCR)**)
 7920 in **JAGS** so that you can evaluate the effectiveness of this misspecified model for
 7921 situations that interest you.

7922 8.3.2 Analysis of Efford's possum trapping data

7923 We provide an analysis here of data from a study of brushtail possums in New
 7924 Zealand. The data are available with the **R** package **secr** (Efford et al., 2009a); see
 7925 the help file **?possum** after loading the **secr** package. Originally the data were an-
 7926 alyzed by Efford et al. (2005), and a detailed description of the data set is available
 7927 in the help file, from which we summarize:

7928 *Brushtail possums (*Trichosurus vulpecula*) are an unwanted invasive species in New
 7929 Zealand. Although most abundant in forests, where they occasionally exceed densities
 7930 of 15/ha, possums live wherever there are palatable food plants and shelter.*

7931 To load the possum data, execute the following commands:

```
7932 > library(secr)
7933 > data(possum)
```

7934 The study area encompasses approximately 300 ha, and 180 live traps were orga-
 7935 nized in 5 distinct grids, shown in Fig. 8.2. Each square arrangement of traps
 7936 consisted of 36 traps with a spacing of 20 m. Thus the squares are 180 m on a side.
 7937 Individuals were captured, tagged, and released over 5 days during April, 2002. A
 7938 noteworthy aspect of this study is that it involves replicated grids selected in some
 7939 fashion from within a prescribed region. From an analysis standpoint, we could
 7940 adopt the use of the multi-session models which we used previously to analyze the
 7941 ovenbird data. This would be useful if we had covariates at the trapping grid level
 7942 that we wanted to model. Alternatively, we could pool the data from all of the grids
 7943 and analyze them jointly as if they were based on a single trapping grid (with 180
 7944 traps) which is clearly a reasonable view in this case. In doing this sort of pooling,
 7945 there is an implicit assumption that N_t (t indexing trapping grid in this case) is
 7946 Poisson distributed, with constant mean (Royle, 2004a; Royle et al., 2012c) which
 7947 we also address in Chapt. 13.

7948 The data file **possumCH** contains 112 encounter histories, and we analyze those
 7949 here although the last 8 of those are recaptures treated as new individuals². The en-
 7950 counter process is not strictly a single-catch multinomial process because, as noted

²M. Efford, personal communication

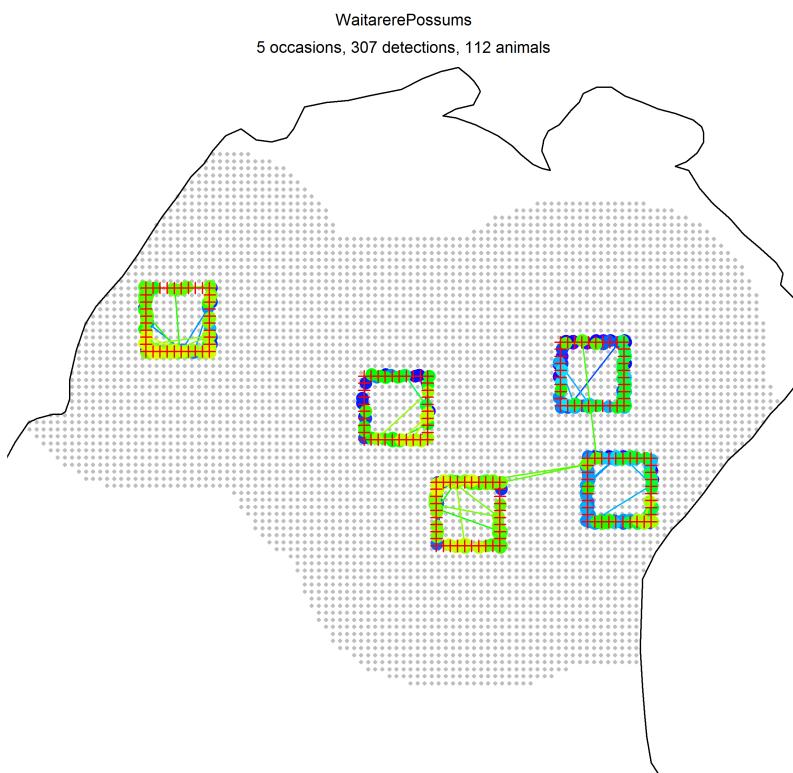


Figure 8.2. Trapping grids used in possum study from Efford et al. (2005), data are contained in the R package `secr` (Efford, 2011), refer to the help file `?possum` for additional details of this study.

in the `possum` help file “One female possum was twice captured at two sites on one day, having entered a second trap after being released; one record in each pair was selected arbitrarily and discarded.” which is a similar situation to what might happen in bird mist net studies, as a bird might fly into a net upon release from another. By discarding the two extra-capture events, we can satisfactorily view these data as single-catch data, for which `secr` uses the independent multinomial likelihood (M. Efford, pers. comm.). If multiple, same-session captures were common, then it might be worth developing a model for n_{ik} = the number of captures of individual i during sample occasion k , in order to make use of all captures.

For our Bayesian analysis here, we used a rectangular state-space which doesn’t account for any geographic boundaries of the survey region, but we note that a habitat mask is included in `secr` and it could be used in a Bayesian analysis. Whether or not we use the mask is probably immaterial as long as we understand the predictions of N or D over the water don’t mean anything biological and we probably wouldn’t report such predictions. The **JAGS** model specification is based on that of the ovenbird analysis given previously, and so we don’t reproduce the model here. The **R/JAGS** script is called `SCRpossum`, which is in the `scrbook` package. The results are summarized in Table 8.3.

Table 8.3. Results of fitting the independent multinomial observation model to the possum trapping data. Strictly speaking, the trapping device is a “single-catch” trap, and the model represents an intentional misspecification. Density is reported in individuals per ha (Dha). Posterior summaries were obtained using **JAGS** with 3 chains, each with 2000 iterations, discarding the first 1000 as burn-in, to yield a total of 3000 posterior samples.

Parameter	Mean	SD	2.5%	50%	97.5%	Rhat	n.eff
N	235.407	17.435	204.000	235.000	270.000	1.009	340
Dha	1.549	0.115	1.343	1.547	1.777	1.009	340
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
ψ	0.783	0.062	0.666	0.782	0.903	1.008	340
σ	52.020	2.675	47.067	51.933	57.585	1.001	2800

The estimated density (posterior mean) is about 1.53 possums/ha. To obtain the `secr` results for the equivalent null model, we execute the following command

```
7971 > secr.fit( capthist = possumCH, trace = F )
```

which produces (edited) summary output:

```
7973 [... some output deleted ...]
7974
7975 Fitted (real) parameters evaluated at base levels of covariates
7976      link   estimate SE.estimate      lcl      ucl
7977 D       log    1.6988930  0.17352645  1.3913904  2.0743547
```

```

7978 g0    logit  0.1968542  0.02256272  0.1563319  0.2448321
7979 sigma   log 51.4689114  2.59981905  46.6204139  56.8216500
7980
7981 [... some output deleted ...]

```

7982 As we've discussed previously, there are many reasons for why there might be
 7983 differences between Bayesian and likelihood estimates. But even among likelihood
 7984 estimates – any time you run a model there is some numerical integration going
 7985 on which requires some specific choices of how to do the integration (see Chapt.
 7986 5). For now we just observe that the estimated density is certainly in the ballpark
 7987 (compared to those in Table. 8.3), and so too is the estimated σ .

8.4 ACOUSTIC SAMPLING

7988 The last decade has seen an explosion of technology that benefits the study of animal
 7989 populations. This includes DNA sampling methods that allow for identification from hair or scat,
 7990 camera trapping and identification software that allow efficient sampling of many mammals, and the resulting statistical technology that helps us
 7991 to make sense of such data (Borchers and Efford, 2008; Royle and Young, 2008;
 7992 Efford et al., 2009b; Gopalaswamy et al., 2012b; Sollmann et al., 2012; Chandler
 7993 and Royle, In press). One other extremely promising technology area is that of
 7994 acoustic sampling using microphones or recording devices. That is, instead of having
 7995 cameras record encounters, or humans pick up scat, we can establish an array of
 7996 (usually) electronic recording devices which, instead of establishing a visual identity
 7997 of individuals, record a vocal expression of each individual. In this context, Efford
 7998 et al. (2009b) referred to audio recorders as “signal strength proximity detectors”
 7999 to distinguish them from other types of proximity detections, including camera
 8000 traps, which are *visual* proximity detector. Using audio records, the spatial pattern
 8001 of the *signal strength* at the different audio recorders or microphones can be used
 8002 for inference about density (Dawson and Efford, 2009; Efford et al., 2009b) in the
 8003 same way as the spatial pattern of detections is used in the types of SCR models
 8004 we have discussed so far. The basic technical formulation of these models comes
 8005 from Efford et al. (2009b), and it was applied to field study of birds by Dawson
 8006 and Efford (2009). In that study, recording devices were organized in groups of 4
 8007 (in a square pattern), with an array of 5×15 such clusters of 4, separated by 100
 8008 m (300 total recorder locations). This data set, called `signalCH`, is provided with
 8009 the `secr` package along with some sample analyses and help files. See Efford and
 8010 Dawson (2010), a version of the document `secr-sound.pdf` (that also comes with
 8011 the `secr` package) which you can access directly from the main help file (`?secr`).

8012 Our development here mostly follows Efford et al. (2009b), but we change some
 8013 notation to be consistent with our previous material. Let $S(\mathbf{x}, \mathbf{u})$ be the strength
 8014 of a signal emanating from signal location \mathbf{u} , as recorded by a device at location
 8015 \mathbf{x} . Just as ordinary SCR models represent a model of *encounter frequency* as a

8017 function of distance, in acoustic models, the acoustic SCR model is a model of
 8018 sound attenuation as a function of distance. In particular, the acoustic models
 8019 assumes that S (or a suitable transformation) declines with distance d from the
 8020 origin of the sound, to the recording device. In the context of spatial sampling
 8021 of animals, the origin is the actual location of some individual animal, and the
 8022 recording device is something we nailed to a tree, or mounted on a post. For
 8023 example, a model of sound attenuation used by Dawson and Efford (2009) is the
 8024 following:

$$S(\mathbf{x}, \mathbf{u}) = \alpha_0 + \alpha_1 d(\mathbf{x}, \mathbf{u}) + \epsilon \quad (8.4.1)$$

8025 where $\epsilon \sim \text{Normal}(0, \sigma_s^2)$. In many standard situations, S will be measured in
 8026 decibels, which can be any value on the real line. In the conduct of acoustic sampling
 8027 and the development of custom models for your own situation, it would probably
 8028 be helpful to know something about sound dynamics and signal processing. In this
 8029 model, the parameters α_0 , α_1 and σ_s^2 are to be estimated. We abbreviate the set
 8030 of parameters by θ for short.

8031 The basic structure of an acoustic SCR study is not really much different from
 8032 ordinary SCR studies. Just as ordinary SCR models require that individuals be
 8033 encountered at > 1 trap, these acoustic models require that individuals be heard
 8034 at > 1 recorder. Therefore, the acoustic signals (calls or vocalizations) must be
 8035 reconcilable and, in fact, reconciled successfully by the investigator. In practice,
 8036 this would require associating signals that occur at the same instant with the same
 8037 individual (or making a decision one way or the other). Further, if individuals are
 8038 actively moving during the sample period (that recorders are functioning) then in-
 8039 dividuals might be double-counted, thereby biasing estimates of density. In general,
 8040 the models produce an estimate of density of *sources*, and how that is interpreted
 8041 depends on whether individuals are stationary or mobile, and other things. In par-
 8042 ticular, if multiple survey occasions are used (e.g., on different days), then modeling
 8043 movement of individuals would be essential in order to interpret estimates of den-
 8044 sity meaningfully. Models that allow some movement should be possible (see Sec.
 8045 8.4.3 below, and Chaps. 14 and 15).

8046 8.4.1 The signal strength model

8047 We assert that an individual is detected if S exceeds a threshold, c . The reason
 8048 for introducing this threshold c is that sound recorders will always record some
 8049 background sound, and so effective use of the acoustic SCR models requires spec-
 8050 ification of the threshold of measured signal below which the record is censored
 8051 (non-detection occurs) because the recorded sound is assumed to be background
 8052 noise. So we assert that an individual is detected if $S > c$ which occurs with prob-
 8053 ability $\Pr(S > c)$, the encounter probability. To expand on and formalize this, let
 8054 S_{ij} be the observed value of S for animal i at detector j . The encounter probability
 8055 is $\Pr(S_{ij} > c)$ which is $\Pr(S_{ij} > c) = 1 - \Pr(S_{ij} < c)$, so that, if we standardize the

8056 variate we have

$$1 - \Pr\left(\frac{(S_{ij} - \mathbb{E}(S))}{\sigma_s} < \frac{(c - \mathbb{E}(S))}{\sigma_s}\right)$$

8057 This probability calculation requires evaluation of the CDF of a standard normal
 8058 variate say, $\eta = (S_{ij} - \mathbb{E}(S))/\sigma_s$, being less than $\gamma(\boldsymbol{\theta}) = (c - \mathbb{E}(S))/\sigma_s$, which is
 8059 a function of all the parameters α_0 , α_1 , σ_s^2 and also the individual location \mathbf{u} and
 8060 trap location \mathbf{x} . We'll identify it by $\gamma(\boldsymbol{\theta}, \mathbf{x}, \mathbf{u})$ when we need to be explicit about
 8061 those things. We can compute $\Pr(S_{ij} > c) = 1 - \Pr(\eta < \gamma(\boldsymbol{\theta}, \mathbf{x}, \mathbf{u}))$ easily using any
 8062 software package including **R** which has a standard function, `pnorm`, for computing
 8063 the normal cdf. To be more precise, we'll use the $\Phi()$ to represent the normal
 8064 cdf. Therefore, an individual is encountered whenever $S_{ij} > c$ which happens with
 8065 probability $\Pr(S_{ij} > c) = 1 - \Phi(\gamma(\boldsymbol{\theta}, \mathbf{x}, \mathbf{u}))$.

8066 Naturally this quantity should depend on *where* an individual is located at the
 8067 time of recording – what we call it's instantaneous location, say \mathbf{u} , to distinguish it
 8068 from it's home-range center \mathbf{s} (but we outline a model below that contains both \mathbf{u}
 8069 and \mathbf{s}), and also the trap \mathbf{x} , so we index the quantity γ by those two quantities, in
 8070 addition to the parameters α_0 , α_1 and σ_s . The probability of detection is therefore

$$p_{ij} = p(\alpha_0, \alpha_1, \sigma | \mathbf{x}_j, \mathbf{u}_i) = 1 - \Phi(\gamma(\cdot))$$

8071 where \mathbf{u}_i is the instantaneous location of individual i and \mathbf{x}_j is the location of trap
 8072 j . We'll suppose here that the random variables \mathbf{u}_i have state-space \mathcal{U} ³.

8073 How do we interpret this probability? Well, two things have to happen for an
 8074 individual to be encountered by a trap: (1) it has to vocalize; (2) the microphone
 8075 has to record a signal $> c$. These two things together are a product of biological and
 8076 environmental factors which could include time of day, wind direction and speed,
 8077 or maybe rain, humidity and other things. The bottom line is a lot of factors
 8078 are balled up in whether or not the microphone records a sound greater than the
 8079 threshold.

8080 The observations from an acoustic survey are the signal strength measurements,
 8081 and the likelihood of the observed signal strength from individual i at detection
 8082 device j can be specified by noting that the likelihood is the normal pdf for the
 8083 observed signal *if* the signal strength is $> c$ and, otherwise, the contribution to the
 8084 likelihood is $\Phi(\gamma(\cdot))$ (see Eq. 8 of Efford et al. (2009b)).

$$\Pr(S_{ij} | \mathbf{u}_i) = \Phi(\gamma(\cdot))^{1-I(S_{ij}>c)} \text{Normal}(S_{ij}; \alpha_0, \alpha_1, \sigma_s, \mathbf{x}_j, \mathbf{u}_i)^{I(S_{ij}>c)}$$

8085 We can use this as the basis for constructing the binomial-form of the likelihood
 8086 as we did in Chapt. 5, which involves the number of individuals not encountered,
 8087 n_0 . The probability that an individual is *not* captured is equal to the probability

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

8088 that its signal strength doesn't exceed c at any microphone. The probability of not
 8089 being captured at a microphone \mathbf{x}_j is:

$$1 - p_{\mathbf{u},j} = \Phi(\gamma(\cdot))$$

8090 and therefore the probability of not being captured at any microphone is:

$$\Pr(\text{all } S_{\mathbf{u},j} < c | \mathbf{u}) = \prod_{j=1}^J (1 - p_{\mathbf{u},j}) = \prod_{j=1}^J \Phi(\gamma(\cdot, \mathbf{x}_j, \mathbf{u}))$$

8091 and therefore the marginal probability of not being captured is

$$\pi_0 = [\text{all } S_{\mathbf{u},j} < c | \boldsymbol{\alpha}] = \int_{\mathcal{U}} \left\{ \prod_{j=1}^J \Phi(\gamma(\boldsymbol{\theta}, \mathbf{x}_j, \mathbf{u})) \right\} d\mathbf{u}$$

8092 which can be used to construct the binomial form of the likelihood as we did in
 8093 Chapt. 5 (see Eq. 5.2.1).

8094 8.4.2 Implementation in **secr**

8095 Fitting acoustic encounter models in **secr** is no more difficult than other SCR
 8096 models. There is a handy manual (**secr-sound.pdf**) with examples (Efford and
 8097 Dawson, 2010) which comes with the **secr** package. The basic process is that
 8098 **make.capthist** will make a **capthist** object from a 3-dimensional encounter array
 8099 – which is a binary array indicating whether each individual was detected or not
 8100 at each recorder/microphone. In the case of signal strength data, **secr** handles the
 8101 case where # occasions = 1, i.e., the recorders obtained data for a single sample
 8102 occasion, but this is not a general requirement of the model for signal strength
 8103 data (see next section). The “signal” attribute of the **capthist** object contains the
 8104 signal strength in decibels. The best way to include the signal attribute is to use
 8105 **make.capthist** in the usual way, providing it with the encounter data and trap
 8106 data and, in addition, the variable “cutval” (which is c in our notation above) and
 8107 then provide the signal strength data as an extra column of the **capthist** object.
 8108 See **?make.capthist** for details.

8109 8.4.3 Implementation in **BUGS**

8110 We don't know of any Bayesian applications of acoustic SCR models, although
 8111 we imagine that implementation of such models in the **BUGS** engines should be
 8112 achievable. It seems easy enough to write down a general hierarchical model that
 8113 would accommodate sampling on repeated occasions. Let \mathbf{s}_i be the home range
 8114 center, and let \mathbf{u}_{ik} the instantaneous location of individual i during sample occasion
 8115 k (see Chapt. 14 for similar models). The model for \mathbf{u}_{ik} can be specified conditional

on \mathbf{s}_i . For example, we could assume that \mathbf{u}_{ik} are bivariate normal draws with mean \mathbf{s}_i and some variance σ_u^2 . Then, conditional on \mathbf{u}_{ik} an individual produces a signal according to the signal attenuation model (Eq. 8.4.1), or perhaps some other model. Then we generate the binary encounter data by truncating the observed signal at c . This general model then is an example of an SCR model in which parameters of a movement model are identifiable (see Sec. 2.6) because there is direct information about movement outcomes from the sampling method, unlike other types of encounter methods (e.g., camera traps) for which animal locations are restricted to a set of fixed, pre-determined points where traps are located. Other types of SCR methods allow for movement information too, including some of the search-encounter models (Chapt. 14).

Instead of developing a Bayesian version of this model here, we leave it to the reader to explore simulating data and devising a Bayesian implementation of the acoustic model in one of the **BUGS** engines. Note that for a single occasion, you can simulate the data using the two stage model (having both \mathbf{s} and \mathbf{u}) or you can simulate \mathbf{u} uniformly without dealing with \mathbf{s} in the model. The kernel of the **BUGS** model specification should resemble the following snippet:

```
8133 model {
8134   # Ignoring loops and data augmentation
8135   u[i,1] ~ dunif(xlim[1], xlim[2])
8136   u[i,2] ~ dunif(ylim[1], ylim[2])
8137   mu[i,j] <- alpha0 + alpha1*d[i,j]
8138   #####
8139   ### JAGS has this T() truncation feature
8140   S[i,j] ~ dnorm(mu[i,j], 1/sigma^2)T(c,Inf)
8141   #####
8142   gamma[i,j] <- (c - mu[i,j])/sigma
8143   p[i,j] <- 1 - pnorm(gamma[i,j], 0, 1) # JAGS has pnorm() function
8144   y[i,j] ~ dbern(p[i,j])
8145 }
```

8146 8.4.4 Other types of acoustic data

Efford and Dawson (2010) noted that various other types of acoustic data might arise for which SCR-like models would be useful⁴. For example, we could measure the *time of arrival* of a vocal queue of some sort at multiple recorders to estimate the number and origin of N queues. Another example is that where we measure *direction* to a queue from multiple devices and do, effectively, a type of statistical triangulation to the multiple but unknown number of sources. This has direct relevance to types of double or multiple-observer sampling that people do in field studies of birds. Normally 2 observers stand in close proximity and record birds,

⁴Some of the following is also related to material presented by D.L. Borchers at the ISEC 2012 conference in Norway.

reconciling their detections after data collection. An SCR-based formulation of the double-observer method has two observers (or more) standing some distance apart, e.g., 50 or 100 meters, and marking individual birds on a map (or at least a direction) and a time of detection. The SCR/double-observer method could be applied to such data.

8.5 SUMMARY AND OUTLOOK

In this chapter we extended SCR models to accommodate alternative models for the observation process, including Poisson and multinomial models. Along with the binomial model described in Chapt. 4, this sequence of models will accommodate a substantial majority of contemporary spatial capture-recapture problems, including the 4 main types of encounter data: binary encounters, multinomial trials from “multi-catch” and “single-catch” (Efford, 2004, 2011; Royle and Gardner, 2011) trap systems, and Poisson encounter frequency data from devices that can record multiple encounters of the same individual at a device. We summarize the standard observation models and the corresponding `secr` terminology in Table 8.4. What we refer to as search-encounter (or area-search) models (see Chapt. 14) are distinct from most of the other classes in that the observation location can also be random (in contrast to traps, where the location is fixed by design). This auxiliary data is informative about an intermediate process related to movement (Royle and Young, 2008).

Table 8.4. Different observation models, where we discuss them in this book, and what the corresponding `secr` terminology is

observation model	Where in this book?	<code>secr</code> name
Bernoulli	Chapt. 4	<code>proximity</code>
Poisson	Sec. 8.1	<code>count</code>
Multinomial (ind)	Sec. 8.2	<code>multi-catch</code>
Multinomial (dep)	Sec. 8.3	<code>single-catch</code>
Acoustic	Sec. 8.4	<code>signal</code>
Search-encounter	Chapt. 14	<code>polygon</code> (in part)

There is a need for other types of encounter models that arise in practice. We identify a few of them here, although we neglect a detailed development of them at the present time or, in some cases, put that off until later chapters: (1) Removal systems – Sometimes traps kill individuals and SCR models can handle that. This can be viewed as a kind of open model, with mortality only, and we handle such models (in part) in Chapt. 15; (2) There are models for which only specific summary statistics are observable (Chandler and Royle, In press; Sollmann et al., 2012) which we cover in Chaps. 17 - 18; (3) We can have multiple observation methods working together as in Gopalaswamy et al. (2012b).

There remains much research to be done to formalize models for certain obser-

vation systems. For example, while we think one will usually be able to analyze single-catch systems using the multi-catch model, or even the Bernoulli model if encounter probability is sufficiently low, a formalization of the single-catch model would be a useful development and, we believe, it should be achievable using one or another of the **BUGS** engines. In addition, classical “trapping webs” (Anderson et al., 1983; Wilson and Anderson, 1985a; Jett and Nichols, 1987; Parmenter and MacMahon, 1989; Link and Barker, 1994) have been around for quite some time and it seems like they are amenable to formulation as a type of SCR model although we have not pursued that development simply because trapping webs are rarely used in practice.

8194
8195

8196

9

SAMPLING DESIGN

9.1 GENERAL CONSIDERATIONS

- 8197 9.1.1 Model-based not design-based
- 8198 9.1.2 Sampling space or sampling individuals?
- 8199 9.1.3 Scope of inference vs. state-space

9.2 STUDY DESIGN FOR (SPATIAL) CAPTURE-RECAPTURE

9.3 TRAP SPACING AND ARRAY SIZE RELATIVE TO ANIMAL MOVEMENT

- 8200 9.3.1 Example: Black bears from Pictured Rocks National Lakeshore:

8201 **9.3.2 Final musings: SCR models, trap spacing and array size**

9.4 SPACING OF TRAPS WITH TELEMETERED INDIVIDUALS

9.5 SAMPLING OVER LARGE SCALES

9.6 MODEL-BASED SPATIAL DESIGN

8202 **9.6.1 Formalization of the Design Problem for SCR Studies**

8203 **9.6.2 An Optimal Design Criterion for SCR**

(9.6.1)

(9.6.2)

(9.6.3)

8204 **9.6.3 Optimization of the criterion**

8205 **9.6.4 Illustration**

9.7 COVARIATE MODELS

9.8 SUMMARY AND OUTLOOK

8206

Part III

8207

8208

Advanced SCR Models

8209
8210

8211
8212

10

MODELING SPATIAL VARIATION IN DENSITY

10.1 HOMOGENEOUS POINT PROCESS REVISITED

10.2 INHOMOGENEOUS POINT PROCESSES

(10.2.1)

(10.2.2)

(10.2.3)

(10.2.4)

(10.2.5)

(10.2.6)

10.3 OBSERVED POINT PROCESSES

10.4 FITTING INHOMOGENEOUS POINT PROCESS SCR MODELS

8213 **10.4.1 Continuous space**

8214 **10.4.2 Discrete space**

10.5 ECOLOGICAL DISTANCE AND DENSITY COVARIATES

10.6 THE JAGUAR DATA

10.7 SUMMARY AND OUTLOOK

8215
8216

11

MODELING LANDSCAPE CONNECTIVITY

8217

11.1 SHORTCOMINGS OF EUCLIDEAN DISTANCE MODELS

11.2 LEAST-COST PATH DISTANCE

(11.2.1)

(11.2.2)

(11.2.3)

8218 11.2.1 Example of Computing Cost-weighted distance

11.3 SIMULATING SCR DATA USING ECOLOGICAL DISTANCE

11.4 LIKELIHOOD ANALYSIS OF ECOLOGICAL DISTANCE MODELS

(11.4.1)

8219 **11.4.1 Example of SCR with Least-Cost Path**

11.5 BAYESIAN ANALYSIS

11.6 SIMULATION EVALUATION OF THE MLE

8220 **11.6.1 Simulation Results**

11.7 DISTANCE IN AN IRREGULAR PATCH

8221 **11.7.1 Basic Geographic Analysis in R**

11.8 SUMMARY AND OUTLOOK

8222
8223

12

8224
8225
8226

INTEGRATING RESOURCE SELECTION WITH SPATIAL CAPTURE-RECAPTURE MODELS

8227
8228
8229

12.1 A SIMPLE MODEL OF SPACE USAGE

12.1.1 Poisson use model

12.1.2 Thinning

12.1.3 Capture-recapture Data

12.2 THE JOINT RSF/SCR LIKELIHOOD

12.3 APPLICATION: NEW YORK BLACK BEAR STUDY

12.4 SIMULATION STUDY

12.5 SUMMARY AND OUTLOOK

8230
8231

8232
8233

13

STRATIFIED POPULATIONS: MULTI-SESSION AND MULTI-SITE DATA

13.1 DATA STRUCTURE

13.2 MULTINOMIAL ABUNDANCE MODELS

(13.2.1)

(13.2.2)

(13.2.3)

(13.2.4)

(13.2.5)

8234 **13.2.1 Observation Models**

8235 **13.2.2 Simulating group structured capture-recapture data**

8236 **13.2.3 Fitting in BUGS**

8237 **13.2.4 Approach B modeling ψ**

13.3 SPATIAL CAPTURE-RECAPTURE

(13.3.1)

13.4 APPLICATION

8238 **13.4.1 Results**

13.5 TOPICS IN MULTI-SESSION MODELS

- 8239 **13.5.1 Temporal models**
- 8240 **13.5.2 Dependence – is it a problem?**
- 13.6 MULTI-SESSION MODELS IN SECR**
- 8241 **13.6.1 Ovenbird data in WinBUGS?**
- 8242 **13.6.2 Converse data in secr?**
- 13.7 SUMMARY AND OUTLOOK**

8243
8244

8245
8246

14

MODELS FOR SEARCH-ENCOUNTER DATA

14.1 SEARCH-ENCOUNTER SAMPLING DESIGNS

14.2 A MODEL FOR SEARCH-ENCOUNTER DATA

(14.2.1)

(14.2.2)

(14.2.3)

8247 14.2.1 Ecological process model

8248 14.2.2 Other stuff

14.3 EXAMPLES

8249 14.3.1 Hard plot boundaries

8250 14.3.2 Analysis of other protocols

14.4 DESIGN 3: AD HOC IMPLEMENTATION OF DESIGN 1.

14.5 CAPRICAILLIE CRAP

8251 14.5.1 model

14.6 DESIGN 4 – NO LOCATION INFO

14.7 SUMMARY AND OUTLOOK

8252
8253

8254

15

OPEN POPULATION MODELS

15.1 INTRODUCTION

8255 15.1.1 Overview of Population Dynamics

8256 15.1.2 Animal movement related to population demography

8257 15.1.3 Basic assumptions of JS and CJS models

15.2 TRADITIONAL JOLLY-SEBER MODELS

8258 15.2.1 Data Augmentation for the Jolly-Seber Model

8259 15.2.2 Mist-netting example

8260 15.2.3 Shortcomings of the traditional JS models

15.3 SPATIAL JOLLY-SEBER MODELS

(15.3.1)

- 8261 **15.3.1 Mist-netting example**
- 15.4 TRADITIONAL CJS MODELS**
- 8262 **15.4.1 Migratory fish example**
- 15.5 MULTI-STATE CJS MODELS**
- 8263 **15.5.1 Migratory fish example**
- 15.6 SPATIAL CJS MODELS**
- 8264 **15.6.1 Migratory fish example**
- 15.7 MOVING ACTIVITY CENTERS**
- 8265 **15.7.1 Migratory Fish Example Notes**
- 15.8 SUMMARY AND OUTLOOK**



8266

Part IV

8267

8268

Super-Advanced SCR Models

8269
8270

16

8271
8272

DEVELOPING MARKOV CHAIN MONTE CARLO SAMPLERS

8273 **16.0.1 Why build your own MCMC algorithm?**

16.1 MCMC AND POSTERIOR DISTRIBUTIONS

(16.1.1)

(16.1.2)

16.2 TYPES OF MCMC SAMPLING

8274 **16.2.1 Gibbs sampling**

(16.2.1)

8275 **16.2.2 Metropolis-Hastings sampling**

8276 **16.2.3 Metropolis-within-Gibbs**

8277 **16.2.4 Rejection sampling and slice sampling**

16.3 MCMC FOR CLOSED CAPTURE-RECAPTURE MODEL MH

16.4 MCMC ALGORITHM FOR MODEL SCR0

8278 **16.4.1 SCR model with binomial encounter process**

8279 **16.4.2 Looking at model output**

8280 **Markov chain time series plots**

8281 **16.4.3 Posterior density plots**

8282 **16.4.4 Serial autocorrelation and effective sample size**

8283 **16.4.5 Summary results**

8284 **16.4.6 Other useful commands**

16.5 MANIPULATING THE STATE-SPACE

16.6 INCREASING COMPUTATIONAL SPEED

8285 **16.6.1 Parallel computing**

8286 **16.6.2 Using C++**

16.7 SUMMARY AND OUTLOOK



8287
8288

17

8289
8290

SPATIAL CAPTURE-RECAPTURE FOR UNMARKED POPULATIONS

**17.1 EXISTING MODELS FOR INFERENCE ABOUT DENSITY IN
UNMARKED POPULATIONS**

17.2 SPATIAL CORRELATION AS INFORMATION

17.3 DATA

17.4 MODEL

(17.4.1)

(17.4.2)

17.5 NORTHERN PARULA EXAMPLE

17.6 IMPROVING PRECISION WITH PRIOR INFORMATION

17.7 DESIGN ISSUES

8291 **17.7.1 How Much Correlation Is Enough?**

8292 **17.7.2 Linear Designs**

8293 **17.7.3 Quadrat counts**

17.8 ALTERNATIVE OBSERVATION MODELS

8294 **17.8.1 Spatial point process models**

17.9 CONCLUSION

8295

8296

8297

8298

8299

18

SPATIAL MARK-RESIGHT MODELS FOR PARTIALLY IDENTIFIABLE POPULATIONS

18.1 BACKGROUND

8300 **18.1.1 Types of partial ID data**

8301 **18.1.2 A short history of mark-resight models**

(18.1.1)

8302

18.2 KNOWN NUMBER OF MARKED INDIVIDUALS

8303 **18.2.1 MCMC for a spatial mark-resight model**

8304 **18.2.2 Binomial encounter model**

18.3 UNKNOWN NUMBER OF MARKED INDIVIDUALS**18.4 IMPERFECT IDENTIFICATION OF MARKED INDIVIDUALS****18.5 HOW MUCH INFORMATION DO MARKED AND UNMARKED
INDIVIDUALS CONTRIBUTE?****18.6 INCORPORATING TELEMETRY DATA****18.7 SUMMARY AND OUTLOOK**

8305
8306

8307

8308

19

2012: A SPATIAL CAPTURE-RECAPTURE ODYSSEY

19.1 10 THESIS OR DISSERTATION TOPICS

19.2 THREE DIMESIONAL SPACE

19.3 GREGARIOUS SPECIES

8309

Part V

8310

8311

Appendices

8312 **APPENDIX I - USEFUL SOFTWARE AND**
8313 **R PACKAGES**

8315 Throughout this book we have used a suite of software and R packages, all of which
8316 are freely available online. To make life a little easier for you, here we provide you
8317 with a list of all software and R packages, download links and some (hopefully)
8318 helpful tips regarding their installation.

19.4 WINBUGS

8319 Although **WinBUGS** (Gilks et al., 1994) is becoming increasingly obsolete with
8320 the faster and more flexible **OpenBUGS** and **JAGS**, there are still situations
8321 in which the program comes in handy. The .exe file can be downloaded from
8322 <http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/contents.shtml>. On 32 bit machines
8323 you can just go ahead and double-click on the .exe file and follow the installation
8324 instructions on the screen. On 64 bit machines, according to the BUGS project you
8325 should download a zip file (from the same page) and unzip it into a folder of your
8326 choice. There are a couple of additional steps to make BUGS run. First, you need
8327 to obtain a key (which is free and valid for life) here: 'http://www.mrc-bsu.cam.ac.uk/bugs/WinBUGS14_immortality_key.txt'. The key comes
8328 with instructions on how to activate it. Second, you need to update the basic
8329 **WinBUGS** version to the most current one (which is from August 2007)
8330 following the instructions given here: 'http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/WinBUGS14_cumulative_patch_No3_06_08_07_RELEASE.txt'. **Win-**
8331 **BUGS** is ready to use after quitting and re-opening it. Remember that **Win-**
8332 **BUGS** only runs on Windows machines. Also, there appears to be a problem
8333 installing the program in Vista, although we have no personal experience with this.

8336 **19.4.1 WinBUGS through R**

8337 While you can run **WinBUGS** as a standalone application, we recommend you ac-
8338 cess it from within **R** using the package **R2WinBUGS** (Sturtz et al., 2005), so you can
8339 conveniently process your output, make graphs etc. **R2WinBUGS** also allows you to
8340 run models in **OpenBUGS** (see below). You can install the package from within

8341 **R** directly from a cran mirror. In addition to the usual package help document
8342 (<http://cran.r-project.org/web/packages/R2WinBUGS/R2WinBUGS.pdf>) you can
8343 also download a short manual with some examples (''http://voteview.com/bayes_beach/R2WinBUGS.pdf'').

19.5 OPENBUGS

8345 **OpenBUGS** is the up-to-date version of **WinBUGS** and can be downloaded
8346 here: ''<http://www.openbugs.info/w/Downloads>'' (Windows, Mac and Linux
8347 versions are available). The name '**OpenBUGS**' refers to the software being open
8348 source, so users do not need to download a license key, like they have to for **Win-**
8349 **BUGS** (although the license key for **WinBUGS** is free and valid for life). For
8350 Windows, install by double-clicking on the .exe file and following the instructions
8351 on the installer screen. Compared to **WinBUGS**, **OpenBUGS** has more built-
8352 in functions. The method of how to determine the right updater for each model
8353 parameter has changed and the user can manually control the MCMC algorithm
8354 used to update model parameters. Several other changes have been implemented
8355 in **OpenBUGS** and a detailed list of differences between the two **BUGS** versions,
8356 can be found at <http://www.openbugs.info/w/OpenVsWin>. We have encountered
8357 convergence problems with simple scr models in this program. There is an extensive
8358 help archive for both **WinBUGS** and **OpenBUGS** and you can subscribe to a
8359 mailing list, where people pose and answer questions of how to use these programs
8360 at <http://www.mrc-bsu.cam.ac.uk/bugs/overview/list.shtml>

8361 19.5.1 OpenBUGS through R

8362 Like **WinBUGS**, **OpenBUGS** can be used as a standalone application or through
8363 **R**. There are several packages that allow **R** to interface with **OpenBUGS**, all of
8364 which can be installed directly from a cran mirror:

8365 **R2WinBUGS**: One of the options in the `bugs()` call is `program`, which lets you
8366 specify either **WinBUGS** or **OpenBUGS**. This is a convenient option because
8367 after having worked through some of this book you will likely be familiar with the
8368 format of `bugs()` output and other functions of the **R2WinBUGS** package.

8369 **R2OpenBUGS**: **R2OpenBUGS** (Sturtz et al., 2005) is very similar to, and actually
8370 based on, **R2WinBUGS** and it is unclear to us what can be gained by using the
8371 former over the latter. Arguments of the `bugs()` call differ slightly between the
8372 two packages and given that **R2WinBUGS** allows for the use of both **OpenBUGS**
8373 and **WinBUGS** it is probably easiest to stick with it.

8374 **BRugs**: **BRugs** (Thomas et al., 2006) can be installed from within **R** directly from
8375 a cran mirror. In addition to the help document at ''<http://www.biostat.umn>.

8376 `edu/~brad/software/BRugs/BRugs_9_21_07.pdf`’’, there is a **WinBUGS** style
8377 manual you can access at ’’<http://www.rni.helsinki.fi/openbugs/OpenBUGS/Docu/BRugs%20Manual.html>’’. **BRugs** has the convenient feature that all pieces of
8378 a **BUGS** analysis can be run from within **R**, including checking the model syntax,
8380 something that requires opening the **BUGS** GUI with other packages.

19.6 JAGS

8381 **JAGS** (Just Another Gibbs Sampler) (Plummer, 2003) runs scr models consider-
8382 ably faster than **WinBUGS**, does not have the convergence problem with simple
8383 scr models we have encountered in **OpenBUGS** but similar to the latter program,
8384 is flexible and constantly updated. Writing a **JAGS** model is virtually identical to
8385 writing a **WinBUGS** model. However, some functions may have slightly different
8386 names and you can look up available functions and their use in the **JAGS** man-
8387 ual. One potential downside is that **JAGS** can be very particular when it comes
8388 to initial values. These may have to be set as close to truth as possible for the
8389 model to start. Although **JAGS** lets you run several parallel Markov chains, this
8390 characteristic interferes with the idea of using overdispersed initial values for the
8391 different chains. Also, we have found that when running models, sometimes **JAGS**
8392 crashes for unclear reasons, taking **R** down with it. Oftentimes, in order to make
8393 it run again you’ll have to go through downloading and installing it again (remove
8394 the non-functioning version first).

8395 **JAGS** has a variety of functions that are not available in **WinBUGS**. For
8396 example, **JAGS** allows you to supply observed data for some deterministic func-
8397 tions of unobserved variables. In **BUGS** we cannot supply data to logical nodes.
8398 Another useful feature is that the adaptive phase of the model (the burn-in) is run
8399 separately from the sampling from the stationary Markov chains. This allows you
8400 to easily add more iterations to the adaptive phase if necessary without the need to
8401 start from 0. There are other, more subtle differences and there is an entire manual
8402 section on differences between **JAGS** and **OpenBUGS**.

8403 **JAGS** is available for download at ’’<http://sourceforge.net/projects/mcmc-jags/files/>’’, together with the R package **rjags** (Plummer, 2011), which
8404 allows running **JAGS** through **R**, user and installation manuals and examples. At
8405 this site **JAGS** is available for Windows and Mac; Linux binaries are distributed
8406 separately and you can find links to various sources here: ’’<http://mcmc-jags.sourceforge.net/>’’. **JAGS** comes with a 32 bit and a 64 bit version and can be
8407 installed by double-clicking on the .exe file and following the instructions on the in-
8408 staller screen. For questions and problems concerning **JAGS** there is a forum online
8409 at <http://sourceforge.net/projects/mcmc-jags/forums/forum/610037>.
8411

19.6.1 JAGS through R

Unlike the two **BUGS** programs, **JAGS** does not have a GUI interface but a command line interface that can be used to run the program as a standalone application. **JAGS** will solely perform the MCMC simulation; analyzing and summarizing the output has to be done outside of **JAGS**. To run **JAGS** through **R** you have two options.

rjags: As mentioned above, **rjags** (Plummer, 2011) can be found together with **JAGS** and was developed/is being maintained by the inventor of **JAGS**, which means it is guaranteed to stay up to date when/as **JAGS** changes. The package can be installed from a cran mirror and the help document can be accessed at '<http://cran.r-project.org/web/packages/rjags/rjags.pdf>'

R2jags: Alternatively, the package **R2jags** (Su and Yajima, 2011) provides a means of accessing **JAGS** through **R**. We prefer **rjags** for the reason named above, as well as because it stores data in a more memory-efficient way and has better **plot()** and **summary()** methods.

19.7 R

At the time of the preparation of this list, **R** for Windows is at version 2.15.0, which can be downloaded at <http://cran.r-project.org/bin/windows/base/> This site also contains helpful tips on how to install **R** in Windows Vista, how to update **R** packages etc. Installation of **R** in Windows is straightforward: download the .exe file, double-click on it and follow the instructions of the Windows installer. The later versions of **R** come with versions for both 64 bit and 32 bit machines. The **R** site ('<http://mirrors.softliste.de/cran/>') has an extensive FAQ section Hornik (2011), which includes instructions on how to install R on Unix and Mac computers.

19.7.1 R packages

This section provides an alphabetical list of useful **R** packages. There is a large number of **R** packages and by no means is this list intended to be complete in terms of what is useful. Rather, we list packages that we are familiar with and that we employ at one point or the other in this book. Unless explicitly stated otherwise, all packages can be installed directly from within **R** trough a cran mirror.

adapt: **adapt** (Genz et al., 2007) is a package for multidimensional numerical integration. The package has been removed from the CRAN repository but can be obtained from '<http://cran.r-project.org/src/contrib/Archive/adapt/>'.

coda: **coda** (Plummer et al., 2006) lets you summarize and perform diagnostics on mcmc output. For a list and description of functions, see the manual at '<http://cran.r-project.org/web/packages/coda/coda.pdf>'.

8448 **gdistance**: **gdistance** (van Etten, 2011) is a package for calculating distances
8449 and routes on geographical grids and can be used to calculate least cost path
8450 surfaces. Manual at '<http://cran.r-project.org/web/packages/gdistance/gdistance.pdf>' .

8452 **igraph**: **igraph** (Csardi and Nepusz, 2006) provides routines for graphs and net-
8453 work analysis. Manual at '<http://cran.r-project.org/web/packages/igraph/igraph.pdf>' .

8455 **inline**: **inline** (Sklyar et al., 2010) allows the user to define R functions with in-
8456 lined **C**, **C++** or **Fortran** code. Manual at <http://cran.r-project.org/web/packages/inline/inline.pdf>.

8458 **maps**: **maptools** (?) is a library of maps. Manual at '<http://cran.r-project.org/web/packages/maps/index.html>' .

8460 **maptools**: **maptools** (Lewin-Koh et al., 2011) provides a set of tools for reading
8461 and manipulating spatial data, especially ESRI shapefiles. Manual at '<http://cran.r-project.org/web/packages/maptools/maptools.pdf>' .

8463 **R2cuba**: **R2cuba** (Hahn et al., 2010) is another package for multidimensional integra-
8464 tion. Manual at '<http://cran.r-project.org/web/packages/R2Cuba/R2Cuba.pdf>' .

8466 **raster**: **raster** (van Etten, 2012) provides functions for geographic analysis and
8467 modeling with raster data. Manual at '<http://cran.r-project.org/web/packages/raster/raster.pdf>' .

8469 **Rcpp**: **Rcpp** (Eddelbuettel and François, 2011) provides R functions as well as a
8470 **C++** library which facilitate the integration of **R** and **C++**. Manual at <http://cran.r-project.org/web/packages/Rcpp/Rcpp.pdf>.

8472 **RcppArmadillo**: **RcppArmadillo** (François et al., 2011) is a templated **C++** linear
8473 algebra library, integrating the **Armadillo** library and **R**. Manual at <http://cran.r-project.org/web/packages/RcppArmadillo/RcppArmadillo.pdf>.

8475 **reshape**: **reshape** (Wickham and Hadley, 2007) allows you to easily manipulate,
8476 summarize and reshape data. Manual at '<http://cran.r-project.org/web/packages/reshape/reshape.pdf>' .

8478 **rgeos**: **rgeos** (Bivand and Rundel, 2011) provides many useful functions for spatial
8479 operations such as intersecting or buffering spatial features. Manual at '<http://cran.r-project.org/web/packages/rgeos/rgeos.pdf>' .

8481 **SCRbayes**: (Russell et al., 2012)

8482 **secr: secr** (Efford et al., 2009a)

8483 **shapefiles: shapefiles** (Stabler, 2006) allows you to read and write ESRI shape-
8484 files (i.e. shapefiles you would use in **ArcGIS**). Manual at '<http://cran.r-project.org/web/packages/shapefiles/shapefiles.pdf>'.

8486 **snow, snowfall:** **snow** (Tierney et al., 2011) and **snowfall** (Knaus, 2010) pro-
8487 vide functionality for parallel computing. The latter is a more user-friendly wrap-
8488 per around the former. Manuals at <http://cran.r-project.org/web/packages/snowfall/snowfall.pdf> and <http://cran.r-project.org/web/packages/snow/snow.pdf>.

8491 **sp: sp** (Pebesma and Bivand, 2011) is a package for plotting, selecting, subsetting
8492 etc. spatial data. **sp** and **spatstat** (see below) are complementary in many ways
8493 and data formats can be easily converted between the two packages. Manual at
8494 '<http://cran.r-project.org/web/packages/sp/sp.pdf>'.

8495 **SPACECAP:** (Gopalaswamy et al., 2012a)

8496 **spatstat: spatstat** (Baddeley and Turner, 2005) is an extensive package for an-
8497 alyzing spatial data. We use it, for example, to generate random points within a
8498 state space that cannot be described as a rectangle but consists of a (or several)
8499 arbitrary polygon(s). Manual at '<http://cran.r-project.org/web/packages/spatstat/spatstat.pdf>'.

8501 **unmarked:**

8502

8503 #####

8504 **References**

8505

BIBLIOGRAPHY

- 8508 Aitkin, M. (1991), "Posterior bayes factors," *Journal of the Royal Statistical Society. Series B (Methodological)*, 53, 111–142.
- 8509
- 8510 Alho, J. M. (1990), "Logistic regression in capture-recapture models," *Biometrics*, 46, 623–635.
- 8511
- 8512 Alpízar-Jara, R. and Pollock, K. H. (1996), "A combination line transect and capture-recapture sampling model for multiple observers in aerial surveys," *Environmental and Ecological Statistics*, 3, 311–327.
- 8513
- 8514
- 8515 Amstrup, S. C., McDonald, T. L., and Manly, B. F. J. (2005), *Handbook of capture-recapture analysis*, Princeton Univ Pr.
- 8516
- 8517 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.
- 8518
- 8519
- 8520 Arnason, A., Schwarz, C., and Gerrard, J. (1991), "Estimating closed population size and number of marked animals from sighting data," *Journal of Wildlife Management*, 55, 716–730.
- 8521
- 8522
- 8523 Baddeley, A. and Turner, R. (2005), "Spatstat: an R package for analyzing spatial point patterns," *Journal of Statistical Software*, 12, 1–42, ISSN 1548-7660.
- 8524
- 8525 Bales, S., Hellgren, E., Leslie Jr, D., and Hemphill Jr, J. (2005), "Dynamics of a recolonizing population of black bears in the Ouachita Mountains of Oklahoma," *Wildlife Society Bulletin*, 33, 1342–1351.
- 8526
- 8527
- 8528 Balme, G. A., Slotow, R., and Hunter, L. T. B. (2010), "Edge effects and the impact of non-protected areas in carnivore conservation: leopards in the Phinda–Mkuze Complex, South Africa," *Animal Conservation*, 13, 315–323.
- 8529
- 8530
- 8531 Bartmann, R. M., White, G. C., Carpenter, L. H., and Garrott, R. A. (1987), "Aerial Mark-Recapture Estimates of Confined Mule Deer in Pinyon-Juniper Woodland," *The Journal of Wildlife Management*, 51, 41–46.
- 8532
- 8533
- 8534 Berger, J., Liseo, B., and Wolpert, R. (1999), "Integrated likelihood methods for eliminating nuisance parameters," *Statistical Science*, 14, 1–28.
- 8535
- 8536 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.
- 8537
- 8538
- 8539 Bibby, C. J. and Buckland, S. T. (1987), "Bias of bird census results due to detectability varying with habitat," *Acta Ecologica*, 8, 103–112.
- 8540

- 8541 Bibby, C. J., Burgess, N. D., and Hill, D. A. (1992), "Bird census techniques:
8542 Academic Press," *London, UK*, 257.
- 8543 Bivand, R. and Rundel, C. (2011), *rgeos: Interface to Geometry Engine - Open*
8544 *Source (GEOS)*, r package version 0.1-8.
- 8545 Blair, W. (1940), "A study of prairie deer-mouse populations in southern Michigan,"
8546 *American Midland Naturalist*, 24, 273–305.
- 8547 Bolker, B. (2008), *Ecological models and data in R*, Princeton Univ Pr.
- 8548 Borchers, D. L. (2012), "A non-technical overview of spatially explicit capture–
8549 recapture models," *Journal of Ornithology*, 152, 1–10.
- 8550 Borchers, D. L., Buckland, S. T., and Zucchini, W. (2002), *Estimating animal*
8551 *abundance: closed populations*, vol. 13, Springer Verlag.
- 8552 Borchers, D. L. and Efford, M. G. (2008), "Spatially explicit maximum likelihood
8553 methods for capture–recapture studies," *Biometrics*, 64, 377–385.
- 8554 Borchers, D. L., Zucchini, W., and Fewster, R. M. (1998), "Mark-recapture models
8555 for line transect surveys," *Biometrics*, 54, 1207–1220.
- 8556 Boulanger, J. and McLellan, B. (2001), "Closure violation in DNA-based mark–
8557 recapture estimation of grizzly bear populations," *Canadian Journal of Zoology*,
8558 79, 642–651.
- 8559 Bowden, D. C. (1993), "A simple technique for estimating population size. Technical
8560 Report 93/12." Tech. rep., Department of Statistics, Colorado State University,
8561 Fort Collins, Colorado, USA.
- 8562 Brooks, S. P., Catchpole, E. A., and Morgan, B. J. T. (2000), "Bayesian Animal
8563 Survival Estimation," *Statistical Science*, 15, 357–376.
- 8564 Buckland, S., Anderson, D., Burnham, K., Laake, J., Borchers, D., and L. T. (2001),
8565 *Introduction to distance sampling: estimating abundance of biological populations*,
8566 Oxford, UK: Oxford University Press.
- 8567 Buckland, S. T. (2004), *Advanced distance sampling*, Oxford University Press, USA.
- 8568 Burnham, K., Anderson, D., and Laake, J. (1980), "Estimation of density from line
8569 transect sampling of biological populations," *Wildlife monographs*, 3–202.
- 8570 Burnham, K., Anderson, D., White, G., Brownie, C., and Pollock, K. (1987),
8571 "Design and analysis methods for fish survival experiments based on release–
8572 recapture," *American Fisheries Society Monograph 5. American Fisheries Society*,
8573 *Bethesda Maryland. 1987. 737.*
- 8574 Burnham, K. P. and Anderson, D. R. (2002), *Model selection and multimodel in-*
8575 *ference: a practical information-theoretic approach*, Springer Verlag.
- 8576 Burnham, K. P. and Overton, W. S. (1978), "Estimation of the size of a closed
8577 population when capture probabilities vary among animals," *Biometrika*, 65,
8578 625.
- 8579 Burt, W. (1943), "Territoriality and home range concepts as applied to mammals,"
8580 *Journal of mammalogy*, 24, 346–352.
- 8581 Casella, G. and Berger, R. L. (2002), *Statistical inference*, Duxbury Press.
- 8582 Chandler, R. B. and Royle, J. A. (In press), "Spatially-explicit models for inference
8583 about density in unmarked or partially marked populations," *Annals of Applied*

- 8584 *Statistics*.
- 8585 Chandler, R. B., Royle, J. A., and King, D. (2011), "Inference about density and
8586 temporary emigration in unmarked populations," *Ecology*, 92, 1429–1435.
- 8587 Clobert, J., Danchin, E., Dhondt, A., and Nichols, J. (2001), *Dispersal*, Oxford.
- 8588 Converse, S., White, G., Farris, K., and Zack, S. (2006), "Small mammals and forest
8589 fuel reduction: national-scale responses to fire and fire surrogates," *Ecological
8590 Applications*, 16, 1717–1729.
- 8591 Converse, S. J. and Royle, J. A. (2012), "Dealing with incomplete and variable
8592 detectability in multi-year, multi-site monitoring of ecological populations," in
8593 *Design and Analysis of Long-term Ecological Monitoring Studies*, eds. Gitzen,
8594 R. R., Millspaugh, J. J., Cooper, A. B., and Licht, D. S., Cambridge University
8595 Press, pp. 426–442.
- 8596 Cooch, E. and White, G. (2006), "Program MARK: a gentle introduction," *available
8597 online with the MARK programme*, 7.
- 8598 Coull, B. A. and Agresti, A. (1999), "The Use of Mixed Logit Models to Reflect
8599 Heterogeneity in Capture-Recapture Studies," *Biometrics*, 55, 294–301.
- 8600 Cressie, N. (1991), *Statistics for spatial data*, Wiley Series in Probability and Math-
8601 ematical Statistics.
- 8602 Csardi, G. and Nepusz, T. (2006), "The igraph software package for complex net-
8603 work research," *InterJournal, Complex Systems*, 1695.
- 8604 Dawson, D. K. and Efford, M. G. (2009), "Bird population density estimated from
8605 acoustic signals," *Journal of Applied Ecology*, 46, 1201–1209.
- 8606 DeGraaf, R. M. and Yamasaki, M. (2001), *New England wildlife: habitat, natural
8607 history, and distribution*, University Press of New England.
- 8608 Dice, L. R. (1938), "Some census methods for mammals," *Journal of Wildlife Man-
8609 agement*, 2, 119–130.
- 8610 Dixon, P. (2002), "Bootstrap resampling," *Encyclopedia of environmetrics*.
- 8611 Dorazio, R. M. (2007), "On the choice of statistical models for estimating occurrence
8612 and extinction from animal surveys," *Ecology*, 88, 2773–2782.
- 8613 Dorazio, R. M. and Royle, J. A. (2003), "Mixture models for estimating the size
8614 of a closed population when capture rates vary among individuals," *Biometrics*,
8615 59, 351–364.
- 8616 Durban, J. and Elston, D. (2005), "Mark-recapture with occasion and individual
8617 effects: abundance estimation through Bayesian model selection in a fixed dimen-
8618 sional parameter space," *Journal of agricultural, biological, and environmental
8619 statistics*, 10, 291–305.
- 8620 Eddelbuettel, D. and François, R. (2011), "Rcpp: Seamless R and C++ Integra-
8621 tion," *Journal of Statistical Software*, 40, 1–18.
- 8622 Efford, M. (2011), *secr: Spatially explicit capture-recapture models*, R package ver-
8623 sion 2.3.1.
- 8624 Efford, M. G. (2004), "Density estimation in live-trapping studies," *Oikos*, 106,
8625 598–610.
- 8626 Efford, M. G., Borchers, D. L., and Byrom, A. E. (2009a), "Density estimation by

- 8627 spatially explicit capture–recapture: likelihood-based methods,” *Modeling demo-*
8628 *graphic processes in marked populations*, 255–269.
- 8629 Efford, M. G. and Dawson, D. K. (2009), “Effect of distance-related heterogeneity
8630 on population size estimates from point counts,” *The Auk*, 126, 100–111.
8631 — (2010), “SECR for acoustic data,” .
- 8632 Efford, M. G., Dawson, D. K., and Borchers, D. L. (2009b), “Population density
8633 estimated from locations of individuals on a passive detector array,” *Ecology*, 90,
8634 2676–2682.
- 8635 Efford, M. G., Dawson, D. K., and Robbins, C. S. (2004), “DENSITY: software for
8636 analysing capture-recapture data from passive detector arrays,” *Animal Biodi-*
8637 *versity and Conservation*, 27, 217–228.
- 8638 Efford, M. G. and Fewster, R. M. (2012), “Estimating population size by spatially
8639 explicit capture–recapture,” *Oikos*.
- 8640 Efford, M. G., Warburton, B., Coleman, M. C., and Barker, R. J. (2005), “A field
8641 test of two methods for density estimation,” *Wildlife Society Bulletin*, 33, 731–
8642 738.
- 8643 F Dormann, C., M McPherson, J., B Araújo, M., Bivand, R., Bolliger, J., Carl, G.,
8644 G Davies, R., Hirzel, A., Jetz, W., Daniel Kissling, W., et al. (2007), “Methods to
8645 account for spatial autocorrelation in the analysis of species distributional data:
8646 a review,” *Ecography*, 30, 609–628.
- 8647 Farnsworth, G. L., Pollock, K. H., Nichols, J. D., Simons, T. R., E., H. J., and
8648 R., S. J. (2002), “A removal model for estimating detection probabilities from
8649 point-count surveys,” *The Auk*, 119, 414–425.
- 8650 Fienberg, S. E., Johnson, M. S., and Junker, B. W. (1999), “Classical multilevel and
8651 Bayesian approaches to population size estimation using multiple lists,” *Journal*
8652 *of the Royal Statistical Society of London A*, 163, 383–405.
- 8653 Fiske, I. J. and Chandler, R. B. (2011), “unmarked: An R package for fitting
8654 hierarchical models of wildlife occurrence and abundance,” *Journal Of Statistical*
8655 *Software*, 43, 1–23.
- 8656 Foster, R. J. and Harmsen, B. J. (2012), “A critique of density estimation from
8657 camera-trap data,” *The Journal of Wildlife Management*, 76, 224–236.
- 8658 François, R., Eddelbuettel, D., and Bates, D. (2011), *RcppArmadillo: Rcpp inte-*
8659 *gration for Armadillo templated linear algebra library*, r package version 0.2.25.
- 8660 García-Alaníz, N., Naranjo, E. J., and Mallory, F. F. (2010), “Hair-snares: A
8661 non-invasive method for monitoring felid populations in the Selva Lacandona,
8662 Mexico,” *Tropical Conservation Science*, 3, 403–411.
- 8663 Gardner, B., Reppucci, J., Lucherini, M., and Royle, J. (2010a), “Spatially explicit
8664 inference for open populations: estimating demographic parameters from camera-
8665 trap studies,” *Ecology*, 91, 3376–3383.
- 8666 Gardner, B., Royle, J. A., and Wegan, M. T. (2009), “Hierarchical models for
8667 estimating density from DNA mark-recapture studies,” *Ecology*, 90, 1106–1115.
- 8668 Gardner, B., Royle, J. A., Wegan, M. T., Rainbolt, R. E., and Curtis, P. D. (2010b),
8669 “Estimating black bear density using DNA data from hair snares,” *The Journal*

- 8670 *of Wildlife Management*, 74, 318–325.
- 8671 Garshelis, D. L. and Hristienko, H. (2006), “State and provincial estimates of Amer-
8672 ican black bear numbers versus assessments of population trend,” *Ursus*, 17, 1–7.
- 8673 Gelman, A. (2006), “Prior distributions for variance parameters in hierarchical
8674 models,” *Bayesian analysis*, 1, 515–533.
- 8675 Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004), *Bayesian data
8676 analysis, second edition.*, Bocan Raton, Florida, USA: CRC/Chapman & Hall.
- 8677 Gelman, A., Meng, X. L., and Stern, H. (1996), “Posterior predictive assessment
8678 of model fitness via realized discrepancies,” *Statistica Sinica*, 6, 733–759.
- 8679 Gelman, A. and Rubin, D. B. (1992), “Inference from iterative simulation using
8680 multiple sequences,” *Statistical Science*, 7, 457–511.
- 8681 Geman, S. and Geman, D. (1984), “Stochastic relaxation, Gibbs distributions, and
8682 the Bayesian restoration of images,” *IEEE Transactions on Pattern Analysis and
8683 Machine Intelligence*, PAMI-6, 721–741.
- 8684 Genz, A. S., Meyer, M. R., Lumley, T., and Maechler, M. (2007), “The adapt
8685 Package. R package version 1.0-4.” .
- 8686 Gilks, W. R., Thomas, A., and Spiegelhalter, D. J. (1994), “A Language and Pro-
8687 gram for Complex Bayesian Modelling,” *Journal of the Royal Statistical Society.
8688 Series D (The Statistician)*, 43, 169–177, ArticleType: primary_article / Issue
8689 Title: Special Issue: Conference on Practical Bayesian Statistics, 1992 (3) / Full
8690 publication date: 1994 / Copyright 1994 Royal Statistical Society.
- 8691 Gopalaswamy, A. M. (2012), “Capture-recapture models, spatially explicit,” in
8692 *Encyclopedia of Environmentrics Second Edition*, eds. El-Shaarawi, A. H. and
8693 Piegorsch, W., John Wiley and Sons Ltd, Chichester, UK.
- 8694 Gopalaswamy, A. M., Royle, A. J., Hines, J., Singh, P., Jathanna, D., Kumar,
8695 N. S., and Karanth, K. U. (2012a), “Program SPACECAP: software for estimat-
8696 ing animal density using spatially explicit capturerecapture models,” *Methods in
8697 Ecology and Evolution*, online early, r package version 1.0.4.
- 8698 Gopalaswamy, A. M., Royle, J. A., Delampady, M., Nichols, J. D., Karanth, K. U.,
8699 and Macdonald, D. W. (2012b), “Density estimation in tiger populations: com-
8700 bining information for strong inference,” *Ecology*, 93, 1741–1751.
- 8701 Greig-Smith, P. (1964), *Quantitative plant ecology*, Butterworths (Washington).
- 8702 Hahn, T., Bouvier, A., and Kiêu, K. (2010), *R2Cuba: Multidimensional Numerical
8703 Integration*, r package version 1.0-6.
- 8704 Hall, R. J., Henry, P. F. P., and Bunck, C. M. (1999), “Fifty-year trends in a box
8705 turtle population in Maryland,” *Biological Conservation*, 88, 165–172.
- 8706 Hanski, I. A. (1999), *Metapopulation Ecology*, Oxford Univiversity Press.
- 8707 Hawkins, C. E. and Racey, P. A. (2005), “Low population density of a tropical forest
8708 carnivore, Cryptoprocta ferox: implications for protected area management,”
8709 *Oryx*, 39, 35–43.
- 8710 Hayes, R. J. and Buckland, S. T. (1983), “Radial distance models for the line
8711 transect method.” *Biometrics*, 39, 29–42.
- 8712 Hayne, D. (1950), “Apparent home range of Microtus in relation to distance between

- 8713 traps," *Journal of mammalogy*, 31, 26–39.
- 8714 Hayne, D. W. (1949), "An Examination of the Strip Census Method for Estimating
8715 Animal Populations," *The Journal of Wildlife Management*, 13, 145–157.
- 8716 Hedley, S. L., Buckland, S. T., and Borchers, D. L. (1999), "Spatial modelling from
8717 line transect data," *Journal of Cetacean Research and Management*, 1, 255–264.
- 8718 Hestbeck, J. B. and Malecki, R. A. (1989), "Mark-resight estimate of Canada Goose
8719 midwinter numbers," *Journal of Wildlife Management*, 53, 749–752.
- 8720 Higdon, D. (1998), "A process-convolution approach to modelling temperatures in
8721 the North Atlantic Ocean," *Environmental and Ecological Statistics*, 5, 173–190.
- 8722 Hobbs, N. T. (2011), "An Ecological Modeler's Primer on JAGS," .
- 8723 Holdenried, R. (1940), "A population study of the long-eared chipmunk (*Eutamias*
8724 *quadrimaculatus*) in the central Sierra Nevada," *Journal of Mammalogy*, 21, 405–
8725 411.
- 8726 Hornik, K. (2011), "The R FAQ," ISBN 3-900051-08-9.
- 8727 Huggins, R. M. (1989), "On the statistical analysis of capture experiments,"
8728 *Biometrika*, 76, 133.
- 8729 Hurlbert, S. H. (1984), "Pseudoreplication and the design of ecological field exper-
8730 iments," *Ecological monographs*, 54, 187–211.
- 8731 Illian, J., Penttinen, A., Stoyan, H., and Stoyan, D. (2008), *Statistical analysis and*
8732 *modelling of spatial point patterns*, Wiley.
- 8733 Ivan, J. (2012), "Density, demography, and seasonal movements of snowshoe hares
8734 in central Colorado," Ph.D. thesis, Colorado State University.
- 8735 Ivan, J., White, G., and Shenk, T. (2013a), "Using auxiliary telemetry information
8736 to estimate animal density from capture-recapture data," *Ecology*.
- 8737 — (2013b), "Using simulation to compare methods for estimating density from
8738 capture-recapture data," *Ecology*.
- 8739 Jackson, R., Roe, J., Wangchuk, R., and Hunter, D. (2006), "Estimating Snow
8740 Leopard Population Abundance Using Photography and Capture-Recapture
8741 Techniques," *Wildlife Society Bulletin*, 34, 772–781.
- 8742 Jett, D. A. and Nichols, J. D. (1987), "A Field Comparison of Nested Grid and
8743 Trapping Web Density Estimators," *Journal of Mammalogy*, 68, 888–892.
- 8744 Johnson (2010), "A Model-Based Approach for Making Ecological Inference from
8745 Distance Sampling Data," *Biometrics*, 66, 310318.
- 8746 Johnson, D. H. (1999), "The insignificance of statistical significance testing," *The*
8747 *journal of wildlife management*, 63, 763–772.
- 8748 Kadane, J. B. and Lazar, N. A. (2004), "Methods and criteria for model selection,"
8749 *Journal of the American Statistical Association*, 99, 279–290.
- 8750 Karanth, K. U. (1995), "Estimating tiger *Panthera tigris* populations from camera-
8751 trap data using capture–recapture models," *Biological Conservation*, 71, 333–338.
- 8752 Karanth, K. U. and Nichols, J. D. (1998), "Estimation of Tiger Densities in India
8753 Using Photographic Captures and Recaptures," *Ecology*, 79, 2852–2862.
- 8754 — (2000), "Ecological status and conservation of tigers in India," *WCS, US Fish*
8755 *and Wildlife Service. Centre for Wildlife Studies. Bangalore, India*, 123.

- 8756 — (2002), *Monitoring tigers and their prey: a manual for researchers, managers*
8757 *and conservationists in {T}ropical {A}sia*, Bangalore, India: Centre for Wildlife
8758 Studies.
- 8759 Kass, R. and Wasserman, L. (1996), “The selection of prior distributions by formal
8760 rules,” *Journal of the American Statistical Association*, 91, 1343–1370.
- 8761 Kays, R. W., Slauson, K. M., Long, R. A., MacKay, P., Zielinski, W. J., and Ray,
8762 J. C. (2008), “Remote cameras.” *Noninvasive survey methods for carnivores*,
8763 110–140.
- 8764 Kelly, M., Noss, A., Di Bitetti, M., Maffei, L., Arispe, R., Paviolo, A., De Angelo,
8765 C., and Di Blanco, Y. (2008), “Estimating puma densities from camera trapping
8766 across three study sites: Bolivia, Argentina, and Belize,” *Journal of Mammalogy*,
8767 89, 408–418.
- 8768 Kendall, K. C., Stetz, J. B., Boulanger, J., Macleod, A. C., Paetkau, D., and
8769 White, G. C. (2009), “Demography and genetic structure of a recovering grizzly
8770 bear population,” *The Journal of Wildlife Management*, 73, 3–16.
- 8771 Kendall, W. L., Nichols, J. D., and Hines, J. E. (1997), “Estimating Temporary Em-
8772 igration Using Capture-Recapture Data with Pollock’s Robust Design,” *Ecology*,
8773 78, 563–578.
- 8774 Kéry, M. (2010), *Introduction to WinBUGS for Ecologists: Bayesian Approach to*
8775 *Regression, ANOVA, Mixed Models and Related Analyses*, Academic Press.
- 8776 Kéry, M. (2011), “Towards the modelling of true species distributions,” *Journal of*
8777 *Biogeography*, 38, 617–618.
- 8778 Kéry, M., Gardner, B., and Monnerat, C. (2010), “Predicting species distributions
8779 from checklist data using site-occupancy models,” *Journal of Biogeography*, 37,
8780 1851–1862.
- 8781 Kéry, M., Gardner, B., Stoeckle, T., Weber, D., and Royle, J. A. (2011), “Use of
8782 Spatial Capture-Recapture Modeling and DNA Data to Estimate Densities of
8783 Elusive Animals,” *Conservation Biology*, 25, 356–364.
- 8784 Kéry, M., Royle, J. A., and Schmid, H. (2005), “Modeling avian abundance from
8785 replicated counts using binomial mixture models,” *Ecological Applications*, 15,
8786 1450–1461.
- 8787 Kéry, M. and Schaub, M. (2012), *Bayesian Population Analysis Using WinBugs*,
8788 Academic Press.
- 8789 King, R. and Brooks, S. (2001), “On the Bayesian analysis of population size,”
8790 *Biometrika*, 88, 317–336.
- 8791 King, R., Brooks, S., and Coulson, T. (2008), “Analyzing Complex Capture-
8792 Recapture Data in the Presence of Individual and Temporal Covariates and
8793 Model Uncertainty,” *Biometrics*, 64, 1187–1195.
- 8794 Knaus, J. (2010), *snowfall: Easier cluster computing (based on snow)*., r package
8795 version 1.84.
- 8796 Kohn, M., York, E., Kamradt, D., Haught, G., Sauvajot, R., and Wayne, R. (1999),
8797 “Estimating population size by genotyping faeces,” *Proceedings of the Royal So-*
8798 *ciety of London. Series B: Biological Sciences*, 266, 657–663.

- 8799 Krebs, C. J. (1999), *Ecological methodology*, Menlo Park, CA: Benjamin/Cummings.
8800
8801 Kucera and Barrett (2011), “missing,” *missing*, missing.
8802 Kuo, L. and Mallick, B. (1998), “Variable selection for regression models,” *Sankhy: The Indian Journal of Statistics, Series B*, 60, 65–81.
8803
8804 Laird, N. M. and Ware, J. H. (1982), “Random-effects models for longitudinal data,” *Biometrics*, 38, 963–974.
8805
8806 Langtimm, C. A., Dorazio, R. M., Stith, B. M., and Doyle, T. J. (2011), “New aerial survey and hierarchical model to estimate manatee abundance,” *The Journal of Wildlife Management*, 75, 399–412.
8807
8808 Le Cam, L. (1990), “Maximum likelihood: an introduction,” *International Statistical Review/Revue Internationale de Statistique*, 58, 153–171.
8809
8810 Lebreton, J. D., Burnham, K. P., Clobert, J., and Anderson, D. R. (1992), “Modeling Survival and Testing Biological Hypotheses Using Marked Animals: A Unified Approach with Case Studies,” *Ecological Monographs*, 62, 67–118.
8811
8812 Lele, S. R. and Keim, J. L. (2006), “Weighted distributions and estimation of resource selection probability functions,” *Ecology*, 87, 3021–3028.
8813
8814 Lele, S. R., Moreno, M., and Bayne, E. (2012), “Dealing with detection error in site occupancy surveys: What can we do with a single survey?” *Journal of Plant Ecology*.
8815
8816 Lewin-Koh, N. J., Bivand, R., contributions by Edzer J. Pebesma, Archer, E., Baddeley, A., Bibiko, H.-J., Dray, S., Forrest, D., Friendly, M., Giraudoux, P., Golicher, D., Rubio, V. G., Hausmann, P., Hufthammer, K. O., Jagger, T., Luque, S. P., MacQueen, D., Niccolai, A., Short, T., Stabler, B., and Turner, R. (2011), *maptools: Tools for reading and handling spatial objects*, r package version 0.8-10.
8817
8818 Link, W. (2003), “Nonidentifiability of Population Size from Capture-Recapture Data with Heterogeneous Detection Probabilities,” *Biometrics*, 59, 1123–1130.
8819
8820 Link, W. A. and Barker, R. J. (1994), “Density Estimation Using the Trapping Web Design: A Geometric Analysis,” *Biometrics*, 50, 733–745.
8821
8822 — (2006), “Model weights and the foundations of multimodel inference,” *Ecology*, 87, 2626–2635.
8823
8824 — (2010), *Bayesian Inference: With Ecological Applications*, London, UK: Academic Press.
8825
8826 Link, W. A. and Eaton, M. J. (2011), “On thinning of chains in MCMC,” *Methods in Ecology and Evolution*, 3, 112–115.
8827
8828 Link, W. A., Yoshizaki, J., Bailey, L. L., and Pollock, K. H. (2010), “Uncovering a latent multinomial: analysis of markrecapture data with misidentification,” *Biometrics*, 66, 178–185.
8829
8830 Liu and Wu (1999), “Parameter expansion for data augmentation,” *J Am Stat Assoc*, 94, 1264–1274.
8831
8832 Lukacs, P. M. and Burnham, K. P. (2005), “Estimating population size from DNA-based closed capture-recapture data incorporating genotyping error,” *Journal of*

- 8842 *Wildlife Management*, 69, 396–403.
- 8843 Lunn, D., Spiegelhalter, D., Thomas, A., and Best, N. (2009), “The BUGS project:
8844 Evolution, critique, and future directions,” *Statistics in Medicine*, 28, 3049–3067.
- 8845 Mace, R., Minta, S., Manley, T., and Aune, K. (1994), “Estimating grizzly bear
8846 population size using camera sightings,” *Wildlife Society Bulletin*, 22, 74–83.
- 8847 MacEachern, S. N. and Berliner, L. M. (1994), “Subsampling the Gibbs sampler,”
8848 *American Statistician*, 48, 188–190.
- 8849 MacKay, P., Smith, D. A., Long, R. A., Parker, M., Long, R. A., MacKay, P.,
8850 Zielinski, W. J., and Ray, J. C. (2008), “Scat detection dogs,” *Noninvasive survey
8851 methods for carnivores*, 183–222.
- 8852 MacKenzie, D. I., Nichols, J. D., Lachman, G. B., Droege, S., Royle, J. A., and
8853 Langtimm, C. A. (2002), “Estimating site occupancy rates when detection prob-
8854 abilities are less than one,” *Ecology*, 83, 2248–2255.
- 8855 MacKenzie, D. I., Nichols, J. D., Royle, J. A., Pollock, K. H., Bailey, L. L., and
8856 Hines, J. E. (2006), *Occupancy estimation and modeling: inferring patterns and
8857 dynamics of species occurrence*, Academic Press.
- 8858 Magoun, A. J., Long, C. D., Schwartz, M. K., Pilgrim, K. L., Lowell, R. E., and
8859 Valkenburg, P. (2011), “Integrating motion-detection cameras and hair snags for
8860 wolverine identification,” *The Journal of Wildlife Management*, 75, 731–739.
- 8861 Manly, B., McDonald, L., Thomas, D., McDonald, T., and Erickson, W. (2002),
8862 *Resource selection by animals: statistical design and analysis for field studies*,
8863 Springer, 2nd ed.
- 8864 Marques, T., Thomas, L., Ward, J., DiMarzio, N., and Tyack, P. (2009), “Es-
8865 timating cetacean population density using fixed passive acoustic sensors: An
8866 example with Blainvilles beaked whales,” *The Journal of the Acoustical Society
8867 of America*, 125, 1982.
- 8868 McCarthy, M. A. (2007), *Bayesian Methods for Ecology*, Cambridge: Cambridge
8869 University Press.
- 8870 McClintock, B. and Hoeting, J. (2010), “Bayesian analysis of abundance for bino-
8871 mial sighting data with unknown number of marked individuals,” *Environmental
8872 and Ecological Statistics*, 17, 317–332.
- 8873 McClintock, B. and White, G. (2012), “From NOREMARK to MARK: software for
8874 estimating demographic parameters using mark-resight methodology,” *Journal of
8875 Ornithology*, 152, 641–650.
- 8876 McClintock, B., White, G., Antolin, M., and Tripp, D. (2009a), “Estimating abun-
8877 dance using mark-resight when sampling is with replacement or the number of
8878 marked individuals is unknown,” *Biometrics*, 65, 237–246.
- 8879 McClintock, B., White, G., and Burnham, K. (2006), “A robust design mark-resight
8880 abundance estimator allowing heterogeneity in resighting probabilities,” *Journal
8881 of agricultural, biological, and environmental statistics*, 11, 231–248.
- 8882 McClintock, B. T., White, G. C., Burnham, K. P., and Pryde, M. A. (2009b),
8883 “A generalized mixed effects model of abundance for mark-resight data when
8884 sampling is without replacement,” in *Modeling demographic processes in marked*

- 8885 *populations*, ed. Thomson, D., New York: Springer, pp. 271–289.
- 8886 McCullagh, P. and Nelder, J. A. (1989), *Generalized linear models*, Chapman &
8887 Hall/CRC.
- 8888 Millar, R. B. (2009), “Comparison of hierarchical Bayesian models for overdispersed
8889 count data using DIC and Bayes’ Factors,” *Biometrics*, 65, 962–969.
- 8890 Mills, L. S., Citta, J. J., Lair, K. P., Schwartz, M. K., and Tallmon, D. A. (2000),
8891 “Estimating animal abundance using noninvasive DNA sampling: promise and
8892 pitfalls,” *Ecological Applications*, 10, 283–294.
- 8893 Minta, S. and Mangel, M. (1989), “A Simple Population Estimate Based on Simula-
8894 tion for Capture-Recapture and Capture-Resight Data,” *Ecology*, 70, 1738–1751.
- 8895 Møller, J. and Waagepetersen, R. P. (2004), *Statistical inference and simulation
8896 for spatial point processes*, Chapman & Hall/CRC.
- 8897 Mohr, C. (1947), “Table of equivalent populations of North American small mam-
8898 mals,” *American midland naturalist*, 37, 223–249.
- 8899 Moldenhauer, R. R. and Regelski, D. J. (1996), “Northern Parula (*Parula ameri-
8900 cana*),” in *The Birds of North America Online*, ed. Poole, A., Ithaca, NY: Cornell
8901 Lab of Ornithology.
- 8902 Molinari-Jobin, A., Kéry, M., Marboutin, E., Marucco, F., Zimmermann, F., Molini-
8903 nari, P., Frick, H., Wölfl, S., Bled, F., Breitenmoser-Würsten, C., Fuxjäger, C.,
8904 Huber, T., I., K., Kos, I., Manfred Wölfl, M., and Breitenmoser, U. (2013),
8905 “Mapping range dynamics from opportunistic data: spatio-temporal distribu-
8906 tion modeling of lynx *Lynx lynx* L. in the Alps,” *Biological Conservation*, xx,
8907 xxxx–xxxx.
- 8908 Neal, A., White, G., Gill, R., Reed, D., and Olterman, J. (1993), “Evaluation
8909 of mark-resight model assumptions for estimating mountain sheep numbers,”
8910 *Journal of Wildlife Management*, 57, 436–450.
- 8911 Neal, A. K. (1990), “Evaluation of mark-resight population estimates using simula-
8912 tions and field data from mountain sheep.” M.S. thesis, Colorado State University,
8913 Fort Collins, Colorado, USA.
- 8914 Nelder, J. A. and Wedderburn, R. W. M. (1972), “Generalized linear models,”
8915 *Journal of the Royal Statistical Society. Series A (General)*, 135, 370–384.
- 8916 Nichols, J. D., Hines, J. E., Sauer, J. R., Fallon, F. W., Fallon, J. E., and Heglund,
8917 P. J. (2000), “A double-observer approach for estimating detection probability
8918 and abundance from point counts,” *The Auk*, 117, 393–408.
- 8919 Nichols, J. D. and Karanth, K. U. (2002), “Statistical concepts: assessing spatial
8920 distributions,” in *Monitoring tigers and their prey: a manual for researchers,
8921 managers and conservationists in Tropical Asia*, eds. Karanth, K. U. and Nichols,
8922 J. D., Bangalore, India: Centre for Wildlife Studies, pp. 29–38.
- 8923 Niemi, A. and Fernández, C. (2010), “Bayesian Spatial Point Process Modeling
8924 of Line Transect Data,” *Journal of Agricultural, Biological, and Environmental
8925 Statistics*, 15, 327–345.
- 8926 Norris, J. L. and Pollock, K. H. (1996), “Nonparametric MLE under two closed
8927 capture-recapture models with heterogeneity,” *Biometrics*, 52, 639–649.

- 8928 O'Brien, T. (2011), "Abundance, density and relative abundance: A conceptual
8929 framework," in *Camera traps in animal ecology: methods and analyses*, eds.
8930 O'Connell, A. F. J., Nichols, J. D., and Karanth, U., Tokyo, Japan: Springer
8931 Verlag, pp. 71–96.
- 8932 O'Connell, A. F., Nichols, J. D., and Karanth, U. K. (2010), *Camera traps in*
8933 *animal ecology: Methods and analyses*, Springer.
- 8934 O'Hara, R. and Sillanpää, M. (2009), "A review of Bayesian variable selection
8935 methods: what, how and which," *Bayesian Analysis*, 4, 85–118.
- 8936 Otis, D. L., Burnham, K. P., White, G. C., and Anderson, D. R. (1978), "Statistical
8937 inference from capture data on closed animal populations," *Wildlife monographs*,
8938 3–135.
- 8939 Ovaskainen, O. (2004), "Habitat-specific movement parameters estimated using
8940 mark-recapture data and a diffusion model," *Ecology*, 85, 242–257.
- 8941 Ovaskainen, O., Rekola, H., Meyke, E., and Arjas, E. (2008), "Bayesian methods for
8942 analyzing movements in heterogeneous landscapes from mark-recapture data,"
8943 *Ecology*, 89, 542–554.
- 8944 Parmenter, R. R. and MacMahon, J. A. (1989), "Animal Density Estimation Using
8945 a Trapping Web Design: Field Validation Experiments," *Ecology*, 70, 169–179.
- 8946 Parmenter, R. R., Yates, T. L., Anderson, D. R., Burnham, K. P., Dunnum, J. L.,
8947 Franklin, A. B., Friggens, M. T., Lubow, B. C., Miller, M., Olson, G. S., and
8948 Others (2003), "Small-mammal density estimation: A field comparison of grid-
8949 based vs. web-based density estimators," *Ecological Monographs*, 73, 1–26.
- 8950 Pebesma, E. and Bivand, R. (2011), *Package 'sp'*, r package version 0.9-91.
- 8951 Pledger, S. (2004), "Unified maximum likelihood estimates for closed capture-
8952 recapture models using mixtures," *Biometrics*, 56, 434–442.
- 8953 Plummer, M. (2003), "JAGS: A program for analysis of Bayesian graphical models
8954 using Gibbs sampling," in *Proceedings of the 3rd International Workshop on*
8955 *Distributed Statistical Computing (DSC 2003)*. March, pp. 20–22.
- 8956 — (2009), "rjags: Bayesian graphical models using mcmc. R package version 1.0
8957 3-12," .
- 8958 — (2011), "rjags: Bayesian graphical models using mcmc. R package version 3-5,"
8959 .
- 8960 Plummer, M., Best, N., Cowles, K., and Vines, K. (2006), "CODA: Convergence
8961 Diagnosis and Output Analysis for MCMC," *R News*, 6, 7–11.
- 8962 Pollock, K. H. (1982), "A Capture-Recapture Design Robust to Unequal Probability
8963 of Capture." *Journal of Wildlife Management*, 46, 752–757.
- 8964 Pollock, K. H., Nichols, J. D., Brownie, C., and Hines, J. E. (1990), "Statistical
8965 inference for capture-recapture experiments," *Wildlife monographs*, 3–97.
- 8966 Reich, B. J., Gardner, B., and Wilting, A. (2012), "A spatial capture-recapture
8967 model for territorial species," *Biometrics in review*, xx, xx–xxx.
- 8968 Robert, C. P. and Casella, G. (2010), *Introducing Monte Carlo Methods with R*,
8969 New York, USA: Springer.
- 8970 Royle, J., Kéry, M., and Guélat, J. (2011a), "Spatial capture-recapture models for

- 8971 search-encounter data," *Methods in Ecology and Evolution*, 2, 602–611.
- 8972 Royle, J. A. (2004a), "Generalized estimators of avian abundance from count survey
8973 data," *Animal Biodiversity and Conservation*, 27, 375–386.
- 8974 — (2004b), "N-Mixture Models for Estimating Population Size from Spatially
8975 Replicated Counts," *Biometrics*, 60, 108–115.
- 8976 — (2006), "Site occupancy models with heterogeneous detection probabilities,"
8977 *Biometrics*, 62, 97–102.
- 8978 — (2008), "Modeling individual effects in the Cormack–Jolly–Seber model: a state–
8979 space formulation," *Biometrics*, 64, 364–370.
- 8980 — (2009), "Analysis of capture-recapture models with individual covariates using
8981 data augmentation," *Biometrics*, 65, 267–274.
- 8982 Royle, J. A. and Chandler, R. B. (2012), "Integrating Resource Selection Informa-
8983 tion with Spatial Capture-Recapture," *arXiv preprint arXiv:1207.3288*.
- 8984 Royle, J. A., Chandler, R. B., Gazenski, K. D., and Graves, T. A. (2012a), "Spa-
8985 tial capture-recapture for jointly estimating population density and landscape
8986 connectivity," *Ecology*, to appear.
- 8987 Royle, J. A., Chandler, R. B., Sun, C., and Fuller, A. (2012b), "Integrating Resource
8988 Selection Information with Spatial Capture-Recapture," *needs updated MEE*.
- 8989 Royle, J. A. and Converse, S. J. (in review), "Hierarchical spatial capture-recapture
8990 models: Modeling population density from replicated capture-recapture experi-
8991 ments," *Ecology*.
- 8992 Royle, J. A., Converse, S. J., and Link, W. A. (2012c), "Data Augmentation for
8993 Hierarchical Capture-recapture Models," *arXiv preprint arXiv:1211.5706*.
- 8994 Royle, J. A., Dawson, D. K., and Bates, S. (2004), "Modeling abundance effects in
8995 distance sampling," *Ecology*, 85, 1591–1597.
- 8996 Royle, J. A. and Dorazio, R. M. (2006), "Hierarchical models of animal abundance
8997 and occurrence," *Journal of Agricultural, Biological, and Environmental Statis-
8998 tics*, 11, 249–263.
- 8999 — (2008), *Hierarchical modeling and inference in ecology: the analysis of data from
9000 populations, metapopulations and communities*, Academic Press.
- 9001 — (2012), "Parameter-expanded data augmentation for Bayesian analysis of
9002 capture-recapture models," *Journal of Ornithology*, 152, S521–S537.
- 9003 Royle, J. A., Dorazio, R. M., and Link, W. A. (2007), "Analysis of multinomial
9004 models with unknown index using data augmentation," *Journal of Computational
9005 and Graphical Statistics*, 16, 67–85.
- 9006 Royle, J. A. and Dubovský, J. A. (2001), "Modeling spatial variation in waterfowl
9007 band-recovery data," *The Journal of wildlife management*, 65, 726–737.
- 9008 Royle, J. A. and Gardner, B. (2011), "Hierarchical models for estimating density
9009 from trapping arrays," in *Camera traps in animal ecology: methods and analyses*,
9010 eds. O'Connel, A. F. J., Nichols, J. D., and Karanth, U., Tokyo, Japan: Springer
9011 Verlag, pp. 163–190.
- 9012 Royle, J. A., Karanth, K. U., Gopalaswamy, A. M., and Kumar, N. S. (2009a),
9013 "Bayesian inference in camera trapping studies for a class of spatial capture–

- 9014 recapture models," *Ecology*, 90, 3233–3244.
- 9015 Royle, J. A. and Kéry, M. (2007), "A Bayesian state-space formulation of dynamic
9016 occupancy models." *Ecology*, 88, 1813–1823.
- 9017 Royle, J. A. and Link, W. A. (2006), "Generalized site occupancy models allowing
9018 for false positive and false negative errors," *Ecology*, 87, 835–841.
- 9019 Royle, J. A., Magoun, A. J., Gardner, B., Valkenburg, P., and Lowell, R. E. (2011b),
9020 "Density estimation in a wolverine population using spatial capture–recapture
9021 models," *The Journal of Wildlife Management*, 75, 604–611.
- 9022 Royle, J. A. and Nichols, J. D. (2003), "Estimating abundance from repeated
9023 presence-absence data or point counts," *Ecology*, 84, 777–790.
- 9024 Royle, J. A., Nichols, J. D., Karanth, K. U., and Gopalaswamy, A. M. (2009b),
9025 "A hierarchical model for estimating density in camera-trap studies," *Journal of
9026 Applied Ecology*, 46, 118–127.
- 9027 Royle, J. A. and Young, K. V. (2008), "A Hierarchical Model For Spatial Capture–
9028 Recapture Data," *Ecology*, 89, 2281–2289.
- 9029 Russell, R. E., Royle, J. A., Desimone, R., Schwartz, M. K., Edwards, V. L.,
9030 Pilgrim, K. P., and McKelvey, K. S. (2012), "Estimating abundance of mountain
9031 lions from unstructured spatial samples," *Journal of Wildlife Management*.
- 9032 Rutledge, M. (2012), "XXX to be determined XXX," Ph.D. thesis, North Carolina
9033 State University.
- 9034 Salom-Pérez, R., Carrillo, E., Sáenz, J., and Mora, J. (2007), "Critical condition of
9035 the jaguar *Panthera onca* population in Corcovado National Park, Costa Rica,"
9036 *Oryx*, 41, 51.
- 9037 Sanathanan, L. (1972), "Estimating the size of a multinomial population," *The
9038 Annals of Mathematical Statistics*, 43, 142–152.
- 9039 Schofield, M. and Barker, R. (2008), "A unified capture-recapture framework,"
9040 *Journal of agricultural, biological, and environmental statistics*, 13, 458–477.
- 9041 Schwartz, M. K. and Monfort, S. L. (2008), *Genetic and endocrine tools for carni-
9042 vore surveys*, Island Press Washington, DC, USA.
- 9043 Seber, G. A. F. (1982), *The estimation of animal abundance and related parameters*,
9044 Macmillan Publishing Co.
- 9045 Sepúlveda, M. A., Bartheld, J. L., Monsalve, R., Gómez, V., and Medina-Vogel,
9046 G. (2007), "Habitat use and spatial behaviour of the endangered Southern river
9047 otter (*Lontra provocax*) in riparian habitats of Chile: conservation implications,"
9048 *Biological Conservation*, 140, 329–338.
- 9049 Sillett, S., Chandler, R. B., Royle, J. A., Kéry, M., and Morrison, S. (2012), "Hierar-
9050 chical distance sampling models to estimate population size and habitat-specific
9051 abundance of an island endemic," *Ecological Applications*.
- 9052 Simons, T. R., Pollock, K. H., Wetters, J. M., Alldredge, M. W., Pacifici, K., and
9053 Brewster, J. (2009), "Sources of Measurement Error, Misclassification Error, and
9054 Bias in Auditory Avian Point Count Data," in *Modeling Demographic Processes
9055 In Marked Populations*, eds. Thomson, D. L., Cooch, E. G., and Conroy, M. J.,
9056 Boston, MA: Springer US, vol. 3, pp. 237–254.

- 9057 Skalski, J. R., Millspaugh, J. J., and Spencer, R. D. (2005), "Population estimation and biases in paintball, mark-resight surveys of elk," *Journal of Wildlife Management*, 69, 1043–1052.
- 9058
- 9059
- 9060 Sklyar, O., Murdoch, D., Smith, M., Eddelbuettel, D., and François, R. (2010), *inline: Inline C, C++, Fortran function calls from R*, r package version 0.3.8.
- 9061
- 9062 Smith, M. H., Blessing, R., Chelton, J. G., Gentry, J. B., Golley, F. B., and McGinnis, J. T. (1971), "Determining density for small mammal populations using a grid and assessment lines," *Acta Theriologica*, 16, 105–125.
- 9063
- 9064
- 9065 Soisalo, M. K. and Cavalcanti, S. M. C. (2006), "Close-up Space in "Radio-telemetry"," *Biological Conservation*, 129, 487–496.
- 9066
- 9067 Sollmann, R., Furtado, M. M., Gardner, B., Hofer, H., Jacomo, A. T. A., Trzes, N. M., and Silveira, L. (2011), "Improving density estimates for elusive carnivores: Accounting for sex-specific detection and movements using spatial capture-recapture models for jaguars in central Brazil," *Biological Conservation*, 144, 1017–1024.
- 9068
- 9069
- 9070
- 9071
- 9072 Sollmann, R., Gardner, B., Chandler, R. B., Shindle, D., Onorato, D. P., Royle, J. A., and O'Connell, A. F. (in revision), "Using multiple data sources provides density estimates for endangered Florida panther," *Journal of Applied Ecology*.
- 9073
- 9074
- 9075 Sollmann, R., Gardner, B., Parsons, A., Stocking, J., McClintock, B., Simons, T., Pollock, K., and O'Connell, A. (2012), "A spatial mark-resight model augmented with telemetry data," *Ecology*.
- 9076
- 9077
- 9078 Sólymos, P., Lele, S., and Bayne, E. (2012), "Conditional likelihood approach for analyzing single visit abundance survey data in the presence of zero inflation and detection error," *Environmetrics*, 23, 197–205.
- 9079
- 9080
- 9081 Spiegelhalter, D., Best, N., Carlin, B., and Van Der Linde, A. (2002), "Bayesian measures of model complexity and fit," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64, 583–639.
- 9082
- 9083
- 9084 Stabler, B. (2006), *shapefiles: Read and Write ESRI Shapefiles*, r package version 0.6.
- 9085
- 9086 Stanley, T. and Burnham, K. (1999), "A closure test for time-specific capture-recapture data," *Environmental and Ecological Statistics*, 6, 197–209.
- 9087
- 9088 Stoyan, D. and Penttinen, A. (2000), "Recent Applications of Point Process Methods in Forestry Statistics," *Statistical Science*, 15, 61–78.
- 9089
- 9090 Sturtz, S., Ligges, U., and Gelman, A. (2005), "R2WinBUGS: A Package for Running WinBUGS from R," *Journal of Statistical Software*, 12, 1–16.
- 9091
- 9092 Su, Y.-S. and Yajima, M. (2011), *R2jags: A Package for Running jags from R*, r package version 0.02-17.
- 9093
- 9094 Taberlet, P. and Bouvet, J. (1992), "Bear conservation genetics," *Nature*, 358, 197–197.
- 9095
- 9096 Tanner, M. A. and Wong, W. H. (1987), "The calculation of posterior distributions by data augmentation," *J Am Stat Assoc*, 82, 528–540.
- 9097
- 9098 Thomas, A., O'Hara, B., Ligges, U., and Sturtz, S. (2006), "Making BUGS Open," *R News*, 6, 12–17.
- 9099

- 9100 Thomas, M. (1949), "A generalization of Poisson's binomial limit for use in ecology,"
9101 *Biometrika*, 36, 18–25.
- 9102 Thompson, C., Royle, J. A., and Garner, J. (2012), "A framework for inference
9103 about carnivore density from unstructured spatial sampling of scat using detector
9104 dogs," *The Journal of Wildlife Management*, 76, 863–871.
- 9105 Tierney, L., Rossini, A. J., Li, N., and Sevcikova, H. (2011), *snow: Simple Network
9106 of Workstations*, r package version 0.3-7.
- 9107 Tilman, D. and Kareiva, P. (1997), *Spatial ecology: the role of space in population
9108 dynamics and interspecific interactions*, vol. 30, Princeton University Press.
- 9109 Tobler, M. W., Hibert, F., Debeir, L., and Hansen, C. (2012), "Density and sus-
9110 tainable harvest estimates for the lowland tapir in the Amazon of French Guiana
9111 using a spatial capture-recapture model," *none yet, none, none*.
- 9112 Trolle, M. and Kéry, M. (2003), "Estimation of ocelot density in the Pantanal using
9113 capture-recapture analysis of camera-trapping data," *Journal of Mammalogy*, 84,
9114 607–614.
- 9115 —— (2005), "Camera-trap study of ocelot and other secretive mammals in the north-
9116 ern Pantanal," *Mammalia*, 69, 409–416.
- 9117 Tyre, A. J., Tenhumberg, B., Field, S. A., Niejalke, D., Parris, K., and Possing-
9118 ham, H. P. (2003), "Improving precision and reducing bias in biological surveys:
9119 estimating false-negative error rates," *Ecological Applications*, 13, 1790–1801.
- 9120 Valiere, N. and Taberlet, P. (2000), "Urine collected in the field as a source of DNA
9121 for species and individual identification," *Molecular Ecology*, 9, 2150–2152.
- 9122 van Etten, J. (2011), *Package gdistance*, r package version 1.1-2.
- 9123 van Etten, R. J. H. . J. (2012), *raster: Geographic analysis and modeling with raster
9124 data*, r package version 1.9-67.
- 9125 Venables, W. and Ripley, B. (2002), *Modern applied statistics with S*, Springer
9126 verlag.
- 9127 Venables, W., Smith, D., and Team, R. D. C. (2012), "An introduction to R," .
- 9128 Ver Hoef, J. and Boveng, P. (2007), "Quasi-poisson vs. negative binomial regression:
9129 How should we model overdispersed count data?" *Ecology*, 88, 2766–2772.
- 9130 Ver Hoef, J. M. (2012), "Who Invented the Delta Method?" *The American Statis-
9131 tician*, 66, 124–127.
- 9132 Wallace, R. B., Gomez, H., Ayala, G., and Espinoza, F. (2003), "Camera trapping
9133 for jaguar (*Panthera onca*) in the Tuichi Valley, Bolivia," *Journal of Neotropical
9134 Mammalogy*, 10, 133–139.
- 9135 Wegan, M., Curtis, P., Rainbolt, R., and Gardner, B. (2012), "Temporal sampling
9136 frame selection in DNA-based capture mark-recapture investigations." *Ursus*, 23,
9137 42 – 51.
- 9138 Wegan, M. T. (2008), "Aversive conditioning, population estimation, and habitat
9139 preference of black bears (*Ursus americanus*) on Fort Drum Military Installation
9140 in northern New York," Ph.D. thesis, Cornell University, Jan.
- 9141 White, G. (1996), "NOREMARK: population estimation from mark-resighting sur-
9142 veys," *Wildlife Society Bulletin*, 24, 50–52.

- 9143 White, G. and Bennetts, R. (1996), "Analysis of frequency count data using the
9144 negative binomial distribution," *Ecology*, 77, 2549–2557.
- 9145 White, G. and Shenk, M. (2001), "Poplation estimation with radio-marked inividu-
9146 als." in *Radio tracking adn animal populations*, eds. Millspaugh, J. and Marzluff,
9147 J., San Diego, USA: Academic Press, pp. 329–350.
- 9148 White, G. C., Anderson, D. R., Burnham, K. P., and Otis, D. (1982), *Capture-
9149 recapture and removal methods for sampling closed populations*, Los Alamos: Los
9150 Alamos National Laboratory.
- 9151 White, G. C. and Garrot, R. (1990), *Analysis of wildlife radiolocation data*, New
9152 York, USA: Academic Press.
- 9153 White, G. C. and Shenk, T. M. (2000), *Population estimation with radio-marked
9154 animals*, San Diego, California: Academic Press.
- 9155 Whitman, J., Ballard, W., and Gardner, C. (1986), "Home range and habitat use
9156 by wolverines in southcentral Alaska," *The Journal of wildlife management*, 460–
9157 463.
- 9158 Wickham and Hadley (2007), "Reshaping data with the reshape package," *Journal
9159 of Statistical Software*, 21.
- 9160 Williams, B. K., Nichols, J. D., and Conroy, M. J. (2002), *Analysis and management
9161 of animal populations: modeling, estimation, and decision making*, Academic Pr.
- 9162 Wilson, K. R. and Anderson, D. R. (1985a), "Evaluation of a Density Estimator
9163 Based on a Trapping Web and Distance Sampling Theory," *Ecology*, 66, 1185–
9164 1194.
- 9165 — (1985b), "Evaluation of Two Density Estimators of Small Mammal Population
9166 Size," *Journal of Mammalogy*, 66, 13–21.
- 9167 Wolpert, R. L. and Ickstadt, K. (1998), "Poisson/gamma random field models for
9168 spatial statistics," *Biometrika*, 85, 251 –267.
- 9169 Woods, J. G., Paetkau, D., Lewis, D., McLellan, B. N., Proctor, M., and Strobeck,
9170 C. (1999), "Genetic tagging of free-ranging black and brown bears," *Wildlife
9171 Society Bulletin*, 27, 616–627.
- 9172 Wright, J., Barker, R., Schofield, M., Frantz, A., Byrom, A., and Gleeson, D.
9173 (2009), "Incorporating Genotype Uncertainty into Mark–Recapture-Type Models
9174 For Estimating Abundance Using DNA Samples," *Biometrics*, 65, 833–840.
- 9175 Yang, H. C. and Chao, A. (2005), "Modeling Animals' Behavioral Response by
9176 Markov Chain Models for Capture–Recapture Experiments," *Biometrics*, 61,
9177 1010–1017.
- 9178 Yoshizaki, J., Pollock, K. H., Brownie, C., and Webster, R. A. (2009), "Modeling
9179 misidentification errors in capture-recapture studies using photographic identifi-
9180 cation of evolving marks," *Ecology*, 90, 3–9.
- 9181 Zuur, A. F., Ieno, E. N., Walker, N. J., Saveliev, A. A., and Smith, G. M. (2009),
9182 *Mixed effects models and extensions in ecology with R*, Springer Verlag.
- 9183 Zylstra, E., Steidl, R., and Swann, D. (2010), "Evaluating survey methods for mon-
9184 itoring a rare vertebrate, the Sonoran desert tortoise," *The Journal of Wildlife
9185 Management*, 74, 1311–1318.