

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

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

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

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

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

202	9.1.7	Count detector models in the secr package	256
203	9.2	Independent Multinomial Observations	256
204	9.2.1	Multinomial resource selection models	258
205	9.2.2	Simulating data and analysis using JAGS	258
206	9.2.3	Multinomial relationship to the Poisson	262
207	9.2.4	Avian mist-netting example	262
208	9.3	Single-catch traps	269
209	9.3.1	Inference for single-catch systems	270
210	9.3.2	Analysis of Efford's possum trapping data	270
211	9.4	Acoustic sampling	273
212	9.4.1	The signal strength model	274
213	9.4.2	Implementation in secr	276
214	9.4.3	Implementation in BUGS	276
215	9.4.4	Other types of acoustic data	277
216	9.5	Summary and Outlook	277
217	10	Sampling Design	279
218	10.1	General Considerations	279
219	10.1.1	Model-based not design-based	279
220	10.1.2	Sampling space or sampling individuals?	279
221	10.1.3	Scope of inference vs. state-space	279
222	10.2	Study design for (spatial) capture-recapture	279
223	10.3	Trap spacing and array size relative to animal movement	279
224	10.3.1	Example: Black bears from Pictured Rocks National Lakeshore: .	279
225	10.3.2	Final musings: SCR models, trap spacing and array size . . .	280
226	10.4	Spacing of traps with telemetered individuals	280
227	10.5	Sampling over large scales	280
228	10.6	Model-based Spatial Design	280
229	10.6.1	Formalization of the Design Problem for SCR Studies . . .	280
230	10.6.2	An Optimal Design Criterion for SCR	280
231	10.6.3	Optimization of the criterion	280
232	10.6.4	Illustration	280
233	10.7	Covariate models	280
234	10.8	Summary and Outlook	280
235	III	Advanced SCR Models	281
237	11	Modeling Spatial Variation in Density	283
238	11.1	Homogeneous point process revisited	283
239	11.2	Inhomogeneous point processes	283
240	11.3	Observed Point Processes	284

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

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

320	IV Super-Advanced SCR Models	303
321	17 Developing Markov Chain Monte Carlo Samplers	305
322	17.0.1 Why build your own MCMC algorithm?	305
323	17.1 MCMC and posterior distributions	305
324	17.2 Types of MCMC sampling	305
325	17.2.1 Gibbs sampling	305
326	17.2.2 Metropolis-Hastings sampling	306
327	17.2.3 Metropolis-within-Gibbs	306
328	17.2.4 Rejection sampling and slice sampling	306
329	17.3 MCMC for closed capture-recapture Model Mh	306
330	17.4 MCMC algorithm for model SCR0	306
331	17.4.1 SCR model with binomial encounter process	306
332	17.4.2 Looking at model output	306
333	17.4.3 Posterior density plots	306
334	17.4.4 Serial autocorrelation and effective sample size	306
335	17.4.5 Summary results	306
336	17.4.6 Other useful commands	306
337	17.5 Manipulating the state-space	306
338	17.6 Increasing computational speed	306
339	17.6.1 Parallel computing	306
340	17.6.2 Using C++	306
341	17.7 Summary and Outlook	306
342	18 Spatial Capture-Recapture for Unmarked Populations	309
343	18.1 Existing Models for Inference About Density in Unmarked Populations	309
344	18.2 Spatial Correlation as Information	309
345	18.3 Data	309
346	18.4 Model	309
347	18.5 Northern Parula Example	309
348	18.6 Improving Precision with Prior Information	309
349	18.7 Design issues	310
350	18.7.1 How Much Correlation Is Enough?	310
351	18.7.2 Linear Designs	310
352	18.7.3 Quadrat counts	311
353	18.8 Alternative Observation Models	311
354	18.8.1 Spatial point process models	311
355	18.9 Conclusion	312
356	19 Spatial Mark-Resight Models for partially identifiable populations	315
357	19.1 Background	316
358	19.1.1 Types of partial ID data	316

360	19.1.2 A short history of mark-resight models	318
361	19.2 Known number of marked individuals	319
362	19.2.1 MCMC for a spatial mark-resight model	320
363	19.2.2 Binomial encounter model	322
364	19.3 Unknown number of marked individuals	323
365	19.4 Imperfect identification of marked individuals	326
366	19.5 How much information do marked and unmarked individuals con-	
367	tribute?	328
368	19.6 Incorporating telemetry data	331
369	19.7 Summary and Outlook	334
370	20 2012: A Spatial Capture-Recapture Odyssey	337
371	20.1 10 thesis or dissertation topics	338
372	20.2 Three dimesional space	338
373	20.3 Gregarious species	338
374	V Appendices	339
375	20.4 WinBUGS	341
376	20.4.1 WinBUGS through R	341
377	20.5 OpenBUGS	342
378	20.5.1 OpenBUGS through R	342
379	20.6 JAGS	342
380	20.6.1 JAGS through R	343
381	20.7 R	344
382	20.7.1 R packages	344
383	Bibliography	347

Preface

384

385

388

Part I

389

390

Background and Concepts

391
392

393

1

INTRODUCTION

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

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

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

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

1.1 THE STUDY OF POPULATIONS BY CAPTURE-RECAPTURE

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

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

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

LIONS AND TIGERS AND BEARS, OH MY: GENESIS OF SPATIAL CAPTURE-RECAPTURE DATA⁵

460 spatial capture-recapture (SCR) models¹.

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

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

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

486 1.2.1 Camera trapping

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

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

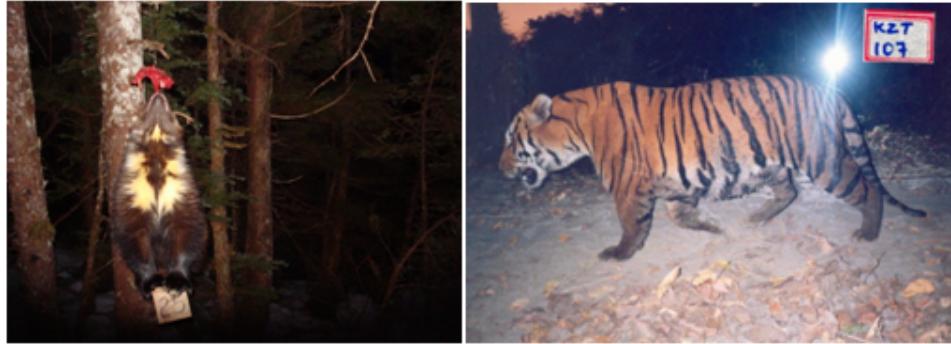


Figure 1.1. Left: Wolverine being encounter by a camera trap (*Photo credit: Audrey Magoun*). Right: Tiger encountered by camera trap (*Photo credit: Ullas Karanth*).

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

508 **1.2.2 DNA sampling**

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



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

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

1.2.3 Acoustic sampling

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

1.2.4 Search-encounter methods

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



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

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

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

1.3 CAPTURE-RECAPTURE FOR MODELING ENCOUNTER PROBABILITY

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

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

1.3.1 Example: Fort Drum bear study

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

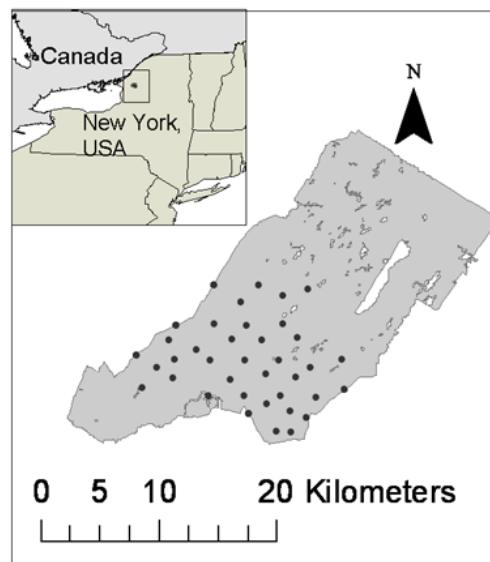


Figure 1.4. Locations of hair snares on Fort Drum, New York, operated during the summer of 2006 to sample black bears.

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

582 The key issue in using such a closed population model regards how we should
583 interpret this estimate of $N = 49.19$ bears. Does it represent the entire population
584 of Fort Drum? Certainly not – the trapping array covers less than half of Fort
585 Drum as we see in Fig. 1.4. So to get at the total bear population size of Fort
586 Drum, we would have to convert our \hat{N} to an estimate of density and extrapolate.
587 To get at density, then, should we assert that N applies to the southern half of
588 Fort Drum below some arbitrary line? Surely bears move on and off of Fort Drum
589 without regard to hypothetical boundaries. Without additional information there
590 is simply no way of converting this estimate of N to density, and hence it is really
591 not meaningful biologically. To resolve this problem, we will adopt the customary
592 approach of converting N to D by buffering the convex hull around the trap array.
593 The convex hull has area 157.135 km². We follow Bales et al. (2005) in buffering
594 the convex hull of the trap array by the radius of the mean female home range size.

595 The mean female home range radius was estimated (Wegan, 2008) for this study
596 region to be 2.19 km, and the area of the convex hull buffered by 2.19 km is
597 277.01 km². (**R** commands to compute the convex hull, buffer it, and compute the
598 area are given in the **R** package **scrbook** which accompanies the book). Hence,
599 the estimated density here is approximately 0.178 bears/km² using the estimated
600 population size obtained by model M_0 . We could assert that the problem has been
601 solved, go home, and have a beer. But then, on the other hand, maybe we should
602 question the use of the estimated home range radius – after all, this is only the
603 female home range radius and the home ranges change for many reasons. Instead,
604 we may decide to rely on a buffer width based on one-half mean maximum distance
605 moved (MMDM) estimated from the actual hair snare data as is more customary
606 (Dice, 1938). In that case the buffer width is 1.19 km, and the resulting estimated
607 density is increased to 0.225 bears/km² about 27 % larger. But wait – some studies
608 actually found the full MMDM (Parmenter et al., 2003) to be a more appropriate
609 measure of movement (e.g. Soisalo and Cavalcanti (2006)). So maybe we should use
610 the full MMDM which is 2.37 km, pretty close to the telemetry-based estimate and
611 therefore providing a similar estimate of density (0.171 bears/km²). So in trying to
612 decide how to buffer our trap array we have already generated 3 density estimates.
613 The crux of the matter is obvious: Although it is intuitive that N should scale with
614 area – the number of bears should go up as area increases and go down as area
615 decreases – in this ad hoc approach of accounting for animal movement N remains

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

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

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

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

650 So what are we left with? Our density estimates span a range from 0.17 to
651 0.43 bears/km² depending on which estimator of N we use and what buffer strip
652 we apply. Should we feel strongly about one or the other? Which buffer should
653 we prefer? AIC favors model M_h , but did it adequately account for the differ-
654 ences in exposure of individuals to the trap array? Are we happy with a purely
655 phenomenological model for heterogeneity? It assumes that all individuals are in-
656 dependent and identically distributed (*iid*) draws from some distribution, but does
657 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-

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

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

1.4 HISTORICAL CONTEXT: A BRIEF SYNOPSIS

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

698 1.4.1 Buffering

699 The standard approach to estimating density even now is to estimate N using
700 conventional closed population models (Otis et al., 1978) and then try to associate
701 with this estimate some specific sampled area, say A , the area which is contributing
702 individuals to the population for which N is being estimated. The strategy is to
703 define A by placing a buffer of say W around the trap array or some polygon which
704 encloses the trap array. The historical context is succinctly stated by (O'Brien,
705 2011) from which we draw this description:

706 “At its most simplistic, A may be described by a concave polygon defined by connect-
707 ing the outermost trap locations (A_{tp} ; Mohr (1947)). This assumes that animals do
708 not move from outside the bounded area to inside the area or vice versa. Unless the
709 study is conducted on a small island or a physical barrier is erected in the study area
710 to limit movement of animals, this assumption is unlikely to be true. More often, a
711 boundary area of width W (A_w) is added to the area defined by the polygon A_{tp} to
712 reflect the area beyond the limit of the traps that potentially is contributing animals
713 to the abundance estimate (Otis et al., 1978). The sampled area, also known as the
714 effective area, is then $A(W) = A_{tp} + A_w$. Calculation of the buffer strip width (W)
715 is critical to the estimation of density and is problematic because there is no agreed
716 upon method of estimating W . Solutions to this problem all involve ad hoc methods
717 that date back to early attempts to estimate abundance and home ranges based on
718 trapping grids (see Hayne, 1949). Dice (1938) first drew attention to this problem
719 in small mammal studies and recommended using one-half the diameter of an aver-
720 age home range. Other solutions have included use of inter-trap distances (Blair,
721 1940; Burt, 1943), mean movements among traps, maximum movements among traps
722 (Holdenried, 1940; Hayne, 1949), nested grids (Otis et al., 1978), and assessment lines
723 (Smith et al., 1971).”

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

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

1.4.2 Temporary emigration

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

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

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

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

1.5 EXTENSION OF CLOSED POPULATION MODELS

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

800 1.5.1 Towards spatial explicitness: Efford’s formulation

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

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

Recently, there has been a flurry of effort devoted to formalizing inference under this model-based framework for the analysis of spatial capture-recapture data (Borchers and Efford, 2008; Royle and Gardner, 2011; Borchers, 2012; Gopalanwamy, 2012). There are two distinct lines of work which adopt the model-based formulation in terms of the underlying point process but differ primarily by the manner in which inference is achieved. One approach (Borchers and Efford, 2008) uses classical inference based on likelihood (see Chapt. 6), and the other (Royle and Young, 2008) adopts a Bayesian framework for inference (Chapts. 5 and 17).

1.5.2 Abundance as the aggregation of a point process

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

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

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

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

853 **1.5.3 The activity center concept**

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

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

876 **1.5.4 The state-space**

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

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

892 **1.5.5 Abundance and density**

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

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

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

1.6 CHARACTERIZATION OF SCR MODELS

908 Formulation of capture-recapture models conditional on the latent point process is
 909 the critical and unifying element of *all* SCR models. However, SCR models differ
 910 in how the underlying process model is formulated, and its complexity. Most of the
 911 development and application of SCR models has focused on their use to estimate
 912 density and touting the fact that they resolve certain specific technical problems.
 913 Related to the use of ordinary capture-recapture models. This is achieved with a sim-
 914 ple process model being a basic point process of independently distributed points.
 915 At the same time, there are models of CR data that focus exclusively on *movement*
 916 modeling, or models with explicit dynamics (Ovaskainen, 2004; Ovaskainen et al.,
 917 2008). Conceptually, these are akin to spatial versions of so-called Cormack-Jolly-
 918 Seber (CJS) models in the traditional capture-recapture literature, except they
 919 involve explicit mathematical models of movement based on diffusion or Brownian
 920 motion. Finally, there are now a very small number of papers that focus on *both*
 921 movement and density simultaneously (Royle and Young, 2008; Royle et al., 2011a;

922 Royle and Chandler, 2012) or population dynamics and density (Gardner et al.,
923 2010b).

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

1.7 SUMMARY AND OUTLOOK

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

947 As we have shown already, SCR models are not simply an extension of a tech-
948 nique to resolve certain technical problems. Rather, they provide a coherent, flex-
949 ible framework for making ecological processes explicit in models of individual en-
950 counter history data, and for studying animal populations processes such as individ-
951 ual movement, resource selection, space usage, population dynamics, and density.
952 Historically, researchers studied these questions independently, using ostensibly un-
953 related study designs and statistical procedures. For example, resource selection
954 function (RSF) models for resource selection, state-space models for movement,
955 density using closed capture-recapture methods, and population dynamics with
956 various “open” capture-recapture models. SCR can bring all of these problems
957 together into a single unified framework for modeling and inference. Most impor-
958 tantly, spatial capture-recapture models promise the ability to integrate explicit
959 ecological theories directly into the models so that we can directly test hypoth-
960 eses about either space usage (e.g., Chapt. 13), landscape connectivity (Chapt.
961 12), movement, or spatial distribution (Chapt. 11). We imagine that, in the near

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

967
968

2

969

STATISTICAL MODELS AND SCR

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

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

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

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

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

2.1 RANDOM VARIABLES AND PROBABILITY DISTRIBUTIONS

1008 2.1.1 Stochasticity in ecology

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

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

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

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

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

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

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

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

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

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

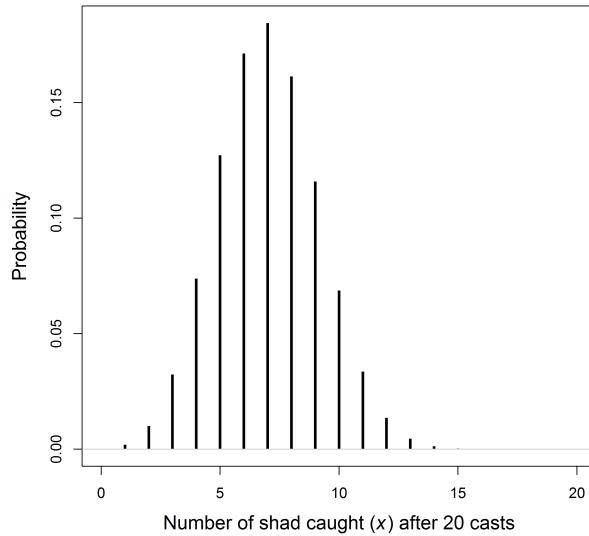


Figure 2.1. The binomial probability mass function with $N = 20$ and $p = 0.35$.

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

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

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

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

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

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

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	$x \sim \text{Gamma}(a, b)$	$\frac{b^a}{\Gamma(a)} x^{a-1} \exp(-bx)$	$x \in [0, \infty]$	a/b	a/b^2
Multivariate Normal	$\mathbf{x} \sim \text{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	$(2\pi)^{-k/2} \boldsymbol{\Sigma} ^{-1/2} \exp(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}))$	$x_k \in [-\infty, \infty]$	$\boldsymbol{\mu}$	$\boldsymbol{\Sigma}$

1089 **2.1.2 Properties of probability distributions**

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

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

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

1106 Another feature of probability distributions is that they can be used to compute
 1107 important summaries of random variables. The two most important summaries
 1108 are the expected value, $\mathbb{E}(x)$, and the variance $\text{Var}(x)$. The expected value, or
 1109 mean, can be thought of as the average of a very large sample from the specified
 1110 distribution. For example, one way of approximating the expected values of a
 1111 binomial distribution with $K = 20$ trials and $p = 0.35$ can be implemented in
 1112 **R** using:

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

1115 For most probability distributions used in this book, the expected values are known
 1116 exactly, as shown in Table 2.1, and thus we don't need to resort to such Monte Carlo
 1117 approximations. For instance, the expected value of the binomial distribution is
 1118 exactly $\mathbb{E}(x) = Kp = 20 \times 0.35 = 7$. In this case, it happens to take an integer
 1119 value, but this is not a necessary condition, even for discrete random variables.

1120 A more formal definition of an expected value is the average of all possible
 1121 values of the random variable, weighted by their probabilities. For continuous
 1122 random variables, this weighted average is found by integration:

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

1123 For example, if $[x]$ is normally distributed with mean 3 and unit variance, we could
 1124 find the expected value using the following code.

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

1127 Of course, the mean *is* the expected value of the normal distribution, so we didn't
 1128 need to compute the integral but, the point is, that Eq. 2.1.3 is generic. For
 1129 discrete random variables, the expected value is found by summation rather than
 1130 integration:

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

1131 where the summation is over all possible values of x . Earlier we approximated the
 1132 expected value of the binomial distribution with $K = 20$ trials and $p = 0.35$ by
 1133 taking a Monte Carlo average. Eq. 2.1.4 let's us find the exact answer, using this
 1134 bit of R code:

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

1137 This is great. But of what use is it? One very important concept to understand is
 1138 that when we fit models, we are often modeling changes in the expected value of
 1139 some random variable. For example, in Poisson regression, we model the expected
 1140 value of the random variable, which may be a function of environmental variables.

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

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

2.2 COMMON PROBABILITY DISTRIBUTIONS

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

1169 **2.2.1 The binomial distribution**

1170 The binomial distribution plays a critical role in ecology. It is used for purposes
 1171 as diverse as modeling count data, survival probability, occurrence probability, and
 1172 capture probability, just to name a few. To describe the properties of the binomial
 1173 distribution, and related distributions, we will introduce a new example. Suppose
 1174 we are conducting a bird survey at a site in which $N = 10$ chestnut-sided warblers
 1175 (*Setophaga pensylvanica*) occur, and each of these individuals has a detection prob-
 1176 ability of $p = 0.5$. The binomial distribution is the natural choice for describing
 1177 the number of individuals that we would expect to detect (n) in this situation, and
 1178 using our notation, we can write the model as: $n \sim \text{Bin}(10, 0.5)$. When $p < 1$,
 1179 we can expect that we will observe a different number of warblers on each of K
 1180 replicate survey occasions. To see this, we simulate data under this simple model
 1181 with $K = 3$.

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

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

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

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

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

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

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

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

1212 2.2.2 The Bernoulli distribution

1213 Above, we showed 3 alternatives to `dbinom` for evaluating the binomial pmf. These
 1214 three commands differed only in how they computed the binomial coefficient, which
 1215 we needed because of the numerous ways in which we could observe $n = 5$ given
 1216 $N = 10$. To conceptualize this, let y_i be a binary variable indicating if individual i
 1217 was detected or not. Hence, given that 5 individuals were detected, the vector of
 1218 individual detections could be something like $\mathbf{y} = (0, 0, 1, 1, 1, 1, 0, 0, 0)$, indicating
 1219 that we detected individuals 3-7 but not 1-2 or 8-10. For $N = 10$ and $n = 5$,
 1220 the binomial coefficient tells us that there are 252 possible vectors \mathbf{y} with 5 ones.
 1221 However, when $N \equiv 1$, this term drops from the pmf and the result is the pmf for
 1222 the Bernoulli distribution. That is, the Bernoulli distribution is simply the binomial
 1223 distribution when $N \equiv 1$. Alternatively, we could say that the binomial distribution
 1224 is the outcome of N iid Bernoulli trials. We use the standard abbreviation “iid”
 1225 to mean *independent, identically distributed*.

1226 The utility of the Bernoulli distribution is evident when we imagine that not all
 1227 of the chestnut-sided warblers have the same detection probability. Thus, if some
 1228 individuals can be detected with probability 0.3 and others have a 0.7 detection
 1229 probability, then the model $n \sim \text{Bin}(N, p)$ is no longer an accurate description of
 1230 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

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

1231 This states that individual i is detected with probability p_i , and the observed count
 1232 is the sum of the N Bernoulli outcomes.

1233 An important point is that the individual-specific data y_{ik} can only be observed
 1234 if the individuals are uniquely distinguishable, such as when they are marked by
 1235 biologists with color bands. In such cases, the Bernoulli distribution allows us
 1236 to model variation in detection probability among individuals and thus would be
 1237 preferable to the binomial distribution, which assumes that each of the N indi-
 1238 viduals have the same p . For this reason, the Bernoulli distribution, as simple as
 1239 it is, is of paramount importance in capture-recapture models, including spatial
 1240 capture-recapture models in which there is virtually always substantial and impor-
 1241 tant variation in capture probability among individuals. Indeed, it could be said
 1242 that the Bernoulli model is the canonical model in capture-recapture studies, and
 1243 most of the different flavors of capture-recapture models differ primarily in how p_i
 1244 is specified.

1245 The Bernoulli pmf is given by $p^n(1-p)^{1-n}$ and hence we do not need canned
 1246 functions to facilitate its evaluation. Of course, if you wanted to, you could always
 1247 use `dbinom` with the `size` argument set to 1. For example, `dbinom(1, 1, 0.3)`
 1248 returns the Bernoulli probability of observing $n = 1$ given $p = 0.3$.

1249 2.2.3 The multinomial and categorical distributions

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

1261 The multinomial distribution is widely used to model data from traditional,
 1262 non-spatial capture-recapture studies. Earlier we let y_{ik} denote a binary random
 1263 variable indicating if warbler i was detected on survey k . The vector of observations
 1264 for an individual, \mathbf{y}_i , is often referred to as the individual's "encounter history".

1265 The number of possible encounter histories depends on K , the number of survey
 1266 occasions. Specifically, there are 2^K possible encounter histories¹. If we tabulate the
 1267 number of individuals with each encounter history, the frequencies can be modeled
 1268 using the multinomial distribution.

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

```
1282 > caphist.probs <- c("11"=0.09, "10"=0.21, "01"=0.21, "00"=0.49)
1283 > drop(rmultinom(1, 10, caphist.probs))
1284 11 10 01 00
1285 0 3 2 5
```

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

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

1299 Just as the Bernoulli distribution is the elemental form of the binomial distri-
 1300 bution (being the case $N = 1$), the categorical distribution is essentially equivalent
 1301 to the multinomial distribution with size parameter $N \equiv 1$. The only difference is
 1302 that, rather than returning a vector with a single element equal to 1, it returns the
 1303 element *location* where the 1 occurs. For example, if $\mathbf{y} = (0, 0, 1, 0)$ is an outcome

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 $2^K - 1$

1304 of a multinomial distribution with $N = 1$, then the categorical outcome would be
 1305 3 because the 1 is located in third position in the vector. Thus, in spatial capture-
 1306 recapture models, we might use either the multinomial distribution with $N = 1$
 1307 or the categorical distribution. The various **BUGS** engines describe the categor-
 1308 ical distribution by the declaration `dcat` and, in **R**, we can simulate categorical
 1309 outcomes using the function `sample` or as so:

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

1312 2.2.4 The Poisson distribution

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

1321 The Poisson distribution is related to both the binomial and multinomial dis-
 1322 tributions, and the following three bits of trivia are occasionally worth knowing.
 1323 First, it is the limit of the binomial distribution as $N \rightarrow \infty$ and $p \rightarrow 0$, which
 1324 means that for high values of N and low values of p , $\text{Pois}(N \times p)$ is approximately
 1325 equal to $\text{Bin}(N, p)$. Second, if $\{n_1 \sim \text{Pois}(\lambda_1), \dots, n_K \sim \text{Pois}(\lambda_K)\}$ then the vector
 1326 of counts is multinomial, $\{n_1, \dots, n_K\} \sim \text{Multinom}(\sum_k n_k, \{\frac{\lambda_1}{\sum_k \lambda_k}, \dots, \frac{\lambda_K}{\sum_k \lambda_k}\})$.
 1327 Third, the sum of two Poisson random variables $x_1 \sim \text{Pois}(\lambda_1)$ and $x_2 \sim \text{Pois}(\lambda_2)$
 1328 is also Poisson: $x_1 + x_2 \sim \text{Pois}(\lambda_1 + \lambda_2)$.

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

1340 2.2.5 The uniform distribution

1341 The lowly uniform distribution is a continuous distribution whose only two pa-
 1342 rameters are the lower and upper bounds that restrict the possible values of the

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

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

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

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

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

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

1369 2.2.6 Other distributions

1370 The other continuous distributions that are regularly encountered in SCR models
 1371 are primarily used as priors in Bayesian analyses, and thus we will avoid a lengthy
 1372 discussion of their properties. The normal distribution, also called the Gaussian
 1373 distribution, is perhaps the most widely recognized and applied probability model
 1374 in statistics, but it plays only a minor role in SCR models other than as a model for
 1375 signal strength in acoustic SCR models (Efford et al., 2009b; Dawson and Efford,
 1376 2009), and see Sec. 9.4. Nonetheless, it is the canonical prior for any continuous
 1377 random variable with infinite support, and thus it is often used as a prior when
 1378 applying Bayesian methods. One common usage is as a prior for the β coefficients
 1379 of a linear model defining some parameter as a function of covariates (usually on

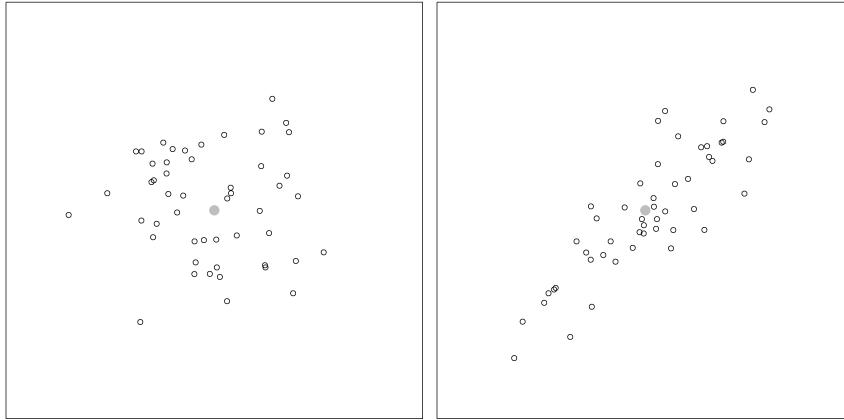


Figure 2.2. Two realized point patterns from the bivariate normal distribution.

1380 a transformed scale). An example, including a cautionary note, is provided in
 1381 Sec. 3.5.1. Be aware that although the normal distribution is typically parameter-
 1382 ized in terms of the variance parameter σ^2 , in the **BUGS** language, the inverse of
 1383 the variance, or precision, is used instead, $\tau = 1/\sigma^2$. In **R**, the `dnorm` function
 1384 requires the standard deviation σ , rather than the variance σ^2 .

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

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

1401 Fig. 2.2 shows the simulated points.

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

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

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

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

2.3 STATISTICAL INFERENCE AND PARAMETER ESTIMATION

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

1431 Suppose we count oak trees at J sites, and the resulting data $\{y_1, \dots, y_J\}$ can
 1432 be assumed to be *iid* outcomes from some distribution, such as the Poisson with
 1433 unknown parameter λ . We want to estimate this parameter. In classical inference,
 1434 the only uncertainty about λ is that attributable to sampling. For instance, we can
 1435 imagine repeatedly sampling the population (sites in this example) and obtaining
 1436 sample-specific estimates of λ . Typically, we entertain the idea that there are an
 1437 infinite number of possible samples and so we could obtain an infinite number of
 1438 estimates: $\{\hat{\lambda}_1, \hat{\lambda}_2, \dots, \hat{\lambda}_\infty\}$. If these estimates are produced using the method
 1439 of maximum likelihood, and as n tends to infinity, the distribution of estimates,
 1440 called the sampling distribution, will be normally distributed with $E(\hat{\lambda}) = \lambda$. The
 1441 standard deviation of the sampling distribution is called the standard error, which
 1442 can also be estimated as part of the maximum likelihood procedure. Of course, we

1443 almost always have just a single sample of data, and hence a single $\hat{\lambda}$ and a single
 1444 estimate of the standard error. However, under the assumption of a normally
 1445 distributed sampling distribution, we can construct a confidence interval that will
 1446 include the true value of λ with coverage probability $1 - \alpha$, where α is a prescribed
 1447 value like 0.05. An important point is that there is no uncertainty associated with
 1448 the actual parameter—it is regarded as a fixed value, and hence probability is only
 1449 used to characterize the estimator via its sampling distribution.

1450 Maximum likelihood is heuristically a method of finding the most “likely” value
 1451 of λ , given the observed data, and of characterizing the variance of the sampling dis-
 1452 tribution. Of course, it also applies to cases where the observations are multivariate,
 1453 or the probability distribution is a function of multiple parameters. Endless num-
 1454 bers of textbooks and online resources are available for those interested in a detailed
 1455 explanation of maximum likelihood. For our purposes, we wish to keep it simple
 1456 and focus on *how* to do it. The first step is to define the likelihood function, which
 1457 is the joint distribution of the data regarded as a function of the parameter(s). If
 1458 the joint distribution of the observations is denoted by $[y_1, y_2, \dots, y_n | \lambda]$, we usually
 1459 denote the likelihood by flipping the arguments: $\mathcal{L}(\lambda | \mathbf{y}) = [\lambda | y_1, y_2, \dots, y_n]$.

1460 If the observations are *iid*, the likelihood simplifies to

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

1461 where $[y_i | \lambda]$ is a probability distribution, like those discussed in the previous sec-
 1462 tions. For example, if y_i is Poisson distributed, then $[y_i | \lambda] = \text{Poisson}(\lambda) = \frac{\lambda^{y_i} e^{-\lambda}}{y_i!}$.
 1463 Although likelihoods are typically shown on the natural scale, we almost always
 1464 maximize the logarithm of the likelihood to avoid computational problems that
 1465 arise when multiplying very small probabilities. Thus, we rewrite Eq. 2.3.1 as

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

1466 Here is some simple **R** code to simulate independent Poisson outcomes and esti-
 1467 mate λ (as though we did not know it) using the method of maximum likelihood.
 1468 Actually, we will minimize the negative log-likelihood because it is equivalent and
 1469 is the default for **R**’s optimizers like `optim` and `nls`.

```
1470 > lambda <- 3                      # Actual parameter value
1471 > y1 <- rpois(100, lambda)        # Realized values (data)
1472 > negLogLike1 <- function(par) -sum(dpois(y1, par, log=TRUE))
1473 > starting.value <- c('lambda'=1)
1474 > optim(starting.value, negLogLike1)$par # MLE
1475   lambda
1476 3.039844
```

1477 Explicitly maximizing the likelihood, numerically, isn't actually necessary here because
 1478 the MLE of λ is given by the mean of the observations. A more interesting
 1479 example is when there are covariates of λ . For example, suppose λ is a function of
 1480 elevation and vegetation height according to: $\log(\lambda_i) = \beta_0 + \beta_1 ELEV_i + \beta_2 VEGHT_i$.
 1481 This is a standard Poisson regression problem, with likelihood:

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

1482 This likelihood is almost identical to the previous one except that λ is now a
 1483 function, and so we need to estimate the parameters of the function, i.e. the β 's.
 1484 Some code to fit this model to simulated data is shown here:

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

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

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

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

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

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

2.4 JOINT, MARGINAL, AND CONDITIONAL DISTRIBUTIONS

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

```
1538 > X <- 0:20 # All possible values of X
1539 > Y <- 0:10 # All possible values of Y
1540 > lambda <- 0.6
```

```

1541 > p <- plogis(-0.62 + -2*Y) # p as function of Y
1542 > round(p,2)
1543 [1] 0.35 0.07 0.01 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
1544 > joint <- matrix(NA, length(X), length(Y))
1545 > rownames(joint) <- paste("X=", X, sep="")
1546 > colnames(joint) <- paste("Y=", Y, sep="")
1547 >
1548 > # Joint distribution [X,Y]
1549 > for(i in 1:length(Y)) {
1550 +   joint[,i] <- dbinom(X, 20, p[i]) * dpois(Y[i], lambda)
1551 +
1552 > round(joint,2)
1553   Y=0  Y=1  Y=2  Y=3  Y=4  Y=5  Y=6  Y=7  Y=8  Y=9  Y=10
1554 X=0  0.00 0.08 0.08 0.02  0  0  0  0  0  0  0
1555 X=1  0.00 0.12 0.02 0.00  0  0  0  0  0  0  0
1556 X=2  0.01 0.08 0.00 0.00  0  0  0  0  0  0  0
1557 X=3  0.02 0.04 0.00 0.00  0  0  0  0  0  0  0
1558 X=4  0.04 0.01 0.00 0.00  0  0  0  0  0  0  0
1559 X=5  0.07 0.00 0.00 0.00  0  0  0  0  0  0  0
1560 X=6  0.09 0.00 0.00 0.00  0  0  0  0  0  0  0
1561 X=7  0.10 0.00 0.00 0.00  0  0  0  0  0  0  0
1562 X=8  0.09 0.00 0.00 0.00  0  0  0  0  0  0  0
1563 X=9  0.06 0.00 0.00 0.00  0  0  0  0  0  0  0
1564 X=10 0.04 0.00 0.00 0.00  0  0  0  0  0  0  0
1565 X=11 0.02 0.00 0.00 0.00  0  0  0  0  0  0  0
1566 X=12 0.01 0.00 0.00 0.00  0  0  0  0  0  0  0
1567 X=13 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
1568 X=14 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
1569 X=15 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
1570 X=16 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
1571 X=17 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
1572 X=18 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
1573 X=19 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
1574 X=20 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0

```

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

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

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

for discrete random variables, and

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

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

```
1584 > margX <- rowSums(joint)
1585 > round(margX, 2)
1586   X=0  X=1  X=2  X=3  X=4  X=5  X=6  X=7  X=8  X=9  X=10  X=11  X=12  X=13  X=14
1587 0.18 0.14 0.09 0.05 0.05 0.07 0.09 0.10 0.09 0.06 0.04 0.02 0.01 0.00 0.00
1588 X=15  X=16  X=17  X=18  X=19  X=20
1589 0.00 0.00 0.00 0.00 0.00 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 .

```
1592 > XgivenY <- joint/matrix(margY, nrow(joint), ncol(joint), byrow=TRUE)
1593 > round(XgivenY, 2)
1594   Y=0  Y=1  Y=2  Y=3  Y=4  Y=5  Y=6  Y=7  Y=8  Y=9  Y=10
1595   X=0  0.00 0.25 0.82 0.97  1  1  1  1  1  1  1
1596   X=1  0.00 0.36 0.16 0.03  0  0  0  0  0  0  0
1597   X=2  0.01 0.25 0.02 0.00  0  0  0  0  0  0  0
1598   X=3  0.03 0.11 0.00 0.00  0  0  0  0  0  0  0
1599   X=4  0.07 0.03 0.00 0.00  0  0  0  0  0  0  0
1600   X=5  0.13 0.01 0.00 0.00  0  0  0  0  0  0  0
1601   X=6  0.17 0.00 0.00 0.00  0  0  0  0  0  0  0
1602   X=7  0.18 0.00 0.00 0.00  0  0  0  0  0  0  0
```

1605	X=8	0.16	0.00	0.00	0.00	0	0	0	0	0	0	0
1606	X=9	0.12	0.00	0.00	0.00	0	0	0	0	0	0	0
1607	X=10	0.07	0.00	0.00	0.00	0	0	0	0	0	0	0
1608	X=11	0.03	0.00	0.00	0.00	0	0	0	0	0	0	0
1609	X=12	0.01	0.00	0.00	0.00	0	0	0	0	0	0	0
1610	X=13	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
1611	X=14	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
1612	X=15	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
1613	X=16	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
1614	X=17	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
1615	X=18	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
1616	X=19	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
1617	X=20	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0

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

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

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

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

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

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

2.5 HIERARCHICAL MODELS AND INFERENCE

1631 The term hierarchical modeling (or hierarchical model) has become something of
 1632 a buzzword over the last decade with hundreds of papers published in ecological
 1633 journals using that term. So then, what exactly is a hierarchical model, anyhow?
 1634 Obviously, this term stems from the root “hierarchy” which means:

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

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

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

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

1643

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

1644 where y_i = observation of presence/absence at a site i and z_i = occurrence status
 1645 ($z_i = 1$ if a species occurs at site i and $z_i = 0$ if not). Note that if $p = 1$, then we
 1646 would perfectly observe z and the model would no longer be hierarchical—it would
 1647 be a simple logistic regression model. Note also that this hierarchical model has an
 1648 important conceptual distinction between other types of classical multi-level models
 1649 such as repeated measures on subjects, in that z_i is an actual state of nature. In
 1650 that sense, z is a random variable that is the outcome of a “real” process. Royle
 1651 and Dorazio (2008) used the term *explicit* hierarchical model to describe this type of
 1652 model to distinguish from hierarchical models (*implicit* hierarchical models) where
 1653 the latent variables don’t correspond to an actual state of nature—but rather just
 1654 soak up variation that is unmodeled by explicit elements of the model. At best,
 1655 latent variables in such models are surrogates for something of ecological relevance
 1656 (“time effects”, “space effects” etc.).

1657 With these examples, we expand on our definition of a hierarchical model as we
 1658 will use it in this book:

1659 **Definition: Hierarchical Model:** A model with explicit component models that de-
 1660 scribe variation in the data due to (spatial/temporal) variation in *ecological process*,
 1661 and due to *imperfect observation* of the process.

1662 Most models considered in this book describe the encounter of individuals con-
 1663 ditional on the “activity center” of the individual, which is a latent variable (i.e.,
 1664 unobserved random effect). The definition of an activity center will be context-
 1665 dependent as discussed in Chapt. 5, but often it can be thought of as an individual’s
 1666 home range center. The collection of these latent variables represents the outcome
 1667 of an ecological process describing how individuals distribute themselves over the
 1668 landscape. Moreover, how individuals are encountered in traps is, in some cases,
 1669 the result of a model governing movement. As such, these models are examples of
 1670 hierarchical models that contain formal model components representing both eco-
 1671 logical process and also the observation of that process. That is, they are explicit
 1672 hierarchical models (Royle and Dorazio, 2008) as opposed to implicit hierarchical
 1673 models.

2.6 CHARACTERIZATION OF SCR MODELS

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

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

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

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

1702 Let’s begin with the model $[\mathbf{s}]$ that describes the distribution of the activity
 1703 centers of each animal in the spatial region \mathcal{S} (the state-space as we called it previ-
 1704 ously). As will be explained in Chapt. 5 and Chapt. 11, $[\mathbf{s}]$ defines a spatial point
 1705 process, which may be inhomogeneous if there exists spatial variation in density, or
 1706 it may be homogeneous if density is constant throughout \mathcal{S} . In the later case, we can
 1707 write $[\mathbf{s}] = \text{Uniform}(\mathcal{S})$, which is to say that the N activity centers are uniformly
 1708 distributed in the polygon \mathcal{S} . A point process is also a model for the number of indi-
 1709 viduals in the population N . So we could write $[\mathbf{s}|\mu]$ where μ is an intensity param-
 1710 eter defined as the number of points per unit area. In other words, μ is population
 1711 density, and we often model population size as either $N \sim \text{Poisson}(\mu A(\mathcal{S}))$, where
 1712 $A(\mathcal{S})$ is the area of the state-space; or, $N \sim \text{Binomial}(M, \psi)$ where $\psi = \mu A(\mathcal{S})/M$

1713 and M is some large integer used simply as a convenience measure when conducting
 1714 Bayesian analysis. As it turns out, there is very little practical difference in the
 1715 Poisson prior versus a binomial models for N (Chapt. 11).

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

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

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

1746 Eq. 2.6.1 is a compact description of the the basic components of a SCR model,
 but it is also rather vague. The previous four paragraphs added enough extra detail
 so that we can now describe a specific SCR model. Perhaps the simplest SCR model
 is this:

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

1746 These “assumptions” are statistical statements of three basic hypotheses that (1)

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

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

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

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

1779 Having briefly explained each of the model components in Eq. 2.6.1, and having
 1780 shown how a subset of these components results in a basic SCR model, we can
 1781 now discuss other relevant arrangements. Examples include: (1) Classical distance
 1782 sampling (Buckland et al., 2001; Borchers et al., 2002), (2) Spatial capture-recapture
 1783 models with fixed arrays of traps (Efford, 2004; Borchers and Efford, 2008; Royle
 1784 et al., 2009a,b; Gardner et al., 2010a; Royle et al., 2011b), and (3) Search-encounter
 1785 models (Royle and Young, 2008; Royle et al., 2011a). We will now elaborate on
 1786 some of these distinctions.

1787 1. **Distance sampling.** The last 2 stages of the hierarchy are confounded
 1788 (implicitly) and so analysis is based on the model $[y|\mathbf{u}][\mathbf{u}]$. The “process
 1789 model” is that of “uniformity”: $\mathbf{u} \sim \text{Uniform}(\mathcal{S})$.

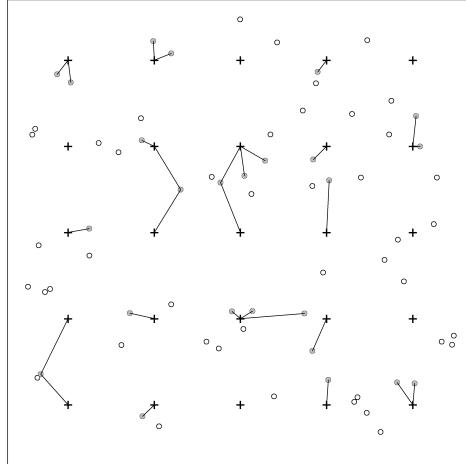


Figure 2.3. Population of $N = 69$ home-range centers (s , circles) and 25 trap locations (x , crosses). Lines connect activity centers to the traps where the individuals were detected. As in many SCR models, movement outcomes (u) are ignored.

1790 2. **Spatial capture-recapture model with a fixed array of traps.** SCR
 1791 models appear to have little in common with distance sampling because ob-
 1792 servations are made only at a pre-defined set of discrete locations—where
 1793 traps are placed. However, the models are closely related in terms of our
 1794 hierarchical representation above. In SCR models based on fixed arrays, we
 1795 cannot estimate both $\Pr(y = 1|u)$ and $\Pr(u|s)$ —the probability that an in-
 1796 dividual “moves to u ” cannot be separated from the probability that it is
 1797 detected given that it moves to u , because of the fact that the observation
 1798 locations are fixed by design. Formally, such SCR models confound $[y|u]$
 1799 with $[u|s]$ so that the observation model arises as:

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

1800 This confounding happens because SCR sampling is spatially biased—restricted
 1801 to a fixed pre-determined set of locations. Conversely, distance sampling
 1802 confounds $[u|s][s]$ because, essentially, there is only a single realization of the
 1803 encounter process. It is probably reasonable to assume that $\Pr(y = 1|u) = 1$
 1804 or at least it is locally constant for most devices (e.g., cameras, etc..), and
 1805 thus the detection model will have the interpretation in terms of movement
 1806 (see Chapt. 13 and 12).

1807 3. **Search-encounter models.** What we call “search-encounter” models (Royle

1808 and Young, 2008; Royle et al., 2011a) are kind of a hybrid model combining
1809 features of SCR models and features of distance sampling. Like distance
1810 sampling they allow for encounters in continuous space which provide di-
1811 rect observations from $[\mathbf{u}|\mathbf{s}]$. Thus, the hierarchical model is fully identified.
1812 These models are described in Chapt. chapt.search-encounter.

2.7 SUMMARY AND OUTLOOK

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

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

1833
1834

1835

3

GLMS AND BAYESIAN ANALYSIS

1836 A major theme of this book is that spatial capture-recapture models are, for the
1837 most part, just generalized linear models (GLMs) wherein the covariate, distance
1838 between trap and home range center, is partially or fully unobserved – and therefore
1839 regarded as a random effect. Outside of capture-recapture, such models are usually
1840 referred to as generalized linear mixed models (GLMMs) and, therefore, SCR mod-
1841 els can be thought of as a specialized type of GLMM. Naturally then, we should
1842 consider analysis of these slightly simpler models in order to gain some experience
1843 and, hopefully, develop a better understanding of spatial capture-recapture models.

1844 In this chapter, we consider classes of GL(M)Ms – Poisson and binomial (i.e.,
1845 logistic regression) models – that will prove to be enormously useful in the analysis
1846 of capture-recapture models of all kinds. Many readers are likely familiar with these
1847 models already because they are among the most useful models in ecology and,
1848 as such, have received considerable attention in many introductory and advanced
1849 texts. We focus on them here in order to introduce the readers to the analysis of
1850 such models in **R** and **WinBUGS** or **JAGS**, which we will translate directly to
1851 the analysis of SCR models in subsequent chapters.

1852 Bayesian analysis is convenient for analyzing GL(M)Ms because it allows us to
1853 work directly with the conditional model – i.e., the model that is conditional on the
1854 random effects, using computational methods known as Markov chain Monte Carlo
1855 (MCMC). Learning how to do Bayesian analysis of GLMs and GLMMs using the
1856 **BUGS** language is, in part, the purpose of this chapter. We focus here on the use of
1857 **WinBUGS** because it is the most popular “**BUGS** engine”. However, later in the
1858 book we transition to another popular **BUGS** engine known as **JAGS** (Plummer,
1859 2009) which stands for *Just Another Gibbs Sampler*. For most of our purposes, the
1860 specification of models in either platform is the same, but **JAGS** is under active
1861 development at the present time while **WinBUGS** no longer is, having transitioned

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

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

3.1 GLMS AND GLMMS

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

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

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

1901 A key aspect of GLMs is that $g(\mathbb{E}(y))$ is assumed to be a linear function of the
 1902 predictor variable(s), here x , with unknown parameters, here β_0 and β_1 , to be
 1903 estimated. In standard GLMs, the variance of y is a function V of the mean of y :
 1904 $\text{Var}(y) = V(\mu)$ (see below for examples). As an example, a Poisson GLM posits
 1905 that $y \sim \text{Poisson}(\lambda)$ with $\mathbb{E}(y) = \lambda$ and usually the model for the mean is specified
 1906 using the *log link function* by

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

1907 The variance function is $V(y_i) = \lambda_i$. To see how a Poisson GLM works, use the **R**
 1908 code below to simulate some data and then estimate the parameters:

```
1909 > set.seed(13)
1910 > n <- 100          # set sample size
1911 > beta0 <- -2       # set intercept term
1912 > beta1 <- 1.5      # set coefficient
1913 > x <- rnorm(n, 0,1) # generate a predictor variable, x
1914
1915 > linpred <- beta0 + beta1*x # calculate linear predictor of E(y)
1916 > y <- rpois(n, exp(linpred)) # generate observations from model
```

1917 The **R** function `glm()` fits a GLM to the data we just generated and returns estimates of
 1918 β_0 and β_1 , which we see are fairly close to the data generating values above:

```
1919 > glm(y ~ 1 + x, family='poisson')      # the fit model
```

1920 This produces the output:

```
1921 Call: glm(formula = y ~ 1 + x, family = "poisson")
1922
1923 Coefficients:
1924 (Intercept)      x
1925     -2.007      1.446
1926
1927 [... some output deleted ...]
```

1928 In this summary output, the maximum likelihood estimates (MLEs) of the regression
 1929 parameters β_0 and β_1 are labeled “Coefficients.” We see that these are not too different
 1930 from the data-generating values (-2 and 1.5, respectively).

1931 The binomial GLM posits that $y_i \sim \text{Binomial}(K, p)$ where K is the fixed sample size
 1932 parameter and $\mathbb{E}(y_i) = K \times p_i$. Usually the model for the mean is specified using the *logit*
 1933 *link function* according to

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

1934 Where $\text{logit}(p) = \log(p/(1-p))$. The inverse-logit function, consequently, is $\text{logit}^{-1}(p) =$
 1935 $\exp(p)/(1 + \exp(p))$.

1936 A GLMM is the extension of GLMs to accommodate “random effects”. Often this
 1937 involves adding a normal random effect to the linear predictor. One simple example is
 1938 using a random intercept, α :

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

1939 where

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

1940 Many other probability distributions and formulations of the linear predictor might be
 1941 considered. GLMMs are enormously useful in ecological modeling applications for mod-
 1942 eling variation due to subjects, observers, spatial or temporal stratification, clustering,
 1943 and dependence that arises from any kind of group structure and, of course, because SCR
 1944 models prove to be a type of GLM with a random effect, but one that does not enter the
 1945 mean linearly.

3.2 BAYESIAN ANALYSIS

1946 Bayesian analysis is less familiar to many ecological researchers because they are often
 1947 educated only in the classical statistical paradigm of frequentist inference. But advances
 1948 in technology and increasing exposure to the benefits of Bayesian analysis are fast mak-
 1949 ing Bayesians out of people or at least making Bayesian analysis an acceptable, general
 1950 alternative to classical, frequentist inference.

1951 Conceptually, the main thing about Bayesian inference is that it uses probability
 1952 directly to characterize uncertainty about things we don't know. "Things", in this case,
 1953 are parameters of models and, just as it is natural to characterize uncertain outcomes of
 1954 stochastic processes using probability, it seems natural also to characterize information
 1955 about unknown parameters using probability. At least this seems natural to us and, we
 1956 think, most ecologists either explicitly adopt that view or tend to fall into that point
 1957 of view naturally. Conversely, frequentists use probability in many different ways, but
 1958 never to characterize uncertainty about parameters¹. Instead, frequentists use probability
 1959 to characterize the behavior of *procedures* such as estimators or confidence intervals (see
 1960 below). It is surprising that people readily adopt a philosophy of statistical inference in
 1961 which the things you don't know (i.e., parameters) should *not* be regarded as random
 1962 variables, so that, as a consequence, one cannot use probability to characterize one's state
 1963 of knowledge about them.

1964 3.2.1 Bayes' rule

1965 As its name suggests, Bayesian analysis makes use of Bayes' rule in order to make direct
 1966 probability statements about model parameters. Given two random variables z and y ,
 1967 Bayes' rule relates the two conditional probability distributions $[z|y]$ and $[y|z]$ by the
 1968 relationship:

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

1969 Bayes' rule itself is a mathematical fact and there is no debate in the statistical community
 1970 as to its validity and relevance to many problems. Generally speaking, these distributions
 1971 are characterized as follows: $[y|z]$ is the conditional probability distribution of y given z ,
 1972 $[z]$ is the marginal distribution of z and $[y]$ is the marginal distribution of y . In the context
 1973 of Bayesian inference we usually associate specific meanings in which $[y|z]$ is thought of
 1974 as "the likelihood", $[z]$ as the "prior" and so on. We leave this for later because here the
 1975 focus is on this expression of Bayes' rule as a basic fact of probability.

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

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

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

Mathematically, this is

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

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

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

3.2.2 Principles of Bayesian inference

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

2011 all unknown quantities of a model as realizations of random variables – this includes data,
 2012 latent variables, and also parameters. Classical (non-Bayesian) analysts sometimes object
 2013 to regarding parameters as outcomes of random variables. Classically, parameters are
 2014 thought of as “fixed but unknown” (using the terminology of classical statistics). Indeed,
 2015 a common misunderstanding on the distinction between Bayesian and frequentist infer-
 2016 ence goes something like this “in frequentist inference parameters are fixed but unknown
 2017 but in a Bayesian analysis parameters are random.” At best this is a sad caricature of the
 2018 distinction and at worst it is downright wrong. In Bayesian analysis the parameters are
 2019 also unknown and, in fact, there is a single data-generating value of each parameter, and
 2020 so they are also fixed. The difference is that the fixed but unknown values are regarded
 2021 as having been generated from some probability distribution. Specification of that prob-
 2022 ability distribution is necessary to carry out Bayesian analysis, but it is not required in
 2023 classical frequentist inference.

2024 To see the general relevance of Bayes’ rule in the context of statistical inference, let y
 2025 denote observations - i.e., data - and let $[y|\theta]$ be the observation model (often colloquially
 2026 referred to as the “likelihood”). Suppose θ is a parameter of interest having (prior)
 2027 probability distribution $[\theta]$ (also simply referred to as the prior). These are combined to
 2028 obtain the posterior distribution using Bayes’ rule, which is:

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

2029 Asserting the general relevance of Bayes’ rule to all statistical problems, we can conclude
 2030 that the two main features of Bayesian inference are that: (1) parameters, θ , are regarded
 2031 as realizations of a random variable and, as a result, (2) inference is based on the prob-
 2032 ability distribution of the parameters given the data, $[\theta|y]$, which is called the posterior
 2033 distribution. This is the result of using Bayes’ rule to combine the “likelihood” and the
 2034 prior distribution. The key concept is regarding parameters as realizations of a random
 2035 variable because, once you admit this conceptual view, this leads directly to the posterior
 2036 distribution, a very natural quantity upon which to base inference about things we don’t
 2037 know - including parameters of statistical models. In particular, $[\theta|y]$ is a probability
 2038 distribution for θ and therefore we can make direct probability statements to characterize
 2039 uncertainty about θ .

2040 The denominator of our invocation of Bayes’ rule, $[y]$, is the marginal distribution of
 2041 the data y . We note without further remark right now that, in many practical problems,
 2042 this can be an enormous pain to compute. The main reason that the Bayesian paradigm
 2043 has become so popular in the last 20 years or so is because methods have been developed
 2044 for characterizing the posterior distribution that do not require that we possess a math-
 2045 ematical understanding of $[y]$. This means we never have to compute it or know what it
 2046 looks like, or know anything specific about it.

2047 While we can understand the conceptual basis of Bayesian inference merely by under-
 2048 standing Bayes’ rule – that’s really all there is to it – it is not so easy to understand the
 2049 basis of classical frequentist inference. What is mostly coherent in frequentist inference is
 2050 the manner in which procedures are evaluated – the performance of a given procedure is
 2051 evaluated by “averaging over” hypothetical realizations of y , regarding the *estimator* as a
 2052 random variable. For example, if $\hat{\theta}$ is an estimator of θ then the frequentist is interested
 2053 in $E_y(\hat{\theta}|y)$ which is used to characterize bias. If the expected value of $\hat{\theta}$, when averaged
 2054 over realizations of y , is equal to θ , then $\hat{\theta}$ is unbiased.

2055 The view of parameters as being random variables allows Bayesians to use probability
2056 to make direct probability statements about parameters. Frequentist inference procedures
2057 do not permit direct probability statements to be made about parameter values. Instead,
2058 the view of parameters as fixed constants and estimators as random variables leads to
2059 interpretations that are not so straightforward. For example confidence intervals having
2060 the interpretation “95% probability that the interval contains the true value” and p-values
2061 being “the probability of observing an outcome of the test statistic as extreme or more
2062 than the one observed.” These are far from intuitive interpretations to most people.
2063 Moreover, this is conceptually problematic to some because we will never get to observe
2064 the hypothetical realizations that characterize the performance of our procedure.

2065 While we do tend to favor Bayesian inference for the conceptual simplicity (parameters
2066 are random, posterior inference), we mostly advocate for a pragmatic non-partisan
2067 approach to inference because, frankly, some of the frequentist methods are actually very
2068 convenient in certain situations, and will generally yield very similar inferences about
2069 parameters, as we will see in later chapters.

2070 3.2.3 Prior distributions

2071 The prior distribution $[\theta]$ is an important feature of Bayesian inference. As a conceptual
2072 matter, the prior distribution characterizes “prior beliefs” or “prior information” about
2073 a parameter. Indeed, an oft-touted benefit of Bayesian analysis is the ease with which
2074 prior information can be included in an analysis. However, more commonly, the prior
2075 is chosen to express a lack of prior information, even if previous studies have been done
2076 and even if the investigator does in fact know quite a bit about a parameter. This is
2077 because the manner in which prior information is embodied in a prior (and the amount
2078 of information) is usually very subjective and thus the result can wind up being very
2079 contentious; e.g., different investigators might report different results based on subjective
2080 assessments of prior information. Thus it is usually better to “let the data speak” and
2081 use priors that reflect absence of information beyond the data set being analyzed. An
2082 example for an uninformative prior is a Uniform(0, 1) for a probability, or a Uniform($-\infty$,
2083 ∞) (also called a “flat” or “improper” prior) for an unbounded continuous parameter.
2084 Alternatively, people use “diffuse priors”; these contain some information, but (ideally)
2085 not enough to exert meaningful influence on the posterior. An example for a diffuse prior
2086 could be a normal distribution with a large standard deviation.

2087 But still the need occasionally arises to embody prior information or beliefs about a
2088 parameter formally into the estimation scheme. In SCR models we often have a parameter
2089 that is closely linked to “home range size” and thus auxiliary information on the home
2090 range size of a species can be used as prior information, which may improve parameter
2091 estimation (e.g., see Chandler and Royle (In press); also Chapt. 18).

2092 At times the situation arises where a prior can inadvertently impose substantial effect
2093 on the posterior of a parameter, and that is not desirable. For example, we use data
2094 augmentation to deal with the fact that the population size N is an unknown parameter
2095 (Royle et al., 2007) which is equivalent to imposing a Binomial(M, ψ) prior on N for some
2096 integer M (see Sec. 4.2). One has to take care to make sure that M is sufficiently large so
2097 as to not affect the posterior distribution on N (see Fig. 17.4.3, and also Kéry and Schaub
2098 (2012, Ch. 5)). Another situation that we have to be careful of is that prior distributions

2099 are *not* invariant to transformation of the parameter, and therefore neither are posterior
 2100 distributions (Link and Barker, 2010, Sec. 6.2.1). Thus, a prior that is ostensibly non-
 2101 informative on one scale, may be very informative on another scale. For example, if we
 2102 have a flat prior on $\text{logit}(p)$ for some probability parameter p , this is very different from
 2103 having a Uniform(0,1) prior on p . We show an example where this makes a difference in
 2104 Chapt. 5. Nonetheless, it is always possible to assess the influence of prior choice, and
 2105 it is often the case (with sufficient data and a structurally identifiable model) that the
 2106 influence of priors is negligible.

2107 **3.2.4 Posterior inference**

2108 In Bayesian inference, we are not focusing on estimating a single point or interval but
 2109 rather on characterizing a whole distribution – the posterior distribution – from which
 2110 one can report any summary of interest. A point estimate might be the posterior mean,
 2111 median, mode, etc.. In many applications in this book, we will compute 95% Bayesian
 2112 confidence intervals using the 2.5% and 97.5% quantiles of the posterior distribution. For
 2113 such intervals, it is correct to say $\Pr(L < \theta < U) = 0.95$. That is, “the probability that θ
 2114 lies between L and U is 0.95”.

2115 As an example, suppose we conducted a Bayesian analysis to estimate detection prob-
 2116 ability (p) of some species at a study site, and we obtained a posterior distribution of
 2117 beta(20,10) for the parameter p . The following R commands demonstrate how we make
 2118 inferences based upon summaries of the posterior distribution:

```
2119 > post.median <- qbeta(0.5, 20, 10)
2120 [1] 0.6704151
2121
2122 > post.95ci <- qbeta(c(0.025, 0.975), 20, 10)
2123 [1] 0.4916766 0.8206164
```

2124 Thus, we can state that there is a 95% probability that θ lies between 0.49 and 0.82. Fig.
 2125 3.1 shows the posterior along with the summary statistics. It is not a subtle thing that
 2126 such statements cannot be made using frequentist methods, although people tend to say
 2127 it anyway and not really understand why it is wrong or even that it is wrong.

2128 **3.2.5 Small sample inference**

2129 The posterior distribution is an exhaustive summary of the state-of-knowledge about an
 2130 unknown quantity. It is *the* posterior distribution - not an estimate of that thing. It is
 2131 also not, usually, an approximation except to within Monte Carlo error (in cases where
 2132 we use simulation to calculate it, see Sec. 3.5.2). One of the great virtues of Bayesian
 2133 analysis which is not widely appreciated is that posterior inference is not “asymptotic”,
 2134 which is to say, valid in a limiting sense as the sample size tends to infinity. Rather,
 2135 posterior inference is valid for *any* sample size and, in particular, *the* sample size on-hand.
 2136 Conversely, almost all frequentist procedures are based on asymptotic approximations to
 2137 the procedure which is being employed.

2138 There seems to be a prevailing view in statistical ecology that classical likelihood-based
 2139 procedures are virtuous because of the availability of simple formulas and procedures for

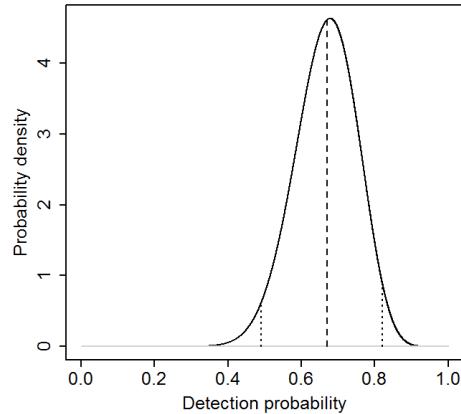


Figure 3.1. Probability density plot of a hypothetical posterior distribution of $\text{beta}(20,10)$; dashed lines indicate mean and upper and lower 95% interval

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

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

²Actual quote from a referee

2162 may be all that is available for species that are extraordinarily important for conservation
2163 and management. The Bayesian framework for inference provides a valid, rigorous, and
2164 flexible framework that is theoretically justifiable in arbitrary sample sizes. This is not to
2165 say that one will obtain precise estimates of density or other parameters, just that your
2166 inference is coherent and justifiable from a conceptual and technical statistical point of
2167 view. That is, for example when we estimate the density D of some animal population,
2168 we report the posterior probability $\Pr(D|data)$ which is easily interpretable and just what
2169 it is advertised to be and we don't need to do a simulation study to evaluate how well
2170 the reported $\Pr(D|data)$ deviates from the "true" $\Pr(D|data)$ because they are the same
2171 quantity.

3.3 CHARACTERIZING POSTERIOR DISTRIBUTIONS BY MCMC SIMULATION

2172 In practice, it is not really feasible to ever compute the marginal probability distribution
2173 [y], the denominator resulting from application of Bayes' rule (Eq. 3.2.1). For decades
2174 (even centuries!) this impeded the adoption of Bayesian methods by practitioners. Or,
2175 the few Bayesian analyses done were based on asymptotic normal approximations to the
2176 posterior distribution. While this was useful from a theoretical and technical standpoint
2177 and, practically, it allowed people to make the probability statements that they naturally
2178 would like to make, it was kind of a bad joke around the Bayesian water-cooler to, on
2179 one hand, criticize classical statistics for being, essentially, completely ad hoc in their
2180 approach to things but then, on the other hand, have to devise various approximations to
2181 what they were trying to characterize. The advent of Markov chain Monte Carlo (MCMC)
2182 methods has made it easier to calculate posterior distributions for just about any problem
2183 to sufficient levels of precision.

2184 Broadly speaking, MCMC is a class of methods for drawing random samples (i.e.,
2185 simulating from or just "sampling") from the target posterior distribution. Thus, even
2186 though we might not recognize the posterior as a named distribution or be able to analyze
2187 its features analytically, e.g., devise mathematical expressions for the mean and variance,
2188 we can use these MCMC methods to obtain a large sample from the posterior and then
2189 use that sample to characterize features of the posterior. What we do with the sample
2190 depends on our intentions – typically we obtain the mean or median for use as a point
2191 estimate, and take a confidence interval based on Monte Carlo estimates of the quantiles.

2192 3.3.1 What goes on under the MCMC hood

2193 We will develop and apply MCMC methods in some detail for spatial capture-recapture
2194 models in Chapt. 17. Here we provide a simple illustration of some basic ideas related to
2195 the practice of MCMC.

2196 A type of MCMC method relevant to most problems is Gibbs sampling (Geman and
2197 Geman, 1984) which we address in more detail in Chapt. 17. Gibbs sampling involves iter-
2198 ative simulation from the "full conditional" distributions (also called conditional posterior
2199 distributions). The full conditional distribution for an unknown quantity is the conditional
2200 distribution of that quantity given every other random variable in the model - the data
2201 and all other parameters (see Sec. 3.3.2 for rules of how to construct full conditionals).

2202 For example, for a normal regression model ³ with $y \sim \text{Normal}(\beta_0 + \beta_1(x - \bar{x}), \sigma^2)$ where
 2203 lets say σ^2 is known, the full conditionals are, using “bracket notation”,

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

2204 and

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

2205 We might use our knowledge of probability to identify these mathematically. In particular,
 2206 by Bayes' Rule, $[\beta_0|y, \beta_1] = [y|\beta_0, \beta_1][\beta_0|\beta_1]/[y|\beta_1]$ and similarly for $[\beta_1|y, \beta_0]$. For
 2207 example, if we have priors for $[\beta_0] = \text{Normal}(\mu_{\beta_0}, \sigma_{\beta_0}^2)$ and $[\beta_1] = \text{Normal}(\mu_{\beta_1}, \sigma_{\beta_1}^2)$ then
 2208 some algebra reveals that

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

2209 where $\tau = 1/\sigma^2$ and $\tau_{\beta_0} = 1/\sigma_{\beta_0}^2$ (the inverse of the variance is sometimes called *precision*),
 2210 and $w = \tau n / (\tau n + \tau_{\beta_0})$. We see in this case that the posterior mean is a *precision-weighted*
 2211 sum of the sample mean \bar{y} and the prior mean μ_{β_0} , and the posterior *precision* is the
 2212 sum of the precision of the likelihood and that of the prior. These results are typical of
 2213 many classes of problems. In particular, note that as the prior precision tends to 0, i.e.,
 2214 $\tau_{\beta_0} \rightarrow 0$, then the posterior of β_0 tends to $\text{Normal}(\bar{y}, \sigma^2/n)$. We recognize the variance of
 2215 this distribution as that of the variance of the sampling distribution of \bar{y} and its mean is
 2216 in fact the MLE of β_0 for this model. The conditional posterior of β_1 has a very similar
 2217 form:

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

2218 which might look slightly unfamiliar, but note that if $\tau_{\beta_1} = 0$, then the mean of this
 2219 distribution is the familiar $\hat{\beta}_1$, and the variance is, in fact, the sampling variance of $\hat{\beta}_1$.
 2220 The MCMC algorithm for this model has us simulate in succession, repeatedly, from
 2221 those two distributions. See Gelman et al. (2004) for more examples of Gibbs sampling
 2222 for the normal model, and we also provide another example in Chapt. 17. A conceptual
 2223 representation of the MCMC algorithm for this simple model is therefore:

Algorithm: Gibbs Sampling for linear regression

```

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

```

2225 As we just saw for this simple “normal-normal” model, it is sometimes possible to
 2226 specify the full conditional distributions analytically. In general, when certain so-called
 2227 conjugate prior distributions are used, which have an analytic form that, in a statistical

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

2228 sense, “matches” the likelihood, then the form of the full conditional distributions is also
 2229 similar to that of the observation model. In this normal-normal case, the normal distribu-
 2230 tion for the mean parameters is the conjugate prior for the normal observation model, and
 2231 thus the full-conditional distributions are also normal. This is convenient because, in such
 2232 cases, we can simulate directly from them using standard methods (or **R** functions). But,
 2233 in practice, we don’t really ever need to know such things because most of the time we
 2234 can get by using a simple algorithm, called the Metropolis-Hastings (henceforth “MH”)
 2235 algorithm, to obtain samples from these full conditional distributions without having to
 2236 recognize them as specific, named, distributions. This gives us enormous freedom in devel-
 2237 oping models and analyzing them without having to resolve them mathematically because
 2238 to implement the MH algorithm we need only identify the full conditional distribution up
 2239 to a constant of proportionality, that being the marginal distribution in the denominator
 2240 (e.g., $[y|\beta_1]$ above).

2241 We will talk about the Metropolis-Hastings algorithm shortly, and we will use it ex-
 2242 tensively in the analysis of SCR models (e.g., Chapt. 17).

2243 3.3.2 Rules for constructing full conditional distributions

2244 The basic strategy for constructing full-conditional distributions for devising MCMC al-
 2245 gorithms can be reduced conceptually to a couple of basic steps summarized as follows:

- 2246 **(step 1)** Identify all stochastic components of the model and collect their probability
 2247 distributions;
- 2248 **(step 2)** Express the full conditional in question as proportional to the product of all
 2249 probability distributions identified in step 1;
- 2250 **(step 3)** Remove the ones that don’t have the focal parameter in them.
- 2251 **(step 4)** Do some algebra on the result in order to identify the resulting probability
 2252 distribution function (pdf) or mass function (pmf).

2253 Of the 4 steps, the last of those is the main step that requires quite a bit of statistical
 2254 experience and intuition because various algebraic tricks can be used to reshape the mess
 2255 into something recognizable – i.e., a standard, named distribution. But step 4 is not
 2256 necessary if we decide instead to use the Metropolis-Hastings algorithm as described below.

2257 In the context of our simple linear regression model that we’ve been working with,
 2258 to characterize $[\beta_0|y, \beta_1]$ we first apply step 1 and identify the model components as:
 2259 $[y|\beta_0, \beta_1]$, with prior distributions $[\beta_0]$ and $[\beta_1]$. Step 2 has us write $[\beta_0|y, \beta_1] \propto [y|\beta_0, \beta_1][\beta_0][\beta_1]$.
 2260 Step 3: We note that $[\beta_1]$ is not a function of β_0 and therefore we remove it to obtain
 2261 $[\beta_0|y, \beta_1] \propto [y|\beta_0, \beta_1][\beta_0]$. Similarly, applying step 2 and 3 for β_1 we obtain $[\beta_1|y, \beta_0] \propto$
 2262 $[y|\beta_0, \beta_1][\beta_1]$. We apply step 4 and manipulate these algebraically to arrive at the re-
 2263 sult (which we provided in Eqs. 3.3.1 and 3.3.2) or, alternatively, we can sample them
 2264 indirectly using the Metropolis-Hastings algorithm, which we discuss now.

2265 3.3.3 Metropolis-Hastings algorithm

2266 The Metropolis-Hastings (MH) algorithm is a completely generic method for sampling
 2267 from any distribution, say $[\theta]$. In our applications, $[\theta]$ will typically be the full conditional
 2268 distribution of θ . While we sometimes use Gibbs sampling, we seldom use “pure” Gibbs

2269 sampling because full conditionals do not always take the form of known distributions we
 2270 can sample from directly. In such cases, we use MH to sample from the full conditional
 2271 distributions. When the MH algorithm is used to sample from full conditional distributions
 2272 of a Gibbs sampler the resulting hybrid algorithm is called *Metropolis-within-Gibbs*. In
 2273 Sec. 3.6.3 we will construct such an algorithm for a simple class of models. We discuss
 2274 both the Gibbs and the MH algorithm, as well as their hybrid in more depth in Chapt.
 2275 17.

2276 The MH algorithm generates candidate values for the parameter(s) we want to estimate
 2277 from some proposal or candidate-generating distribution that may be conditional on the
 2278 current value of the parameter, denoted by $h(\theta^*|\theta^{t-1})$. Here, θ^* is the *candidate* or
 2279 proposed value and θ^{t-1} is the value of θ at the previous time step, i.e., at iteration $t - 1$
 2280 of the MCMC algorithm. The proposed value is accepted with probability

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

2281 which is called the MH acceptance probability. This ratio can sometimes be > 1 in which
 2282 case we set it equal to 1. It is useful to note that $h()$ can be any probability distribution.

2283 In the context of using the MH algorithm to do MCMC (in which case the target
 2284 distribution is a full-conditional or posterior distribution), an important fact is, no matter
 2285 the choice of $h()$, we can compute the MH acceptance probability directly because the
 2286 marginal distribution of y cancels from both the numerator and denominator of r . This
 2287 is the magic of the MH algorithm.

3.4 BAYESIAN ANALYSIS USING THE BUGS LANGUAGE

2288 We won't be too concerned with devising our own MCMC algorithms for every analysis,
 2289 although we will do that a few times for fun. More often, we will rely on the freely available
 2290 software package **WinBUGS** or **JAGS** for doing this. We will always execute these
 2291 **BUGS** engines from within **R** using the **R2WinBUGS** (Sturtz et al., 2005) or, for **JAGS**,
 2292 the **R2jags** (Su and Yajima, 2011) or **rjags** (Plummer, 2009) packages. **WinBUGS** and
 2293 **JAGS** are MCMC black boxes that take a pseudo-code description (i.e., written in the
 2294 **BUGS** language) of all of the relevant stochastic and deterministic elements of a model
 2295 and generate an MCMC algorithm for that model. But you never get to see the algorithm.
 2296 Instead, **WinBUGS/JAGS** will run the algorithm and return the Markov chain output
 2297 - the posterior samples of model parameters.

2298 The great thing about using the **BUGS** language is that it forces you to become
 2299 intimate with your statistical model - you have to write each element of the model down,
 2300 admit (explicitly) all of the various assumptions, understand what the actual probability
 2301 assumptions are and how data relate to latent variables and data and latent variables
 2302 relate to parameters, and how parameters relate to one another.

2303 While we normally use **WinBUGS**, we note that **OpenBUGS** is the current active
 2304 development tree of the **BUGS** project. See Kéry (2010) and Kéry and Schaub (2012,
 2305 especially Appendix 1) for more on practical analysis in **WinBUGS**. Those books should
 2306 be consulted for a more comprehensive introduction to using **WinBUGS**. Recently we
 2307 have migrated many of our analyses to **JAGS** (Plummer, 2009), which we adopt later in

2308 the book. You can refer to Hobbs (2011) for an ecological introduction to **JAGS**. Next,
 2309 we provide an example of a Bayesian analysis using **WinBUGS**.

2310 **3.4.1 Linear regression in WinBUGS**

2311 We provide a brief introductory example of a normal regression model using a small
 2312 simulated data set. The following commands are executed from within your **R** workspace.
 2313 First, simulate a covariate x and observations y having prescribed intercept, slope and
 2314 variance:

```
2315 > x <- rnorm(10)
2316 > mu <- -3.2 + 1.5*x
2317 > y <- rnorm(10, mu, sd=4)
```

2318 The **BUGS** model specification for a normal regression model is written within **R** as
 2319 a character string input to the command `cat()` and then dumped to a text file named
 2320 `normal.txt`:

```
2321 > cat("
2322   model{
2323     for (i in 1:10){
2324       y[i] ~ dnorm(mu[i],tau)      # the likelihood
2325       mu[i] <- beta0 + beta1*x[i]  # the linear predictor
2326     }
2327     beta0 ~ dnorm(0,.01)         # prior distributions
2328     beta1 ~ dnorm(0,.01)
2329     sigma ~ dunif(0,100)
2330     tau <- 1/(sigma*sigma)      # tau is the precision
2331   }                                # and a derived parameter
2332 ",file="normal.txt")
```

2333 Alternatively, you can write the model specifications directly within a text file and save it
 2334 in your current working directory, but we do not usually take that approach in this book.

2335 The **BUGS** dialects⁴ parameterize the normal distribution in terms of the mean and
 2336 inverse-variance, called the precision. Thus, `dnorm(0,.01)` implies a variance of 100.
 2337 We typically use diffuse normal priors for mean parameters, β_0 and β_1 in this case, but
 2338 sometimes we might use uniform priors with suitable bounds $-B$ and $+B$. Also, we
 2339 typically use a Uniform($0, B$) prior on standard deviation parameters (Gelman, 2006).
 2340 But sometimes we might use a gamma prior on the precision parameter τ . In a **BUGS**
 2341 model file, every variable referenced in the model description has to be either data, which
 2342 will be input (see below), a random variable which must have a probability distribution
 2343 associated with it using the tilde character “~” (a.k.a. “twiddle”) or it has to be a derived
 2344 parameter connected to variables and data using an assignment arrow: “<-”.

2345 To fit the model, we need to describe various data objects to **WinBUGS**. In particular,
 2346 we create an **R** list object called `data` which are the data objects identified in the **BUGS**
 2347 model file. In the example, the data consist of two objects which exist as y and x in the

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

2348 **R** workspace and also in the **WinBUGS** model definition. We also create an **R** function
 2349 that produces a list of starting values, **inits**, that get sent to **WinBUGS**. In general,
 2350 starting values are optional. We recommend to always provide reasonable starting values
 2351 where possible, both for structural parameters and also random effects⁵. Finally, we
 2352 identify the names of the parameters (labeled correspondingly in the **WinBUGS** model
 2353 specification) that we want **WinBUGS** to save the MCMC output for. In this example,
 2354 we will “monitor” the parameters β_0 , β_1 , σ and τ . **WinBUGS** is executed using the
 2355 **R** command **bugs()**. We set the option **debug=TRUE** if we want the **WinBUGS** GUI to
 2356 stay open (useful for analyzing MCMC output and looking at the **WinBUGS** error log).
 2357 Also, we set **working.dir=getwd()** so that **WinBUGS** output files and the log file are
 2358 saved in the current **R** working directory (note that sometimes you will need to specify the
 2359 place where you installed **WinBUGS** within the **bugs()** call, using the **bugs.directory**
 2360 argument). All of these activities together look like this:

```
2361 > library(R2WinBUGS)      # "load" the R2WinBUGS package
2362 > data <- list( y=y, x=x)
2363 > inits <- function()
2364 > list ( beta1=rnorm(1),beta0=rnorm(1),sigma=runif(1,0,2) )
2365 > parameters <- c("beta0","beta1","sigma","tau")
2366 > out <- bugs(data, inits, parameters, "normal.txt", n.thin=1, n.chains=2,
2367   n.burnin=2000, n.iter=6000, debug=TRUE,working.dir=getwd())
```

2368 Note that the previously created objects defining data, initial values and parameters to
 2369 monitor are passed to the function **bugs()**. In addition, various other things are declared:
 2370 The number of parallel Markov chains (**n.chains**), the thinning rate (**n.thin**), the number
 2371 of burn-in iterations (**n.burnin**) and the total number of iterations (**n.iter**). To develop
 2372 a detailed understanding of the various parameters and settings used for MCMC, consult
 2373 a basic reference such as Kéry (2010). We also come back to these issues in the following
 2374 section (3.5) and in Chapt. 17. A common question is “how should my data be formatted?”
 2375 That depends on how you describe the model in the **BUGS** language, and how your data
 2376 are input into **R**. There is no unique way to describe any particular model and so you have
 2377 some flexibility. We talk about data format further in the context of capture-recapture
 2378 models and SCR models in Chapt. 5 and elsewhere.

2379 You should execute all of the commands given above and then close the **WinBUGS**
 2380 GUI, and the data will be read back into **R** (or specify **debug=FALSE** in the **bugs()** call).
 2381 We don’t want to give instructions on how to navigate and use the GUI – but you can
 2382 fire up **WinBUGS** and read the help files, or see Chapt. 4 from Kéry (2010) for a brief
 2383 introduction. The **print** command applied to the object **out** prints some basic summary
 2384 output (this is slightly edited):

```
2385 > print(out,digits=2)
2386 Inference for Bugs model at "normal.txt", fit using WinBUGS,
2387 2 chains, each with 6000 iterations (first 2000 discarded)
```

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

```

2388 n.sims = 8000 iterations saved
2389      mean   sd 2.5% 25% 50% 75% 97.5% Rhat n.eff
2390 beta0    -6.62 1.64 -9.77 -7.63 -6.64 -5.63 -3.29     1  4200
2391 beta1     0.81 1.20 -1.63  0.09  0.80  1.54  3.24     1  5100
2392 sigma     4.99 1.56  2.93  3.92  4.66  5.70  8.85     1  8000
2393 tau       0.05 0.03  0.01  0.03  0.05  0.07  0.12     1  8000
2394 deviance 58.72 3.21 55.06 56.35 57.85 60.26 67.15     1  6200
2395
2396 For each parameter, n.eff is a crude measure of effective sample size,
2397 and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
2398
2399 DIC info (using the rule, pD = Dbar-Dhat)
2400 pD = 2.5 and DIC = 61.3

```

2401 In the **WinBUGS** output you see a column called “Rhat”, as well as one called
2402 “n.eff”. These are convergence diagnostics (the \hat{R} or Brooks-Gelman-Rubin statistic
2403 and the effective sample size) and we will discuss those in the following section, 3.5.2.
2404 DIC is the deviance information criterion (Spiegelhalter et al. (2002), see section 3.9)
2405 which some people use in a manner similar to AIC although it is recognized to have some
2406 problems in hierarchical models (Millar, 2009). We consider use of DIC in the context of
2407 SCR models in Chapt. 8.

3.5 PRACTICAL BAYESIAN ANALYSIS AND MCMC

2408 The mere execution of a Bayesian analysis using the **BUGS** language, as demonstrated
2409 with the linear regression example, is fairly straight forward. There are, however, a number
2410 of really important practical issues to be considered in any Bayesian analysis and we cover
2411 some of these briefly here before we move on to implementing slightly more complex
2412 GL(M)Ms in a Bayesian framework.

2413 **3.5.1 Choice of prior distributions**

2414 Bayesian analysis requires that we choose prior distributions for all of the structural pa-
2415 rameters of the model (we use the term structural parameter to mean all parameters that
2416 aren’t customary thought of as latent variables). We will strive to use priors that are
2417 meant to express little or no prior information - default or customary “non-informative”
2418 or diffuse priors. This will be $\text{Uniform}(a, b)$ priors for parameters that have a natural
2419 bounded support and, for parameters that live on the real line we use either (1) diffuse
2420 normal priors, as we did in the linear regression example above; (2) improper uniform
2421 priors which have unbounded support, e.g., $[\theta] \propto 1$, or (3) sometimes even a bounded
2422 $\text{Uniform}(a, b)$ prior, if that greatly improves the performance of **WinBUGS** or other
2423 software doing the MCMC for us. In **WinBUGS** a prior with low precision, τ , where
2424 $\tau = 1/\sigma^2$, such as $\text{Normal}(0, .01)$ will typically be used. Of course $\tau = 0.01$ ($\sigma^2 = 100$)
2425 might be very informative for a regression parameter depending on its magnitude and
2426 scaling of x . Therefore, we recommend that predictor variables (covariates) *always* be
2427 standardized to have mean 0 and variance 1.

2428 **Lack of invariance of priors to transformation.** Clearly there are a lot of choices
 2429 for ostensibly non-informative priors, and the degree of non-informativeness depends on
 2430 the parameterization. For example, a natural non-informative prior for the intercept of a
 2431 logistic regression

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

2432 would be a very diffuse normal prior, $[\beta_0] = \text{Normal}(0, \text{Large})$ or even $\beta_0 \sim \text{Uniform}(-\text{Large}, \text{Large})$.
 2433 However, we might also use a prior on the parameter $p_0 = \text{logit}^{-1}(\beta_0)$, which is $\Pr(y=1)$
 2434 for the value $x=0$. Since p_0 is a probability a natural choice is $p_0 \sim \text{Uniform}(0, 1)$. These
 2435 priors are very different in their implications. For example, if we choose the normal prior
 2436 for β_0 with variance $\text{Large} = 5^2$ and look at the implied prior for p_0 we have the result
 shown in Fig. 3.2 which looks nothing like a $\text{Uniform}(0, 1)$ prior. These two priors can

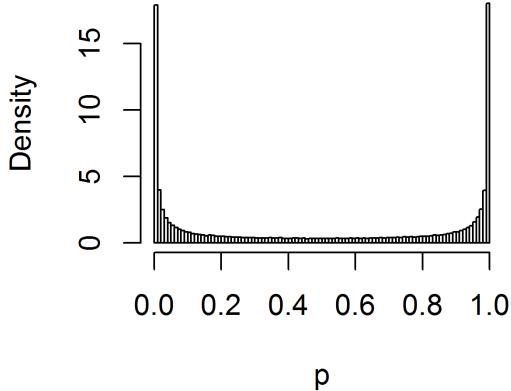


Figure 3.2. Implied prior for $p_0 = \exp(\beta_0)/(1 + \exp(\beta_0))$ if $\beta_0 \sim \text{Normal}(0, 5^2)$.

2437 affect results (see Sec. 4.4.2 for an illustration of this for a real data set), yet they are
 2438 both sensible non-informative priors. Despite this, it is often the case that priors will have
 2439 little or no impact on the results. Choice of priors and parameterization is very much
 2440 problem-specific and often largely subjective. Moreover, it also affects the behavior of
 2441 MCMC algorithms and therefore the analyst needs to pay some attention to this issue
 2442 and possibly try different things out. Most standard Bayesian analysis books address
 2443 issues related to specification and effect of prior distribution choice in some depth. Some
 2444 good references include Kass and Wasserman (1996), Gelman (2006) and Link and Barker
 2445 (2010).

2447 **3.5.2 Convergence and so-forth**

2448 Once we have carried out an analysis by MCMC, there are many other practical issues
 2449 that we have to confront. One characteristic of MCMC sampling is that Markov chains
 2450 take some time to converge to their stationary distribution - in our case the posterior
 2451 distribution for some parameter given data, $[\theta|y]$. Only when the Markov chain has
 2452 reached its stationary distribution, the generated samples can be used to characterize the
 2453 posterior distribution. Thus, one of the most important issues we need to address is “have
 2454 the chains converged?” Since we do not know what the stationary posterior distribution
 2455 of our Markov chain should look like (this is the whole point of doing an MCMC analysis),
 2456 we effectively have no means to assess whether or not it has truly converged to this desired
 2457 distribution. Most MCMC algorithms only guarantee that, eventually, the samples being
 2458 generated will be from the target posterior distribution, but no-one can tell us how long
 2459 this will take. Also, you only know the part of your posterior distribution that the Markov
 2460 chain has explored so far – for all you know the chain could be stuck in a local maximum,
 2461 while other maxima remain completely undiscovered. Acknowledging that there is truly
 2462 nothing we can do to ever prove convergence of our MCMC chains, there are several things
 2463 we can do to increase the degree of confidence we have about the convergence of our chains.
 2464 Some problems are easily detected using simple plots, such as a time-series plot, where
 2465 parameter values of each MCMC iteration are plotted against the number of iterations.
 2466 Fig. 3.3 shows the time series plots for the three parameters – β_0 , β_1 and σ – from our
 2467 linear regression example, taken from the **WinBUGS** GUI before closing it to return to
 2468 **R**.

2469 Typically a period of transience is observed in the early part of the MCMC algorithm,
 2470 and this is usually discarded as the “burn-in” period. In our linear regression example,
 2471 within the `bugs()` call we set the burn-in period as 2000 iterations so these are auto-
 2472 matically removed by **WinBUGS** and are not part of the output (but Fig. 3.6 shows a
 2473 time-series plot that starts at iteration 0 with a clearly visible burn-in period). The quick
 2474 diagnostic to whether convergence has been achieved is that your Markov chains look
 2475 “grassy” – this seems a reasonable statement for the plots in Fig. 3.3. Another way to
 2476 check convergence is to update the parameters some more and see if the posterior changes.
 2477 If the chains have converged to the posterior, the posterior mean, confidence intervals, and
 2478 other summaries should be relatively static as we continue to run the algorithm. Yet an-
 2479 other option, and one generally implemented in **WinBUGS**, is to run several Markov
 2480 chains and to start them off at different initial values that are over-dispersed relative to
 2481 the posterior distribution. Such initial values help to explore different areas of the param-
 2482 eter space simultaneously; if, after a while, all chains oscillate around the same average
 2483 value, chances are good that they indeed converged to the posterior distribution. Gelman
 2484 and Rubin came up with the so-called “R-hat” statistic (\hat{R}) or Brooks-Gelman-Rubin
 2485 statistic that essentially compares within-chain and between-chain variance to check for
 2486 convergence of multiple chains (Gelman et al., 1996). The R-hat statistic should be close
 2487 to 1 if the Markov chains have converged and sufficient posterior samples have been ob-
 2488 tained. For the linear regression example, we ran two parallel chains (also specified in the
 2489 `bugs()` call) and **WinBUGS** returns the \hat{R} statistic for us as part of the summary model
 2490 output. If you look back to Sec. 3.4.1 you see that $\hat{R} = 1$ for all parameters of the linear
 2491 model. In practice, $\hat{R} \leq 1.2$ may be good enough for some problems. For some models you
 2492 can’t actually realize a low \hat{R} . E.g., if the posterior is a discrete mixture of distributions

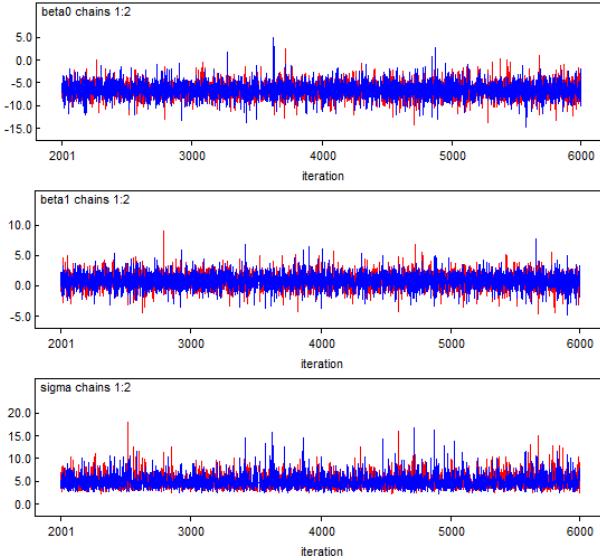


Figure 3.3. Time-series plots for parameters from a linear regression run in **WinBUGS** using two parallel Markov chains.

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

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

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

2513 (MacEachern and Berliner, 1994; Link and Eaton, 2011). Practical considerations might
 2514 necessitate thinning, even though it is statistically inefficient. For example, in models
 2515 with many parameters or other unknowns being tabulated, the output files might be
 2516 enormous and unwieldy to work with. In such cases, thinning is perfectly reasonable. In
 2517 many cases, how well the Markov chains mix is strongly influenced by parameterization,
 2518 standardization of covariates, and the prior distributions being used. Some things work
 2519 better than others, and the investigator should experiment with different settings and
 2520 remain calm when things don't work out perfectly.

2521 **Is the posterior sample large enough?** The subsequent samples generated from
 2522 a Markov chain are not *independent* samples from the posterior distribution, due to the
 2523 correlation among samples introduced by the Markov process⁶ and the sample size has
 2524 to be adjusted to account for the autocorrelation in subsequent samples (see Chapt. 8 in
 2525 Robert and Casella (2010) for more details). This adjusted sample size is referred to as the
 2526 effective sample size. Checking the degree of autocorrelation in your Markov chains and
 2527 estimating the effective sample size your chain has generated should be part of evaluating
 2528 your model output. **WinBUGS** will automatically return the effective sample size for
 2529 all monitored parameters, as we saw in our linear regression example (the “n.eff” column
 2530 of the summary output). If you find that your supposedly long Markov chain has only
 2531 generated a very short effective sample, you should consider a longer run. What exactly
 2532 constitutes a reasonable effective sample size is hard to say. A more palpable measure
 2533 of whether you've run your chain for enough iterations is the time-series or Monte Carlo
 2534 error - the “noise” introduced into your samples by the stochastic MCMC process. The
 2535 MC error is printed by default in summaries produced in the **WinBUGS** GUI, which
 2536 can be reproduced in **R** using `bugs.log('log.txt')$stats` (note that “log.txt” refers
 2537 to a model log file that **WinBUGS** automatically creates in the working directory; it is
 2538 overwritten with every new model you run unless you save it under a different name).

```
2539 > bugs.log('log.txt')$stats
2540 $stats
2541      mean      sd   Mcerror    2.5%   median   97.5% start sample
2542 beta0    -6.64700 1.60300 0.0179400 -9.7140 -6.70800 -3.2730 2001 8000
2543 beta1     0.82100 1.19000 0.0116800 -1.4900  0.82560  3.1800 2001 8000
2544 deviance  58.66000 3.08800 0.0506800 55.0700 57.93000 66.8400 2001 8000
2545 sigma      4.96800 1.52300 0.0248300  2.9350  4.68100  8.7410 2001 8000
2546 tau       0.05074 0.02677 0.0003651  0.0131  0.04564  0.1162 2001 8000
```

2547 When using **JAGS** the `summary` command will automatically produce the MC error
 2548 (which is called “Time-series SE” in **JAGS**). You want the MC error to be smallish relative
 2549 to the magnitude of the parameter and what smallish means will depend on the purpose
 2550 of the analysis. For a preliminary analysis you might settle for a few percent whereas
 2551 for a final analysis then certainly less than 1% is called for. You can run your MCMC
 2552 algorithm as long as it takes to achieve that. A consequence of the MC error is that even
 2553 for the exact same model, results will usually be slightly different. Thus, as a good rule of
 2554 thumb, you should avoid reporting MCMC results to more than 2 or 3 significant digits!

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

2555 **3.5.3 Bayesian confidence intervals**

2556 The 95% Bayesian confidence interval based on percentiles of the posterior is not a unique
 2557 interval - there are many of them. The so-called “highest posterior density” (HPD) inter-
 2558 val is an alternative, defined as the narrowest interval that contains *at least* 95% of the
 2559 posterior mass. As a result (of the *at least* clause), for discrete parameters, the 95% HPD
 2560 is not often exactly 95% but usually slightly more conservative than nominal.

2561 **3.5.4 Estimating functions of parameters**

2562 A benefit of analysis by MCMC is that we can seamlessly estimate functions of parameters
 2563 by simply tabulating the desired function of the simulated posterior draws. For example,
 2564 if θ is the parameter of interest and let $\theta^{(i)}$ for $i = 1, 2, \dots, M$ be the posterior samples
 2565 of θ . Let $\eta = \exp(\theta)$, then a posterior sample of η can be obtained simply by computing
 2566 $\exp(\theta^{(i)})$ for $i = 1, 2, \dots, M$. Almost all SCR models in this book involve at least 1 derived
 2567 parameter. For example, density D is a derived parameter, being a function of population
 2568 size N and the area A of the underlying state-space of the point process (see Chapt. 5).

2569 **Example: Finding the optimum value of a covariate.** As another example of
 2570 estimating functions of model parameters, suppose that the normal regression model from
 2571 Sec. 3.4.1 had a quadratic response function of the form

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

2572 Then the optimum value of x , i.e., that corresponding to the optimal expected response,
 2573 can be found by setting the derivative of this function to 0 and solving for x . We find that

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

2574 yields that $x_{opt} = -\beta_1/(2 * \beta_2)$. We can just take our posterior draws for β_1 and β_2
 2575 and obtain a posterior sample of x_{opt} by this simple calculation applied to the posterior
 2576 output. As an exercise, take the normal model above and simulate a quadratic response
 2577 and then describe the posterior distribution of x_{opt} .

3.6 POISSON GLMS

2578 The Poisson GLM (also known as “Poisson regression”) is probably the most relevant
 2579 and important class of models in all of ecology. The basic model assumes observations
 2580 $y_i; i = 1, 2, \dots, n$ follow a Poisson distribution with mean λ which we write

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

2581 Commonly y_i is a count of animals or plants at some point in space (“site”) i , and λ
 2582 might vary over sites as well. For example, i might index point count locations in a
 2583 forest, survey route centers, or sample quadrats, or similar, and we are interested in how
 2584 λ depends on site characteristics such as habitat. If covariates are available it is typical to
 2585 model them as linear effects on the log mean. If x_i is some measured covariate associated
 2586 with observation i , then,

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

2587 While we only specify the mean of the Poisson model directly, the Poisson model (and
 2588 all GLMs) has a “built-in” variance which is directly related to the mean. In this case,
 2589 $\text{Var}(y) = \mathbb{E}(y) = \lambda$. Thus the model accommodates a linear increase in variance with the
 2590 mean.

2591 3.6.1 Example: Breeding Bird Survey data

2592 As an example we consider a classical situation in ecology where counts of an organism
 2593 are made at a collection of spatial locations. In this particular example, we have
 2594 mourning dove (*Zenaida macroura*) counts made along North American Breeding Bird
 2595 Survey (BBS) routes in Pennsylvania, USA. A route consists of 50 stops separated by
 2596 0.5 miles. For the purposes here we are defining y_i = route total count and the sample
 2597 location will be marked by the center point of the BBS route. The survey is run annually
 2598 and the data set we analyze is 1966-1998. BBS data can be obtained online at
 2599 <http://www.pwrc.usgs.gov/bbs/>, but the particular chunk of data we will be using here
 2600 is also included in the **scrbook** package (**data(bbsdata)**). We will make use of the whole
 2601 data set shortly but for now we’re going to focus on a specific year of counts (1990) for
 2602 the sake of building a simple model. In 1990 there were 77 active routes; this data set
 2603 contains rows which index the unique route, column 1 is the route ID, columns 2-3 are
 2604 the route coordinates (longitude/latitude), column 4 is a habitat covariate “forest cover”
 2605 (standardized, see below) and the remaining columns are the yearly counts. Years for
 2606 which a survey was not conducted on a route are coded as “NA” in the data matrix. We
 2607 imagine that this will be a typical format for many ecological studies, perhaps with more
 2608 columns representing covariates. To read in the data and display the first few elements of
 2609 the data frame containing the counts, do this:

```
2610 > data(bbsdata)           #  loads data frame 'bbs'  

2611 > bbsdata$counts[1:2,1:6]  

2612  

2613      X     lon     lat    habitat X66 X67  

2614 1 72002 -80.445 41.501 -0.3871372 NA 24  

2615 2 72003 -80.347 41.214 -1.0171629 NA NA
```

2616 It is useful to display the spatial pattern in the observed counts. For that we use a
 2617 spatial dot plot – where we plot the coordinates of the observations and mark the color
 2618 of the plotting symbol based on the magnitude of the count. We have a special plotting
 2619 function for that which is called **spatial.plot()** and it is available with the supplemental
 2620 **R** package **scrbook**. Actually, what we want to do here is plot the log-counts (+1 of
 2621 course) which (Fig. 3.4) display a notable pattern that could be related to something.
 2622 The **R** commands for obtaining this figure are:

```
2623 > library(scrbook)  

2624 > data(bbsdata)  

2625 > library(maps)  

2626  

2627 > y <- bbsdata$counts[, "X90"] # Pick year 1990  

2628 > notna <- !is.na(y)
```

```

2629 > y <- y[notna]
2630 > locs <- bbsdata$counts[notna,c("lon","lat")]
2631 > sz <- y/max(y)
2632
2633 > par(mar=c(3,3,3,6))
2634 > plot(locs,pch=" ",axes=FALSE,xlim=range(locs[,1])+c(-.3,+.3),
2635   ylim=c(range(locs[,2]) + c(-.6,.6)), xlab=" ",ylab=" ")
2636 > map('state', regions='pennsylvania', add=TRUE, lwd=2)
2637 > spatial.plot(bbsdata$counts[notna,2:3], y, cx=1+sz*6, add=TRUE)

```

2638 We can ponder the potential effects that might lead to dove counts being high - corn
 2639 fields, telephone wires, barn roofs along with misidentification of pigeons, these could all
 2640 correlate reasonably well with the observed count of mourning doves. Unfortunately we
 don't have any of that information. However, we do have a measure of forest cover (pro-

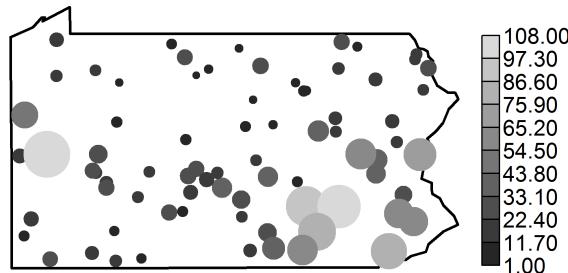


Figure 3.4. Mourning dove counts along North American Breeding Bird Survey routes in Pennsylvania (year = 1990). Plot symbol shading and circle size is proportional to raw count.

```

2641 vided in the data frame bbsdata$habitat) which can be plotted using the spatial.plot
2642 function with the following R commands
2643

```

```

2644 > habdata <- bbsdata$habitat
2645 > map('state',regions="penn",lwd=2)
2646 > I <- matrix(NA, nrow=30, ncol=40)
2647 > I <- matrix(habdata[,"dfor"], ncol=40, byrow=FALSE)
2648 > ux <- unique(habdata[,2])

```

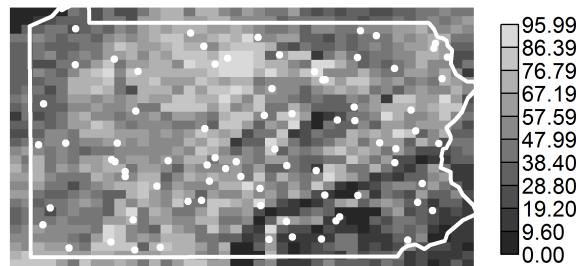


Figure 3.5. Forest cover (percent deciduous) in Pennsylvania. BBS route locations are shown by white dots.

```

2649 > uy <- sort(unique(habdata[,3]))
2650
2651 > par(mar=c(3,3,3,6))
2652 > plot(locs,pch=" ", axes=FALSE, xlim=range(locs[,1])+c(-.3,+.3),
2653   ylim=c(range(locs[,2]) + c(-.6,.6)), xlab=" ",ylab=" ")
2654 > image(ux,uy,rot(I), add=TRUE, col=gray(seq(3,17,,10)/20) )
2655 > map('state', regions='pennsylvania', add=TRUE, lwd=3, col="white")
2656 > image.scale(I, col=gray(seq(3,17,,10)/20) )
2657 > points(locs,pch=20, col="white")

```

2658 The result appears in Fig. 3.5. We see a prominent pattern that indicates high forest
2659 coverage in the central part of the state and low forest cover in the SE. Inspecting the
2660 previous figure of the raw counts suggests a relationship between counts and forest cover
2661 which is perhaps not surprising.

2662 3.6.2 Doing it in WinBUGS

2663 Here we demonstrate how to fit a Poisson GLM in **WinBUGS** using the covariate $x_i =$
2664 forest cover along BBS route i . It is advisable that x_i be standardized in most cases as
2665 this will improve mixing of the Markov chains. We have pre-standardized the forest cover
2666 covariate for the BBS route locations, and so we don't have to worry about that here. To
2667 read the BBS data into **R** and get things set up for **WinBUGS** we issue the following
2668 commands:

```

2669 > library(scrbook)
2670 > data(bbsdata)
2671
2672 > y <- bbsdata$counts[, "X90"] # Pick year 1990
2673 > notna <- !is.na(y)
2674 > y <- y[notna]
2675     ## Forest cover already standardized here:
2676 > habitat <- bbsdata$counts[notna, "habitat"]
2677 > M <- length(y)
2678
2679 > library(R2WinBUGS)           # Load R2WinBUGS
2680 > data <- list (y=y, M=M, habitat=habitat) # Bundle data for WinBUGS

```

2681 Now we write out the Poisson model specification in **WinBUGS** pseudo-code, provide
2682 initial values, identify parameters to be monitored and then execute **WinBUGS**:

```

2683 > cat("
2684 model{
2685   for (i in 1:M){
2686     y[i] ~ dpois(lam[i])
2687     log(lam[i]) <- beta0+beta1*habitat[i]
2688   }
2689   beta0 ~ dunif(-5,5)
2690   beta1 ~ dunif(-5,5)
2691 }
2692 ",file="PoissonGLM.txt")

2693 > inits <- function() list ( beta0=rnorm(1),beta1=rnorm(1) )
2694 > parameters <- c("beta0","beta1")
2695 > out <- bugs(data, inits, parameters, "PoissonGLM.txt", n.thin=2,n.chains=2,
2696   n.burnin=2000,n.iter=6000,debug=TRUE,working.dir=getwd())

```

2697 The **WinBUGS** output can be viewed in **R** using the `print` command:

```

2698 print(out,digits=2)
2699 Inference for Bugs model at "PoissonGLM.txt", fit using WinBUGS,
2700 2 chains, each with 6000 iterations (first 2000 discarded), n.thin = 2
2701 n.sims = 4000 iterations saved
2702      mean    sd   2.5%    25%    50%    75%   97.5% Rhat n.eff
2703 beta0     3.15  0.02   3.10   3.13   3.15   3.17   3.20     1  4000
2704 beta1    -0.50  0.02  -0.54  -0.51  -0.50  -0.48  -0.46     1  4000
2705 deviance 1116.56 1.95 1115.00 1115.00 1116.00 1117.00 1122.00     1  4000

```

2706 3.6.3 Constructing your own MCMC algorithm

2707 At this point it might be helpful to suffer through an example building a custom MCMC
2708 algorithm. Here, we develop an MCMC algorithm for the Poisson regression model, using
2709 a Metropolis-within-Gibbs sampling framework. Building MCMC algorithms is covered in

more detail in Chapt. 17 where you can also find step-by-step instructions for Metropolis-within-Gibbs samplers, should the following section move through all this material too quickly.

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

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

Mathematically, the full conditional is of the form

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

which you can program as an **R** function with arguments β_0 , β_1 and \mathbf{y} without difficulty. The full-conditional for β_1 is:

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

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

Here we will use the so-called random walk candidate generator, which is a Normal proposal distribution, so that, for example, $\beta_0^* \sim \text{Normal}(\beta_0^t, \delta)$ where δ is the standard-deviation of the proposal distribution, which is just a tuning parameter that is set by the user and adjusted to achieve efficient mixing of chains (see Sec. 17.2.2). We remark also that calculations are often done on the log-scale to preserve numerical integrity of things when quantities evaluate to small or large numbers, so keep in mind, for example, $a * b = \exp(\log(a) + \log(b))$ for two positive numbers a and b . The “Metropolis within Gibbs” algorithm for a Poisson regression turns out to be remarkably simple and is given in Panel 3.1. It is also part of the **scrbook** package and you can run 1000 iterations of it by calling `PoisGLMBBS(y=y, habitat=habitat, niter=1000)` (note that y = point count data and $habitat$ = forest cover have to be defined in your **R** workspace as shown in the previous analysis of these data).

The first 300 iterations of the MCMC history of each parameter are shown in Fig. 3.6. These chains are not very appealing but a couple of things are evident: We see that the

```

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

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

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

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

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

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

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

```

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

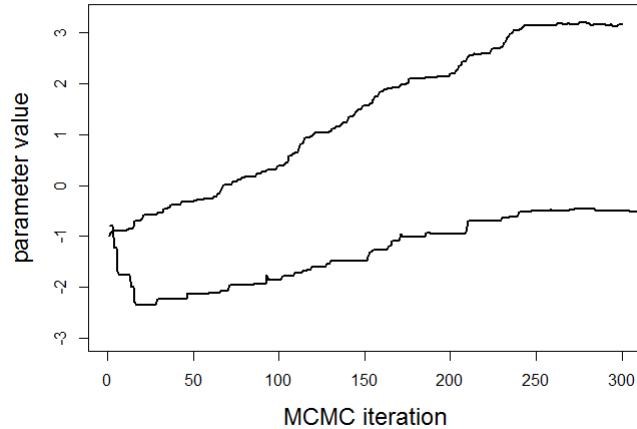


Figure 3.6. First 300 MCMC iterations for the Poisson GLM model parameters β_0 (top) and β_1 (bottom) using a Metropolis-Hastings tuning parameter of $\delta = 0.05$.

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

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

3.7 POISSON GLM WITH RANDOM EFFECTS

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

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

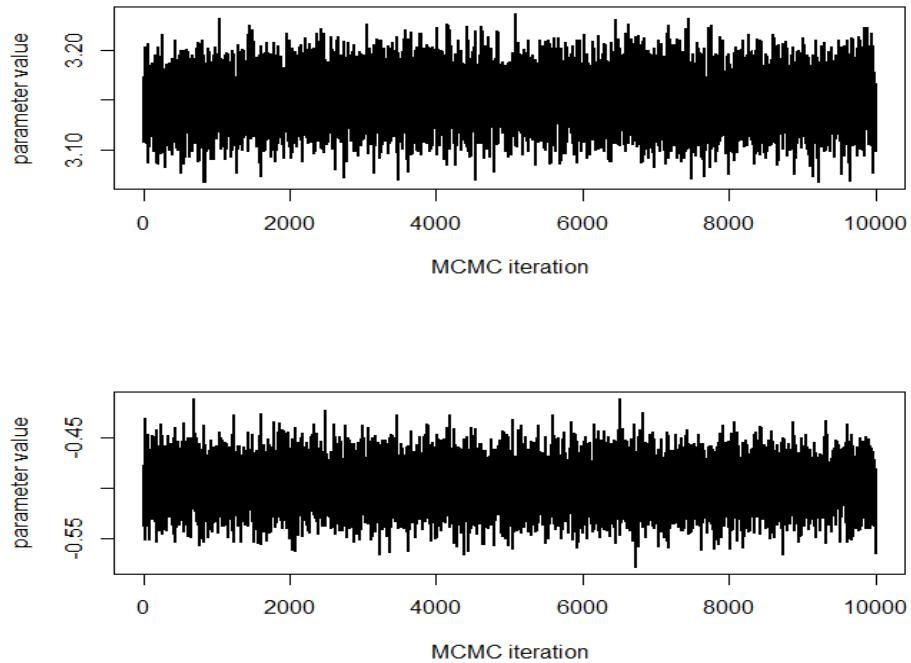


Figure 3.7. Nice grassy plots of 10,000 MCMC iterations for the Poisson GLM model parameters β_0 (top) and β_1 (bottom) using a Metropolis-Hastings tuning parameter of $\delta = 0.05$.

2764 we have:

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

2765 where $\eta_i \sim \text{Normal}(0, \sigma^2)$. In this context, η could represent an error term capturing
 2766 variation in λ_i not accounted for by the covariates, or overdispersion. It is really amazingly
 2767 simple to express this model in the **BUGS** language and have **WinBUGS** (or **JAGS**,
 2768 etc..) draw samples from the posterior distribution. The code for analysis of the BBS
 2769 dove counts is given as follows:

```

2770 > library(scrbook)
2771 ### Grab the BBS Data as before
2772 > data(bbsdata)
2773 ### Set random seed so that results are repeatable
2774 > set.seed(2013)
2775 ### Dump the BUGS model into a file

```

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

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

```

2776 > cat("
2777 model{
2778   for (i in 1:M){  # Observation model, linear predictor, etc..
2779     y[i] ~ dpois(lam[i])
2780     log(lam[i]) <- beta0+ beta1*habitat[i] + eta[i]
2781     frog[i] <- beta1*habitat[i] + eta[i]
2782     eta[i] ~ dnorm(0,tau)
2783   }
2784           # Prior distributions:
2785   beta0 ~ dunif(-5,5)
2786   beta1 ~ dunif(-5,5)
2787   sigma ~ dunif(0,10)
2788   tau <- 1/(sigma*sigma)
2789 }
2790 ",file="model.txt")

2791 > data <- list ("y","M","habitat") # Define the data
2792 > inits <- function()           #      inits and parameters
2793   list ( beta0=rnorm(1), beta1=rnorm(1), sigma=runif(1,0,4))
2794 > parameters <- c("beta0","beta1","sigma","tau")

2795 > library(R2WinBUGS)          # Load and run R2WinBUGS
2796 > out <- bugs (data, inits, parameters, "model.txt", n.thin=2,n.chains=2,
2797   n.burnin=1000, n.iter=5000, debug=TRUE)

```

2799 This produces the posterior summary statistics given in table 3.1. One thing we notice
2800 is that the posterior standard deviations of the regression parameters are much higher,
2801 a result of the extra-Poisson variation allowed for by this model. We would also notice
2802 much less precise predictions of hypothetical new observations.

3.8 BINOMIAL GLMS

2803 Another extremely important class of models in ecology are binomial models. We use
2804 binomial models for count data whenever the observations are counts or frequencies and

it is natural to condition on a “sample size”, say K , the maximum frequency possible in a sample. The random variable, $y \leq K$, is then the frequency of occurrences out of K “trials”. The parameter of the binomial models is p , often called “success probability” which is related to the expected value of y by $\mathbb{E}(y) = pK$. Usually we are interested in modeling covariates that affect the parameter p , and such models are called binomial GLMs, binomial regression models or logistic regression, although logistic regression really only applies when the logistic link is used to model the relationship between p and covariates (see below).

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

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

3.8.1 Binomial regression

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

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

so that if you have covariates in your linear predictor for $\mathbb{E}(y)$ under a Poisson model then they are linear on the complementary log-log link of p . In models of species occurrence it seems natural to view occupancy as being derived from local abundance N (Royle and Nichols, 2003; Royle and Dorazio, 2006; Dorazio, 2007). Therefore, models of local abundance in which $N_i \sim \text{Poisson}(A_i \lambda_i)$ for a habitat patch of area A_i implies a model

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

2844 for occupancy ψ_i of the form

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

2845 We will use the cloglog link in some analyses of SCR models in Chapt. 5 and elsewhere.

2846 3.8.2 Example: waterfowl banding data

2847 The standard binomial modeling problem in ecology is that of modeling species distributions, where $K = 1$ and the outcome is occurrence ($y = 1$) or not ($y = 0$) of some 2848 species. Such examples abound in books (e.g., Royle and Dorazio (2008, ch. 3); Kéry 2849 (2010, ch. 21); Kéry and Schaub (2012, ch. 13)) and in the literature. Therefore, instead, 2850 we will consider an example involving band returns of waterfowl in the upper great plains 2851 including some Canadian provinces, which were analyzed by Royle and Dubovsky (2001). 2852

2853 For these data, y_{it} is the number of mallard (*Anas platyrhynchos*) bands recovered out 2854 of B_{it} birds banded at some location s_i in year t . In this case B_{it} is fixed. Thinking about 2855 recovery rate as being proportional to harvest rate, we use these data to explore geographic 2856 gradients in recovery rate resulting from variability in harvest pressure experienced by 2857 different populations. As such, we fit a basic binomial GLM with a linear response to 2858 geographic coordinates (including an interaction term). Here we provide the part of the 2859 script for creating the model and fitting the model in **WinBUGS**. There are few structural 2860 differences between this model and the Poisson GLM fitted previously. The main things 2861 are due to the data structure (we have a matrix here instead of a vector) and otherwise 2862 we change the distributional assumption to binomial (specified with `dbin`) and then use 2863 the `logit` function to relate the parameter p_{it} to the covariates.

2864 **Dummy variables in BUGS:** In the mallard example, we model the band recovery 2865 probability p_{it} not only as a linear function (on the logit scale) of geographic location, but 2866 also allow for variation in p_{it} with year, t ; $t = 1, 2, \dots, T$. In this particular example there 2867 are $T = 5$ years of data and we could describe the full mallard model with a formula in 2868 terms of “dummy variables.” Dummy variables are binary variables, one variable for each 2869 level of the categorical variable they describe, such that variable for level t takes on the 2870 value 1 if the observation belongs with level t and 0 otherwise. So, the mallard model in 2871 terms of dummy variables for “year” looks like this:

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

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

2872 Here, x_2 to x_5 are the dummy variable vectors of length T that take on the value of 1 2873 when t corresponds to the respective year and 0 otherwise; β_0 is the common intercept 2874 term and corresponds to $t = 1$; $\beta_1 - \beta_4$ describe the difference in p_{it} for each t relative to 2875 $t = 1$.

2876 There is a more concise way of implementing such a model with a categorical covariate 2877 in **BUGS**, namely, by using indexing instead of dummy variables⁸. Essentially, instead of

2878⁸Actually, in some cases a model may mix or converge better depending on whether you choose 2879 a dummy variable or an indexing description of it, although they are structurally equivalent (Kéry, 2880 2010)

estimating the difference in p relative to category 1, we estimate a separate intercept term for each category, so that we have 5 different β_0 parameters indexed by t . This reduces the linear predictor to:

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

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

```

2883 > library(scrbook)
2884 > data(mallard)      # Load mallard data
2885
2886 > cat("
2887 model{
2888   for(t in 1:5){
2889     for (i in 1:nobs){
2890       y[i,t] ~ dbin(p[i,t], B[i,t])
2891       pl[i,t] <- beta0[t]+beta1*X[i,1]+beta2*X[i,2]+beta3*X[i,1]*X[i,2]
2892       p[i,t] <- exp(pl[i,t])/(1+exp(pl[i,t]))
2893     }
2894   }
2895   beta1 ~ dnorm(0,.001)
2896   beta2 ~ dnorm(0,.001)
2897   beta3 ~ dnorm(0,.001)
2898   for(t in 1:5){
2899     beta0[t] ~ dnorm(0,.001)
2900   }
2901 }
2902 ",file="BinomialGLM.txt")

2903 > library(R2WinBUGS)
2904 > data <- list(B=mallard$bandings, y=mallard$recoveries,
2905   X=mallard$locs, nobs=nrow(mallard$locs))
2906 > inits <- function(){ list(beta0=rnorm(5),beta1=0,beta2=0,beta3=0) }
2907 > parms <- list('beta0','beta1','beta2','beta3')
2908 > out <- bugs(data, inits, parms,"BinomialGLM.txt", n.chains=3,
2909   n.iter=2000, n.burnin=1000, n.thin=2, debug=TRUE)

```

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

3.9 BAYESIAN MODEL CHECKING AND SELECTION

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

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

Parameter	Mean	SD	2.5%	50%	97.5%	Rhat	n.eff
beta0[1]	-2.346	0.036	-2.417	-2.346	-2.277	1.001	1500
beta0[2]	-2.356	0.032	-2.420	-2.356	-2.292	1.001	1500
beta0[3]	-2.220	0.035	-2.291	-2.219	-2.153	1.001	1500
beta0[4]	-2.144	0.039	-2.225	-2.143	-2.068	1.000	1500
beta0[5]	-1.925	0.034	-1.990	-1.924	-1.856	1.004	570
beta1	-0.023	0.003	-0.028	-0.023	-0.018	1.001	1500
beta2	0.020	0.006	0.009	0.020	0.031	1.001	1500
beta3	0.000	0.001	-0.002	0.000	0.002	1.001	1500
deviance	1716.001	4.091	1710.000	1715.000	1726.000	1.001	1500

2917 We're against dogma on these issues and think people need to be open-minded about
 2918 such things and recognize that models can be useful whether or not they pass certain
 2919 statistical tests. Some models are intrinsically better than others because they make more
 2920 biological sense or foster understanding or achieve some objective that some bootstrap or
 2921 other goodness-of-fit test can't decide for you. That said, it gives you some confidence if
 2922 your model seems adequate in a purely statistical sense. We provide a very brief overview
 2923 of concepts here, but provide more detailed coverage in Chapt. 8. See also coverage of
 2924 these topics in Kéry (2010) and Link and Barker (2010) for specific context related to
 2925 Bayesian model checking and selection.

2926 3.9.1 Goodness-of-fit

2927 Goodness-of-fit testing is an important element of any analysis because our model repre-
 2928 presents a general set of hypotheses about the ecological and observation processes that
 2929 generated our data. Thus, if our model "fits" in some statistical or scientific sense, then
 2930 we believe it to be consistent with the hypotheses that went into the model. More for-
 2931 mally, we would conclude that the data are *not inconsistent* with the hypotheses, or that
 2932 the model appears adequate. If we have enough data, then of course we will reject any
 2933 set of statistical hypotheses. Conversely, we can always come up with a model that fits
 2934 by making the model extremely complex. Despite this paradox, it seems to us that sim-
 2935 ple models that you can understand should usually be preferred even if they don't fit,
 2936 for example if they embody essential mechanisms central to our understanding of things,
 2937 or if we think that some contributing factors to lack-of-fit are minor or irrelevant to the
 2938 scientific context and intended use of the model. In other words, models can be useful
 2939 irrespective of whether they fit according to some formal statistical test of fit. Yet the
 2940 tension is there to obtain fitting models, and this comes naturally at the expense of models
 2941 that can be easily interpreted and studied and effectively used. Unfortunately, conduct-
 2942 ing a goodness-of-fit test is not always so easy to do. And, moreover, it is never really
 2943 easy (or especially convenient) to decide if your goodness-of-fit test is worth anything. It

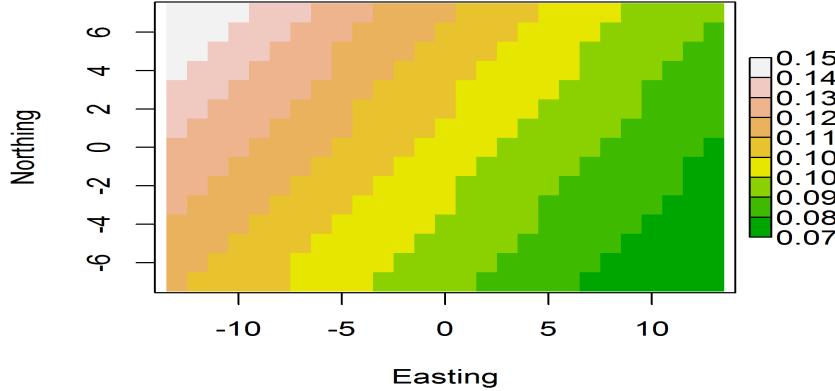


Figure 3.8. Predicted recovery rates of mallard bands in the upper great plains of North America. Note the negative gradient from the NW to the SE.

2944 might have 0 power! Despite this, we recommend attempting to assess model fit in real
 2945 applications, as a general rule, and we provide some basic guidance here and some more
 2946 specific to SCR models in Chapt. 8.

2947 To evaluate goodness-of-fit in Bayesian analyses, we will most often use the Bayesian
 2948 p-value (Gelman et al., 1996). The basic idea is to define a fit statistic or “discrepancy
 2949 measure” and compare the posterior distribution of that statistic to the posterior predictive
 2950 distribution of that statistic for hypothetical perfect data sets for which the model is known
 2951 to be correct. For example, with count frequency data, a standard measure of fit is the
 2952 sum of squares of the “Pearson residuals”,

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

2953 The fit statistic based on the squared residuals computed from the observations is

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

2954 which can be computed at each iteration of a MCMC algorithm given the current values
 2955 of parameters that determine the response distribution. At the same time (i.e., at each
 2956 MCMC iteration), the equivalent statistic is computed for a “new” data set, say \mathbf{y}^{new} ,
 2957 simulated using the current parameter values. From the new data set, we compute the
 2958 same fit statistic:

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

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

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

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

3.9.2 Model selection

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

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

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

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

This approach seems to even work sometimes with fairly complex hierarchical models of a certain form. E.g., Royle (2008) applied it to a random effects model to evaluate the importance of the random effect component of the model. The main problem, which is really a general problem in Bayesian model selection, is that its effectiveness and results will typically be highly sensitive to the prior distribution on the structural parameters (e.g., see Royle and Dorazio (2008, table 3.6)). The reason for this is obvious: If $I_1 = 0$ for the current iteration of the MCMC algorithm, so that β is sampled from the prior distribution, and the prior distribution is very diffuse, then extreme values of β are likely.

2999 Consequently, when the current value of β is far away from the mass of the posterior when
3000 $I_1 = 1$, then the Markov chain may only jump from $I_1 = 0$ to $I_1 = 1$ infrequently. One
3001 seemingly reasonable solution to this problem is to fit the full model to obtain posterior
3002 distributions for all parameters, and then use those as prior distributions in a “model
3003 selection” run of the MCMC algorithm (Aitkin, 1991). This seems preferable to more-or-
3004 less arbitrary restriction of the prior support to improve the performance of the MCMC
3005 algorithm.

3006 A third method that we advocate is subject-matter context. It seems that there are
3007 some situations – some models – where one should not have to do model selection because a
3008 specific model may be necessitated by the biological context of the problem, thus rendering
3009 a formal hypothesis test pointless (Johnson, 1999). Certain aspects of SCR models are
3010 such an example. In SCR models, we will see that “spatial location” of individuals is
3011 an element of the model. The simpler, reduced, model is an ordinary capture-recapture
3012 model which is not spatially explicit (i.e., Chapt. 4), but it seems silly and pointless to
3013 think about actually using the reduced model even if we could concoct some statistical
3014 test to refute the more complex model. The simpler model is manifestly wrong but, more
3015 importantly, not even a plausible data-generating model! Other examples are when effort,
3016 area or sample rate is used as a covariate. One might prefer to have such things in models
3017 regardless of whether or not they pass some statistical litmus test.

3018 Many problems can be approached using one of these methods. In later chapters
3019 (especially Chapt. 8) we will address model selection in specific contexts and we hope
3020 those will prove useful for a majority of the situations you might encounter.

3.10 SUMMARY AND OUTLOOK

3021 GLMs and GLMMs are the most useful statistical methods in all of ecology. The prin-
3022 ciples and procedures underlying these methods are relevant to nearly all modeling and
3023 analysis problems in every branch of ecology. Therefore, understanding how to analyze
3024 these models is an essential skill for the quantitative ecologist to possess. If you under-
3025 stand and can conduct classical likelihood and Bayesian analysis of Poisson and binomial
3026 GL(M)Ms, then you will be successful analyzing and understanding more complex classes
3027 of models that arise. We will see shortly that spatial capture-recapture models are a
3028 type of GL(M)M and thus having a basic understanding of the conceptual origins and
3029 formulation of GL(M)Ms and their analysis is extremely useful.

3030 We note that GL(M)Ms are routinely analyzed by likelihood methods but we have
3031 focused on Bayesian analysis here in order to develop the tools that are less familiar
3032 to most ecologists, and that we will apply in much of the remainder of the book. In
3033 particular, Bayesian analysis of models with random effects is relatively straightforward
3034 because the models are easy to analyze conditional on the random effect, using MCMC.
3035 Thus, we will often analyze SCR models in later chapters by MCMC, explicitly adopting a
3036 Bayesian inference framework. In that regard, the various **BUGS** engines (**WinBUGS**,
3037 **OpenBUGS**, **JAGS**; see also Appendix 1) are enormously useful because they provide
3038 an accessible platform for carrying out analyses by MCMC by just describing the model,
3039 and not having to worry about how to actually build MCMC algorithms. That said, the
3040 **BUGS** language is more important than just to the extent that it enables one to do
3041 MCMC - it is useful as a modeling tool because it fosters understanding, in the sense

3042 that it forces you to become intimate with your model. You have to think about and
3043 write down all of the probability assumptions, and the relationships between observations
3044 and latent variables and parameters in a way that is ecologically sensible and statistically
3045 coherent. Because of this, it focuses your thinking on *model construction*, as M. Kéry says
3046 in his **WinBUGS** book (Kéry, 2010), “**WinBUGS** frees the modeler in you.”

3047 While we have emphasized Bayesian analysis in this chapter, and make primary use of
3048 it through the book, we will provide an introduction to likelihood analysis in Chapt. 6
3049 and use those methods also from time to time. Before getting to that, however, it will be
3050 useful to talk about more basic, conventional closed population capture-recapture models
3051 and such models are the topic of the next chapter.

3052
3053

4

3054

CLOSED POPULATION MODELS

3055 In this chapter we introduce ordinary *non-spatial* capture-recapture (CR) models for es-
3056 timating population size in closed populations. A closed population is one whose size, N ,
3057 does not change during the study. Two forms of closure are often discussed: demographic
3058 closure, meaning that no births or deaths occur, and geographic closure, which states
3059 that no individuals move onto or off of the sampled area during the study. Although few
3060 populations are actually closed except during very short time intervals, closed population
3061 CR models serve as the basis for the development of the rest of the models presented in
3062 this book, including the models for open populations discussed in Chapt. 16.

3063 We begin with the most basic capture-recapture model, colloquially referred to as
3064 “model M_0 ” (Otis et al., 1978), in which encounter probability is strictly constant in all
3065 respects (across individuals, and replicates). This allows us to highlight the basic structure
3066 of closed population models as binomial GLMs. We then consider some important exten-
3067 sions of ordinary closed population models that accommodate various types of “individual
3068 effects” — either in the form of explicit, observed covariates (sex, age, body mass) or
3069 unstructured “heterogeneity” in the form of an individual random effect, which represent
3070 unobserved or unmeasured covariates. A special type of individual covariate models is dis-
3071 tance sampling, which could be thought of as the most primitive spatial capture-recapture
3072 model. All of these different types of closed population models are closely related to bi-
3073 nomial (or logistic) regression-type models. In fact, when N is known, they are precisely
3074 logistic regression models.

3075 We emphasize Bayesian analysis of capture-recapture models and we accomplish this
3076 using a method related to classical “data augmentation” from the statistics literature (e.g.,
3077 Tanner and Wong, 1987). This is a general concept in statistics but, in the context of
3078 capture-recapture models where N is unknown, it has a consistent implementation across
3079 classes of capture-recapture models and one that is really convenient from the standpoint
3080 of doing MCMC (Royle et al., 2007; Royle and Dorazio, 2012). We use data augmentation
3081 throughout this book and thus emphasize its conceptual and technical origins and demon-
3082 strate applications to closed population models. We refer the reader to Kéry and Schaub
3083 (2012, ch. 6) for an accessible and complementary development of Bayesian analysis of

3084 ordinary, i.e., nonspatial closed population models.

4.1 THE SIMPLEST CLOSED POPULATION MODEL: MODEL M_0

3085 To start looking at the simplest capture-recapture model, let's suppose there exists a pop-
 3086 ulation of N individuals which we subject to repeated sampling, say over K "occasions",
 3087 such as trap nights, where individuals are captured, marked, released, and subsequently
 3088 recaptured. We suppose that individual encounter histories are obtained, and these are of
 3089 the form of a sequence of 0's and 1's indicating capture ($y = 1$) or not ($y = 0$) during any
 3090 sampling occasion. As an example, suppose $K = 5$ sampling occasions, then an individual
 3091 captured during occasion 2 and 3 but not otherwise would have an encounter history of
 3092 the form $\mathbf{y} = (0, 1, 1, 0, 0)$. Thus, the observation \mathbf{y}_i for each individual ($i = 1, 2, \dots, N$)
 3093 is a vector having elements denoted by y_{ik} for $k = 1, 2, \dots, K$. Usually this is organized
 3094 as a row of a matrix with elements y_{ik} , see Table 4.1. Except where noted explicitly,
 3095 we suppose that observations are independent within individuals and among individuals.
 3096 Formally, this allows us to say that y_{ik} are independent and identically distributed ("iid")
 3097 Bernoulli random variables and we may write $y_{ik} \sim \text{Bernoulli}(p)$. Consequently, for this
 3098 very simple model in which p is constant (i.e., there are no individual or temporal co-
 3099 variates that affect p) the original binary detection variables can be aggregated into the
 3100 total number of encounters for each individual¹, $y_{i\cdot} = \sum_k y_{ik}$, and the observation model
 3101 changes from a Bernoulli distribution to a binomial distribution based on a sample of size
 3102 K . That is

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

3103 for every individual in the population $i = 1, 2, \dots, N$, where N is the number of individuals
 3104 in the population (i.e., population size).

3105 We emphasize the central importance of the basic Bernoulli encounter model – an
 3106 individual is either encountered in a sample, or not – which forms the cornerstone of
 3107 almost all of classical capture-recapture models, including many spatial capture-recapture
 3108 models discussed in this book.

3109 Evidently, the basic capture-recapture model is a simplistic version of a logistic-
 3110 regression model with only an intercept term ($\text{logit}(p) = \text{constant}$). To say that all
 3111 capture-recapture models are just logistic regressions is a slight over-simplification. In
 3112 fact, we are proceeding here as if we knew N . In practice we don't, of course, and esti-
 3113 mating N is actually the central objective. But, by proceeding as if N were known, we
 3114 can specify a simple model and then deal with the fact that N is unknown using standard
 3115 methods that you are already familiar with (i.e., GLMs - see Chapt. 3).

3116 Assuming individuals in the population are encountered independently, the joint prob-
 3117 ability distribution of the observations is the product of N binomials

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

3118 We emphasize that this expression is conditional on N , in which case we get to observe
 3119 the $y_i = 0$ observations and the resulting data are just iid binomial counts. Because this

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

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

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

3120 is a binomial regression model of the variety described in Chapt. 3, fitting this model
 3121 using a **BUGS** engine poses no difficulty.

3122 Equation 4.1.1 can be simplified even further if we reformat the observations as en-
 3123 counter frequencies. Specifically, let n_k denote the number of individuals captured exactly
 3124 k times after K survey occasions, $n_k = \sum_{i=1}^N I(y_i = k)$ where $I()$ is the indicator func-
 3125 tion evaluating to 1 if its argument is true and 0 otherwise. For sake of illustration, we
 3126 converted the data from Table 4.1 to this format (Table 4.2). What is important to note
 3127 is that if we know N , then we know n_0 , i.e. the number of individuals not captured. In
 3128 this case, an alternative and equivalent expression to Eq. 4.1.1 is

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

3129 where $\pi_k = \Pr(y = k)$ under the binomial model with parameter p and sample size K .
 The essential problem in capture-recapture, however, is that N is *not* known because the

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

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

3130 number of uncaptured individuals (n_0) is unknown. Consequently, the observed capture
 3131 frequencies n_k are no longer independent because n_0 is a function of the other frequencies,
 3132 $n_0 = N - \sum_{k=1}^K n_k$. Hence, their joint distribution is multinomial (e.g., see Illian et al.
 3133 (2008, p. 61)):

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

3135 We gave a general overview of the multinomial distribution in Sec. 2.2. The multino-
 3136 mial distribution is the standard model for discrete responses that can fall into a fixed
 3137 number ($K + 1$ in this case) of possible categories. In the context of capture-recapture,

3138 the multinomial posits a population of N individuals with $K + 1$ possible outcomes de-
 3139 fined by the possible encounter frequencies: encountered $y = 1, 2, \dots, K$ times or not
 3140 encountered at all. These possible outcomes occur with probabilities π_k , which we refer
 3141 to as “cell probabilities” or in the specific context of capture-recapture, encounter history
 3142 probabilities.

3143 To fit the model in which N is *unknown*, we can regard n_0 as a parameter and maximize
 3144 the multinomial likelihood directly. Direct likelihood analysis of the multinomial model is
 3145 straightforward, but that is not always sufficiently useful in practice because we seldom
 3146 are concerned with models for the aggregated encounter history frequencies, which entail
 3147 that capture probabilities are the same for all individuals. In many instances, including
 3148 for spatial capture-recapture (SCR) models, we require a formulation of the model that
 3149 can accommodate individual-level covariates to account for differences in detection among
 3150 individuals, which we address subsequently in this chapter, and also in Chapt. 7.

3151 **4.1.1 The core capture-recapture assumptions**

3152 This basic capture-recapture model – model M_0 – comes with it a host of specific biological
 3153 and statistical assumptions. In addition to the basic assumption of population closure,
 3154 Otis et al. (1978) list the following:

- 3155 1. animals do not lose their marks during the experiment,
- 3156 2. all marks are correctly noted and recorded at each trapping occasion, and
- 3157 3. each animal has a constant and equal probability of capture on each trapping oc-
casional.

3159 The remainder of their classic work is dedicated to relaxing assumption 3. While assump-
 3160 tions 1 and 2 are undoubtedly necessary for inference from basic CR methods to be valid,
 3161 and while they are also assumed by most of the models we present in the following chap-
 3162 ters, we refrain from repeatedly making such statements. Our opinion is that all model
 3163 assumptions are apparent when a model is clearly specified, and it is both redundant and
 3164 impossible to list all the things not allowed by the model. For example, closed population
 3165 models also assume that other sources of error do not occur, but it is not necessary to
 3166 enumerate each possibility. Rather, it is necessary to make clear statements such as

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

3167 This simple model description carries a tremendous amount of information, and it leaves
 3168 very little left to say with respect to assumptions. Although we will not always show
 3169 the *iid* symbol, it will be assumed unless otherwise noted, and this assumption is critical
 3170 for valid inference. It implies that the encounter of one individual does not affect the
 3171 encounter of another individual, and encounter does not affect future encounter. Under
 3172 this assumption, it is easy to write down the likelihood of the parameters and obtain
 3173 parameter estimates; however, whether or not it is true depends upon biological and
 3174 sampling issues. If this assumption is deemed false, the model can be discarded in favor
 3175 of a more realistic alternative. However, once we have settled on our model, statistical
 3176 inference proceeds by assuming the model is truth—not an approximation to truth—but
 3177 actual truth.

3178 In spite of the fact that we assume that all models are truth, but we acknowledge that
 3179 all models are wrong due to their assumptions, assumptions should not be viewed as a
 3180 necessary evil. In fact, one way to view assumptions is as embodiments of our ecological
 3181 hypotheses. If we make these assumptions too complex or too specific, then we will never
 3182 be able to study general phenomena that hold true across space and time. Furthermore,
 3183 in practice, we will rarely have enough data to estimate the parameters of highly complex
 3184 models.

3185 4.1.2 Conditional likelihood

3186 We saw that the closed population model is a simple logistic regression model if N is known
 3187 and, when N is unknown, the model is multinomial with index or sample size parameter
 3188 N . This multinomial model, being conditional on N , is sometimes referred to as the “joint
 3189 likelihood” the “full likelihood” or the “unconditional likelihood” (sometimes “model” in
 3190 place of “likelihood”) (Sanathanan, 1972; Borchers et al., 2002). This formulation differs
 3191 from the so-called “conditional likelihood” approach in which the likelihood of the observed
 3192 encounter histories is devised conditional on the event that an individual is captured at
 3193 least once. To construct this likelihood, we have to recognize that individuals appear
 3194 or not in the sample based on the value of the random variable y_i , that is, if and only
 3195 if $y_i > 0$. The observation model is therefore based on $\Pr(y|y > 0)$. For the simple
 3196 case of model M_0 , the resulting conditional distribution is a “zero truncated” binomial
 3197 distribution which accounts for the fact that we cannot observe the value $y = 0$ in the data
 3198 set. Both the conditional and unconditional models are legitimate modes of analysis in
 3199 all capture-recapture types of studies. They provide equally valid descriptions of the data
 3200 and, for many practical purposes provide equivalent inferences, at least in large sample
 3201 sizes (Sanathanan, 1972).

3202 In this book we emphasize Bayesian analysis of capture-recapture models using data
 3203 augmentation (described in Sec. 4.2 below), which produces yet a third distinct formu-
 3204 lation of capture-recapture models based on the zero-*inflated* binomial distribution that
 3205 we describe in the next section. Thus, there are 3 distinct formulations of the model – or
 3206 modes of analysis – for analyzing all capture-recapture models based on the (1) binomial
 3207 model for the joint or unconditional specification; (2) zero-truncated binomial that arises
 3208 “conditional on n ”; and (3) the zero-inflated binomial that arises under data augmen-
 3209 tation. Each formulation has distinct model parameters (shown in Table 4.3 for model
 3210 M_0).

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

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

4.2 DATA AUGMENTATION

3211 We consider a method of analyzing closed population models using parameter-expanded
 3212 data augmentation (PX-DA), which we abbreviate to “data augmentation” or DA, which
 3213 is useful for Bayesian analysis and, in particular, analysis of models using the various
 3214 **BUGS** engines and other Bayesian model fitting software. Data augmentation is a general
 3215 statistical concept that is widely used in statistics in many different settings. The classical
 3216 reference is Tanner and Wong (1987), but see also Liu and Wu (1999). Data augmentation
 3217 can be adapted to provide a very generic framework for Bayesian analysis of capture-
 3218 recapture models with unknown N . This idea was introduced for closed populations by
 3219 Royle et al. (2007), and has subsequently been applied to a number of different contexts
 3220 including individual covariate models (Royle, 2009), open population models (Royle and
 3221 Dorazio, 2008, 2012; Gardner et al., 2010a), spatial capture-recapture models (Royle and
 3222 Young, 2008; Royle et al., 2009a; Gardner et al., 2009), and many others. Kéry and Schaub
 3223 (2012, Chaps. 6 and 10) provide a good introduction to data augmentation in the context
 3224 of closed and open population models.

3225 Conceptually, the technique of data augmentation represents a reparameterization
 3226 of the “complete data” model – i.e., that conditional on N . The reparameterization
 3227 is achieved by embedding this data set into a larger data set having $M > N$ “rows”
 3228 (individuals) and re-expressing the model conditional on M instead of N . The great thing
 3229 about data augmentation is that we do not need to know N for this reparameterization.
 3230 Although this has a whiff of arbitrariness or even outright ad hockery to it, in the choice
 3231 of M , it is always possible, in practice, to choose M pretty easily for a given problem and
 3232 context and results will be insensitive to choice of M^2 . Then, under data augmentation,
 3233 analysis is focused on the “augmented data set.” That is, we analyze the bigger data set -
 3234 the one having M rows - with an appropriate model that accounts for the augmentation.
 3235 This is achieved by a Bernoulli sampling process that determines whether an individual
 3236 in M is also a member of N . Inference is focused directly on estimating the proportion
 3237 $\psi = E[N]/M$, instead of directly on N , where ψ is the “data augmentation parameter.”

3238 4.2.1 DA links occupancy models and closed population models

3239 There is a close correspondence between so-called “occupancy” models and closed popu-
 3240 lation models (see Royle and Dorazio, 2008, Sec. 5.6). In occupancy models (MacKenzie
 3241 et al., 2002; Tyre et al., 2003) the sampling situation is that M sites, or patches, are sam-
 3242 pled multiple times to assess whether a species occurs at the sites. This yields encounter
 3243 data such as that illustrated in the left panel of Table 4.4. The important problem is that
 3244 a species may occur at a site, but go undetected, yielding an all-zero encounter history for
 3245 the site, which in the case of occupancy studies, are *observed*. However, some of the zero
 3246 vectors will typically correspond to sites where the species in fact *does* occur. Thus, while
 3247 the zeros are observed, there are too many of them and, in a sense, the inference problem
 3248 is to partition the zeros into “structural” (fixed) and “sampling” (or stochastic) zeros,
 3249 where the former are associated with unoccupied sites and the latter with occupied sites
 3250 where the species went undetected. More formally, inference is focused on the parameter
 3251 ψ , the probability that a site is occupied.

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

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

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

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

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

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

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

3289 and we understand this to mean, if $z_i = 0$, then y_i is necessarily 0 because its success
 3290 probability is $z_i p = 0$.

3291 Note that, under data augmentation, N is no longer an explicit parameter of this
 3292 model. In its place, we estimate ψ and functions of the latent variables z . In particular,
 3293 under the assumptions of the zero-inflated model, $z_i \stackrel{iid}{\sim} \text{Bernoulli}(\psi)$; therefore, N is a
 3294 function of these latent variables:

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

3295 Further, we note that the latent z_i parameters *can be* removed from the model by inte-
 3296 gration, in which case the joint probability of the data is

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

3297 Interpreted as a likelihood, we can directly maximize this expression to obtain the MLEs of
 3298 the structural parameters ψ and p or those of other more complex models (e.g., see Royle,
 3299 2006). We could estimate these parameters and then use them to obtain an estimator of
 3300 N using the so-called “Best unbiased predictor” (see Royle and Dorazio, 2012). Normally,
 3301 however, we will analyze the model in its “conditional-on- z ” form using methods of MCMC
 3302 either in the **BUGS** engines or using our own MCMC algorithms (see Chapt. 17).

3303 4.2.2 Model M_0 in BUGS

3304 It is helpful to understand data augmentation by seeing what its effect is on implementing
 3305 model M_0 . For this model, in which we can aggregate the encounter data to individual-
 3306 specific encounter frequencies, the augmented data are given by the vector of frequencies
 3307 $(y_1, \dots, y_n, 0, 0, \dots, 0)$ where the augmented values of $y = 0$ represent the encounter fre-
 3308 quency for potential individuals y_{n+1}, \dots, y_M . The zero-inflated model of the augmented
 3309 data combines the model of the latent variables, $z_i \sim \text{Bernoulli}(\psi)$. The **BUGS** model
 3310 description of the closed population model M_0 is shown in Panel 4.1. The last line of the
 3311 model specification provides the expression for computing N from the data augmentation
 3312 variables z_i . Note that, to improve readability of code snippets (especially of large ones),
 3313 we will sometimes deviate from our standard notation a bit. In this case we use **nind**
 3314 for n (the number of encountered individuals), and $M = nind + nz$ is the total size of the
 3315 augmented data set. In other cases we might also use **nocc** in place of K and **ntraps**
 3316 in place of J . We find that word definitions make code easier to understand, especially
 3317 without having to read surrounding text.

3318 Specification of a more general model in terms of the individual encounter observations
 3319 y_{ik} is not much more difficult than for the individual encounter frequencies. We define
 3320 the observation model by a double loop and change the indexing of quantities accordingly,
 3321 i.e.,

```
3322 for(i in 1:(nind+nz)){
  3323   z[i] ~ dbern(psi)
  3324   for(k in 1:K){
    3325     mu[i,k] <- z[i]*p
```

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

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

```

3326     y[i,k] ~ dbin(mu[i,k],1)
3327   }
3328 }

```

3329 In this manner, it is straightforward to incorporate covariates on p for both individuals
3330 and sampling occasions (see discussion of this below and also Chapt. 7) as well as to devise
3331 other extensions of the model, including models for open populations (see Chapt. 16).

3332 4.2.3 Formal development of data augmentation (DA)

3333 Use of parameter-expanded data augmentation (PX-DA), or DA for short, for solving
3334 inference problems with unknown N can be justified as originating from the choice of a
3335 uniform prior on N . The Uniform(0, M) prior for N is innocuous in the sense that the
3336 posterior associated with this prior is equal to the likelihood for sufficiently large M . One
3337 way of inducing the Uniform(0, M) prior on N is by assuming the following hierarchical
3338 prior:

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

```

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

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

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

```

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

3339 The model assumptions, specifically the multinomial model (Eq. 4.1.3) and Eq. 4.2.2, may
 3340 be combined to yield a reparameterization of the conventional model that is appropriate
 3341 for the augmented data set of known size M :

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

3342 This expression arises by removing N from Eq. 4.1.3 by integrating over the binomial
 3343 prior distribution for N . Thus, the models we analyze under data augmentation arise
 3344 formally by removing the parameter N from the ordinary closed-population model, which
 3345 is conditional on N , by integrating over a binomial prior distribution for N .

3346 Note that the $M - n$ unobserved individuals in the augmented data set have probability
 3347 $\psi\pi(0) + (1 - \psi)$, indicating that these unobserved individuals are a mixture of individuals
 3348 that are sampling zeros ($\psi\pi_0$), and belong to the population of size N , and others that
 3349 are “structural zeros” (occurring in the augmented data set with probability $1 - \psi$). In
 3350 Eq. 4.2.3, N has been eliminated as a formal parameter of the model by marginalization
 3351 (integration) and replaced with the new parameter ψ , the data augmentation parameter.
 3352 However, the full likelihood containing both N and ψ can also be analyzed (see Royle
 3353 et al., 2007).

3354 4.2.4 Remarks on data augmentation

3355 Data augmentation may seem like a strange and mysterious black-box, and likely it is un-
 3356 familiar to most people, even to many of those with substantial experience with capture-

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

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

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

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

4.2.5 Example: Black bear study on Fort Drum

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

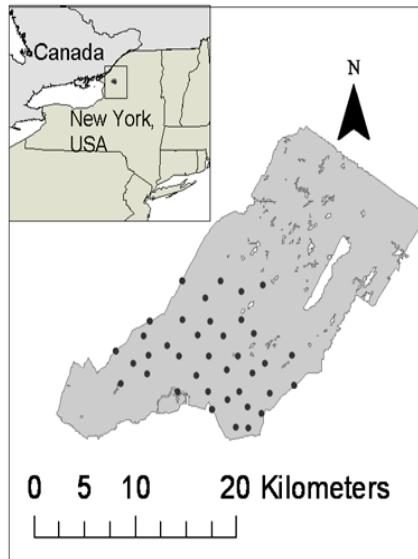


Figure 4.1. Fort Drum Black bear study area and the 38 baited hair snare locations operated for 8 weeks during June and July, 2006.

```

3413 > library(scrbook)
3414 > data(beardata)           # load the bear data and extract components
3415 > trapmat <- beardata$trapmat
3416 > nind <- dim(beardata$bearArray)[1]
3417 > K <- dim(beardata$bearArray)[3]
3418 > ntraps <- dim(beardata$bearArray)[2]
3419

```

```

3420 > M <- 175
3421 > nz <- M-nind
3422 > Yaug <- array(0, dim=c(M,ntraps,K))
3423
3424 > Yaug[1:nind,,] <- beardata$bearArray
3425 > y <- apply(Yaug,c(1,3),sum) # summarize by ind x rep
3426 > y[y>1] <- 1 # toss out multiple encounters per occasion
3427 # b/c traditional CR models ignore space

```

3428 The raw data object, `beardata$bearArray` is a 3-dimensional array $nind \times ntraps \times K$ of individual encounter events (i.e., $y_{ijk} = 1$ if individual i was encountered in trap j during occasion k , and 0 otherwise). For fitting model M_0 (or M_h , see below), it is sufficient to reduce the data to individual encounter frequencies which we have re-labeled “y” above. The **BUGS** model file along with commands to fit the model are as follows:

```

3433 > set.seed(2013) # to obtain the same results each time
3434 > library(R2WinBUGS) # load R2WinBUGS, set-up:
3435 > data0 <- list(y=y, M=M, K=K) # data ....
3436 > params0 <- c('psi','p','N') # parameters ....
3437 > zst <- c(rep(1,nind),rbinom(M-nind, 1, .5)) # inits ....
3438 > inits <- function(){ list(z=zst, psi=runif(1), p=runif(1)) }
3439
3440 > cat("
3441 model{
3442
3443   psi ~ dunif(0, 1)
3444   p ~ dunif(0,1)
3445
3446   for (i in 1:M){
3447     z[i] ~ dbern(psi)
3448     for(k in 1:K){
3449       tmp[i,k] <- p*z[i]
3450       y[i,k] ~ dbin(tmp[i,k],1)
3451     }
3452   }
3453   N<-sum(z[1:M])
3454 }
3455 ",file="modelM0.txt")
3456
3457 ## Run the model:
3458 > fit0 <- bugs(data0, inits, params0, model.file="modelM0.txt",n.chains=3,
3459   n.iter=2000, n.burnin=1000, n.thin=1,debug=TRUE,working.directory=getwd())

```

3460 This produces the following posterior summary statistics:

```

3461 > print(fit0,digits=2)
3462 Inference for Bugs model at "modelM0.txt", fit using WinBUGS,

```

```

3463 3 chains, each with 2000 iterations (first 1000 discarded)
3464 n.sims = 3000 iterations saved
3465      mean    sd   2.5%   25%   50%   75% 97.5% Rhat n.eff
3466 psi     0.29  0.04  0.22  0.26  0.29  0.31  0.36    1 3000
3467 p       0.30  0.03  0.25  0.28  0.30  0.32  0.35    1 3000
3468 N       49.94 1.99 47.00 48.00 50.00 51.00 54.00    1 3000
3469 deviance 489.05 11.28 471.00 480.45 488.80 495.40 513.70    1 3000
3470
3471 [... some output deleted ...]

```

3472 **WinBUGS** did well in choosing an MCMC algorithm for this model – we have $\hat{R} = 1$
 3473 for each parameter, and an effective sample size of 3000, equal to the total number of
 3474 posterior samples³. We see that the posterior mean of N under this model is 49.94 and
 3475 a 95% posterior interval is (48, 54). We revisit these data later in the context of more
 3476 complex models.

3477 In order to obtain an estimate of density, D , we need an area to associate with the
 3478 estimate of N , and in Chapt. 1 we already went through a number of commonly used
 3479 procedures to conjure up such an area, including buffering the trap array by the home range
 3480 radius, often estimated by the mean maximum distance moved (MMDM) (Parmenter
 3481 et al., 2003), 1/2 MMDM (Dice, 1938) or directly from telemetry data (Wallace et al.,
 3482 2003). Typically, the trap array is defined by the convex hull around the trap locations,
 3483 and this is what we applied a buffer to. We computed the buffer by using a telemetry-based
 3484 estimate of the mean female home range radius (2.19 km) (Bales et al., 2005) instead of
 3485 using an estimate based on our relatively more sparse recapture data. For the Fort Drum
 3486 study, the convex hull has an area of 157.135 km², and the buffered convex hull has an
 3487 area of 277.011 km². To create this we used functions contained in the **R** package **rgeos**
 3488 and created a utility function **bcharea** which is in our **R** package **scrbook**. The commands
 3489 are as follows:

```

3490 > library(rgeos)
3491
3492 > bcharea <- function(buff,traplocs){
3493   p1 <- Polygon(rbind(traplocs,traplocs[1,]))
3494   p2 <- Polygons(list(p1=p1),ID=1)
3495   p3 <- SpatialPolygons(list(p2=p2))
3496   p1ch <- gConvexHull(p3)
3497   bp1 <- (gBuffer(p1ch, width=buff))
3498   plot(bp1, col='gray')
3499   plot(p1ch, border='black', lwd=2, add=TRUE)
3500   gArea(bp1)
3501 }
3502
3503 > bcharea(2.19,traplocs=trapmat)

```

3504 The resulting buffered convex hull is shown in Fig. 4.2.

3505 To conjure up a density estimate under model M_0 , we compute the appropriate pos-
 3506 terior summary of the ratio of N and the prescribed area (277.011 km²):

³This is even a little suspicious....

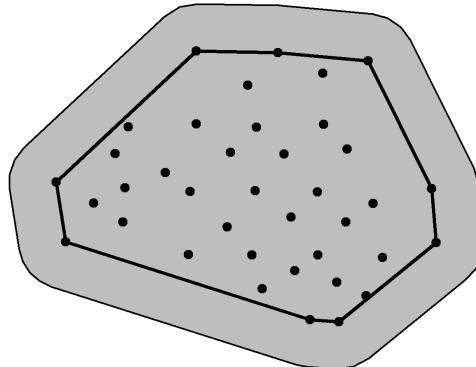


Figure 4.2. Convex hull of the bear hair snare array at Fort Drum, NY, buffered by mean female home range radius (2.19 km).

```

3507 > summary(fit0$sims.list$N/277.011)
3508   Min. 1st Qu. Median Mean 3rd Qu. Max.
3509 0.1697 0.1733 0.1805 0.1803 0.1841 0.2130
3510
3511 > quantile(fit0$sims.list$N/277.011,c(0.025,0.975))
3512   2.5% 97.5%
3513 0.1696684 0.1949381

```

3514 which yields a density estimate of about $0.18 \text{ ind}/\text{km}^2$, and a 95% Bayesian confidence
 3515 interval of $(0.170, 0.195)$. Our estimate of density should be reliable if we have faith in
 3516 our stated value of the “sampled area”. Clearly though this is largely subjective, and not
 3517 something we can formally evaluate (or estimate) from the data based on model M_0 .

4.3 TEMPORALLY VARYING AND BEHAVIORAL EFFECTS

3518 The purpose of this chapter is mainly to emphasize the central importance of the binomial
 3519 model in capture-recapture and so we have considered models for individual encounter
 3520 frequencies—the number of times individuals are captured out of K occasions. Sometimes
 3521 we can’t aggregate the encounter data for each individual, such as when encounter proba-
 3522 bility varies over time among samples. Time-varying responses that are relevant in many

3523 capture-recapture studies are “effort” such as amount of search time, number of observers,
 3524 or trap nights, or encounter probability varying over time, as a function of date or season
 3525 (Kéry et al., 2010) due to species behavior. A common situation in many animal studies
 3526 is that in which there exists a “behavioral response” to trapping (even if the animal is not
 3527 physically trapped).

3528 Behavioral response is an important concept in animal studies because individuals
 3529 might learn to come to baited traps or avoid traps due to trauma related to being encoun-
 3530 tered. There are a number of ways to parameterize a behavioral response to encounter.
 3531 The distinction between persistent and ephemeral was made by Yang and Chao (2005)
 3532 who considered a general behavioral response model of the form:

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

3533 where x_{ik} is a covariate indicator variable of previous capture (i.e., $x_{ik} = 1$ if captured
 3534 in any previous period). Therefore, encounter probability changes depending on whether
 3535 an individual was captured in the immediate previous period (a Markovian or ephemeral
 3536 behavioral response; (Yang and Chao, 2005)), described by the term $\alpha_1 y_{i,k-1}$ or in *any*
 3537 previous period (persistent behavioral response), described by the term $\alpha_2 x_{ik}$. Because
 3538 spatial capture-recapture models allow us to include trap-specific covariates, we can de-
 3539 scribe a 3rd type of behavioral response—a local behavioral response that is trap-specific
 3540 (Royle et al., 2011b). In this local behavioral response, the encounter probability is mod-
 3541 ified for an individual trap depending on previous capture in that trap. Models with
 3542 temporal effects are easy to describe and analyze in the **BUGS** language and we provide
 3543 a number of examples in Chapt. 7 and elsewhere.

4.4 MODELS WITH INDIVIDUAL HETEROGENEITY

3544 Models in which encounter probability varies by individual have a long history in capture-
 3545 recapture and, indeed, this so-called “model M_h ” is one of the elemental capture-recapture
 3546 models in (Otis et al., 1978). Conceptually, we imagine that the individual-specific em-
 3547 counter probability parameters, p_i , are random variables distributed according to some
 3548 probability distribution, $[\theta]$. We denote this basic model assumption as $p_i \sim [\theta]$. This
 3549 type of model is similar in concept to extending a GLM to a GLMM but in the capture-
 3550 recapture context N is unknown. The basic class of models is often referred to as “model
 3551 M_h ” (“h” for heterogeneity), but really this is a broad class of models, each being dis-
 3552 tinguished by the specific distribution assumed for p_i . There are many different varieties
 3553 of model M_h including parametric and various non-parametric approaches (Burnham and
 3554 Overton, 1978; Norris and Pollock, 1996; Pledger, 2004). One important practical matter
 3555 is that estimates of N can be extremely sensitive to the choice of heterogeneity model
 3556 (Fienberg et al., 1999; Dorazio and Royle, 2003; Link, 2003). Indeed, Link (2003) showed
 3557 that in some cases it’s possible to find models that yield precisely the same expected data,
 3558 yet produce wildly different estimates of N . In that sense, N for most practical pur-
 3559 poses is not identifiable across classes of different heterogeneity models, and this should
 3560 be understood before fitting any such model. One solution to this problem is to seek
 3561 to model explicit factors that contribute to heterogeneity, e.g., using individual covariate
 3562 models (See 4.5 below). Indeed, spatial capture-recapture models do just that, by mod-
 3563 eling heterogeneity due to the spatial organization of individuals in relation to traps or

3564 other encounter mechanism. For additional background and applications of model M_h see
 3565 Royle and Dorazio (2008, Chapt. 6) and Kéry and Schaub (2012, Chapt. 6).

3566 We will work with a specific type of model M_h here which is a natural extension of
 3567 the basic binomial observation model of model M_0 so that

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

3568 where μ is a fixed parameter (the mean) to be estimated, and η_i is an individual random
 3569 effect assumed to be normally distributed:

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

3570 We could as well combine these two steps and write $\text{logit}(p_i) \sim \text{Normal}(\mu, \sigma_p^2)$. This
 3571 “logit-normal mixture” was analyzed by Coull and Agresti (1999) and elsewhere. It is
 3572 a natural extension of the basic model with constant p , as a mixed GLMM, and similar
 3573 models occur throughout statistics. It is also natural to consider a beta prior distribution
 3574 for p_i (Dorazio and Royle, 2003) and so-called “finite-mixture” models are also popular
 3575 (Norris and Pollock, 1996; Pledger, 2004). In the latter, individuals are assumed to belong
 3576 to a finite number of latent classes, each of which has its own capture probability.

3577 Model M_h has important historical relevance to spatial capture-recapture situations
 3578 (Karanth, 1995) because investigators recognized that the juxtaposition of individuals with
 3579 the array of trap locations should yield heterogeneity in encounter probability, and thus it
 3580 became common to use some version of model M_h in spatial trapping arrays to estimate
 3581 N . While this doesn’t resolve the problem of not knowing the effective sample area, it
 3582 does yield an estimator that accommodates the heterogeneity in p induced by the spatial
 3583 aspect of capture-recapture studies. To see how this juxtaposition induces heterogeneity,
 3584 we have to understand the relevance of movement in capture-recapture models. Imagine a
 3585 quadrat that can be uniformly searched by a crew of biologists for some species of reptile
 3586 (see Royle and Young (2008)). Figure 4.3 shows a sample quadrat searched repeatedly
 3587 over a period of time. Further, suppose that the species exhibits some sense of spatial
 3588 fidelity in the form of a home range or territory, and individuals move about their home
 3589 range (home range centroids are given by the solid dots) in some kind of random fashion.
 3590 Heuristically, we imagine that each individual in the vicinity of the study area is liable
 3591 to experience variable exposure to encounter due to the overlap of its home range with
 3592 the sampled area - essentially the long-run proportion of times the individual is within
 3593 the sample plot boundaries, say ϕ . We might model the exposure or *availability* of an
 3594 individual to capture by supposing that $a_i = 1$ if individual i is available to be captured
 3595 (i.e., within the survey plot) during any sample, and 0 otherwise. Then, $\Pr(a_i = 1) = \phi$.
 3596 In the context of spatial studies, it is natural that ϕ should depend on *where* an individual
 3597 lives, i.e., it should be individual-specific ϕ_i (Chandler et al., 2011). This system describes,
 3598 precisely, that of “random temporary emigration” (Kendall et al., 1997) where ϕ_i is the
 3599 individual-specific probability of being “available” for capture.

3600 Conceptually, SCR models aim to deal with this problem of variable exposure to sam-
 3601 pling due to movement in the proximity of the trapping array explicitly and formally with
 3602 auxiliary spatial information. If individuals are detected with probability p_0 , *conditional*
 3603 on $a_i = 1$, then the marginal probability of detecting individual i is

$$p_i = p_0 \phi_i$$

3604 so we see clearly that individual heterogeneity in encounter probability is induced as a re-
 3605 sult of the juxtaposition of individuals (i.e., their home ranges) with the sample apparatus
 3606 and the movement of individuals about their home range.

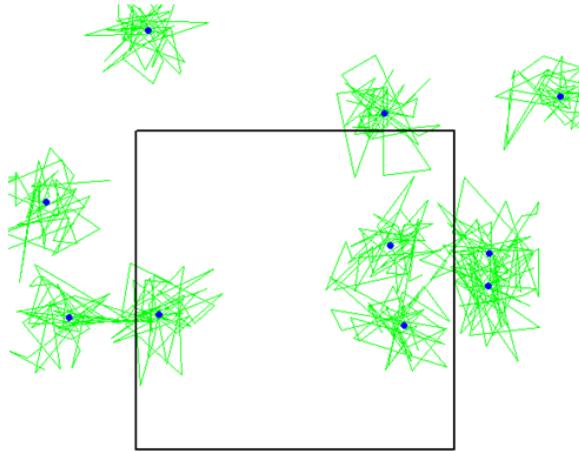


Figure 4.3. A quadrat searched for lizards over some period of time (simulated data). The locations of encounter for each of 10 lizards are connected by lines—the dots are activity centers.

3607 4.4.1 Analysis of model M_h

3608 If N is known, it is worth taking note of the essential simplicity of model M_h as a binomial
 3609 GLMM. This is a type of model that is widely applied throughout statistics using standard
 3610 methods of inference based either on integrated likelihood (Laird and Ware, 1982; Berger
 3611 et al., 1999), which we discuss in Chapt. 6, or standard Bayesian methods. However,
 3612 because N is not known, inference is somewhat more challenging. We address that here
 3613 using Bayesian analysis based on data augmentation. Although we use data augmentation
 3614 in the context of Bayesian methods here, we note that heterogeneity models formulated
 3615 under DA are easily analyzed by conventional likelihood methods as zero-inflated binomial
 3616 mixtures (Royle, 2006) and more traditional analysis of model M_h based on integrated
 3617 likelihood, without using data augmentation, has been considered by Coull and Agresti
 3618 (1999), Dorazio and Royle (2003), and others.

3619 As with model M_0 , we have the Bernoulli model for the zero-inflation variables: $z_i \sim$
 3620 Bernoulli(ψ) and the model of the observations expressed conditional on these latent

3621 variables z_i . For $z_i = 1$, we have a binomial model with individual-specific p_i :

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

3622 and otherwise $y_i | z_i = 0 \sim I(y = 0)$, i.e., a point mass at $y = 0$. Further, we prescribe a
3623 distribution for p_i . Here we assume

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

3624 For prior distributions we assume $p_0 = \text{logit}^{-1}(\mu) \sim \text{Uniform}(0, 1)$ and, for the standard
3625 deviation $\sigma \sim \text{Uniform}(0, B)$ for some large B . Another common default prior is to assume
3626 $\tau = 1/\sigma^2 \sim \text{Gamma}(1, 1)$, although we usually choose $\sigma \sim \text{Uniform}(0, B)$.

3627 4.4.2 Analysis of the Fort Drum data with model M_h

3628 Here we provide an analysis of the Fort Drum bear survey data using the logit-normal
3629 heterogeneity model, and we used data augmentation to produce a data set of $M = 700$
3630 individuals. We have so far mostly used **WinBUGS** but we are now transitioning to
3631 the use of **JAGS** run from within **R** using the useful packages **R2jags** or **rjags**. The
3632 function **jags** from the **R2jags** package runs essentially like the **bugs** function which we
3633 demonstrate here for setting up and running model M_h for the Fort Drum bear data:

```
3634 [...] get data as before ....]
3635
3636 > set.seed(2013)
3637
3638 > cat("
3639 model{
3640   p0 ~ dunif(0,1)           # prior distributions
3641   mup <- log(p0/(1-p0))
3642   sigmap ~ dunif(0,10)
3643   taup <- 1/(sigmap*sigmap)
3644   psi ~ dunif(0,1)
3645
3646   for(i in 1:(nind+nz)){
3647     z[i] ~ dbern(psi)        # zero inflation variables
3648     lp[i] ~ dnorm(mup,taup) # individual effect
3649     logit(p[i]) <- lp[i]
3650     mu[i] <- z[i]*p[i]
3651     y[i] ~ dbin(mu[i],K)    # observation model
3652   }
3653
3654   N<-sum(z[1:(nind+nz)])
3655 }
3656 ",file="modelMh.txt")
3657 > data1 <- list(y=y, nz=nz, nind=nind, K=K)
3658 > params1 <- c('p0','sigmap','psi','N')
```

```

3659 > inits <- function(){ list(z=as.numeric(y>=1), psi=.6, p0=runif(1),
3660   sigmap=runif(1,.7,1.2),lp=rnorm(M,-2)) }
3661 > library(R2jags)
3662 > wabout <- jags(data1, inits, params1, model.file = "modelMh.txt", n.chains = 3,
3663   n.iter = 1010000, n.burnin = 10000, working.directory = getwd())

```

3664 We provide an **R** function `modelMhBUGS` in the package `scrbook` which will fit the
3665 model using either **JAGS** or **WinBUGS** as specified by the user. In addition, for fun,
3666 we construct our own MCMC algorithm using a Metropolis-within-Gibbs algorithm for
3667 model M_h in Chapt. 17, where we also develop MCMC algorithms for spatial capture-
3668 recapture models. Using `modelMhBUGS`, we ran 3 chains of 1 *million* iterations (mixing is
3669 poor for this model and this data set), which produced the posterior distribution for N
3670 shown in Fig. 4.4. Posterior summaries of parameters are given in Table 4.5.

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

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

3671 We used $M = 700$ for this analysis and we note that while the posterior mass of N is
3672 concentrated away from this upper bound (Fig. 4.4), the posterior has an extremely long
3673 right tail, with some MCMC draws at the upper boundary $N = 700$, suggesting that an
3674 even higher value of M may be called for. To characterize the posterior distribution of
3675 density we produce the relevant summaries of the posterior distribution of $D = N/277.11$
3676 (recall the buffered area of the convex hull is 277.11 km^2):

```

3677 > summary(wabout$sims.list$N/277.11)
3678   Min. 1st Qu. Median Mean 3rd Qu. Max.
3679 0.1696 0.2959 0.3681 0.4428 0.4944 2.5260
3680
3681 > quantile(wabout$sims.list$N/277.11,c(0.025,0.50,0.975))
3682   2.5% 50% 97.5%
3683 0.2237379 0.3680849 1.1511674

```

3684 Therefore, the point estimate, characterized by the posterior median, is around 0.37 bears
3685 per square km and a 95% Bayesian credible interval is (0.224, 1.151).

4.4.3 Comparison with MLE

3687 The posterior of N is highly skewed; therefore, we see that the posterior mean ($N = 122.7$)
3688 is considerably higher than the posterior median ($N = 102$). Further, it may be surprising
3689 that these posterior summaries do not compare well with the MLE. We used the **R** code

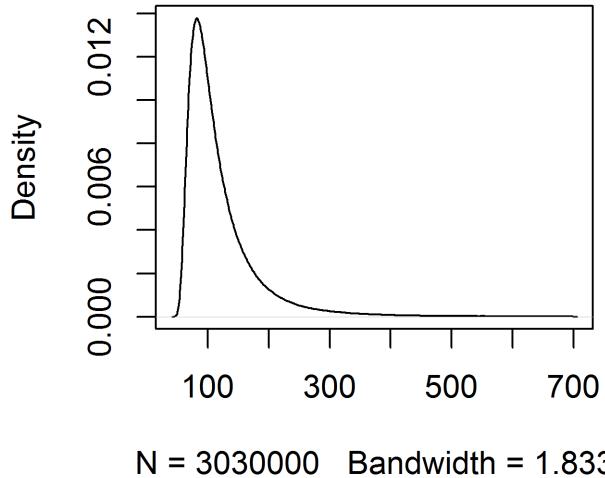


Figure 4.4. Posterior of N for Fort Drum bear study data under the logit-normal version of model M_h .

3690 contained in Panel 6.1 from Royle and Dorazio (2008) to obtain the MLE of $\log(n_0)$,
 3691 the logarithm of the number of uncaptured individuals, is $\widehat{\log(n_0)} = 3.86$ and therefore
 3692 $\hat{N} = \exp(3.86) + 47 = 94.47$, which is larger than the mode shown in Fig. 4.4. To see
 3693 this, we compute the posterior mode, by finding the posterior value of N with the highest
 3694 mass. Because N is discrete, we can use the `table()` function in **R** and find the most
 3695 frequent value⁴. If we want to smooth out some of the Monte Carlo error a bit, we can
 3696 use a smoother of some sort applied to the tabled posterior frequencies of N . Here we use
 3697 a smoothing spline (**R** function `smooth.spline`) with the degree of smoothing chosen by
 3698 cross-validation (the `cv=TRUE` argument):

```
3699 > N <- table(jout$BUGSoutput$sims.list$N)
3700 > xg <- as.numeric(names(N))
3701
3702 > sp <- smooth.spline(xg,N,cv=TRUE)
3703
3704 > sp
```

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

```

3705
3706 Call:
3707 smooth.spline(x = xg, y = N, cv = TRUE)
3708
3709 Smoothing Parameter spar= 0.09339815 lambda= 8.201724e-09 (17 iterations)
3710 Equivalent Degrees of Freedom (Df): 121.1825
3711 Penalized Criterion: 2544481
3712 PRESS: 5903.4

```

3713 We obtain the mode of the smoothed frequencies as follows:

```

3714 sp$x[sp$y==max(sp$y)]
3715 [1] 82

```

3716 We don't dwell too much on the difference between the MLE and features of the posterior, but we do note here that the posterior distribution for the parameters of this model, for the Fort Drum data set, are very sensitive to the prior distributions. In the present case, the use of a Uniform(0, 1) prior for $p_0 = \text{logit}^{-1}(\mu)$ is somewhat informative—in particular, it is not at all “flat” on the scale of μ , and this affects the posterior. We generally always recommend use of a Uniform(0, 1) prior for $\text{logit}^{-1}(\mu)$ in such models. That said, we were surprised at this result, and we experimented with other prior configurations including putting a flat prior on μ directly. This kind of small sample instability has been widely noted in model M_h (Fienberg et al., 1999; Dorazio and Royle, 2003), as has extreme sensitivity to the specific form of model M_h (Link, 2003). In summary, while the mode is well-defined, the data set is relatively sparse and hence inferences are poor and sensitive to model choice.

4.5 INDIVIDUAL COVARIATE MODELS: TOWARD SPATIAL CAPTURE-RECAPTURE

3728 A standard situation in capture-recapture models is when a covariate which is thought
 3729 to influence encounter probability is measured for each individual. These are often called
 3730 “individual covariate models” but, in keeping with the classical nomenclature on closed
 3731 population models, Kéry and Schaub (2012) referred to this class of models as “model
 3732 M_x ” (the x here being an explicit covariate). As with other closed population models, we
 3733 begin with the basic binomial observation model:

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

3734 To model the covariate, we use a logit model for encounter probability of the form:

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

3735 where x_i is the covariate value for individual i and the parameters $\boldsymbol{\alpha} = (\alpha_0, \alpha_1)$ are the
 3736 regression coefficients. Classical examples of covariates influencing detection probability
 3737 are type of animal (juvenile/adult or male/female), a continuous covariate such as body
 3738 mass, or a discrete covariate such as group or cluster size. For example, in models of aerial
 3739 survey data, it is natural to model the detection probability of a group as a function of the
 3740 observation-level individual covariate, “group size” (Royle, 2008; Langtimm et al., 2011).

3741 Model M_x is similar in structure to model M_h , except that the individual effects are
 3742 *observed* for the n individuals that appear in the sample. These models are important
 3743 here because spatial capture-recapture models can be described precisely as a form of
 3744 model M_x , where the covariate describes *where* the individual is located in relation to the
 3745 trapping array. Specifically, SCR models *are* individual covariate models, but where the
 3746 individual covariate is only observed imperfectly (or partially observed) for each captured
 3747 individual. Unlike model M_h , in SCR models (and model M_x) we do have some direct
 3748 information about the latent variable, which comes from the spatial locations/distribution
 3749 of individual recaptures.

3750 Traditionally, estimation of N in model M_x is achieved using methods based on ideas of
 3751 unequal probability sampling (i.e., Horvitz-Thompson estimation⁵; Huggins (1989), Alho
 3752 (1990) and Borchers et al. (2002)). An estimator of N is

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

3753 where \tilde{p}_i is the probability that individual i appeared in the sample. This quantity is
 3754 $\tilde{p}_i = \Pr(y_i > 0)$ and, in closed population capture-recapture models, it can be computed
 3755 as:

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

3756 where p_i is a function of parameters α_0 and α_1 according to Eq. 4.5.1. In practice, pa-
 3757 rameters are estimated from the conditional-likelihood of the observed encounter histories
 3758 which is, for observation y_i ,

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

3759 This derives from a straightforward application of the law of total probability. Conceptually,
 3760 we partition $\Pr(y)$ according to $\Pr(y) = \Pr(y|y > 0)\Pr(y > 0) + \Pr(y|y = 0)\Pr(y = 0)$.
 3761 For any positive value of y the 2nd term is necessarily 0, and so we rearrange to obtain
 3762 $\Pr(y|y > 0) = \Pr(y)/\Pr(y > 0)$ which, in the specific case where $\Pr(y)$ is the binomial
 3763 probability mass function (pmf) produces Eq. 4.5.2.

3764 Here we take a formal model-based approach to Bayesian analysis of such models
 3765 based on the joint likelihood using data augmentation (Royle, 2009). Classical likelihood
 3766 analysis of the so-called “full likelihood” is covered by Borchers et al. (2002). For Bayesian
 3767 analysis of model M_x , because the individual covariate is unobserved for the $n_0 = N - n$
 3768 uncaptured individuals, we require a model to describe variation in x among individuals,
 3769 essentially allowing the sample to be extrapolated to the population. For example, if we
 3770 have a continuous trait measured on each individual, then we might assume that x has a
 3771 normal distribution:

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

3772 Data augmentation can be applied directly to this class of models. In particular, reformu-
 3773 lation of the model under DA yields a basic zero-inflated binomial model of the following

⁵For a quick summary of the idea see:

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

3774 form, for each $i = 1, 2, \dots, M$:

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

3775 Fully spatial capture-recapture models use this formulation with a latent covariate that
 3776 is directly related to the individual detection probability (see next section). As with
 3777 the previous models, implementation is trivial in the **BUGS** language. The **BUGS**
 3778 specification is very similar to that for model M_h , but we require the distribution of the
 3779 covariate to be specified, along with priors for the parameters of that distribution.

3780 **4.5.1 Example: Location of capture as a covariate**

3781 Here we consider a special type of model M_x that is especially relevant to spatial capture-
 3782 recapture. Intuitively, some measure of distance from home range center to traps for an
 3783 individual should be a reasonable covariate to explain heterogeneity in encounter probabil-
 3784 ity, i.e., individuals with more exposure to traps should have higher encounter probabilities
 3785 and vice versa. So we can imagine *estimating* such a quantity, say average distance from
 3786 home range center to “the trap array”, and then using it as an individual covariate in
 3787 capture-recapture models. A version of this idea was put forth by Boulanger and McLel-
 3788 lan (2001) (see also Ivan (2012)), but using the Huggins-Alho estimator and with covariate
 3789 “distance from home range center to edge” of the trapping array, where the home range
 3790 center is estimated by the average capture location. This is intuitively appealing because
 3791 we can imagine, in some kind of an ideal situation where we have a dense grid of traps
 3792 over some geographic region, that the average location of capture would be a decent esti-
 3793 mate (heuristically) of an individual’s home range center. We provide an example of this
 3794 type of approach using a fully model-based analysis of the version of model M_x described
 3795 above, analyzed by data augmentation. We take a slightly different approach than that
 3796 adopted by Boulanger and McLellan (2001). By analyzing the full likelihood and placing
 3797 a prior distribution on the individual covariate, we will resolve the problem of having an
 3798 ill-defined sample area. After you read later chapters of this book, it will be apparent that
 3799 SCR models represent a formalization of this heuristic procedure.

3800 For our purposes here, we define the scalar individual covariate x_i to be the distance
 3801 from the average encounter location of individual i , say \mathbf{s}_i , to the centroid of the trap
 3802 array, \mathbf{x}_0 : $x_i = \|\mathbf{s}_i - \mathbf{x}_0\|$. Note that $\|\mathbf{u}\|$ is standard notation for Euclidean norm or
 3803 magnitude of the vector \mathbf{u} , and we use it throughout the book. In practice, people have
 3804 used distance from edge of the trap array but that is less easy to quantify, as “edge” itself
 3805 is not precisely defined. Conceptually, individuals in the middle of the array should have
 3806 a higher probability of encounter and, as x_i increases, p_i should therefore decrease. We
 3807 note that we have defined \mathbf{s}_i in terms of a sample quantity—the observed mean encounter
 3808 location—which, while ad hoc, is consistent with the use of individual covariate models in
 3809 the literature. For an expansive, dense trapping grid we might expect the sample mean
 3810 encounter location to be a good estimate of home range center but, clearly this is biased
 3811 for individuals that live around the edge (or off) the trapping array.

3812 A key point is that s_i is missing for each individual that is not encountered and so
 3813 x_i is also missing. Therefore, it is a latent variable, and we need to specify a probability
 3814 distribution for it. As a measurement of distance we know it must be positive-valued, and
 3815 it seems sensible that an individual located extremely far from the array of traps would
 3816 not be captured. Therefore, let's assume that x_i is uniformly distributed from 0 to some
 3817 large number, say B , beyond which it would be difficult to imagine an individual being
 3818 captured by the trap array:

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

3819 where B is a specified constant, which we may choose to be arbitrarily large. For example,
 3820 B should be at least a home range diameter past the furthest trap from the centroid of
 3821 the array.

3822 4.5.2 Fort Drum bear study

3823 We have to do a little bit of data processing to fit this individual covariate model to the
 3824 Fort Drum data. We need to compute the individual covariate \mathbf{x}_i (distance from the
 3825 centroid of the trapping array) using the **R** function `spiderplot` provided in `scrbook`.
 3826 This function also produces the keen plot shown in Fig. 4.5 which we call a “spider plot”.
 3827 The **R** commands for obtaining the individual covariate “distance from trap centroid”
 3828 (the variable `xcent` returned by `spiderplot`) and making the spider plot are as follows:

```
3829 > library(scrbook)
3830 > data(beardata)
3831 > toad <- spiderplot(beardata$bearArray,beardata$trapmat)
3832 > xcent <- toad$xcent
```

3833 For the analysis of these data using the individual covariate “distance from centroid”
 3834 we used $x_i \sim \text{Uniform}(0, B)$ with $B = 11.5 \text{ km}^2$, which is about the distance from the
 3835 array center to the furthest trap. Once we choose a value for B , the direct implication is
 3836 that the population size parameter, N , applies to the area within 11.5 units of the trap
 3837 centroid. Therefore, the model associates a precise area within which the population of N
 3838 individuals resides. We will see shortly that N does, in fact, scale with our choice of B to
 3839 reflect the changing area over which the N individuals of the model reside. The **BUGS**
 3840 model specification and **R** commands to package the data and fit the model are as follows:

```
3841 cat("
3842 model{
3843   p0 ~ dunif(0,1)                      # prior distributions
3844   alpha0 <- log(p0/(1-p0))
3845   psi ~ dunif(0,1)
3846   beta ~ dnorm(0,.01)

3847   for(i in 1:(nind+nz)){
3848     xcent[i] ~ dunif(0,B)
3849     z[i] ~ dbern(psi)                    # DA variables
3850     lp[i] <- alpha0 + beta*xcent[i] # individual effect
3851     logit(p[i]) <- lp[i]
```

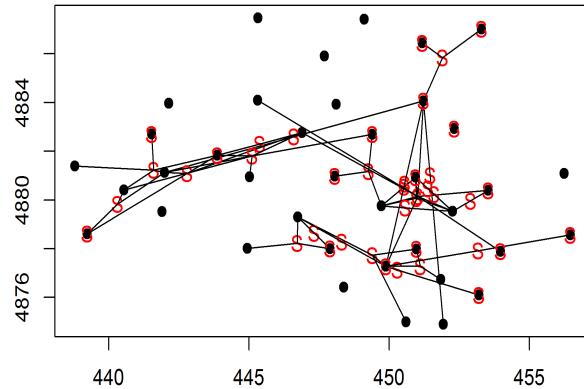


Figure 4.5. Spider plot of the Fort Drum study data. The black dots represent the 47 trap locations with the "S" symbols being the average capture location of each bear. i.e., its estimated home range center. All traps in which a bear was captured are connected to its estimated home range center with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

```

3853     mu[i] <- z[i]*p[i]
3854     y[i] ~ dbin(mu[i],K)           # observation model
3855 }
3856
3857 N <- sum(z[1:(nind+nz)])
3858 }
3859 ",file="modelMcov.txt")

3860 data2 <- list(y=y,nz=nz, nind=nind, K=K, xcent=xcent,B=11.5)
3861 params2 <- c('p0','psi','N','beta')
3862 inits <- function() {list(z=zst, psi=psi, p0=rnorm(1), beta=rnorm(1) ) }
3863 fit2 <- bugs(data2, inits, params2, model.file="modelMcov.txt",
3864                 n.chains=3, n.iter=11000, n.burnin=1000, n.thin=1)

```

3865 This produces the posterior summary statistics in Table 4.6.
 3866 We note that the estimated N is much lower than obtained by model M_h but there
 3867 is a good explanation for this which we discuss in the next section. That issue notwithstanding,
 3868 it is worth pondering how this model could be an improvement (conceptually or technically)
 3869 over some other model/estimator including M_0 and M_h considered previously. Well, for one, we have accounted formally for heterogeneity due to spatial location

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

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

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

4.5.3 Extension of the model

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

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

```

3900 > u <- runif(10000,-1,1)
3901 > v <- runif(10000,-1,1)
3902 > d <- sqrt(u*u+v*v)

```

```

3903 > hist(d[d<1])
3904 > hist(d[d<1],100)
3905 > hist(d[d<1],100,probability=TRUE)
3906 > abline(0,2)

```

3907 It would be useful if we could describe this distribution directly in **BUGS** but there
 3908 is not a built-in way to do so. However, we can implement a discrete version of the pdf⁶.
 3909 To do this, we break B into L distance classes of width δ , with probabilities proportional
 3910 to $2 * x$. In particular, if we denote the cut-points by $g_1 = 0, g_2, \dots, g_{L+1} = B$ and the
 3911 interval midpoints are $m_i = g_{i+1} - \delta$. Then the interval probabilities are, approximately⁷,
 3912 $p_i = \delta(2m_i/B^2)$, which we can compute once and then pass them to **BUGS** as data. The
 3913 **R** commands for doing all of this (noting that we have already loaded and processed the
 3914 Fort Drum bear data) are given in the following **R/BUGS** script:

```

3915 > delta <- .2
3916 > xbin <- xcent%/%delta + 1                      # Put x in bins
3917 > midpts <- seq(delta,Dmax,delta)
3918 > xprobs <- delta*(2*midpts/(B*B))
3919 > xprobs <- xprobs/sum(xprobs)

3920
3921 > cat("
3922 model{
3923 p0 ~ dunif(0,1)                                # Prior distributions
3924 alpha0 <- log(p0/(1-p0))
3925 psi ~ dunif(0,1)
3926 beta ~ dnorm(0,.01)

3927 for(i in 1:(nind+nz)){
3928   xbin[i] ~ dcat(xprobs[])
3929   z[i] ~ dbern(psi)                               # DA variables
3930   lp[i] <- alpha0 + beta*xbin[i]*delta          # Individual covariate model
3931   logit(p[i]) <- lp[i]
3932   mu[i] <- z[i]*p[i]
3933   y[i] ~ dbin(mu[i],K)                          # Observation model
3934 }
3935

3936 N <- sum(z[1:(nind+nz)])                      # N is derived
3937 }
3938 ",file="modelMcov.txt")

```

3940 In the model description, the variable x (observed distance from centroid of the trap
 3941 array) has been rounded or binned (placed into a distance bin) so that the discrete version
 3942 of the pdf of x can be used, as described previously. The new variable labeled **xbin** is
 3943 then the *integer category* in units of δ from 0. Thus, to convert back to distance in the

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

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

3944 expression for `lp[i]`, `xbin[i]` has to be multiplied by δ . To fit the model, keeping in
 3945 mind that the data objects required below have been defined in previous analyses of this
 3946 chapter, we do this:

```
3947 > data2 <- list(y=y, nz=nz, nind=nind, K=K, xbin=xbin, xprobs=xprobs,  

3948   delta=delta)  

3949 > params2 <- c('p0','psi','N','beta')  

3950 > inits <- function() {list(z=z, psi=psi, p0=runif(1),beta=rnorm(1) ) }  

3951 > fit <- bugs(data2, inits, params2, model.file="modelMcov.txt",  

3952   working.directory=getwd(), debug=FALSE, n.chains=3,  

3953   n.iter=11000, n.burnin=1000, n.thin=2)
```

3954 By specification of B , this model induces a clear definition of area in which the popu-
 3955 lation of N individuals reside. The parameter N of the model is the population size that
 3956 applies to the particular value of B and, as such, we will see that N scales with our choice
 3957 of B . This might be disconcerting to some—we can get whatever value of N we want
 3958 by changing B ! However, it is intuitively reasonable that, as we increase the area under
 3959 consideration, there should be more individuals in it. Fortunately, we find empirically,
 3960 that while N is highly sensitive to the prescribed value of B , density appears invariant to
 3961 B as long as B is sufficiently large. We fit the model for a set of values of B from $B = 12$
 3962 (restricting values of x to be in close proximity to the trap array) on up to 20. The results
 3963 are given in Table 4.7.

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

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

3964 We see that the posterior mean and SD of density (individuals per square km) appear
 3965 insensitive to choice of B once we reach about $B = 17$ or so. The estimated density of
 3966 0.25 per km² is actually quite a bit lower than we reported using model M_h for which no
 3967 relevant “area” quantity is explicit in the model (and so we had to make it up). Using
 3968 MLEs of N in conjunction with buffer strips (see Tab. 1.1) our estimates were in the
 3969 range of 0.32 – 0.43 and see Sec. 4.4 above. On the other hand our estimate of $\hat{D} = 0.25$
 3970 here (based on the posterior mean) is higher than that reported from model M_0 using
 3971 the buffered area ($\hat{D} = 0.18$). There is no basis really for comparing or contrasting
 3972 these various estimates. In particular, application of models M_0 and M_h are distinctly
 3973 *not* spatially explicit models—the area within which the population resides is not defined
 3974 under either model. There is therefore no reason at all to think that the estimates produced
 3975 under either closed population model, based on a buffered “trap area”, are justifiable by
 3976 any theory. In fact, we would get exactly the same estimate of N no matter what we declare

3977 the area to be. On the other hand, the individual covariate model uses an explicit model
 3978 for “distance from centroid” that is a reasonable and standard null model—it posits, in the
 3979 absence of direct information, that individual home range centers are randomly distributed
 3980 in space and that probability of detection depends on the distance between home range
 3981 center and the centroid of the trap array. Under this definition of the system, we see that
 3982 density is invariant to the choice of area, which seems like a desirable feature.

3983 **4.5.4 Invariance of density to B**

3984 Under model M_x , and also under models that we consider in later chapters, a general
 3985 property of the estimators is that while N increases with the prescribed area of the model
 3986 (defined by B in this model), we expect that density estimators should be invariant to this
 3987 area. In the model used above, we note that $\text{Area}(B) = \pi B^2$ and $\mathbb{E}(N(B)) = \lambda \text{Area}(B)$
 3988 and thus $\mathbb{E}(\text{Density}(B)) = \lambda$, i.e., constant. This should be interpreted as the *prior*
 3989 density. Absent data, then realizations under the model will have density λ regardless
 3990 of what B is prescribed to be. As we verified empirically above, posterior summaries of
 3991 density are also invariant to B as long as the prescribed area is sufficiently large.

3992 **4.5.5 Toward fully spatial capture-recapture models**

3993 While the use of an individual covariate model resolves two important problems inherent
 3994 in almost all capture-recapture studies (induced heterogeneity and absence of a precise
 3995 relationship between N and area), is not ideal for all purposes because it does not make
 3996 full use of the spatial information in the data set, i.e., the trap locations and the locations
 3997 of each individual encounter, so that we cannot use this model to model trap-specific
 3998 effects (e.g., trap effort or type). Moreover, we applied this model for “data” being the
 3999 average observed encounter location, and equated that summary to the home range center
 4000 s_i . Intuitively, taking the average encounter location as an estimate of home range center
 4001 makes sense but more so when the trapping grid is dense and expansive relative to typical
 4002 home range sizes which might not be reasonable in practice. Moreover, this approach
 4003 also ignored the variable precision with which each s_i is estimated. Finally, it ignores
 4004 that estimates of s_i around the “edge” (however we define that) are biased because the
 4005 observations are truncated—we can only observe locations interior to the array.

4006 However, there is hope to extend this model in order to resolve these remaining defi-
 4007 ciencies. In the next chapter we provide a further extension of this individual covariate
 4008 model that definitively resolves the *ad hoc* nature of the approach we took here. In that
 4009 chapter we build a model in which s_i are regarded as latent variables and the observation
 4010 locations (i.e., trap specific encounters) are linked to those latent variables with an explicit
 4011 model. We note that the model fitted previously could be adapted easily to deal with s_i
 4012 as a latent variable, simply by adding a prior distribution for s_i . This is actually easier,
 4013 and less ad hoc in a number of respects, and you should try it out.

4.6 DISTANCE SAMPLING: A PRIMITIVE SCR MODEL

4014 Distance sampling is a class of methods for estimating animal density from measurements
 4015 of distance from an observer to individual animals (or groups). The basic assumption

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

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

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

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

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

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

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

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

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

along with

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

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

```

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

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

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

```

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

4056 Basic math can be used to argue that if individuals have a uniform distribution in space,
 4057 then the distribution of Euclidean distance is also uniform. In particular, if a transect of
 4058 length L is used and x is distance to the transect then $F(x) = \Pr(X \leq x) = L*x/L*B =$
 4059 x/B and $f(x) = dF/dx = (1/B)$. For measurements of radial distance, we provided the
 4060 analogous argument in the previous section.

4061 The preceding paragraph makes it clear that distance sampling is a special case of
 4062 spatial capture-recapture models, such as those derived from model M_x of the previous
 4063 section, where the encounter probability is related directly to *distance*, which is a reduced
 4064 information summary of *location*, \mathbf{u} . Some intermediate forms of SCR/DS models can
 4065 be described (Royle et al., 2011a). In the context of our general characterization of SCR
 4066 models (Chapt. 2.6), we suggested that every SCR model can be described, conceptually,
 4067 by a hierarchical model of the form:

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

4068 Distance sampling ignores the part of the model pertaining to \mathbf{s} , and deals only with the
 4069 model components for the observed data \mathbf{u} ⁸. Thus, we are left with a hierarchical model
 4070 of the form

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

4071 In contrast, as we will see in the next chapters, many SCR models (Chapt. 5) ignore \mathbf{u}
 4072 and condition on \mathbf{s} , which is not observed:

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

4073 Since $[\mathbf{u}]$ and $[\mathbf{s}]$ are both assumed to be uniformly distributed, these are equivalent models!
 4074 The main differences have to do with interpretation of model components and whether or
 4075 not the latent variables are observable (in distance sampling they are).

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

4076 So why bother with SCR models when distance sampling yields density estimates and
4077 accounts for spatial heterogeneity in detection? For one, imagine trying to collect distance
4078 sampling data on species such as jaguars or tigers! Clearly, distance sampling requires
4079 that one can collect large quantities of distance data, which is not always possible. For
4080 tigers, it is much easier, efficient, and safer to employ camera traps or track plates and
4081 then apply SCR models. Furthermore, as we will see in Chapt. 15, SCR models can make
4082 use of distance data, allowing us to study distribution, movement, and density. Thus,
4083 SCR models are more general and versatile than distance sampling models (which clearly
4084 are a special case), and can accommodate data from virtually all animal survey designs.

4085 **4.6.1 Example: Sonoran desert tortoise study**

4086 We illustrate the application of distance sampling models using data on the Sonoran desert
4087 tortoise (*Gopherus agassizii*), shown in Fig. 4.6, collected along transects in southern
4088 Arizona (see Zylstra et al. (2010) for details). The data are from 120 square transects
4089 having four 250-m sides, although we ignore this detail in our analysis here and regard
4090 them as 1 km transects, and we pooled the detection data from all 120 transects. The
4091 histogram of encounter distances from the 65 encountered individuals is shown in Fig. 4.7



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

4092
4093 Commands for reading in and organizing the data for analysis using **WinBUGS** are
4094 given in the help file `?tortoise` provided with the `scrbook` package. To compute density,

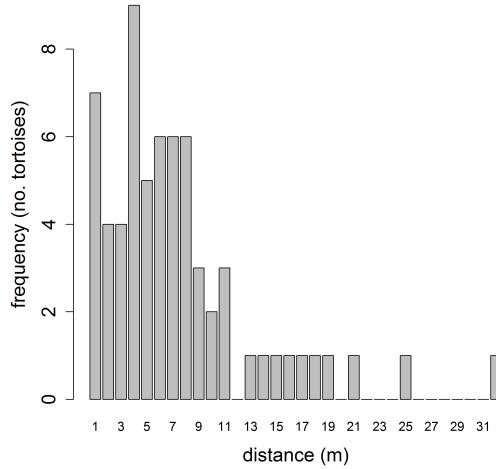


Figure 4.7. Distance histogram of $n = 65$ Sonoran desert tortoise detections from a total of 120 km of survey transect.

4095 the total sampled area of the transects `striparea` is input as data, and computed as:
 4096 120 (transects) multiplied by the length (1000 m) and half-width ($B = 40$ m), then
 4097 multiplied by 2, and divided by 10000 to convert to units of individuals per ha. We also
 4098 provide commands for analyzing the data with `unmarked` (Fiske and Chandler, 2011) using
 4099 hierarchical distance sampling models (Royle et al., 2004).

4100 Posterior summaries for the tortoise data are given in Tab. 4.8. Estimated density
 4101 (posterior mean) is 0.54 individuals per ha and the estimated scale parameter of the
 4102 distance function (posterior mean) is $\sigma = 9.12$ meters. The R-hat statistics of around 1.02
 4103 suggest that slightly longer MCMC simulations might be called for. The posterior mass
 4104 of the data augmentation parameter ψ is located away from the upper bound $\psi = 1$ and
 4105 so the degree of data augmentation appears sufficient.

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

parameter	Mean	SD	2.5%	50 %	97.5%	Rhat	n.eff
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

4.7 SUMMARY AND OUTLOOK

4106 Traditional closed population capture-recapture models are closely related to binomial
4107 generalized linear models. Indeed, the only real distinction is that in capture-recapture
4108 models, the population size parameter N (corresponding also to the size of a hypothetical
4109 “complete” data set) is unknown. This requires special consideration in the analysis of
4110 capture-recapture models. The classical approach to inference recognizes that the observa-
4111 tions don’t have a standard binomial distribution but, rather, a truncated binomial (from
4112 which which the so-called *conditional likelihood* derives) since we only have encounter fre-
4113 quency data on observed individuals. If instead we analyze the models using data augmen-
4114 tation, which arises under a $\text{Uniform}(0, M)$ prior for N , the observations can be modeled
4115 using a zero-inflated binomial distribution. When we deal with the unknown- N problem
4116 using data augmentation then we are left with zero-inflated GLMs and GLMMs instead
4117 of ordinary GLMs or GLMMs. The analysis of such zero-inflated models is practically
4118 convenient, especially using the **BUGS** variants.

4119 Spatial capture-recapture models that we will consider in the rest of the chapters
4120 of this book are closely related to individual covariate models (model M_x). Naturally,
4121 spatial capture-recapture models arise by defining individual covariates based on observed
4122 locations of individuals—we can think of using some function of mean encounter location as
4123 an individual covariate. We did this in a novel way, by using distance to the centroid of the
4124 trapping array as a covariate. We analyzed the *full likelihood* using data augmentation,
4125 and placed a prior distribution on the individual covariate which was derived from an
4126 assumption that individual locations are, *a priori*, uniformly distributed in space. This
4127 assumption provides for invariance of the density estimator to the choice of population
4128 size area (induced by maximum distance from the centroid of the trap array). The model
4129 addressed some important problems in the use of closed population models: it allows for
4130 heterogeneity in encounter probability due to the spatial juxtaposition of individuals with
4131 the array of traps, and it also provides a direct estimate of density because area is a
4132 feature of the model (via the prior on the individual covariate). The model is still not
4133 completely general, however, because it does not make full use of the spatial encounter
4134 histories, which provide direct information about the locations and density of individuals.

4135 A specific individual covariate model that is in widespread use is classical distance
4136 sampling. The model underlying distance sampling is precisely a special kind of SCR
4137 model—but one without replicate samples. Understanding distance sampling and individ-
4138 ual covariate models more broadly provides a solid basis for understanding and analyzing
4139 spatial capture-recapture models. In fact if, instead of placing an explicit model on *dis-*
4140 *tance* in the classical distance sampling model, we were to place the prior distribution on
4141 *location*, s , of each individual, then the form of the distance sampling model more closely
4142 resembles the SCR model we introduce in the next chapter.

4143

Part II

4144

4145

Basic SCR Models

5

FULLY SPATIAL CAPTURE-RECAPTURE MODELS

4150 In the previous chapter, we discussed models that could be viewed as primitive spatial
4151 capture-recapture models. We looked at a basic distance sampling model, and we also
4152 considered a classical individual covariate modeling approach in which we defined a co-
4153 variate to be the distance from the (estimated) home range center to the center of the
4154 trap array. The individual covariate model that we conjured up was “spatial” in the sense
4155 that it included some characterization of where individuals live but, on the other hand,
4156 only a primitive or no characterization of trap location. That said, there is only a small
4157 step from this model to spatial capture-recapture models that we consider in this chapter,
4158 which fully recognize the spatial attribution of both individual animals *and* the locations
4159 of encounter devices.

4160 Capture-recapture models must accommodate the spatial organization of individuals
4161 and the encounter devices because the encounter process occurs at the level of individual
4162 traps. Failure to consider the trap-specific data is one of the key deficiencies with classical
4163 ad-hoc approaches which aggregate encounter information to the resolution of the entire
4164 trap array. We have previously addressed some problems that this causes including induced
4165 heterogeneity in encounter probability, imprecise notation of “sample area” and not being
4166 able to accommodate trap-specific effects or trap-specific missing values. In this chapter
4167 we resolve these issues by developing our first fully spatial capture-recapture model. This
4168 model is not too different from that considered in Sec. 4.5 but, instead of defining the
4169 individual covariate to be distance to the centroid of the array we define J individual
4170 covariates - the distance to *each* trap. And, instead of using estimates of individual
4171 locations \mathbf{s} , we consider a fully hierarchical model in which we regard \mathbf{s} as a latent variable
4172 and impose a prior distribution on it.

4173 In this chapter we investigate the basic spatial capture-recapture model, which we re-
4174 fer to as “model SCR0”, and address some important considerations related to its analysis
4175 in **BUGS**. We demonstrate how to summarize posterior output for the purposes of pro-
4176 ducing density maps or spatial predictions of density. The key aspect of the SCR models

4177 considered in this chapter is the formulation of a model for encounter probability that is
 4178 a function of distance between individual home range center and trap locations. We also
 4179 discuss how encounter probability models are related to explicit models of space usage
 4180 or “home range area.” Understanding this allows us to compute, for example, the area
 4181 used by an individual during some prescribed time. While it is intuitive that SCR models
 4182 should be related to some model of space usage, this has not been discussed much in the
 4183 literature (but see Royle et al. (2012b) which we address further in Chapt. 13).

5.1 SAMPLING DESIGN AND DATA STRUCTURE

4184 In our development here, we will assume a standard sampling design in which an array
 4185 of J traps is operated for K sample occasions (say, nights) producing encounters of n
 4186 individuals. Because sampling occurs by traps and also over time, the most general data
 4187 structure yields temporally *and* spatially indexed encounter histories for *each individual*.
 4188 Thus a typical data set will include an encounter history *matrix* for each individual indicating
 4189 which trap the individual was captured, during each sample occasion. For example,
 4190 suppose we sample at 4 traps over 3 nights. A plausible data set for a single individual
 4191 captured one time in trap 1 on the first night and one time in trap 3 on the 3rd night is:

```
4192     night1 night2 night3
4193 trap1    1    0    0
4194 trap2    0    0    0
4195 trap3    0    0    1
4196 trap4    0    0    0
```

4197 This data structure would be obtained for *each* of the $i = 1, 2, \dots, n$ captured individuals.

4198 We develop models in this chapter for passive detection devices such as “hair snares” or
 4199 other DNA sampling methods (Kéry et al., 2010; Gardner et al., 2010b) and related types of
 4200 sampling devices in which (i) devices (“traps”) may capture any number of individuals (i.e.,
 4201 they don’t fill up); (ii) an individual may be captured in more than one trap during each
 4202 occasion but (iii) individuals can be encountered at most 1 time by each trap during any
 4203 occasion. Hair snares for sampling DNA from bears and other species function according
 4204 to these rules. An individual bear wandering about its territory might come into contact
 4205 with > 1 devices; a device may encounter multiple bears; however, in practice, it will
 4206 often not be possible to attribute multiple visits of the same individual during a single
 4207 occasion (e.g., night) to distinct encounter events. Thus, an individual may be captured
 4208 at most 1 time in each trap during any occasion. While this model, which we refer to
 4209 as SCR0, is most directly relevant to hair snares and other DNA sampling methods for
 4210 which multiple detections of an individual are not distinguishable, we will also make use
 4211 of the model for data that arise from camera-trapping studies. In practice, with camera
 4212 trapping, individuals might be photographed several times in a night but it is common to
 4213 distill such data into a single binary encounter event for reasons discussed later in Chapt.
 4214 9.

4215 The statistical assumptions we make to build a model for these data are that individual
 4216 encounters within and among traps are independent, and this allows us to regard
 4217 individual- and trap-specific encounters as *independent* Bernoulli trials (see next section).
 4218 These basic (but admittedly at this point somewhat imprecise) assumptions define the

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

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

4219 basic spatial capture-recapture model, SCR0. We will make things more precise as we
 4220 develop a formal statistical definition of the model shortly.

5.2 THE BINOMIAL OBSERVATION MODEL

4221 We begin by considering the simple model in which there are no time-varying covariates
 4222 that influence encounter, there are no explicit individual-specific covariates, and there are
 4223 no covariates that influence density. In this case, we can aggregate the binary encounters
 4224 over the K sample occasions and record the total number of encounters out of K . We will
 4225 denote these individual- and trap-specific encounter frequencies by y_{ij} for $i = 1, 2, \dots, n$
 4226 captured individuals and $j = 1, 2, \dots, J$ traps. For example, suppose we observe 6 individuals
 4227 in sampling at 4 traps over 3 nights of sampling then a plausible data set is the 6×4
 4228 matrix of encounters (out of 3 sampling occasions) shown in Table 5.1. We assume that
 4229 y_{ij} are mutually independent outcomes of a binomial random variable which we express
 4230 as:

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

4231 This is the basic model underlying standard closed population models (Chapt. 4) except
 4232 that, in the present case, the encounter frequencies are individual- *and* trap-specific, and
 4233 encounter probability p_{ij} depends on both individual *and* trap.

4234 As we did in Sec. 4.5, we will make explicit the notion that p_{ij} is defined conditional
 4235 on *where* individual i lives. Naturally, we think about defining an individual home range
 4236 and then relating p_{ij} explicitly to a summary of its location relative to each trap. For
 4237 example, the centroid of the individuals home range, or its center of activity (Efford, 2004;
 4238 Borchers and Efford, 2008; Royle and Young, 2008). In what follows, we define \mathbf{s}_i , a two-
 4239 dimensional spatial coordinate, to be the home range or activity center of individual i .
 4240 Then, the SCR model postulates that encounter probability, p_{ij} , is a decreasing function
 4241 of distance between \mathbf{s}_i and the location of trap j , \mathbf{x}_j (also a two-dimensional spatial
 4242 coordinate). A standard model for modeling binomial counts is the logistic regression,
 4243 where we model the dependence of p_{ij} on distance according to:

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

4244 where, here, $\|\mathbf{x}_j - \mathbf{s}_i\|$ is the distance between \mathbf{s}_i and \mathbf{x}_j . We sometimes write $\|\mathbf{x}_j - \mathbf{s}_i\| =$
 4245 $\text{dist}(\mathbf{x}_j, \mathbf{s}_i) = d_{ij}$. Alternatively, a popular model is

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

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

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

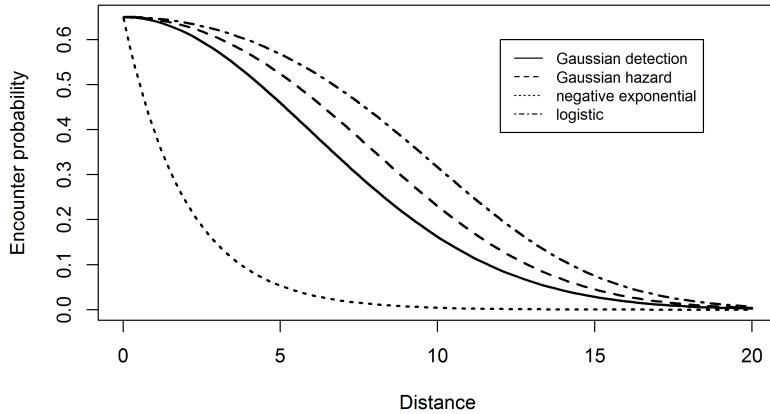


Figure 5.1. Some common encounter probability models showing the characteristic monotone decrease of encounter probability with distance between activity center and trap location.

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

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

4266 but sometimes, for notational simplicity, we abbreviate this by omitting some of the
 4267 arguments to p .

4268 5.2.1 Definition of home range center

4269 We define an individual's home range as *the area used by an organism during some time*
 4270 *period* which has a clear meaning for most species regardless of their biology. We therefore
 4271 define the home range center (or activity center) to be the center of the space that individ-
 4272 ual was occupying (or using) during the period in which traps were active. Thinking about
 4273 it in that way, it could even be observable (almost) as the centroid of a very large number
 4274 of radio fixes over the course of a survey period or a season. Thus, this practical version
 4275 of a home range center in terms of space usage is a well-defined construct regardless of
 4276 whether one thinks the home range itself is a meaningful concept. We use the terms home
 4277 range center and activity center interchangeably, and we recognize that this is a transient
 4278 thing which applies only to a well-defined period of study.

4279 5.2.2 Distance as a latent variable

4280 If we knew precisely every \mathbf{s}_i in the population (and population size N), then the model
 4281 specified by Eqs. 5.2.1 and 5.2.2 would be just an ordinary logistic regression-type of
 4282 a model (with covariate d_{ij}) which we learned how to fit using **WinBUGS** previously
 4283 (Chapt. 3). However, the activity centers are unobservable even in the best possible
 4284 circumstances. In that case, d_{ij} is an unobserved variable, analogous to the situation in
 4285 classical random effects models. We need to therefore extend the model to accommodate
 4286 these random variables with an additional model component – the random effects dis-
 4287 tribution. The customary assumption is the so-called “uniformity assumption,” which is
 4288 to assume that the \mathbf{s}_i are uniformly distributed over space (the obvious next question:
 4289 “which space?” is addressed below). This uniformity assumption amounts to a uniform
 4290 prior distribution on \mathbf{s}_i , i.e., the pdf of \mathbf{s}_i is constant, which we may express

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

4291 As it turns out, this assumption is usually not precise enough to fit SCR models in practice
 4292 for reasons we discuss shortly. We will give another way to represent this prior distribution
 4293 that is more concrete, but depends on specifying the “state-space” of the random variable
 4294 \mathbf{s}_i . The term state-space is a technical way of saying “the space of all possible outcomes”
 4295 of the random variable.

5.3 THE BINOMIAL POINT PROCESS MODEL

4296 In the SCR model, the individual activity centers are unobserved and thus we treat them
 4297 as random effects. Specifically, the collection of individual activity centers $\mathbf{s}_1, \dots, \mathbf{s}_N$
 4298 represents a realization of a *binomial point process* (Illian et al., 2008, p. 61). The
 4299 binomial point process (BPP) is analogous to a Poisson point process in the sense that it
 4300 represents a “random scatter” of points in space – except that the total number of points
 4301 is *fixed*, whereas, in a Poisson point process, it is random (having a Poisson distribution).

4302 As an example, we show in Fig. 5.2 locations of 20 individual activity centers (black
 4303 dots) in relation to a grid of 25 traps. For a Poisson point process the number of such
 4304 points in the prescribed state-space would be random whereas often we will simulate fixed
 4305 numbers of points, e.g., for evaluating the performance of procedures, e.g., how well does
 our estimator perform when $N = 50$?

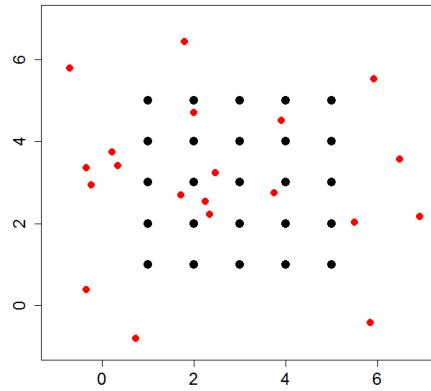


Figure 5.2. Realization (small dots) of a binomial point process with $N = 20$. The large dots represent trap locations.

4306
 4307 It is natural to consider a binomial point process in the context of capture-recapture
 4308 models because it preserves N in the model and thus preserves the linkage directly with
 4309 closed population models. In fact, under the binomial point process model, model M_0
 4310 and other closed models are simple limiting cases of SCR models, i.e., they arise as the
 4311 coefficient on distance (α_1 above) tends to 0.

4312 While we often will express SCR models “conditional-on- N ”, it will sometimes be
 4313 convenient to impose specific prior distributions on N . By assuming N has a binomial
 4314 distribution, we can make use of data augmentation, our preferred tool, for Bayesian
 4315 analysis of the models as in Chapt. 4, thus yielding a methodologically coherent approach
 4316 to analyzing the different classes of models. We might also assume that N has a Poisson
 4317 distribution in some cases (see Chapt. 14). Of course, the two assumptions are closely
 4318 related in the usual limiting sense.

4319 One consequence of having fixed N in the BPP model is that the model is not
 4320 strictly a model of “complete spatial randomness”. This is because, if one forms counts
 4321 $n(A_1), \dots, n(A_k)$ in any set of disjoint regions of the state-space, say A_1, \dots, A_k , then
 4322 these counts are *not* independent. In fact, they have a multinomial distribution (see Illian
 4323 et al., 2008, p. 61). Thus, the BPP model introduces a slight bit of dependence in the
 4324 distribution of points. However, in most situations this will have no practical effect on any
 4325 inference or analysis and, as a practical matter, we will usually regard the BPP model as

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

5.3.1 The state-space of the point process

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

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

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

Prescribing the state-space

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

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

4367 situations this might be an acceptable approach to defining the state-space, i.e., just a
 4368 rectangle around the trap array. Although defining the state-space to be a regular polygon
 4369 has computational advantages (e.g., we can implement this more efficiently in **BUGS** and
 4370 cannot for irregular polygons), a regular polygon induces an apparent problem of admitting
 4371 into the state-space regions that are distinctly non-habitat (e.g., oceans, large lakes, ice
 4372 fields, etc.). It is difficult to describe complex regions in mathematical terms that can be
 4373 used in **BUGS**. As an alternative, we can provide a representation of the state-space as
 4374 a discrete set of points which the **R** package **secr** (Efford, 2011) permits (**secr** uses the
 4375 term “mask” for what we call the state-space). Defining the state-space by a discrete set
 4376 of points is handy because it allows specific points to be deleted or not, depending on
 4377 whether they represent available or suitable habitat (see Sec. 5.10). We can also define
 4378 the state-space as an arbitrary collection of polygons stored as a GIS shapefile which can
 4379 be analyzed easily by MCMC in **R** (see Sec. 17.5), but not so easily in the **BUGS** engines.
 4380 In Sec. 5.10, we provide an analysis of the wolverine camera trapping data, in which we
 4381 define the state-space to be a regular continuous polygon (a rectangle).

4382 **Invariance to the state-space**

4383 We will assert for all models we consider in this book that density is invariant to the size
 4384 and extent of \mathcal{S} , if \mathcal{S} is sufficiently large, and as long as our model relating p_{ij} to \mathbf{s}_i is a
 4385 decreasing function of distance. We can prove this easily by drawing an analogy with a 1-d
 4386 case involving distance sampling. Let y_j be the number of individuals captured in some
 4387 interval $[d_{j-1}, d_j)$, and define $d_J = B$ for some large value of B . The observations from a
 4388 survey are y_1, \dots, y_J and the likelihood is a multinomial likelihood, so the log-likelihood
 4389 is of the form

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

4390 where π_j is the probability of detecting an individual in distance class j , which depends on
 4391 parameters of the detection function (the manner of which is not relevant for the present
 4392 discussion). Choosing B sufficiently large guarantees that $\mathbb{E}(y_J) = 0$ and therefore the
 4393 observed frequency in the “last cell” contributes nothing to the likelihood, in regular
 4394 situations in which the detection function decays monotonically with distance and prior
 4395 density is constant. We can think of B as being related to the state-space in an SCR
 4396 model, as the width of a rectangular state-space with area $B \times L$, L being the length
 4397 of the transect. Thus, if we choose B large enough, then we ensure that the expected
 4398 trap-frequencies beyond B will be 0, and thus contribute nothing to the likelihood.

4399 Sometimes our estimate of density can be affected by choosing \mathcal{S} too small. However,
 4400 this might be sensible if \mathcal{S} is naturally well-defined. As we discussed in Chapt. 1, \mathcal{S} is
 4401 *part of the model*, and thus it is sensible that estimates of density might be sensitive to
 4402 its definition in problems where it is natural to restrict \mathcal{S} . One could imagine, however,
 4403 in specific cases, e.g., a small population with well-defined habitat preferences, that a
 4404 problem could arise because changing the state-space based on differing opinions, and
 4405 GIS layers, might have substantial affects on the density estimate. But this is a real
 4406 biological problem, and a natural consequence of the spatial formalization of capture-
 4407 recapture models – a feature, not a bug or some statistical artifact – and it should be
 4408 resolved with better information, research, and thinking. For situations where there is not
 4409 a natural choice of \mathcal{S} , we should default to choosing \mathcal{S} to be very large in order to achieve

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

5.3.2 Connection to model M_h and distance sampling

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

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

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

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

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

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

It is also worth re-emphasizing that the basic SCR encounter model is a binomial encounter model in which distance is a covariate. As such, it is strikingly similar to classical distance sampling models (Buckland et al., 2001). Both have distance as a covariate but, in classical distance sampling problems, the focus is on the distance between the observer and the animal at an instant in time, not the distance between a trap and an animal’s home range center. As a practical matter, in distance sampling, “distance” is *observed* for those individuals that appear in the sample. Conversely, in SCR problems, it is only imperfectly observed (we have partial information in the form of trap observations). Clearly, it is preferable to observe distance if possible, but distance sampling requires field methods that are not practical in many situations, e.g. when studying carnivores such as

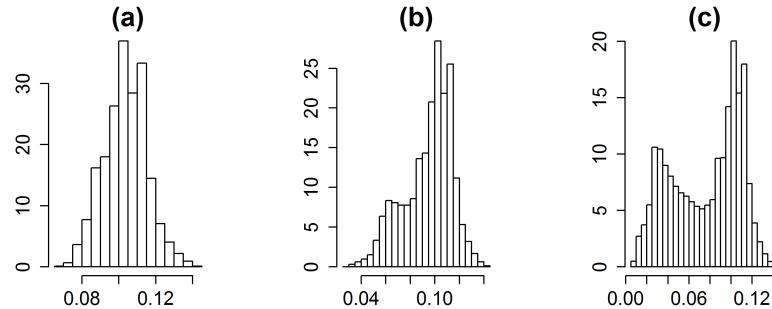


Figure 5.3. Implied distribution of p_i for a population of individuals as a function of the size of the state-space buffer around the trap array. The state-space buffer is 0.2, 0.5 and 1.0 for panels (a), (b), (c), respectively. In each case, the trap array is fixed and centered within a square state-space.

bears or large cats. Furthermore, SCR models allow us to relax many of the assumptions made in classical distance sampling, such as perfect detection at distance zero, and SCR models allow for estimates of quantities other than density, such as home range size, and space usage (see Chaps. 12 and 13).

5.4 THE IMPLIED MODEL OF SPACE USAGE

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

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

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

In practice it might make sense to think about the first component, i.e., $\Pr(\text{encounter}|\text{usage of } \mathbf{x}, \mathbf{s})$ as being a constant (e.g., if traps are located within arbitrarily small grid cells) and then, in that case, the encounter probability model is directly

4471 proportional to this model for individual movements about their home range center deter-
 4472 mining the use frequency of each \mathbf{x} . This is a sensible heuristic model for what ecologists
 4473 would call a central place forager although, as we have stated previously, it may be mean-
 4474 ingful as a description of transient space usage as well (that is, the space usage during the
 4475 period of sampling).

4476 To motivate a specific model for space usage, imagine the area we are interested in
 4477 consists of some large number of small pixels (i.e. we're looking at a discrete representation
 4478 of space), and that we have some kind of perfect observation device (e.g., continuous
 4479 telemetry) so that we observe every time an individual moves into a pixel. After a long
 4480 period of time, we observe an enormous sample size of \mathbf{x} values. We tally those up into
 4481 each pixel, producing the frequency $m(\mathbf{x}, \mathbf{s})$, which is something like the "true" usage of
 4482 pixel \mathbf{x} by individual with activity center \mathbf{s} . So, then, the usage model should be regarded
 4483 as a probability mass function for these counts and, naturally, we regard the counts $m(\mathbf{x}, \mathbf{s})$
 4484 as a multinomial observation with probabilities $\pi(\mathbf{x}|\mathbf{s})$, and prescribe a suitable model for
 4485 $\pi(\mathbf{x}|\mathbf{s})$ that describes how use events should accumulate in space. A natural null model
 4486 for $\pi(\mathbf{x}|\mathbf{s})$ has a decreasing probability of use as \mathbf{x} gets far away from \mathbf{s} ; i.e., animals spend
 4487 more time close to their activity centers than far away. We can regard points used by
 4488 the individual with activity center \mathbf{s} as the realization of a point process with conditional
 4489 intensity:

$$\pi(\mathbf{x}|\mathbf{s}) = \frac{k(\mathbf{x}, \mathbf{s})}{\sum_x k(\mathbf{x}, \mathbf{s})} \quad (5.4.1)$$

4490 where $k(\mathbf{x}, \mathbf{s})$ is any positive function. In continuous space, the equivalent representation
 4491 would be:

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

4492 Clearly the space used by an individual will be proportional to whatever kernel, $k(\mathbf{x}, \mathbf{s})$,
 4493 we plug-in here. If we use a negative exponential function, then this produces a standard
 4494 resource selection function (RSF) model (e.g., Manly et al., 2002, Chapt. 8). But, here
 4495 we use a Gaussian kernel, i.e.,

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

4496 so that contours of the probability of space usage resemble a bivariate normal or Gaussian
 4497 probability distribution function.

4498 To apply this model of space-usage to SCR problems we allow for imperfect detection
 4499 by introducing a non-uniform "thinning rate" of the true counts $m(\mathbf{x}, \mathbf{s})$. This yields,
 4500 precisely, our Gaussian encounter probability model where the thinning rate is our baseline
 4501 encounter probability p_0 for each pixel where we place a trap, and $p = 0$ in each pixel
 4502 where we don't place a trap.

4503 The main take-away point here is that underlying most SCR models is some kind of
 4504 model of space-usage, implied by the specific choice of $k(\mathbf{x}, \mathbf{s})$. Whether or not we have
 4505 perfect sampling devices, the function we use in the encounter probability model equates
 4506 to some conditional distribution of points, a utilization distribution, as in Eq. 5.4.1, from
 4507 which we can compute effective home range area, i.e., the area that contains some percent
 4508 of the mass of a probability distribution proportional to $k(\mathbf{x}, \mathbf{s})$; e.g., 95% of all space used
 4509 by an individual with activity center \mathbf{s} .

4510 **5.4.1 Bivariate normal case**

4511 One encounter model that allows direct analytic computation of home range area is the
 4512 Gaussian encounter probability model

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

4513 For this model, encounter probability is proportional to the kernel of a bivariate normal
 4514 (Gaussian) pdf and so the natural interpretation is that in which movement outcomes (or
 4515 successive locations of an individual) are draws from a bivariate normal distribution with
 4516 standard deviation σ . We say that use of this model implies a bivariate normal model of
 4517 space usage. Under this model we can compute precisely the effective home range area. In
 4518 particular, if use outcomes are bivariate normal, then $\|\mathbf{x} - \mathbf{s}\|^2$ has a chi-square distribution
 4519 with 2 d.f. and the quantity $B(\alpha)$ that encloses $(1 - \alpha)\%$ of all realized distances i.e.,
 4520 $\Pr(d \leq B(\alpha)) = 1 - \alpha$, is $B(\alpha) = \sigma * \sqrt{q(\alpha, 2)}$ where $q(\alpha, 2)$ is the 0.05 chi-square
 4521 critical value on 2 df. For example, to compute $q(.05, 2)$ in R we execute the command
 4522 `qchisq(.95, 2)` which is $q(2, \alpha) = 5.99$. Then, for $\sigma = 1$, $B(\alpha) = 1 * \sqrt{5.99} = 2.447$.
 4523 Therefore 95% of the points used will be within 2.447 (standard deviation) units of the
 4524 home range center. So, in practice, we can estimate σ by fitting the bivariate normal
 4525 encounter probability model to some SCR data, and then use the estimated σ to compute
 4526 the “95% radius”, say $r_{.95} = \sigma\sqrt{5.99}$, and convert this to the 95% use area – the area
 4527 around \mathbf{s} which contains 95% of the movement outcomes – according to $A_{.95} = \pi r_{.95}^2$.

4528 An alternative bivariate normal model is the bivariate normal hazard rate model:

$$p(\mathbf{x}, \mathbf{s}) = 1 - \exp\left(-\lambda_0 * \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{x} - \mathbf{s}\|^2\right)\right) \quad (5.4.2)$$

4529 We use λ_0 here because this parameter, the baseline encounter *rate*, can be > 1 . This arises
 4530 by assuming the latent “use frequency” $m(\mathbf{x}, \mathbf{s})$ is a Poisson random variable with intensity
 4531 $\lambda_0 k(\mathbf{x}, \mathbf{s})$. The model is distinct from our Gaussian encounter model $p(\mathbf{x}, \mathbf{s}) = p_0 k(\mathbf{x}, \mathbf{s})$
 4532 used previously, although we find that they produce similar results in terms of estimates
 4533 of density or 95% use area, as long as baseline encounter probability is low. We discuss
 4534 these two formulations of the bivariate normal model further in Chapt. 9.

4535 **5.4.2 Empirical analysis**

4536 For any encounter model we can compute space usage quantiles empirically by taking a fine
 4537 grid of points and either simulating movement outcomes with probabilities proportional to
 4538 $p(\mathbf{x}, \mathbf{s})$ and accumulating area around \mathbf{s} , or else we can do this precisely by varying $B(\alpha)$
 4539 to find that value within which 95% of all movements are concentrated, i.e., the set of all
 4540 \mathbf{x} such that $\|\mathbf{x} - \mathbf{s}\| \leq B(q)$. Under any detection model, movement outcomes will occur
 4541 in proportion to $p(\mathbf{x}, \mathbf{s})$, as long as the probability of encounter is constant, *conditional on*
 4542 *use*, and so we can define our space usage distribution according to:

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

4543 Given the probabilities $\pi(\mathbf{x}, \mathbf{s})$ for all \mathbf{x} we can find the value of $B(q)$, for any q , such that

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

4544 (here, we use \ni to mean “such that”). We have a function called `hra` in the `scrbook`
 4545 package that computes the home range area for any encounter model and prescribed
 4546 parameter values. The help file for `hra` has an example of simulating some data. The
 4547 following commands illustrate this calculation for two different bivariate normal models
 4548 of space usage:

```

4549 ##
4550 ## Define encounter probability model as R function
4551 ##
4552 > pGauss2 <- function(parms,Dmat){
4553   a0 <- parms[1]
4554   sigma <- parms[2]
4555   lp <- parms[1] -(1/(2*parms[2]*parms[2]))*Dmat*Dmat
4556   p <- 1-exp(-exp(lp))
4557   p
4558 }
4559
4560 > pGauss1 <- function(parms,Dmat){
4561   a0 <- parms[1]
4562   sigma <- parms[2]
4563   p <- plogis(parms[1])*exp( -(1/(2*parms[2]*parms[2]))*Dmat*Dmat )
4564   p
4565 }
4566
4567 ##
4568 ## Execute hra with sigma = .3993
4569 ##
4570 > hra(pGauss1,parms=c(-2,.3993),plot=FALSE,xlim=c(0,6),ylim=c(0,6),
4571       ng=500,tol=.0005)
4572
4573 [1] 0.9784019
4574 radius to achieve 95% of area: 0.9784019
4575 home range area: 3.007353
4576 [1] 3.007353
4577
4578
4579 ## Analytic solution:
4580 ## true sigma that produces area of 3
4581 > sqrt(3/pi)/sqrt(5.99)
4582 [1] 0.3992751

```

4583 What this means is that $B(q) = 0.978$ is the radius that encloses about 95% of all
 4584 movements under the standard bivariate normal encounter model. Therefore, the area is
 4585 about $\pi * .978 * .978 = 3.007$ spatial units. You can change the intercept of the model
 4586 and find that it has no effect. The true (analytic) value of σ that produces a home range
 4587 area of 3.0 is 0.3993 which is the value we initially plugged in to the `hra` function. We
 4588 can improve on the numerical approximation to home range area (get it closer to 3.0) by

4589 increasing the resolution of our spatial grid (increase the `ng` argument) along with the `tol`
 4590 argument.

4591 We can also reverse this process, and find, for any detection model, the parameter
 4592 values that produce a certain $(1 - q)\%$ home range area, which we imagine would be
 4593 useful for doing simulation studies. The function `hra` will compute the value of the scale
 4594 parameter that achieves a certain target $(1 - q)\%$ home range area, by simply providing a
 4595 non-null value of the variable `target.area`. Here we use `target.area = 3.00735` (from
 4596 above) to obtain a close approximation to the value σ we started with (the parameter
 4597 argument is meaningless here):

```
4598 > hra(pGauss1,parms=c(-2,.3993),plot=FALSE,xlim,ylim,ng=500,  

4599   target.area=3.00735,tol=.0005)  

4600  

4601 Value of parm[2] to achieve 95% home range area of 3.00735: 0.3993674
```

4602 5.4.3 Relevance of understanding space usage

4603 One important reason that we need to be able to deduce “home range area” from a
 4604 detection model is so that we can compare different models with respect to a common
 4605 biological currency. Many encounter probability models have some “scale parameter”,
 4606 which we might call σ no matter the model, but this relates to 95% area in a different
 4607 manner under each model. Therefore, we want to be able to convert different models
 4608 to the same currency. Another reason to understand the relationship between models of
 4609 encounter probability and space usage is that it opens the door to combining traditional
 4610 resource selection data from telemetry with spatial capture-recapture data. In Chapt. 13
 4611 we consider this problem, for the case in which a sample of individuals produces encounter
 4612 history data suitable for SCR models and, in addition, we have telemetry relocations on a
 4613 sample of individuals. This is achieved by regarding the two sources of data as resulting
 4614 from the same underlying process of space usage but telemetry data produce “perfect”
 4615 observations, like always-on camera traps blanketing a landscape. We use this idea to
 4616 model the effect of a measured covariate at each pixel, say $C(\mathbf{x})$, on home range size and
 4617 geometry and, hence, the probability of encounter in traps.

4618 5.4.4 Contamination due to behavioral response

4619 Interpretation of encounter probability models as models of animal home range and space
 4620 usage can be complicated by a number of factors, including whether traps are baited or
 4621 not. In the case of baited traps, this might lead to a behavioral response (Sec. 7.2.3)
 4622 which could affect animal space usage. For example, if traps attract animals from a long
 4623 distance, it could make typical home ranges appear larger than normal. More likely, in our
 4624 view, it wouldn’t change the typical size of a range but would change how individuals use
 4625 their range e.g., by moving from baited trap to baited trap, so that observed movement
 4626 distances of individuals are typically larger than normal.

4627 In other cases, the reliance on Euclidean distance in models for encounter probability
 4628 might be unrealistic, and can lead to biased estimates of density (Royle et al., 2012a).
 4629 For example, animals might concentrate their movements along trails, roads, or other

4630 landscape features. In this case, models that accommodate other distance metrics can be
 4631 considered. We present models based on least-cost path in Chapt. 12.

5.5 SIMULATING SCR DATA

4632 It is always useful to simulate data because it allows you to understand the system that
 4633 you're modeling and also calibrate your understanding with specific values of the model
 4634 parameters. That is, you can simulate data using different parameter values until you
 4635 obtain data that "look right" based on your knowledge of the specific situation that
 4636 you're interested in. Here we provide a simple script to illustrate how to simulate spatial
 4637 encounter history data. In this exercise we simulate data for 100 individuals and a 25 trap
 4638 array laid out in a 5×5 grid of unit spacing. The specific encounter model is the Gaussian
 4639 model given above and we used this code to simulate data used in subsequent analyses.
 4640 The 100 activity centers were simulated on a state-space defined by a 8×8 square within
 4641 which the trap array was centered (thus the trap array is buffered by 2 units). Therefore,
 4642 the density of individuals in this system is fixed at 100/64.

```

4643 > set.seed(2013)
4644 # Create 5 x 5 grid of trap locations with unit spacing
4645 > traplocs <- cbind(sort(rep(1:5,5)),rep(1:5,5))
4646 > ntraps <- nrow(traplocs)
4647 # Compute distance matrix:
4648 > Dmat <- e2dist(traplocs,traplocs)

4649

4650

4651 # Define state-space of point process. (i.e., where animals live).
4652 # "buffer" just adds a fixed buffer to the outer extent of the traps.
4653 #
4654 > buffer <- 2
4655 > xlim <- c(min(traplocs[,1] - buffer),max(traplocs[,1] + buffer))
4656 > ylim <- c(min(traplocs[,2] - buffer),max(traplocs[,2] + buffer))

4657

4658 > N <- 100    # population size
4659 > K <- 20    # number nights of effort

4660

4661 > sx <- runif(N,xlim[1],xlim[2])    # simulate activity centers
4662 > sy <- runif(N,ylim[1],ylim[2])
4663 > S <- cbind(sx,sy)
4664 # Compute distance matrix:
4665 > D <- e2dist(S,traplocs) # distance of each individual from each trap

4666

4667 > alpha0 <- -2.5      # define parameters of encounter probability
4668 > sigma <- 0.5        # scale parameter of half-normal
4669 > alpha1 <- 1/(2*sigma*sigma) # convert to coefficient on distance

4670

4671 # Compute Probability of encounter:
4672 #

```

```

4673 > probcap <- plogis(-2.5)*exp( - alpha1*D*D)
4674
4675 # Generate the encounters of every individual in every trap
4676 > Y <- matrix(NA,nrow=N,ncol=ntraps)
4677 > for(i in 1:nrow(Y)){
4678   Y[i,] <- rbinom(ntraps,K,probcap[i,])
4679 }

```

4680 We remind the reader that, in presenting **R** or other code snippets throughout the
 4681 book, we will deviate from our standard variable expressions for some quantities. In
 4682 particular, we sometimes substitute words for integer variable designations: **nind** (for n),
 4683 **ntraps** (for J), and **nocc** (for K). In our opinion this leaves less to be inferred by the
 4684 reader in trying to understand code snippets.

4685 Subsequently we will generate data using this code packaged in an **R** function called
 4686 **simSCR0** in the package **scrbook** which takes a number of arguments including **discard0**
 4687 which, if TRUE, will return only the encounter histories for captured individuals. A second
 4688 argument is **array3d** which, if TRUE, returns the 3-dimensional encounter history array
 4689 instead of the aggregated **nind** \times **ntraps** encounter frequencies (see below). Finally we
 4690 provide a random number seed, **rnd** = 2013 to ensure repeatability of the analysis here.
 4691 We obtain a data set as above using the following command:

```
4692 > data <- simSCR0(discard0=TRUE, array3d=FALSE, rnd=2013)
```

4693 The **R** object **data** is a list, so let's take a look at what's in the list and then harvest some
 4694 of its elements for further analysis below.

```

4695 > names(data)
4696 [1] "Y"      "traplocs" "xlim"      "ylim"      "N"       "alpha0"    "beta"
4697 [8] "sigma"   "K"
4698
4699 ## Grab encounter histories from simulated data list
4700 > Y <- data$Y
4701 ## Grab the trap locations
4702 > traplocs <- data$traplocs

```

4703 5.5.1 Formatting and manipulating real data sets

4704 Conventional capture-recapture data are easily stored and manipulated as a 2-dimensional
 4705 array, an **nind** \times **K** (individuals by sample occasions) matrix, which is maximally informative
 4706 for any conventional capture-recapture model, but not for spatial capture-recapture
 4707 models. For SCR models we must preserve the spatial information in the encounter history
 4708 information. We will routinely analyze data from 3 standard formats:

- 4709 (1) The basic 2-dimensional data format, which is an **nind** \times **ntraps** encounter frequency
 4710 matrix such as that simulated previously. These are the total number of encounters in
 4711 each trap, summed over the K sample occasions.
- 4712 (2) The maximally informative 3-dimensional array, for which we establish here the con-
 4713 convention that it has dimensions **nind** \times **ntraps** \times **K**.

4714 (3) We use a compact format – the “encounter data file” – which we describe below in
 4715 Sec. 5.9.

4716 To simulate data in the most informative format - the “3-d array” - we can use the **R**
 4717 commands given previously but replace the last 4 lines with the following:

```
4718 > Y <- array(NA,dim=c(N,ntraps,K))
4719
4720 > for(i in 1:nrow(Y)){
4721   for(j in 1:ntraps){
4722     Y[i,j,1:K] <- rbinom(K,1,probcap[i,j])
4723   }
4724 }
```

4725 We see that a collection of K binary encounter events are generated for *each* individual
 4726 and for *each* trap. The probabilities of those Bernoulli trials are computed based on the
 4727 distance from each individual’s home range center and the trap (see calculation above),
 4728 and those are housed in the matrix `probcap`. Our data simulator function `simSRC0` will
 4729 return the full 3-d array if `array3d=TRUE` is specified in the function call. To recover the
 4730 2-d matrix from the 3-d array, and subset the 3-d array to individuals that were captured,
 4731 we do this:

```
4732 # Sum over the ‘‘replicates’’ dimension (3rd margin of the array)
4733 > Y2d <- apply(Y,c(1,2),sum)
4734
4735 # Compute how many times each individual was captured
4736 > ncaps <- apply(Y2d,1,sum)
4737
4738 # Keep those individuals that were captured
4739 > Y <- Y[ncaps>0,,]
```

5.6 FITTING MODEL SCR0 IN BUGS

4740 Clearly if we somehow knew the value of N then we could fit this model directly because,
 4741 in that case, it is a special kind of logistic regression model, one with a random effect (**s**)
 4742 that enters into the model in a peculiar fashion, and also with a distribution (uniform)
 4743 which we don’t usually think of as standard for random effects models. So our aim here is
 4744 to analyze the known- N problem, using our simulated data, as an incremental step in our
 4745 progress toward fitting more generally useful models. To begin, we use our simulator to
 4746 grab a data set and then harvest the elements of the resulting object for further analysis.

```
4747 > data <- simSRC0(discard0=FALSE,rnd=2013)
4748 > y <- data$Y
4749 > traplocs <- data$traplocs
4750
4751 # In this case nind=N because we’re doing the known-N problem
4752 #
4753 > nind <- nrow(y)
```

```

4754 > X <- data$traplocs
4755 > J <- nrow(X)    # number of traps
4756 > K <- data$K
4757 > xlim <- data$xlim
4758 > ylim <- data$ylim

```

4759 Note that we specify `discard0 = FALSE` so that we have a “complete” data set, i.e.,
 4760 one with the all-zero encounter histories corresponding to uncaptured individuals. Now,
 4761 within an **R** session, we can create the **BUGS** model file and fit the model using the
 4762 following commands.

```

4763 cat("
4764   model{
4765     alpha0 ~ dnorm(0,.1)
4766     logit(p0) <- alpha0
4767     alpha1 ~ dnorm(0,.1)
4768     sigma <- sqrt(1/(2*alpha1))
4769     for(i in 1:N){      # note N here -- N is KNOWN in this example
4770       s[i,1] ~ dunif(xlim[1],xlim[2])
4771       s[i,2] ~ dunif(ylim[1],ylim[2])
4772       for(j in 1:J){
4773         d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
4774         y[i,j] ~ dbin(p[i,j],K)
4775         p[i,j] <- p0*exp(- alpha1*d[i,j]*d[i,j])
4776       }
4777     }
4778   }
4779 ",file = "SCR0a.txt")

```

4780 This model describes the Gaussian encounter probability model, but it would be trivial
 4781 to modify that to various others including the logistic described above. One consequence
 4782 of using the half-normal is that we have to constrain the encounter probability to be in
 4783 $[0, 1]$ which we do here by defining `alpha0` to be the logit of the intercept parameter `p0`.
 4784 Note that the distance covariate is computed within the **BUGS** model specification given
 4785 the matrix of trap locations, `X`, which is provided to **WinBUGS** as data.

4786 Next we do a number of organizational activities including bundling the data for **Win-**
4787 BUGS, defining some initial values, the parameters to monitor and some basic MCMC
 4788 settings. We choose initial values for the activity centers `s` by generating uniform random
 4789 numbers in the state-space but, for the observed individuals, we replace those values by
 4790 each individual’s mean trap coordinate for all encounters

```

4791 ### Starting values for activity centers, s
4792 > sst <- cbind(runif(nind,xlim[1],xlim[2]),runif(nind,ylim[1],ylim[2]))
4793 > for(i in 1:nind){
4794   if(sum(y[i,])==0) next
4795   sst[i,1] <- mean( X[y[i,>0,1] )
4796   sst[i,2] <- mean( X[y[i,>0,2] )
4797 }

```

```

4798
4799 > data <- list (y=y, X=X, K=K, N=nind, J=J, xlim=xlim, ylim=ylim)
4800 > inits <- function(){
4801   list (alpha0=rnorm(1,-4,.4), alpha1=runif(1,1,2), s=sst)
4802 }
4803
4804 > library(R2WinBUGS)
4805 > parameters <- c("alpha0","alpha1","sigma")
4806 > out <- bugs (data, inits, parameters, "SCR0a.txt", n.thin=1, n.chains=3,
4807   n.burnin=1000,n.iter=2000,debug=TRUE,working.dir=getwd())

```

4808 There is little to say about the preceding operations other than to suggest that you might
 4809 explore the output and investigate additional analyses by running the `simSCR0` script
 4810 provided in the **R** package `scrbook`.

4811 For purposes here, we ran 1000 burn-in and 1000 post-burn-in iterations, and 3 chains,
 4812 to obtain 3000 posterior samples. Because we know N for this particular data set we only
 4813 have 2 parameters of the detection model to summarize (`alpha0` and `alpha1`), along with
 4814 the derived parameter σ , the scale parameter of the Gaussian kernel, i.e., $\sigma = \sqrt{1/(2\alpha_1)}$.
 4815 When the object `out` is produced we print a summary of the results as follows:

```

4816 > print(out,digits=2)
4817 Inference for Bugs model at "SCR0a.txt", fit using WinBUGS,
4818   3 chains, each with 2000 iterations (first 1000 discarded)
4819   n.sims = 3000 iterations saved
4820     mean    sd   2.5%   25%   50%   75%  97.5% Rhat n.eff
4821 alpha0   -2.50  0.22  -2.95  -2.65  -2.48  -2.34  -2.09  1.01   190
4822 alpha1    2.44  0.42   1.64   2.15   2.44   2.72   3.30  1.00   530
4823 sigma     0.46  0.04   0.39   0.43   0.45   0.48   0.55  1.00   530
4824 deviance 292.80 21.16 255.60 277.50 291.90 306.00 339.30 1.01   380
4825
4826
4827 [...some output deleted...]
4828

```

4829 We know the data were generated with `alpha0 = -2.5` and `alpha1 = 2`. The estimates
 4830 look reasonably close to those data-generating values and we probably feel pretty good
 4831 about the performance of the Bayesian analysis and MCMC algorithm that **WinBUGS**
 4832 cooked-up based on our sample size of 1 data set. It is worth noting that the `Rhat`
 4833 statistics indicate reasonable convergence but, as a practical matter, we might choose to
 4834 run the MCMC algorithm for additional time to bring these closer to 1.0 and to increase
 4835 the effective posterior sample size (`n.eff`). Other summary output includes “deviance”
 4836 and related things including the deviance information criterion (DIC). We discuss general
 4837 issues of convergence and other MCMC considerations in Chapt. 17, and DIC and model
 4838 selection in Chapt. 8.

5.7 UNKNOWN N

4839 In all real applications N is unknown. We handled this important issue in Chapt. 4
 4840 using the method of data augmentation (DA) which we apply here to achieve a realistic
 4841 analysis of model SCR0. As with the basic closed population models considered previously,
 4842 we formulate the problem by augmenting our observed data set with a number of “all-
 4843 zero” encounter histories - what we referred to in Chapt. 4 as potential individuals. If
 4844 n is the number of observed individuals, then let $M - n$ be the number of potential
 4845 individuals in the data set. For the 2-dimensional y_{ij} data structure (n individual $\times J$
 4846 traps encounter frequencies) we simply add additional rows of all-zero observations to
 4847 that data set. Because such “individuals” are unobserved, they therefore necessarily have
 4848 $y_{ij} = 0$ for all j . A data set, say with 4 traps and 6 individuals, augmented with 4
 4849 pseudo-individuals therefore might look like this:

```
4850 trap1 trap2 trap3 trap4
4851 [1,] 1 0 0 0
4852 [2,] 0 2 0 0
4853 [3,] 0 0 0 1
4854 [4,] 0 1 0 0
4855 [5,] 0 0 1 1
4856 [6,] 1 0 1 0
4857 [7,] 0 0 0 0
4858 [8,] 0 0 0 0
4859 [9,] 0 0 0 0
4860 [10,] 0 0 0 0
```

4861 We typically have more than 4 traps and, if we’re fortunate, many more individuals in
 4862 our data set.

4863 For the augmented data set, we introduce a set of binary latent variables (the data
 4864 augmentation variables), z_i , and the model is extended to describe $\Pr(z_i = 1)$ which is, in
 4865 the context of this problem, the probability that an individual in the augmented data set
 4866 is a member of the population of size N that was exposed to sampling. In other words,
 4867 if $z_i = 1$ for one of the all-zero encounter histories, this is implied to be a sampling zero
 4868 whereas observations for which $z_i = 0$ are “structural zeros” under the model. Under DA,
 4869 we also express the binomial observation model *conditional on z_i* as follows:

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

4870 where we see that the binomial probability evaluates to 0 if $z_i = 0$ (so y_{ij} is a fixed 0 in
 4871 that case) and evaluates to p_{ij} if $z_i = 1$.

4872 How big does the augmented data set have to be? We discussed this issue in Chapt. 4
 4873 where we noted that the size of the data set is equivalent to the upper limit of a uniform
 4874 prior distribution on N . Practically speaking, it should be sufficiently large so that the
 4875 posterior distribution for N is not truncated. On the other hand, if it is too large then
 4876 unnecessary calculations are being done. An approach to choosing M by trial-and-error
 4877 is indicated. Do a short MCMC run and then consider whether you need to increase M .
 4878 See Chapt. 17 for an example of this. Kéry and Schaub (2012, Chapt. 6) provide an
 4879 assessment of choosing M in closed population models. The useful thing about DA is that

4880 it removes N as an explicit parameter of the model. Instead, N is a derived parameter,
 4881 computed by $N = \sum_{i=1}^M z_i$. Similarly, *density*, D , is also a derived parameter computed
 4882 as $D = N/\text{area}(\mathcal{S})$.

4883 **5.7.1 Analysis using data augmentation in WinBUGS**

4884 We provide a complete **R** script for simulating and organizing a data set, and analyzing
 4885 the data in **WinBUGS**. As before we begin by obtaining a data set using our **simSCR0**
 4886 function and then harvesting the required data objects from the resulting data list. Note
 4887 that we use the **discard0=TRUE** option this time so that we get a “real looking” data set
 4888 with no all-zero encounter histories:

```
4889 ##  
4890 ## Simulate the data and extract the required objects  
4891 ##  
4892 > data <- simSCR0(discard0=TRUE,rnd=2013)  
4893 > y <- data$Y  
4894 > nind <- nrow(y)  
4895 > X <- data$traplocs  
4896 > K <- data$K  
4897 > J <- nrow(X)  
4898 > xlim <- data$xlim  
4899 > ylim <- data$ylim
```

4900 After harvesting the data we augment the data matrix y with $M - n$ all-zero encounter
 4901 histories, and create starting values for the variables z_i and also the activity centers s_i
 4902 of which, for each, we require M values. One thing to take care of in using the **BUGS**
 4903 engines is the starting values for the activity centers. It is usually helpful to start the s_i
 4904 for each observed individual at or near the trap(s) it was captured. All of this happens as
 4905 follows:

```
4906 ## Data augmentation  
4907 > M <- 200  
4908 > y <- rbind(y,matrix(0,nrow=M-nind,ncol=ncol(y)))  
4909 > z <- c(rep(1,nind),rep(0,M-nind))  
4910  
4911 ## Starting values for s  
4912 > sst <- cbind(runif(M,xlim[1],xlim[2]),runif(M,ylim[1],ylim[2]))  
4913 > for(i in 1:nind){  
4914   sst[i,1] <- mean( X[y[i,]>0,1] )  
4915   sst[i,2] <- mean( X[y[i,]>0,2] )  
4916 }
```

4917 Next, we write out the **BUGS** model specification and save it to an external file
 4918 called **SCR0b.txt**. The model specification now includes M encounter histories including
 4919 the augmented potential individuals, the data augmentation parameters z_i , and the data
 4920 augmentation parameter ψ :

```

4921 > cat("
4922 model{
4923   alpha0 ~ dnorm(0,.1)
4924   logit(p0) <- alpha0
4925   alphai ~ dnorm(0,.1)
4926   sigma <- sqrt(1/(2*alphai))
4927   psi ~ dunif(0,1)
4928
4929   for(i in 1:M){
5000     z[i] ~ dbern(psi)
5001     s[i,1] ~ dunif(xlim[1],xlim[2])
5002     s[i,2] ~ dunif(ylim[1],ylim[2])
5003     for(j in 1:J){
5004       d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
5005       y[i,j] ~ dbin(p[i,j],K)
5006       p[i,j] <- z[i]*p0*exp(- alphai*d[i,j]*d[i,j])
5007     }
5008   }
5009   N <- sum(z[])
5010   D <- N/64
5011 }
5012 ",file = "SCR0b.txt")

```

4943 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:

```

4946 > data <- list (y=y, X=X, K=K, M=M, J=J, xlim=xlim, ylim=ylim)
4947 > inits <- function(){
4948   list (alpha0=rnorm(1,-4,.4), alpha1=runif(1,1,2), s=sst, z=z)
4949 }
4950
4951 > library(R2WinBUGS)
4952 > parameters <- c("alpha0","alpha1","sigma","N","D")
4953 > out <- bugs (data, inits, parameters, "SCR0b.txt", n.thin=1,n.chains=3,
4954   n.burnin=1000,n.iter=2000,debug=TRUE,working.dir=getwd())

```

4955 Note the differences in this new **WinBUGS** model with that appearing in the known-
4956 N version – there are not many! The loop over individuals goes up to M now, and there is a
4957 model component for the DA variables z . We are also computing some derived parameters:
4958 population size $N(\mathcal{S})$ is computed by summing up all of the data augmentation variables
4959 z_i (as we've done previously in Chapt. 4) and density, D , is also a derived parameter,
4960 being a function of N . The input data has changed slightly too, as the augmented data
4961 set has more rows to include excess all-zero encounter histories. Previously we knew that
4962 $N = 100$ but in this analysis we pretend not to know N , but think that $N = 200$ is a
4963 good upper bound. This analysis can be run directly using the **SCR0bayes** function once
4964 the **scrbook** package is loaded, by issuing the following commands:

```

4965 > library(scrbook)
4966 > data <- simSCR0(discard0=TRUE,rnd=2013)
4967 > out1 <- SCR0bayes(data,M=200,engine="winbugs",ni=2000,nb=1000)

4968 Summarizing the output from WinBUGS produces:

4969 > print(out1,digits=2)
Inference for Bugs model at "SCR0b.txt", fit using WinBUGS,
 3 chains, each with 2000 iterations (first 1000 discarded)
n.sims = 3000 iterations saved
      mean    sd   2.5%   25%   50%   75%  97.5% Rhat n.eff
4974 alpha0   -2.57  0.23  -3.04  -2.72  -2.56  -2.41  -2.15  1.01   320
4975 alpha1    2.46  0.42   1.63   2.16   2.46   2.73   3.33  1.02   120
4976 sigma     0.46  0.04   0.39   0.43   0.45   0.48   0.55  1.02   120
4977 N        113.62 15.73  86.00 102.00 113.00 124.00 147.00 1.01   260
4978 D         1.78  0.25   1.34   1.59   1.77   1.94   2.30  1.01   260
4979 deviance 302.60 23.67 261.19 285.47 301.50 317.90 354.91 1.00  1400
4980
4981 [...some output deleted...]
4982

```

4983 The **Rhat** statistic (discussed in Secs. 3.5.2 and 17.4.5) for this analysis indicates
 4984 satisfactory convergence. We see that the estimated parameters (α_0 and α_1) are comparable
 4985 to the previous results obtained for the known- N case, and also not too different
 4986 from the data-generating values. The posterior of N overlaps the data-generating value
 4987 substantially.

4988 **Use of other BUGS engines: JAGS**

4989 There are two other popular **BUGS** engines in widespread use: **OpenBUGS** (Thomas
 4990 et al., 2006) and **JAGS** (Plummer, 2003). Both of these are easily called from **R**. **Open-**
 4991 **BUGS** can be used instead of **WinBUGS** by changing the package option in the **bugs**
 4992 call to **package='OpenBUGS'**. **JAGS** can be called using the function **jags()** in package
 4993 **R2jags** which has nearly the same arguments as **bugs()**. Or, it can be executed from the
 4994 **R** package **rjags** (Plummer, 2011) which has a slightly different implementation that we
 4995 demonstrate here as we reanalyze the simulated data set in the previous section (note:
 4996 the same **R** commands are used to generate the data and package the data, inits and
 4997 parameters to monitor). The function **jags.model** is used to initialize the model and run
 4998 the MCMC algorithm for an adaptive period during which tuning of the MCMC algorithm
 4999 might take place. These samples cannot be used for inference. Then the Markov chains
 5000 are updated using **coda.samples()** to obtain posterior samples for analysis, as follows:

```

5001 > jinit <- jags.model("SCR0b.txt", data=data, inits=inits,
5002           n.chains=3, n.adapt=1000)
5003 > jout <- coda.samples(jinit, parameters, n.iter=1000, thin=1)

```

5004 These commands can be executed using the function **SCR0bayes** provided with the **R**
 5005 package **scrbook**. Hobbs (2011) provides a good introduction to ecological modeling with
 5006 **JAGS** which we recommend.

Table 5.2. Posterior mean of model parameters for 4 different models fitted to a single simulated data set, and the effective home range area under each detection model.

	Gaussian	Cloglog	Exponential	Logit
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

5007 5.7.2 Implied home range area

5008 Here we apply the method described in Sec. 5.4 to compute the effective home range
 5009 area under different encounter probability models fit to simulated data. We simulated a
 5010 data set from the Gaussian kernel model as in Sec. 5.7 and then we fitted 4 models to it:
 5011 (1) the true data-generating Gaussian encounter probability model; (2) the “hazard” or
 5012 complementary log-log link model (Eq. 5.4.2); (3) the negative exponential model and (4)
 5013 the logit model (Eq. 5.2.2). We modified the function `SCR0bayes` for this purpose which
 5014 you should be able to do with little difficulty. We fit each model to the same simulated
 5015 data set using **WinBUGS**, based only on 1000 post-burn-in samples and 3 chains, which
 5016 produced the posterior summaries given in Table 5.2. The main thing we see is that, while
 5017 the implied home range area can vary substantially, there are smaller differences in the
 5018 estimated N and hence D .

5019 5.7.3 Realized and expected density

5020 In Bayesian analysis of the SCR model, we estimate a parameter N which is the size of the
 5021 population for the prescribed state-space (presumably the state-space is defined so as to be
 5022 relevant to where our traps were located, so N can be thought of as the size of the sampled
 5023 population). In the context of Efford and Fewster (2012) this is the *realized* population
 5024 size. Conversely, sometimes we see estimates of *expected* population size reported, which
 5025 are estimates of $\mathbb{E}(N)$, the expected size of some hypothetical, unspecified population.
 5026 Usually the distinction between realized and expected population size is not made in SCR
 5027 models, because almost everyone only cares about actual populations – and their realized
 5028 population size.

5029 If you do likelihood analysis of SCR models, then the distinction between realized and
 5030 expected is often discussed by whether the estimator is “conditional on N ” (realized) or
 5031 not (expected). The naming arises because in obtaining the MLE of N , its properties are
 5032 evaluated *conditional* on N – in particular, if the estimator is unbiased then $\mathbb{E}(\hat{N}|N) = N$
 5033 and $\text{Var}(\hat{N}|N) = \tilde{\sigma}_{\hat{N}}^2$ is the sampling variance. This does not conform to any concept or
 5034 quantity that is relevant to Bayesian inference. If we care about N for the population that
 5035 we sampled it is understood to be a realization of a random variable, but the relevance of
 5036 “conditional on N ” is hard to see. Bayesian analysis will provide a prediction of N that
 5037 is based on the posterior $[N|y, \theta]$ – which is certainly *not* conditional on N .

5038 There is a third type of inference objective that is relevant in practice and that is
 5039 prediction of N for a population that was not sampled – i.e., a “new” population. To

elaborate on this, consider a situation in which we are concerned about the tiger population in 2 distinct reserves in India. We do a camera trapping study on one of the reserves to estimate N_1 and we think the reserves are similar and homogeneous so we're willing to apply a density estimate based on N_1 to the 2nd reserve. For the 2nd reserve, do we want a prediction of the realized population size, N_2 , or do we want an estimates of its expected value? We believe the former is the proper quantity for inference about the population size in the 2nd reserve. An estimate of N_2 should include the uncertainty with which the mean is estimated (from reserve 1) and it should also include "process variation" for making the prediction of the latent variable N_2 .

As a practical matter, to do a Bayesian analysis of this you could just define the state-space to be the union of the two state-spaces, increase M so that the posterior of the total population size is not truncated, and then have MCMC generate a posterior sample of individuals on the joint state-space. You can tally-up the ones that are on \mathcal{S}_2 as an estimate of N_2 . Alternatively, we can define $\mu = \psi M/A_1$ and then simulate posterior samples of $N_s \sim \text{Binomial}(M, \mu A_2/M)$ for the new state-space area, A_2 .

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

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

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

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

The first part is estimation error and the 2nd component is the "process variance." If you do Bayesian analysis, then you don't have to worry too much about how to compute variances properly. You decide if you care about N , or its expected value, or predictions of some "new" N , and you tabulate the correct posterior distribution from your MCMC output.

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

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

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

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

5080 In summary, there are 3 basic inference problems that relate to estimating population
5081 size (or density):

- 5082 (1) What is the value of N for some population that was sampled. This is what Efford
5083 and Fewster call “realized N ” In general, we want the uncertainty to reflect having to
5084 estimate n_0 , the part of the population not seen.
5085 (2) We need to estimate N for some population that we didn’t sample but it is “similar”
5086 to the population that we have information on. In this case, we have to account for
5087 both variation in having to estimate parameters of the distribution of N and we have
5088 to account for process variation in N (i.e., due to the stochastic model of N).
5089 (3) In some extremely limited cases we might care about estimating the expected value of
5090 N , $\mathbb{E}(N)$. This is only useful as a hypothetical statement that we might use, e.g., if we
5091 were to establish a new million ha refuge somewhere, then we might say its expected
5092 population size is 200 tigers.

5.8 THE CORE SCR ASSUMPTIONS

5093 It’s always a good idea to sit down and reflect on the meaning of any particular model,
5094 its various assumptions, and what they mean in a specific context. From the statistician’s
5095 point of view, the basic assumption, the omnibus assumption, as in all of statistics, and
5096 for every statistical model, is that “the model is correctly specified”. So, naturally, that
5097 precludes everything that isn’t explicitly addressed by the model. To point this out to
5098 someone seems to cause a lot of anxiety, so we enumerate here what we think are the most
5099 important statistical assumptions of the basic SCR0 model:

- 5100 • **Demographic closure.** The model does not allow for demographic processes. There
5101 is no recruitment or entry into the sampled population. There is no mortality or exit
5102 from the sampled population.
- 5103 • **Geographic closure.** We assume no permanent emigration or immigration from the
5104 state-space. However, we allow for “temporary” movements around the state-space
5105 and variable exposure to encounter as a result. The whole point of SCR models is to
5106 accommodate this dynamic. In ordinary capture-recapture models we have to assume
5107 geographic closure to interpret N in a meaningful way.
- 5108 • **Activity centers are randomly distributed.** That is, uniformity and independence
5109 of the underlying point process s_1, \dots, s_N (see next section).
- 5110 • **Detection is a function of distance.** A detection model that describes how encounter
5111 probability declines as a function of distance from an individual’s home range center.
- 5112 • **Independence of encounters** among individuals. Encounter of any individual is
5113 independent of encounter of each other individual.
- 5114 • **Independence of encounters** of the same individual. Encounter of an individual
5115 in any trap is independent of its encounter in any other trap, and subsequent sample
5116 occasion.

5117 It’s easy to get worried and question the whole SCR enterprise just on the grounds that
5118 these assumptions combine to form such a simplistic model, one that surely can’t describe
5119 the complexity of real populations. On this sentiment, a few points are worth making.
5120 First, you don’t have inherently fewer assumptions by using an ordinary capture-recapture
5121 model but, rather, the SCR model relaxes a number of important assumptions compared

5122 to the non-spatial counterpart. For one, here, we're not assuming that p is constant for all
5123 individuals but rather that p varies substantially as a matter of the spatial juxtaposition of
5124 individuals with traps. So maybe the manner in which p varies isn't quite right, but that's
5125 not an argument that supports doing less modeling. Fundamentally a distance-based
5126 model for p has some basic biological justification in virtually every capture-recapture
5127 study. Secondly, for some of these core assumptions such as uniformity, and independence
5128 of individuals and of encounters, we expect a fair amount of robustness to departures.
5129 They function primarily to allow us to build a model and an estimation scheme and we
5130 don't usually think they represent real populations (of course, no model does!). Third, we
5131 can extend these assumptions in many different ways and we do that to varying extents in
5132 this book, and more work remains to be done in this regard. Forth, we can also evaluate
5133 the reasonableness of the assumptions formally in some cases using standard methods of
5134 assessing model fit (Chapt. 8).

5135 Finally, we return back to our sentiment about the omnibus assumptions which is
5136 that the model is properly specified. This precludes *everything* that isn't in the model.
5137 Sometimes you see in capture-recapture literature statements like "we assume no marks
5138 are lost", "marks are correctly identified" and similar things. We might as well also
5139 assume that, a shopping mall is not built, or a meteor does not crash down into our study
5140 area, the sun does not go super-nova, and so forth. Our point is that we should separate
5141 statistical assumptions about model parameters or aspects of the probability model from
5142 what are essentially logistical or operational assumptions about how we interpret our data,
5143 or based on our ability to conduct the study. It is pointless to enumerate all of the possible
5144 explanations for apparent *departures*, because there are an infinity of such cases.

5.9 WOLVERINE CAMERA TRAPPING STUDY

5145 We provide an illustration of some of the concepts we've introduced previously in this
5146 chapter by analyzing data from a camera trapping data from a study of wolverines *Gulo*
5147 *gulo* (Magoun et al., 2011; Royle et al., 2011b). The study took place in SE Alaska (Fig.
5148 5.4) where 37 cameras were operational for variable periods of time (min = 5 days, max
5149 = 108 days, median = 45 days). A consequence of this is that the number of sampling
5150 occasions, K , is variable for each camera. Thus, we must provide a vector of sample sizes
5151 as data to **BUGS** and modify the model specification in Sec. 5.7 accordingly.

5152 5.9.1 Practical data organization

5153 To carry out an analysis of these data, we require the matrix of trap coordinates and the
5154 encounter history data. We usually store data in 2 distinct data files which contain all
5155 the information needed for an analysis. These files are

- 5156 • The encounter data file (EDF) containing a record of which traps and when each individual
5157 encounter occurred.
- 5158 • The trap deployment file (TDF) which contains the coordinates of each trap, along with
5159 information indicating which sample occasions each trap was operating.

5160 **Encounter Data File (EDF)** – We store the encounter data in the an efficient file
5161 format which is easily manipulated in **R** and easy to create in Excel and other spreadsheets

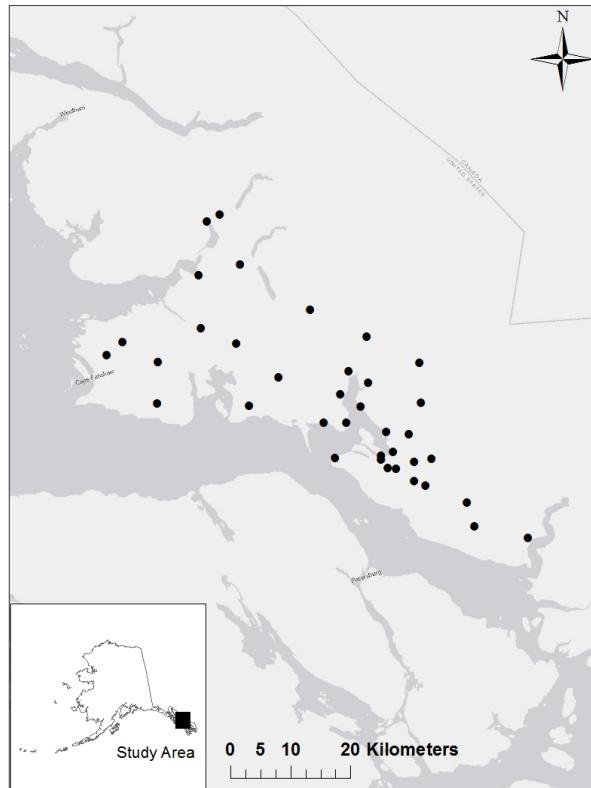


Figure 5.4. Wolverine camera trap locations (black dots) from a study that took place in SE Alaska. See Magoun et al. (2011) for details.

which are widely used for data management. The file structure is a simple matrix with 4 columns, those being: (1) **session ID**: the trap *session* which usually corresponds to a year or a primary period in the context of a Robust Design situation, but it could also correspond to a distinct spatial unit (see Sec. 6.5.4 and Chapt. 14). For a single-year study (as considered here) this should be an integer that is the same for all records; (2) **individual ID**: the individual identity, being an integer from 1 to n (repeated for multiple captures of the same individual) indicating which individual the record (row) of the matrix belongs to; (3) **occasion ID**: The integer sample occasion which generated the record, and (4) **trap ID**: the trap identity, an integer from 1 to J , the number of traps. The structure of the EDF is the same as used in the **secr** package (Efford, 2011) and similar to that used in the **SPACECAP** (Gopalaswamy et al., 2012a), and **SCRbayes** (Russell et al., 2012) packages, both of which have a 3-column format (**trapID**, **indID**,

5174 **sampID**). We note that the naming of the columns is irrelevant as far as anything we do in
 5175 this book, although **secr** and other software may have requirements on variable naming.

5176 To illustrate this format, the wolverine data are available in the package **scrbook** by
 5177 typing:

5178 > **data(wolverine)**

5179 which contains a list having elements **wcaps** (the EDF) and **wtraps** (the TDF). We see
 5180 that **wcaps** has 115 rows, each representing a unique encounter event including the trap
 5181 identity, the individual identity and the sample occasion index (**sample**). The first 5 rows
 5182 of **wcaps** are:

```
5183 > wolverine$wcaps[1:5,]
5184   year individual day trap
5185 [1,]    1          2 127   1
5186 [2,]    1          2 128   1
5187 [3,]    1          2 129   1
5188 [4,]    1         18 130   1
5189 [5,]    1          3 106   2
```

5190 The 1st column here, labeled **year**, is an integer indicating the year or session of the
 5191 encounter. All these data come from a single year (2008) and so **year** is set to 1. Variable
 5192 **individual** is an integer identity of each individual captured, **day** is the sample occasion of
 5193 capture (in this case, the sample occasions correspond to days), and **trap** is the integer trap
 5194 identity. The variable **trapid** will have to correspond to the row of a matrix containing
 5195 the trap coordinates - in this case the TDF file **wtraps** which we describe further below.

5196 Note that the information provided in this encounter data file **wcaps** does not repre-
 5197 sent a completely informative summary of the data. For example, if no individuals were
 5198 captured in a certain trap or during a certain period, then this compact data format will
 5199 have no record. Thus we will need to know *J*, the number of traps, and *K*, the number of
 5200 sample occasions when reformatting this SCR data format into a 2-d encounter frequency
 5201 matrix or 3-d array. In addition, the encounter data file does not provide information
 5202 about which periods each trap was operated. This additional information is also neces-
 5203 sary as the trap-specific sample sizes must be passed to **BUGS** as data. We provide this
 5204 information along with trap coordinates, in the “trap deployment file” (TDF) which is
 5205 described below.

5206 For our purposes, we need to convert the **wcaps** file into the $n \times J$ array of binomial
 5207 encounter frequencies, although more general models might require an encounter-history
 5208 formulation of the model which requires a full 3-d array. To obtain our encounter frequency
 5209 matrix, we do this the hard way by first converting the encounter data file into a 3-d array
 5210 and then summarize to trap totals. We have a handy function **SCR23darray** which takes
 5211 the compact encounter data file, and converts it to a 3-d array, and then we use the **R**
 5212 function **apply** to summarize over the sample occasion dimension (by convention here,
 5213 this is the 2nd dimension). To apply this to the wolverine data in order to compute the
 5214 3-d array we do this:

```
5215 > y3d <- SCR23darray(wolverine$wcaps,wolverine$wtraps)
5216 > y <- apply(y3d,c(1,2),sum)
```

5217 See the help file for more information on **SCR23darray**. The 3-d array is necessary to
 5218 fit certain types of models (e.g., behavioral response) and this is why we sometimes will
 5219 require this maximally informative 3-d data format but, here, we analyze the summarized
 5220 data.

5221 **Trap Deployment File (TDF)** – The other important information needed to fit SCR
 5222 models is the “trap deployment file” (TDF) which provides additional information not
 5223 contained in the encounter data file. The traps file has $K + 3$ columns. The first column is
 5224 assumed to be a trap identifier, columns 2 and 3 are the easting and northing coordinates
 5225 (assumed to be in a Euclidean coordinate system), and columns 4 to $K + 3$ are binary
 5226 indicators of whether each trap was operational during each sample occasion. The first 10
 5227 rows (out of 37) and 10 columns (out of 167) of the trap deployment file for the wolverine
 5228 data are shown as follows:

```
5229 > wolverine$wtraps[1:10,1:10]
5230
5231      Easting Northing 1 2 3 4 5 6 7 8
5232 1   632538  6316012 0 0 0 0 0 0 0 0
5233 2   634822  6316568 1 1 1 1 1 1 1 1
5234 3   638455  6309781 0 0 0 0 0 0 0 0
5235 4   634649  6320016 0 0 0 0 0 0 0 0
5236 5   637738  6313994 0 0 0 0 0 0 0 0
5237 6   625278  6318386 0 0 0 0 0 0 0 0
5238 7   631690  6325157 0 0 0 0 0 0 0 0
5239 8   632631  6316609 0 0 0 0 0 0 0 0
5240 9   631374  6331273 0 0 0 0 0 0 0 0
5241 10  634068  6328575 0 0 0 0 0 0 0 0
```

5242 This tells us that trap 2 was operated during occasions (days) 1-7 but the other traps
 5243 were not operational during those periods. It is extremely important to recognize that
 5244 each trap was operated for a variable period of time and thus the binomial “sample size”
 5245 is different for each, and this needs to be accounted for in the **BUGS** model specification.
 5246 To compute the vector of sample sizes K , and extract the trap locations, we do this:

```
5247 > traps <- wolverine$wtraps
5248 > traplocs <- traps[,1:2]
5249 > K <- apply(traps[,3:ncol(traps)],1,sum)
```

5250 This results in a matrix **traplocs** which contains the coordinates of each trap and a vector
 5251 K containing the number of days that each trap was operational. We now have all the
 5252 information required to fit a basic SCR model in **BUGS**.

5253 Summarizing the data for the wolverine study, we see that 21 unique individuals were
 5254 captured a total of 115 times. Most individuals were captured 1-6 times, with 4, 1, 4, 3, 1,
 5255 and 2 individuals captured 1-6 times, respectively. In addition, 1 individual was captured
 5256 each 8 and 14 times and 2 individuals each were captured 10 and 13 times. The number
 5257 of unique traps that captured a particular individual ranged from 1-6, with 5, 10, 3, 1, 1,
 5258 and 1 individual captured in each of 1 to 6 different traps, respectively, for a total of 50
 5259 unique wolverine-trap encounters. These numbers might be hard to get your mind around

5260 whereas some tabular summary is often more convenient. For that it seems natural to
 5261 tabulate individuals by trap and total encounter frequencies. The spatial information in
 5262 SCR data is based on multi-trap captures, and so, it is informative to understand how
 5263 many unique traps each individual is captured in, and the total number of encounters.
 5264 For the wolverine data, we reproduce Table 1 from Royle et al. (2011b) as Table 5.3.

Table 5.3. Individual frequencies of capture for wolverines captured in camera traps in 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

5265 5.9.2 Fitting the model in WinBUGS

5266 Here we fit the simplest SCR model with the Gaussian encounter probability model,
 5267 although we revisit these data and fit additional models in later chapters. Model SCR0 is
 5268 summarized by the following 4 elements:

- 5269 (1) $y_{ij} | \mathbf{s}_i \sim \text{Binomial}(K, z_i p_{ij})$
- 5270 (2) $p_{ij} = p_0 \exp(-\alpha_1 ||\mathbf{x}_j - \mathbf{s}_i||^2)$
- 5271 (3) $\mathbf{s}_i \sim \text{Uniform}(\mathcal{S})$
- 5272 (4) $z_i \sim \text{Bernoulli}(\psi)$

5273 We assume customary flat priors on the structural (hyper-) parameters of the model,
 5274 $\alpha_0 = \text{logit}(p_0)$, α_1 and ψ .

5275 It remains to define the state-space \mathcal{S} . For this, we nested the trap array (Fig. 5.4)
 5276 in a rectangular state-space extending 20 km beyond the traps in each cardinal direction.
 5277 We scaled the coordinate system so that a unit distance was equal to 10 km, producing a
 5278 rectangular state-space of dimension 9.88×10.5 units ($\text{area} = 10374 \text{ km}^2$) within which
 5279 the trap array was nested. As a general rule, we recommend scaling the state-space so
 5280 that it is defined near the origin $(x, y) = (0, 0)$. While the scaling of the coordinate system
 5281 is theoretically irrelevant, a poorly scaled coordinate system can produce Markov chains
 5282 that mix poorly. The buffer of the state space should be large enough so that individuals
 5283 beyond the state-space boundary are not likely to be encountered (Sec. 5.3.1). To evaluate
 5284 this, we fit models for various choices of a rectangular state-space based on buffers from
 5285 1.0 to 5.0 units (10 km to 50 km). In the **R** package **scrbook** we provide a function
 5286 **wolvSCR0** which will fit model SCR0. For example, to fit the model in **WinBUGS** using
 5287 data augmentation with $M = 300$ potential individuals, using 3 Markov chains each of

Table 5.4. Posterior summaries of SCR model parameters for the wolverine camera trapping data from SE Alaska, using state-space buffers from 10 up to 50 km. Each analysis was based on 3 chains, 12000 iterations, 2000 burn-in, for a total of 30000 posterior samples.

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

5288 12000 total iterations, discarding the first 2000 as burn-in, we execute the following R
 5289 commands:

```
5290 > library(scrbook)
5291 > data(wolverine)
5292 > traps <- wolverine$wtraps
5293 > y3d <- SCR23darray(wolverine$wcaps,wolverine$wtraps)
5294 > wolv <- wolvSCR0(y3d,traps,nb=2000,ni=12000,buffer=1,M=300)
```

5295 The argument **buffer** determines the buffer size of the state-space in the scaled units
 5296 (i.e., 10 km). Note that this analysis takes between 1-2 hours on many machines (in 2013)
 5297 so we recommend testing it with lower values of M and fewer iterations. The posterior
 5298 summaries are shown in Table 5.9.2.

5299 5.9.3 Summary of the wolverine analysis

5300 We see that the estimated density is roughly consistent as we increase the state-space
 5301 buffer from 15 to 55 km. We do note that the data augmentation parameter ψ (and,
 5302 correspondingly, N) increase with the size of the state space in accordance with the deter-
 5303 ministic relationship $N = D * A$. However, density is more or less constant as we increase
 5304 the size of the state-space beyond a certain point. For the 10 km state-space buffer, we see
 5305 a slight effect on the posterior distribution of D because the state-space is not sufficiently
 5306 large. The full results from the analysis based on 20 km state-space buffer are given in
 5307 Table 5.5.

5308 Our point estimate of wolverine density from this study, using the posterior mean from
 5309 the state-space based on the 20 km buffer, is approximately 5.77 individuals/1000 km²
 5310 with a 95% posterior interval of [3.86, 8.29]. Density is estimated imprecisely which might
 5311 not be surprising given the low sample size ($n = 21$ individuals!). This seems to be a
 5312 basic feature of carnivore studies although it should not (in our view) preclude the study
 5313 of their populations by capture-recapture nor attempts to estimate density or vital rates.

Table 5.5. Posterior summaries of SCR model parameters for the wolverine camera trapping data from SE Alaska. The model was run with the trap array centered in a state-space with a 20 km rectangular buffer.

parameter	mean	SD	2.5%	25%	50%	75%	97.5%	Rhat
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

5314 It is worth thinking about this model, and these estimates, computed under a rect-
 5315 angular state space roughly centered over the trapping array (Fig. 5.4). Does it make
 5316 sense to define the state-space to include, for example, ocean? What are the possible
 5317 consequences of this? What can we do about it? There's no reason at all that the state
 5318 space has to be a regular polygon – we defined it as such here strictly for convenience and
 5319 for ease of implementation in **WinBUGS** where it enables us to specify the prior for the
 5320 activity centers as uniform priors for each coordinate. While it would be possible to define
 5321 a more realistic state-space using some general polygon GIS coverage, it might take some
 5322 effort to implement that in the **BUGS** language but it is not difficult to devise custom
 5323 MCMC algorithms to do that (see Chapt. 17). Alternatively, we recommend using a
 5324 discrete representation of the state-space – i.e., approximate \mathcal{S} by a grid of G points. We
 5325 discuss this in Sec. 5.10.

5326 5.9.4 Wolverine space usage

5327 The parameter α_1 is related to the home range radius (Sec. 5.4). For the Gaussian model
 5328 we interpret the scale parameter σ , related to α_1 by $\alpha_1 = 1/(2\sigma^2)$, as the radius of a
 5329 bivariate normal model of space usage. In this case $\sigma = 0.64$ standardized units (10 km),
 5330 which corresponds to $0.64 \times 10 = 6.4$ km. It can be argued then that 95% of space used
 5331 by an individual is within $6.4 \times \sqrt{5.99} = 15.66$ km of the home range center. The effective
 5332 “home range area” is then the area of this circle, which is $\pi \times 15.66^2 = 770.4$ km². Using
 5333 our handy function **hra** we do this:

```
5334 hra(pGauss1,parms=c(-2,1/(2*.64*.64)),xlim=c(-1,7),ylim=c(-1,7))  

5335 [1] 7.731408
```

5337 which is in units of 100 km², so 773.1. The difference in this case is due to numerical
 5338 approximation of our all-purpose tool **hra**. This home range size is relatively huge for
 5339 measured home ranges, which range between 100 and 535 km² (Whitman et al., 1986).

5340 Royle et al. (2011b) reported estimates for σ in the range 6.3 – 9.8 km depending on
 5341 the model, which isn't too different than here¹. However, these estimates are larger than

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 $\text{cloglog}(p_{ij}) = \alpha_0 -$

5342 the typical home range sizes suggested in the literature. One possible explanation is that
 5343 if a wolverine is using traps as a way to get yummy chicken, so it's moving from trap to
 5344 trap instead of adhering to "normal" space usage patterns, then the implied home range
 5345 size might not be worth much biologically. Thus, interpretation of detection models in
 5346 terms of home range area depends on some additional context or assumptions, such as
 5347 that traps don't effect individual space usage patterns. As such, we caution against direct
 5348 biological interpretations of home range area based on σ , although SCR models can be
 5349 extended to handle more general, non-Euclidean, patterns of space usage. See Chaps. 12
 5350 and 13.

5351 We can calibrate the desired size of the state-space by looking at the estimated home
 5352 range radius of the species. We should target a buffer of width 2 to $3 \times \sigma$ in order that
 5353 the probability of encountering an individual is very close to 0 beyond the prescribed
 5354 state-space. Essentially, by specifying a state-space, we're setting $p = 0$ for individuals
 5355 beyond the prescribed state-space. For the wolverine data, with σ in the range of 6-9 km,
 5356 a state-space buffer of 20 km is sufficiently large.

5.10 USING A DISCRETE HABITAT MASK

5357 The SCR model developed previously in this chapter assumes that individual activity
 5358 centers are distributed uniformly over the prescribed state-space. Clearly this will not
 5359 always be a reasonable assumption. In Chapt. 11, we develop models that allow explicitly
 5360 for non-uniformity of the activity centers by modeling covariate effects on density. A
 5361 simplistic method of affecting the distribution of activity centers, which we address here,
 5362 is to modify the shape and organization of the state-space explicitly. For example, we
 5363 might be able to classify the state-space into distinct blocks of habitat and non-habitat.
 5364 In that case we can remove the non-habitat from the state-space and assume uniformity of
 5365 the activity centers over the remaining portions judged to be suitable habitat. There are
 5366 several ways to approach this: We can use a grid of points to represent the state-space, i.e.,
 5367 by the set of coordinates s_1, \dots, s_G , and assign equal probabilities to each possible value.
 5368 Alternatively, we can retain the continuous formulation of the state-space but attempt
 5369 to describe constraints analytically, or we can use polygon clipping methods to enforce
 5370 constraints on the state-space in the MCMC analysis. We focus here on the formulation of
 5371 the basic SCR model in terms of a discrete state-space but in Chapt. 17 we demonstrate
 5372 the latter approach based on using polygon operations to define an irregular state-space.
 5373 Use of a discrete state-space can be computationally expensive in **WinBUGS**. That said,
 5374 it isn't too difficult to perform the MCMC calculations in **R** (discussed in Chapt. 17).
 5375 The **R** package **SPACECAP** (Gopalswamy et al., 2012a) arose from the **R** implementation
 5376 of the SCR model in Royle et al. (2009a).

5377 While clipping out non-habitat seems like a good idea, we think investigators should
 5378 go about this very cautiously. We might prefer to do it when non-habitat represents a
 5379 clear-cut restriction on the state-space such as a reserve boundary or a lake, ocean or
 5380 river. But, having the capability to do this also causes people to start defining "habitat"
 5381 vs. "non-habitat" based on their understanding of the system whereas it can't be known

$\frac{1}{2\sigma^2} * d_{ij}^2$, 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.

5382 whether the animal being studied has the same understanding. Moreover, differentiating
 5383 the landscape by habitat or habitat quality must affect the geometry and morphology of
 5384 home ranges (see Chapt. 13) much more so than the plausible locations of activity centers.
 5385 That is, a home range centroid could, in actual fact, occur in a shopping mall parking lot
 5386 if there is pretty good habitat around the shopping mall, so there is probably no sense
 5387 preclude it as the location for an activity center. It would generally be better to include
 5388 some definition of habitat quality in the model for the detection probability (Royle et al.,
 5389 2012a) which we address in Chaps. 12 and 13.

5390 5.10.1 Evaluation of coarseness of habitat mask

5391 The coarseness of the state-space should not really have much of an effect on estimates
 5392 if the grain is sufficiently fine relative to typical animal home range sizes. Why is this?
 5393 We have two analogies that can help us understand. First is the relationship to model
 5394 M_h . As noted in Sec. 5.3.2 above, we can think about SCR models as a type of finite
 5395 mixture (Norris and Pollock, 1996; Pledger, 2004) where we are fortunate to be able to
 5396 obtain direct information about which group individuals belong to (group being location
 5397 of activity center). In the standard finite mixture models we typically find that a small
 5398 number of groups (e.g., 2 or 3 at the most) can explain high levels of heterogeneity and
 5399 are adequate for most data sets of small to moderate sample sizes. We therefore expect a
 5400 similar effect in SCR models when we discretize the state-space. We can also think about
 5401 discretizing the state-space as being related to numerical integration where we find (see
 5402 Chapt. 6) that we don't need a very fine grid of support points to evaluate the integral to
 5403 a reasonable level of accuracy. We demonstrate this here by reanalyzing simulated data
 5404 using a state-space defined by a different number of support points. We provide an **R**
 5405 script called **SCR0bayesDss** in the **R** package **scrbook**. We note that for this comparison
 5406 we generated the actual activity centers as a continuous random variable and thus the
 5407 discrete state-space is, strictly speaking, an approximation to truth. That said, we regard
 5408 all state-space specifications as approximations to truth in the sense that they represent
 5409 a component of the SCR model.

5410 As with our **R** function **SCR0bayes**, the modification **SCR0bayesDss** will use either
 5411 **WinBUGS** or **JAGS**. In addition, it requires a grid resolution argument (**ng**) which
 5412 is the dimension of 1 side of a square state-space. To execute this function we do, for
 5413 example:

```
5414 > library(scrbook)
5415 > data <- simSCR0(discard0=TRUE,rnd=2013)    # generate data set
5416
5417 # run with JAGS
5418 > out1 <- SCR0bayesDss(data,ng=8,M=200,engine="jags",ni=2000,nb=1000)
5419
5420 # run with WinBUGS
5421 > out2 <- SCR0bayesDss(data,ng=8,M=200,engine="winbugs",ni=2000,nb=1000)
```

5422 We fit this model to the same simulated data set for 6×6 , 9×9 , 12×12 , 15×15
 5423 state-space grids. For **WinBUGS**, we used 3 chains of 5000 total length with 1000 burn-
 5424 in, which yields 12000 total posterior samples. Summary results are shown in Table 5.6.

Table 5.6. Comparison of the effect of state-space grid coarseness on estimates of N for a simulated data set. Posterior summaries and run time are given. Results obtained using **WinBUGS** run from R2WinBUGS.

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

5425 The results are broadly consistent except for the 6×6 case. We see that the run time
 5426 increases with the size of the state-space grid (not unexpected), such that we imagine it
 5427 would be impractical to run models with more than a few hundred state-space grid points.
 5428 We found (not shown here) that the runtime of **JAGS** is much faster and, furthermore,
 5429 relatively *constant* as we increase the grid size. We suspect that **WinBUGS** is evaluating
 5430 the full-conditional for each activity center at all G possible values whereas it may be
 5431 that **JAGS** is evaluating the full-conditional only at a subset of values or perhaps using
 5432 previous calculations more effectively. While this might suggest that one should always
 5433 use **JAGS** for this analysis, we found in our analysis of the wolverine (next section) that
 5434 **JAGS** could be extremely sensitive to starting values, producing MCMC algorithms that
 5435 often simply do not work for some problems, so be careful when using **JAGS**. To improve
 5436 its performance, always start the latent activity centers at values near where individuals
 5437 were captured. The performance of either should improve if we compute the full distance
 5438 matrix outside of **BUGS** and pass it as data, although we haven't fully evaluated this
 5439 approach.

5440 5.10.2 Analysis of the wolverine camera trapping data

5441 We reanalyzed the wolverine data using discrete state-space grids with points spaced by
 5442 2, 4 and 8 km (see Fig. 5.5). These were constructed from a 40 km buffered state-space,
 5443 and deleting the points over water (see Royle et al., 2011b). Our interest in doing this
 5444 was to evaluate the relative influence of grid resolution on estimated density because the
 5445 coarser grids will be more efficient from a computational stand-point and so we would
 5446 prefer to use them, but only if there is no strong influence on estimated density. The
 5447 posterior summaries for the 3 habitat grids are given in Table 5.7. We see that the
 5448 density estimates are quite a bit larger than obtained in our analysis (Table 5.9.2) based
 5449 on a rectangular, continuous state-space. We also see that there are slight differences
 5450 depending on the resolution of the state-space grid. Interestingly, the effectiveness of the
 5451 MCMC algorithms, as measured by effective sample size (`n.eff`) is pretty remarkably
 5452 different. Furthermore, the finest grid resolution (2 km spacing) took about 6 days to run
 5453 and thus it would not be practical for large problems or with many models.

5.11 SUMMARIZING DENSITY AND ACTIVITY CENTER LOCATIONS

5454 One of the most useful aspects of SCR models is that they are parameterized in terms of
 5455 individual locations – i.e., *where* each individual lives – and, thus, we can compute many

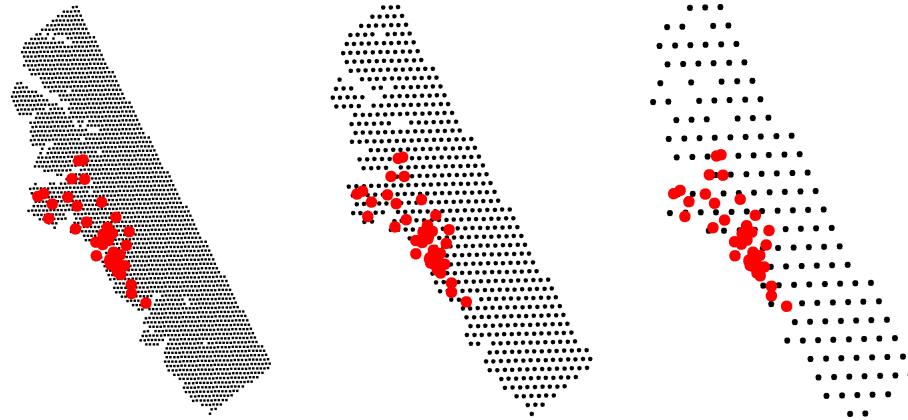


Figure 5.5. Three habitat mask grids used in the comparison of the effect of pixel size on the estimated density surface of wolverines. The 3 cases are 2 (left), 4 (center) and 8 (right) km spacing of state-space points, extending 40 km from the vicinity of the trap array.

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

5.11.1 Constructing density maps

Because SCR models are spatially-explicit, it is natural to want to summarize the results of fitting a model by producing a map of density. Using Bayesian analysis by MCMC, it is most easy to make a map of *realized* density. We can do this by tallying up the number of activity centers \mathbf{s}_i in pixels of arbitrary size and then producing a nice multi-color spatial plot of the result. Specifically, let $B(\mathbf{x})$ indicate a pixel centered at \mathbf{x} then

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

(here, $I(arg)$ is the indicator function which evaluates to 1 if arg is true, and 0 otherwise) is the population size of pixel $B(\mathbf{x})$, and $D(\mathbf{x}) = N(\mathbf{x})/\|B(\mathbf{x})\|$ is the local density. Note

Table 5.7. Posterior summaries for the wolverine camera trapping data, using model SCR0, with a Gaussian hazard encounter probability model, and a discrete habitat mask of 3 different resolutions: 2, 4 and 8 km. Parameters are λ_0 = baseline encounter rate, $p_0 = 1 - \exp(-\lambda_0)$, σ is the scale parameter of the Gaussian kernel, ψ is the data augmentation parameter, N and D are population size and density, respectively. Models fitted using **WinBUGS**, 3 chains, each with 11000 iterations (first 1000 discarded) producing 30000 posterior samples.

2 km spacing										
	Mean	SD	2.5%	25%	50%	75%	97.5%	Rhat	n.eff	
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	

that these $N(\mathbf{x})$ parameter are just “derived parameters” as we normally obtain from posterior output using the appropriate Monte Carlo average (see Chapt. 3).

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

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

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

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

```
5484 > xg <- seq(xlim[1], xlim[2], , 50)
5485 > yg <- seq(ylim[1], ylim[2], , 50)
```

5486 **Step 2:** Extract the MCMC histories for the activity centers and the data augmentation
 5487 variables. Note that these are each $N \times \text{niter}$ matrices. Here we do this assuming that
 5488 **WinBUGS** was run producing the **R** object named **out**:

```
5489 > Sxout <- out$sims.list$s[,1]
5490 > Syout <- out$sims.list$s[,2]
5491 > z <- out$sims.list$z
```

5492 **Step 3:** We associate each coordinate with the proper pixel using the **R** command **cut()**.
 5493 Note that we keep only the activity centers for which $z = 1$ (i.e., individuals that belong
 5494 to the population of size N):

```
5495 > Sxout <- cut(Sxout[z==1], breaks=xg, include.lowest=TRUE)
5496 > Syout <- cut(Syout[z==1], breaks=yg, include.lowest=TRUE)
```

5497 **Step 4:** Use the **table()** command to tally up how many activity centers are in each
 5498 $B(x)$:

```
5499 > Dn <- table(Sxout, Syout)
```

5500 **Step 5:** Use the **image()** command to display the resulting matrix.

```
5501 > image(xg, yg, Dn/nrow(z), col=terrain.colors(10))
```

5502 It is worth emphasizing here that density maps will not usually appear uniform despite
 5503 that we have assumed that activity centers are uniformly distributed. This is because
 5504 the observed encounters of individuals provide direct information about the location of
 5505 the $i = 1, 2, \dots, n$ activity centers and thus their “estimated” locations will be affected
 5506 by the observations. In a limiting sense, were we to sample space intensely enough,
 5507 every individual would be captured a number of times and we would have considerable
 5508 information about all N point locations. Consequently, the uniform prior would have
 5509 almost no influence at all on the estimated density surface in this limiting situation.
 5510 Thus, in practice, the influence of the uniformity assumption decreases as the fraction of
 5511 the population encountered, and the total number of encounters per individual, increases.

5512 **On the non-intuitiveness of `image()`** – the **R** function **image()**, invoked for a
 5513 matrix M by **image(M)**, might not be very intuitive to some – it plots $M[1, 1]$ in the lower
 5514 left corner. If you want $M[]$ to be plotted “as you look at it” then $M[1, 1]$ should be in the
 5515 upper left corner. We have a function **rot()** which does that. If you do **image(rot(M))**
 5516 then it puts it on the monitor as if it was a map you were looking at. You can always
 5517 specify the x - and y -labels explicitly as we did above.

5518 **Spatial dot plots** – A cruder version of the density map can be made using our
 5519 “spatial dot map” function **spatial.plot** (in **scrbook**). This function requires, as input,
 5520 point locations and the value to be displayed. A simplified version of this function is as
 5521 follows:

```

5522 > spatial.plot <- function(x,y){
5523   nc <- as.numeric(cut(y,20))
5524   plot(x,pch=" ")
5525   points(x,pch=20,col=topo.colors(20)[nc],cex=2)
5526   image.scale(y,col=topo.colors(20))
5527 }
5528 #
5529 # To execute the function do this:
5530 #
5531 > spatial.plot(cbind(xg,yg), Dn/nrow(z))

```

5.11.2 Example: Wolverine density map

5532 We return to the wolverine study which took place in 2008 in SE Alaska (Fig. 5.4) and
 5533 we produce a density map of wolverines from that analysis. We include the function
 5534 **SCRdensity** which requires a specific data structure as shown below. In particular, we
 5535 have to package up the MCMC history for the activity centers and the data augmentation
 5536 variables z into a list. This also requires that we add those variables to the parameters-
 5537 to-be-monitored list when we pass things to **BUGS**.

5538 We used the posterior output from the wolverine model fitted previously to compute
 5539 a relatively coarse version of a density map, using 100 pixels in a 10×10 grid (Fig. 5.6
 5540 top panel) and using 900 pixels arranged in a 30×30 grid (Fig. 5.6 lower panel) for a
 5541 fine-scale map. The **R** commands for producing such a plot (for a short MCMC run) are
 5542 as follows:

```

5544 > library(scrbook)
5545 > data(wolverine)
5546 > traps <- wolverine$wtraps
5547 > y3d <- SCR23darray(wolverine$wcaps,wolverine$wtraps)
5548
5549 # this takes 341 seconds on a standard CPU circa 2011
5550 > out <- wolvSCRO(y3d,traps,nb=1000,ni=2000,buffer=1,M=100,keepz=TRUE)
5551
5552 > Sx <- out$sims.list$s[,1]
5553 > Sy <- out$sims.list$s[,2]
5554 > z <- out$sims.list$z
5555 > obj <- list(Sx=Sx,Sy=Sy,z=z)
5556 > tmp <- SCRdensity(obj,nx=10,ny=10,scalein=100,scaleout=100)

```

5557 In these figures density is expressed in units of individuals per 100 km^2 , while the area of
 5558 the pixels is about 103.7 km^2 and 11.5 km^2 , respectively. That calculation is based on:

```

5559 > total.area <- (ylim[2]-ylim[1])*(xlim[2]-xlim[1])*100
5560 > total.area/(10*10)
5561 [1] 103.7427
5562 > total.area/(30*30)
5563 [1] 11.52697

```

5564 A couple of things are worth noting: First is that as we move away from “where the
5565 data live” – away from the trap array – we see that the density approaches the mean
5566 density. This is a property of the estimator as long as the detection function decreases
5567 sufficiently rapidly as a function of distance. Relatedly, it is also a property of statistical
5568 smoothers such as splines, kernel smoothers, and regression smoothers – predictions tend
5569 toward the global mean as the influence of data diminishes. Another way to think of it is
5570 that it is a consequence of the prior, which imposes uniformity, and as you get far away
5571 from the data, the predictions tend to the expected constant density under the prior.
5572 Another thing to note about this map is that density is not 0 over water (although the
5573 coastline is not shown). This might be perplexing to some who are fairly certain that
5574 wolverines do not like water. However, there is nothing about the model that recognizes
5575 water from non-water and so the model predicts over water *as if* it were habitat similar to
5576 that within which the array is nested. But, all of this is OK as far as estimating density
5577 goes and, furthermore, we can compute valid estimates of N over any well-defined region
5578 which presumably wouldn’t include water if we so wished. Alternatively, areas covered by
5579 water could be masked out, which we discuss in the next section.

5580 **5.11.3 Predicting where an individual lives**

5581 The density maps in the previous section show the expected number of individuals per
5582 unit area. A closely related problem is that of producing a map of the probable location
5583 of a specific individual’s activity center. For any observed encounter history, we can easily
5584 generate a posterior distribution of \mathbf{s}_i for individual i . In addition, for an individual that
5585 is *not* captured, we can use the MCMC output to produce a corresponding plot of where
5586 such an individual might live, say \mathbf{s}_{n+1} . Obviously, all such uncaptured individuals (for
5587 $i = n + 1, \dots, N$) should have the same posterior distribution. To illustrate, we show the
5588 posterior distribution of \mathbf{s}_1 , the activity center for the individual labeled 1 in the data
5589 set, in Fig. 5.7. This individual was captured a single time at trap 30 which is circled
5590 in Fig. 5.7. We see that the posterior distribution is affected by traps of capture *and*
5591 traps of non-capture in fairly intuitive ways. In particular, because there are other traps
5592 in close proximity to trap 30, in which individual 1 was *not* captured, the model pushes
5593 its activity center away from the trap array. The help file for **SCRdensity** shows how to
5594 calculate Fig. 5.7.

5.12 EFFECTIVE SAMPLE AREA

5595 One of the key issues in using ordinary capture recapture models which we’ve brought up
5596 over and over again is this issue that the area which is sampled by a trapping array is
5597 unknown – in other words, the N that is estimated by capture-recapture models does not
5598 have an explicit region of space associated with it. Classically this has been addressed in
5599 the ad hoc way of prescribing an area that contains the trap array, usually by adding a
5600 buffer of some width, which is not estimated as part of the capture-recapture model. In
5601 SCR models we avoid the problem of not having an explicit linkage between N and “area”,
5602 by prescribing explicitly the area within which the underlying point process is defined – the
5603 state-space of the point process. This state-space is *not* the effective sample (or sampled)
5604 area (ESA) – it is desirable that it be somewhat larger than the ESA, whatever that may

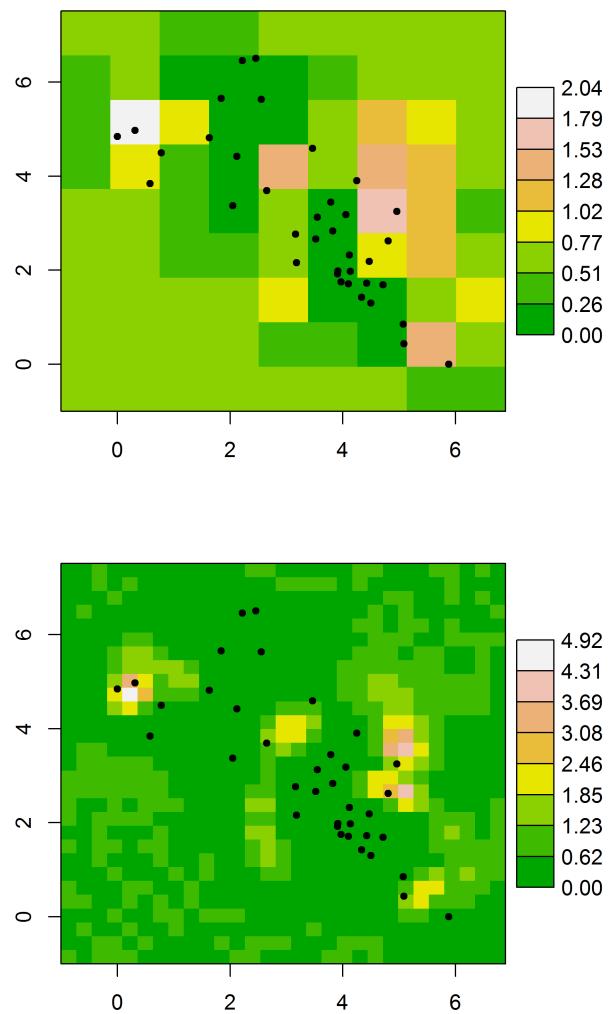


Figure 5.6. Density of wolverines (individuals per 100 km²) in SE Alaska in 2007 based on model SCR0. Map grid cells are about 103.7 km² (top panel) and 11.5 km² (bottom panel) in area. Dots are the trap locations.

be, in the sense that individuals at the edge of the state-space have no probability of being captured, but as part of the SCR model we don't need to try to estimate or otherwise characterize the ESA explicitly.

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

$$D = N/A = n/ESA$$

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

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

Our following development assumes that D is constant, but these calculations can be generalized to allow for D to vary spatially. Imagine our habitat mask for the wolverine data, or the bins we just used to produce a density map, then we can write $\mathbb{E}(n)$ according to

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

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

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

and (plugging this into the expression above for ESA)

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

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

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

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

$$ESA = a \sum_s 1 - (1 - p(\mathbf{s}))^K \tag{5.12.1}$$

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

For the wolverine camera trapping data, we used the 2×2 km habitat mask and the posterior means of p_0 and σ (see Sec. 5.10.2) to compute the probability of encounter for

5634 each \mathbf{s} of the mask points. The result is shown graphically in Fig. 5.8. The ESA is the
 5635 sum of the values plotted in that figure multiplied by 4, the area of each pixel. For the
 5636 wolverine study, the result is 2507.152 km². We note that the probability of encounter
 5637 declines rapidly to 0 as we move away from the periphery of the camera traps, indicating
 5638 the state-space constructed from a 40 km buffered trap array was indeed sufficient for the
 5639 analysis of these data. An R script for producing this figure is in the **wolvESA** function of
 5640 the **scrbook** package.

5.13 SUMMARY AND OUTLOOK

5641 In this chapter, we introduced the simplest SCR model – “model SCR0” – which is an ordinary
 5642 capture-recapture model like model M_0 , but augmented with a set of latent individual
 5643 effects, \mathbf{s}_i , which relate encounter probability to some sense of individual location using a
 5644 covariate, “distance”, from \mathbf{s}_i to each trap location. Thus, individuals in close proximity
 5645 to a trap will have a higher probability of encounter, and *vice versa*. The explicit modeling
 5646 of individual locations and distance in this fashion resolves classical problems related to
 5647 estimating density: unknown sample area, and heterogeneous encounter probability due
 5648 to variable exposure to traps.

5649 SCR models are closely related to classical individual covariate models (“model M_x ”,
 5650 as introduced in Chapt. 4), but with imperfect information about the individual covariate.
 5651 Therefore, they are also not too dissimilar from standard GLMMs used throughout
 5652 statistics and, as a result, we find that they are easy to analyze using standard MCMC
 5653 methods encased in black boxes such as **WinBUGS** or **JAGS**. We will also see that they
 5654 are easy to analyze using likelihood methods, which we address in Chapt. 6.

5655 Formal consideration of the collection of individual locations ($\mathbf{s}_1, \dots, \mathbf{s}_N$) is fundamental
 5656 to all models considered in this book. In statistical terminology, we think of the
 5657 collection of points $\{\mathbf{s}_i\}$ as a realization of a point process. Because SCR models formally
 5658 link individual encounter history data to an underlying point process, we can obtain formal
 5659 inferences about the point process. For example, we showed how to produce a density
 5660 map (Fig. 5.6), or even a probability map for an individual’s home range center (Fig.
 5661 5.7). We can also use SCR models as the basis for doing more traditional point process
 5662 analyses, such as testing for “complete spatial randomness” (CSR) (see Chapt. 8), and
 5663 computing other point process summaries (Illian et al., 2008).

5664 Part of the promise, and ongoing challenge, of SCR models is to develop models that
 5665 reflect interesting biological processes, for example interactions among points or temporal
 5666 dynamics in point locations. In this chapter we considered the simplest possible point
 5667 process model in which points are independent and uniformly (“randomly”) distributed
 5668 over space. Despite the simplicity of this model, it should suffice in many applications of
 5669 SCR models, although we do address generalizations in later chapters. Moreover, even
 5670 though the *prior* distribution on the point locations is uniform, the realized pattern may
 5671 deviate markedly from uniformity as the observed encounter data provide information to
 5672 impart deviations from uniformity. Thus, estimated density maps will typically appear
 5673 distinctly non-uniform (as we saw in the wolverine example). In applications of the basic
 5674 SCR model, we find that this simple *a priori* model can effectively reflect or adapt to
 5675 complex realizations of the underlying point process. For example, if individuals are
 5676 highly territorial then the data should indicate this in the form of individuals not being

5677 encountered in the same trap – the resulting posterior distribution of point locations should
5678 therefore reflect non-independence. Obviously the complexity of posterior estimates of the
5679 point pattern will depend on the quantity of data, both number of individuals and captures
5680 per individual. Because the point process is such an integral component of SCR models,
5681 the state-space of the point process plays an important role in developing SCR models.
5682 As we emphasized in this chapter, the state-space is part of the model. It can have an
5683 influence on parameter estimates and other inferences, such as model selection (see chapter
5684 8).

5685 One concept we introduced in this chapter, which has not been discussed much in
5686 the literature on SCR models, is the manner in which the encounter probability model
5687 relates to a model of space usage by individuals. The standard SCR models of encounter
5688 probability can all be motivated as simplistic models of space usage and movement, in
5689 which individuals make random use decisions from a probability distribution proportional
5690 to the encounter probability model. This both clarifies the simplicity of the underlying
5691 model of space usage and also suggests a direct extension to produce more realistic models,
5692 which we discuss in Chapt. 13. We consider some other important extensions of the basic
5693 SCR model in later chapters. For example, we consider models that include covariates that
5694 vary by individual, trap, or over time (Chapt. 7), spatial covariates on density (Chapt.
5695 11), open populations (Chapt. 16), and methods for model assessment and selection
5696 (Chapt. 8) among other topics. We also consider technical details of maximum likelihood
5697 (Chapt. 6) and Bayesian (Chapt. 17) estimation, so that the interested reader can develop
5698 or extend methods to suit their own needs.

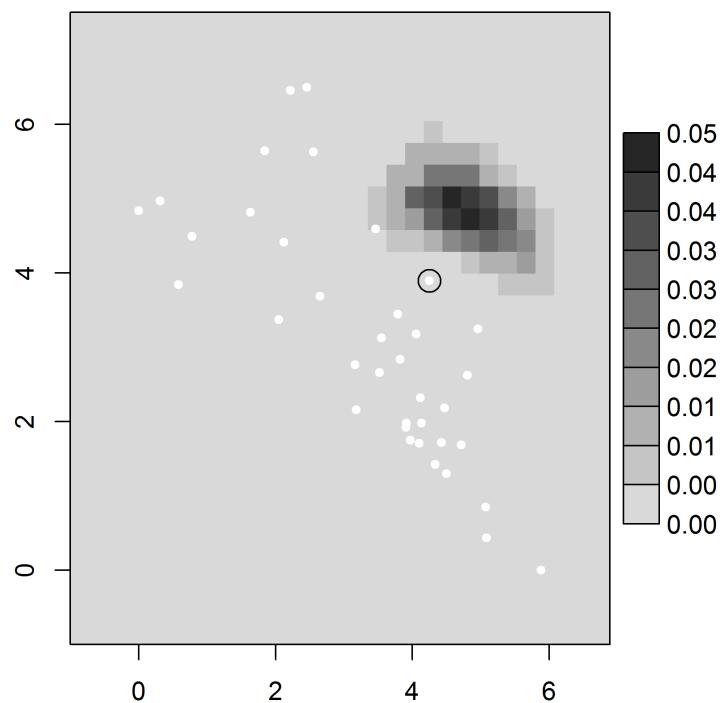


Figure 5.7. Posterior probability distribution of s_1 , the activity center for individual 1 in the wolverine data set. This individual was captured a single time in one trap (trap 30) which is circled. White dots are trap locations.

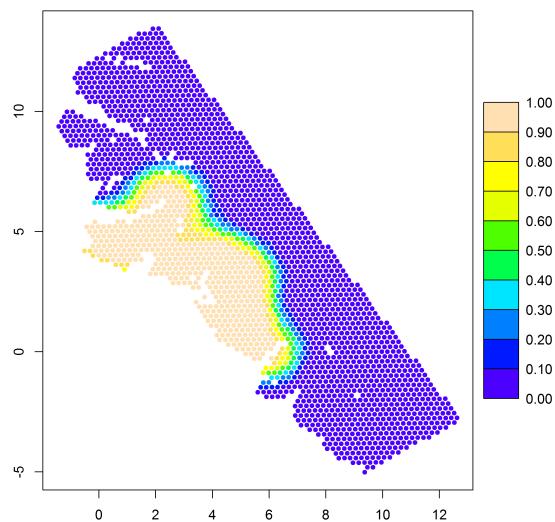


Figure 5.8. Probability of encounter used in computing effective sampled area for the wolverine camera trapping array, using the parameter estimates (posterior means) for the 2×2 km habitat mask.

5699

5700

5701

5702

6

LIKELIHOOD ANALYSIS OF SPATIAL CAPTURE-RECAPTURE MODELS

5703 We have so far mainly focused on Bayesian analysis of spatial capture-recapture models.
 5704 And, in the previous chapters we learned how to fit some basic spatial capture-recapture
 5705 models using a Bayesian formulation of the models analyzed in **BUGS** engines including
 5706 **WinBUGS** and **JAGS**. Despite our focus on Bayesian analysis, it is instructive to de-
 5707 velop the basic concepts and ideas behind classical analysis based on likelihood methods
 5708 and frequentist inference for SCR models. We recognized earlier (Chapt. 5) that SCR
 5709 models are versions of binomial (or other) GLMs, but with random effects (i.e., GLMMs).
 5710 Throughout statistics, such models are routinely analyzed by likelihood methods. In par-
 5711 ticular, likelihood analysis is based on the integrated or marginal likelihood in which the
 5712 random effects are removed, by integration, from the conditional-on-s likelihood (s being
 5713 the individual activity center). This has been the approach taken by Borchers and Ef-
 5714 ford (2008); Dawson and Efford (2009) and related papers. Therefore, in this chapter, we
 5715 provide some conceptual and technical foundation for likelihood-based analysis of spatial
 5716 capture-recapture models.

5717 We will show here that it is straightforward to compute the maximum likelihood esti-
 5718 mates (MLE) for SCR models by integrated likelihood. We develop the MLE framework
 5719 using **R**, and we also provide a basic introduction to the **R** package **secr** (Efford, 2011)
 5720 which does likelihood analysis of SCR models (see also the stand-alone program **DEN-**
 5721 **SITY** (Efford et al., 2004)). To set the context for likelihood analysis of SCR models,
 5722 we first analyze the SCR model when N is known because, in that case, analysis is no
 5723 different at all than a standard GLMM. We generalize the model to allow for unknown N
 5724 using both conventional ideas based on the “full likelihood” (e.g., Borchers et al., 2002)
 5725 and also using a formulation based on data augmentation. We obtain the MLEs for the
 5726 SCR model from the wolverine camera trapping study (Magoun et al., 2011) analyzed in
 5727 previous chapters to compare/contrast the results.

6.1 MLE WITH KNOWN N

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

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

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

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

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

$$y_{ij}|\mathbf{s}_i \sim \text{Binomial}(K, p_{\boldsymbol{\alpha}}(\mathbf{x}_j, \mathbf{s}_i)) \quad (6.1.1)$$

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

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

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

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

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

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

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

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

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

5757 We emphasize that two independence assumptions are explicit in this development: independence of trap-specific encounters within individuals and also independence among
 5758 individuals. In particular, this would only be valid when individuals are not physically
 5759 restrained or removed upon capture, and when traps do not “fill up.”

5760 The key operation for computing the likelihood is solving a 2-dimensional integration
 5761 problem. There are some general purpose **R** packages that implement a number of multi-
 5762 dimensional integration routines including **adapt** (Genz et al., 2007) and **R2cuba** (Hahn
 5763 et al., 2010). In practice, we won’t rely on these extraneous **R** packages (except see Chapt.
 5764 11 for an application of **R2cuba**) but instead will use perhaps less efficient methods in which
 5765 we replace the integral with a summation over an equal area mesh of points on the state-
 5766 space \mathcal{S} and explicitly evaluate the integrand at each point. We invoke the rectangular
 5767 rule for integration here¹ in which we evaluate the integrand on a regular grid of points
 5768 of equal area and compute the average of the integrand over that grid of points. Let
 5769 $u = 1, 2, \dots, nG$ index a grid of nG points, \mathbf{s}_u , where the area of grid cells is constant, say
 5770 A . In this case, the integrand, i.e., the marginal pmf of \mathbf{y}_i , is approximated by

$$[\mathbf{y}_i | \boldsymbol{\alpha}] = \frac{1}{nG} \sum_{u=1}^{nG} [\mathbf{y}_i | \mathbf{s}_u, \boldsymbol{\alpha}] \quad (6.1.2)$$

5772 This is a specific case of the general expression that could be used for approximating
 5773 the integral for any arbitrary distribution $[\mathbf{s}]$. The general case is

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

5774 Under the uniformity assumption, $[\mathbf{s}] = 1/A(\mathcal{S})$ and thus the grid-cell area cancels in the
 5775 above expression to yield Eq. 6.1.2. The rectangular rule for integration can be seen as
 5776 an application of the Law of Total Probability for a discrete random variable \mathbf{s} , having
 5777 nG unique values with equal probabilities $1/nG$.

5778 6.1.1 Implementation (simulated data)

5779 Here we will illustrate how to carry out this integration and optimization based on the
 5780 integrated likelihood using simulated data (i.e., see Sec. 5.5). Using **simSCR0** we simulate
 5781 data for 100 individuals and an array of 25 traps laid out in a 5×5 grid of traps having unit
 5782 spacing. The specific encounter model is the Gaussian model. The 100 activity centers
 5783 were simulated on a state-space defined by an 8×8 square within which the trap array was
 5784 centered (thus the trap array is buffered by 2 units). Therefore, the density of individuals
 5785 in this system is fixed at 100/64. In the following set of **R** commands we generate the
 5786 data and then harvest the required data objects:

```
5787 ## simulate a complete data set (perfect detection)
5788 > data <- simSCR0(discard0=FALSE, rnd=2013)
5789   ## extract the objects that we need for analysis
5790 > y <- data$Y
```

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

```

5791 > traplocs <- data$traplocs
5792 > nind <- nrow(y) ## in this case nind=N
5793 > J <- nrow(traplocs)
5794 > K <- data$K
5795 > xlim <- data$xlim
5796 > ylim <- data$ylim

```

5797 Now, we need to define the integration grid, say **G**, which we do with the following set of
 5798 **R** commands (here, **delta** is the grid spacing):

```

5799 > delta <- .2
5800 > xg <- seq(xlim[1]+delta/2,xlim[2]-delta/2,by=delta)
5801 > yg <- seq(ylim[1]+delta/2,ylim[2]-delta/2,by=delta)
5802 > npix <- length(xg)           # valid for square state-space only
5803 > G <- cbind(rep(xg,npix),sort(rep(yg,npix)))
5804 > nG <- nrow(G)

```

5805 In this case, the integration grid is set up as a grid with spacing $\delta = 0.2$ which produces,
 5806 for our example, a 40×40 grid of points for evaluating the integrand if the state-space
 5807 buffer is set at 2. We note that the integration grid is set-up here to correspond exactly
 5808 to the state-space used in simulating the data. However, in practice, we wouldn't know
 5809 this, and our estimate of N (for the unknown case, see below) would be sensitive to choice
 5810 of the extent of the integration grid. As we've discussed previously, density, which is N
 5811 standardized by the area of the state-space, will not be so sensitive in most cases.

5812 We are now ready to compute the conditional-on-s likelihood and carry out the
 5813 marginalization described by Eq. 6.1.2. We need to do this by defining an **R** function
 5814 that computes the likelihood for the integration grid, as a function of the data objects
 5815 **y** and **traplocs** which were created above. However, it is a bit untidy to store the grid
 5816 information in your workspace, and define the likelihood function in a way that depends
 5817 on these things that exist in your workspace. Therefore, we build the **R** function so that
 5818 it computes the integration grid *within* the function, thereby avoiding potential problems
 5819 if our trapping grid locations change, or if we want to modify the state-space buffer easily.
 5820 We therefore define the function, called **intlik1**, to which we pass the data objects and
 5821 other information necessary to compute the marginal likelihood. This function is available
 5822 in the **scrbook** package (use **?intlik1** at the **R** prompt). The code is reproduced here:

```

5823 intlik1 <- function(parm,y=y,X=traplocs, delta=.2, ssbuffer=2){
5824
5825   Xl <- min(X[,1]) - ssbuffer ## These lines of code are setting up the
5826   Xu <- max(X[,1]) + ssbuffer ## support for the integration which is
5827   Yu <- max(X[,2]) + ssbuffer ## the same as the state-space of "s"
5828   Yl <- min(X[,2]) - ssbuffer
5829   xg <- seq(Xl+delta/2,Xu-delta/2,,length=npix)
5830   yg <- seq(Yl+delta/2,Yu-delta/2,,length=npix)
5831   npix<- length(xg)
5832
5833   G <- cbind(rep(xg,npix),sort(rep(yg,npix)))

```

```

5834   nG <- nrow(G)
5835   D <- e2dist(X,G)
5836
5837   alpha0 <- parm[1]
5838   alpha1 <- exp(parm[2]) # alpha1 restricted to be positive here
5839
5840   probcap <- plogis(alpha0)*exp(-alpha1*D*D)
5841   Pm <- matrix(NA,nrow=nrow(probcap),ncol=ncol(probcap))
5842           # Frequency of all-zero encounter histories
5843   n0 <- sum(apply(y,1,sum)==0)
5844           # Encounter histories with at least 1 detection
5845   ymat <- y[apply(y,1,sum)>0,]
5846   ymat <- rbind(ymat,rep(0,ncol(ymat)))
5847   lik.marg <- rep(NA,nrow(ymat))
5848
5849   for(i in 1:nrow(ymat)){
5850       ## Next line: log conditional likelihood for ALL possible values of s
5851       Pm[1:length(Pm)] <- dbinom(rep(ymat[i,],nG),K,probcap[1:length(Pm)],
5852                                     log=TRUE)
5853       ## Next line: sum the log conditional likelihoods, exp() result
5854       ## same as taking the product
5855       lik.cond <- exp(colSums(Pm))
5856       ## Take the average value == computing marginal
5857       lik.marg[i] <- sum(lik.cond*(1/nG))
5858   }
5859   ## n0 = number of all-0 encounter histories
5860   nv <- c(rep(1,length(lik.marg)-1),n0)
5861   return( -1*(sum(nv*log(lik.marg)) ) )
5862 }
```

5863 We emphasize that this function (and subsequent) are not meant to be general-purpose
 5864 routines for solving all of your SCR problems but, rather, they are meant for illustrative
 5865 purposes – so you can see how the integrated likelihood is constructed and how we connect
 5866 it to data and other information that is needed.

5867 The function `intlik1` accepts as input the encounter history matrix, `y`, the trap locations,
 5868 `X`, and the state-space buffer. This allows us to vary the state-space buffer and easily
 5869 evaluate the sensitivity of the MLE to the size of the state-space. Note that we have a
 5870 peculiar handling of the encounter history matrix `y`. In particular, we remove the all-zero
 5871 encounter histories from the matrix and tack-on a single all-zero encounter history as the
 5872 last row which then gets weighted by the number of such encounter histories (`n0`). This is
 5873 a bit long-winded and strictly unnecessary when N is known, but we did it this way be-
 5874 cause the extension to the unknown- N case is now transparent (as we demonstrate in the
 5875 following section). The matrix `Pm` holds the log-likelihood contributions of each encounter
 5876 frequency for each possible state-space location of the individual. The log contribu-
 5877 tions are summed up and the result exponentiated on the next line, producing `lik.cond`, the
 5878 conditional-on-s likelihood (Eq. 6.1.1 above). The marginal likelihood (`lik.marg`) sums
 5879 up the conditional elements weighted by the probabilities [`s`] (Eq. 6.1.2 above).

5880 This is a fairly primitive function which doesn't allow much flexibility in the data
 5881 structure. For example, it assumes that K , the number of replicates, is constant for each
 5882 trap. Further, it assumes that the state-space is a square. We generalize this to some
 5883 extent later in this chapter.

5884 Here is the **R** command for maximizing the likelihood using **nlm** (the function **optim**
 5885 could also be used) and saving the results into an object called **frog**. The output is a list
 5886 of the following structure and these specific estimates are produced using the simulated
 5887 data set:

```
5888 # should take 15-30 seconds
5889
5890 > starts <- c(-2,2)
5891 > frog <- nlm(intlik1,starts,y=y,X=traplocs,delta=.1,ssbuffer=2,hessian=TRUE)
5892 > frog
5893
5894 $minimum
5895 [1] 297.1896
5896
5897 $estimate
5898 [1] -2.504824 2.373343
5899
5900 $gradient
5901 [1] -2.069654e-05 1.968754e-05
5902
5903 $hessian
5904 [,1]      [,2]
5905 [1,] 48.67898 -19.25750
5906 [2,] -19.25750 13.34114
5907
5908 $code
5909 [1] 1
5910
5911 $iterations
5912 [1] 11
```

5913 Details about this output can be found on the help page for **nlm**. We note briefly that
 5914 **frog\$minimum** is the negative log-likelihood value at the MLEs, which are stored in the
 5915 **frog\$estimate** component of the list. The order of the parameters is as they are defined
 5916 in the likelihood function so, in this case, the first element (value = -2.504824) is the
 5917 logit transform of p_0 and the second element (value = 2.373343) is the value of α_1 the
 5918 "coefficient" on distance-squared. The Hessian is the observed Fisher information matrix,
 5919 which can be inverted to obtain the variance-covariance matrix using the command:

```
5920 > solve(frog$hessian)
```

5921 It is worth drawing attention to the fact that the estimates are slightly different than
 5922 the Bayesian estimates reported previously in Sec. 5.6. There are several reasons for this.
 5923 First Bayesian inference is based on the posterior distribution and it is not generally the

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

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

6.2 MLE WHEN N IS UNKNOWN

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

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

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

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

5961 (2) We compute the probability of an all-0 encounter history:

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

5962 (3) The combinatorial term: $\binom{N}{n}$. Then, the marginal likelihood has this form:

$$\mathcal{L}(\boldsymbol{\alpha}, n_0 | \mathbf{y}) = \frac{N!}{n! n_0!} \left\{ \prod_{i=1}^n [\mathbf{y}_i | \boldsymbol{\alpha}] \right\} \pi_0^{n_0}. \quad (6.2.1)$$

5963 This is discussed in Borchers and Efford (2008, p. 379) as the conditional-on- N form of the
 5964 likelihood – we also call it the “binomial form” of the likelihood because of its appearance.

5965 Operationally, things proceed much as before: We compute the marginal probability
 5966 of each observed \mathbf{y}_i , i.e., by removing the latent \mathbf{s}_i by integration. In addition, we com-
 5967 pute the marginal probability of the “all-zero” encounter history \mathbf{y}_{n+1} , and make sure to
 5968 weight it n_0 times. We accomplish this by “padding” the data set with a single encounter
 5969 history having $y_{n+1,j} = 0$ for all traps $j = 1, 2, \dots, J$. Then we be sure to include the
 5970 combinatorial term in the likelihood or log-likelihood computation. We demonstrate this
 5971 shortly. To analyze a specific case, we’ll simulate our fake data set (simulated using the
 5972 parameters given above). To set some things up in our workspace we do this:

```
5973 ## Obtain a simulated data set
5974 > data <- simSCRO(discard0=TRUE, rnd=2013)
5975
5976 ## Extract the items we need for analysis
5977 > y <- data$Y
5978 > nind <- nrow(y)
5979 > traplocs <- data$traplocs
5980 > J <- nrow(traplocs)
5981 > K <- data$K
```

5982 Recall that these data are simulated by default with $N = 100$, on an 8×8 unit state-
 5983 space representing the trap locations buffered by 2 units, although you can modify the
 5984 simulation script easily.

5985 As before, the likelihood is defined in the **R** workspace as an **R** function, **intlik2**,
 5986 which takes an argument being the unknown parameters of the model and additional
 5987 arguments as prescribed. In particular, we provide the encounter history matrix **y**, the
 5988 trap locations **traplocs**, the spacing of the integration grid (argument **delta**) and the
 5989 state-space buffer. Here is the new likelihood function:

```
5990 intlik2 <- function(parm,y=y,X=traplocs,delta=.3,ssbuffer=2){
5991
5992   Xl <- min(X[,1]) - ssbuffer
5993   Xu <- max(X[,1]) + ssbuffer
5994   Yu <- max(X[,2]) + ssbuffer
5995   Yl <- min(X[,2]) - ssbuffer
5996
5997   xg <- seq(Xl+delta/2,Xu-delta/2,delta)
```

```

5998 yg <- seq(Yl+delta/2,Yu-delta/2,delta)
5999 npix.x <- length(xg)
6000 npix.y <- plength(yg)
6001 area <- (Xu-Xl)*(Yu-Yl)/((npix.x)*(npix.y))
6002 G <- cbind(rep(xg,npix.y),sort(rep(yg,npix.x)))
6003 nG <- nrow(G)
6004 D <- e2dist(X,G)
6005 # extract the parameters from the input vector
6006 alpha0 <- parm[1]
6007 alpha1 <- exp(parm[2])
6008 n0 <- exp(parm[3]) # note parm[3] lives on the real line
6009 probcap <- plogis(alpha0)*exp(-alpha1*D*D)
6010 Pm <- matrix(NA,nrow=nrow(probcap),ncol=ncol(probcap))
6011 ymat <- rbind(y,rep(0,ncol(y)))
6012
6013 lik.marg <- rep(NA,nrow(ymat))
6014 for(i in 1:nrow(ymat)){
6015   Pm[1:length(Pm)] <- (dbinom(rep(ymat[i,],nG),K,probcap[1:length(Pm)],
6016                                     log=TRUE))
6017   lik.cond <- exp(colSums(Pm))
6018   lik.marg[i] <- sum(lik.cond*(1/nG) )
6019 }
6020 nv <- c(rep(1,length(lik.marg)-1),n0)
6021 ## part1 here is the combinatorial term.
6022 ## math: log(factorial(N)) = lgamma(N+1)
6023 part1 <- lgamma(nrow(y)+n0+1) - lgamma(n0+1)
6024 part2 <- sum(nv*log(lik.marg))
6025 return( -1*(part1+ part2) )
6026 }

```

6027 To execute this function for the data that we created with `simSCR0`, we execute the
 6028 following command (saving the result in our friend `frog`). This results in the usual output,
 6029 including the parameter estimates, the gradient, and the numerical Hessian which is useful
 6030 for obtaining asymptotic standard errors (see below):

```

6031 > starts <- c(-2.5,0,4)
6032 > frog <- nlm(intlik2,starts,hessian=TRUE,y=y,X=traplocs,delta=.2,ssbuffer=2)
6033
6034 Warning message:
6035 In nlm(intlik2, starts, hessian = TRUE, y = y, X = traplocs, delta = 0.2, :
6036 NA/Inf replaced by maximum positive value
6037
6038 > frog
6039 $minimum
6040 [1] 113.5004
6041
6042 $estimate

```

```
6043 [1] -2.538333 0.902807 4.232810
6044
6045 [... additional output deleted ...]
```

6046 Executing `nlm` here usually produces one or more **R** warnings due to numerical calculations
 6047 happening on extremely small or large numbers (calculation of p near the edge of the
 6048 state-space), and they also happen if a poor parameterization is used which produces
 6049 evaluations of the objective function beyond the boundary of the parameter space (e.g.,
 6050 $n_0 < 0$). Such numerical warnings can often be minimized or avoided altogether by picking
 6051 judicious starting values of parameters or properly transforming or scaling the parameters
 6052 but, in general, they can be ignored. You will see from the `nlm` output that the algorithm
 6053 performed satisfactory in minimizing the objective function. The estimate of population
 6054 size, \hat{N} , for the state-space (using the default state-space buffer) is

```
6055 > Nhat <- nrow(y) + exp(4.2328) #### This is n + MLE of n0
6056 > Nhat
6057 [1] 110.9099
```

6058 Which differs from the data-generating value ($N = 100$), as we might expect for a single
 6059 realization. We usually will present an estimate of uncertainty associated with this MLE
 6060 which we can obtain by inverting the Hessian. Note that $\text{Var}(\hat{N}) = n + \text{Var}(\hat{n}_0)$. Since
 6061 we have parameterized the model in terms of $\log(n_0)$ we use the delta method² described
 6062 in Williams et al. (2002, Appendix F4) (see also Ver Hoef, 2012) to obtain the variance
 6063 on the scale of n_0 as follows:

```
6064 > (exp(4.2328)^2)*solve(frog$hessian)[3,3]
6065 [1] 260.2033
6066
6067 > sqrt(260)
6068 [1] 16.12452
```

6069 Therefore, the asymptotic “Wald-type” confidence interval for N is $110.91 \pm 1.96 \times 16.125 =$
 6070 $(79.305, 142.515)$. To report this in terms of density, we scale appropriately by the area
 6071 of the prescribed state-space which is 64 units of area (i.e., an 8×8 square). Our MLE
 6072 of D is $\hat{D} = 110.91/64 = 1.733$ individuals per square unit. To get the standard error
 6073 for \hat{D} we need to divide the SE for \hat{N} by the area of the state-space, and so $\text{SE}(\hat{D}) =$
 6074 $(1/64) * 16.12452 = 0.252$.

6075 6.2.1 Integrated likelihood under data augmentation

6076 The likelihood analysis developed in the previous sections is based on the likelihood in
 6077 which N (or n_0) is an explicit parameter. This is usually called the “full likelihood” or
 6078 sometimes “unconditional likelihood” (Borchers et al., 2002) because it is the likelihood
 6079 for all individuals in the population, not just those which have been captured, i.e., not that
 6080 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>

6081 likelihood using data augmentation, replacing the parameter N with ψ (e.g., see Sec. 7.1.6
 6082 Royle and Dorazio, 2008, for an example). We don't go into detail here, but we note that
 6083 the likelihood under data augmentation is a zero-inflated binomial mixture – precisely an
 6084 occupancy type model (Royle, 2006). Thus, while it is possible to carry out likelihood
 6085 analysis of models under data augmentation, we primarily advocate data augmentation
 6086 for Bayesian analysis.

6087 6.2.2 Extensions

6088 We have only considered basic SCR models with no additional covariates. However,
 6089 in practice, we are interested in covariate effects including “behavioral response”, sex-
 6090 specificity of parameters, and potentially others. Some of these can be added directly to
 6091 the likelihood if the covariate is fixed and known for all individuals captured or not. An
 6092 example is a behavioral response, which amounts to having a covariate $x_{ik} = 1$ if individual
 6093 i was captured prior to occasion k and $x_{ik} = 0$ otherwise. For uncaptured individuals,
 6094 $x_{ik} = 0$ for all k . Royle et al. (2011b) called this a global behavioral response because the
 6095 covariate is defined for all traps, no matter the trap in which an individual was captured.
 6096 We could also define a *local* behavioral response which occurs at the level of the trap, i.e.,
 6097 $x_{ijk} = 1$ if individual i was captured in trap j prior to occasion k , etc... Trap-specific
 6098 covariates such as trap type or status, or time-specific covariates such as date, are eas-
 6099 ily accommodated as well. As an example, Kéry et al. (2010) develop a model for the
 6100 European wildcat *Felis silvestris* in which traps are either baited or not (a trap-specific
 6101 covariate with only 2 values), and also encounter probability varies over time in the form
 6102 of a quadratic seasonal response. We consider models with behavioral response or fixed
 6103 covariates in Chapt. 7. The integrated likelihood routines we provided above can be
 6104 modified directly for such cases, which we leave to the interested reader to investigate.

6105 Sex-specificity is more difficult to deal with since sex is not known for uncaptured
 6106 individuals (and sometimes not even for all captured individuals). To analyze such models,
 6107 we do Bayesian analysis of the joint likelihood using data augmentation (Gardner et al.,
 6108 2010b; Russell et al., 2012), discussed further in Chapt. 7. For such covariates (i.e., that
 6109 are not fixed and known for all individuals), it is somewhat more challenging to do MLE
 6110 based on the joint likelihood as we have developed above. Instead it is more conventional
 6111 to use what is colloquially referred to as the “Huggins-Alho” type model which is one of
 6112 the approaches taken in the software package **secr** (Efford, 2011). We introduce the **secr**
 6113 package in Sec. 6.5 below.

6.3 CLASSICAL MODEL SELECTION AND ASSESSMENT

6114 In most analyses, one is interested in choosing from among various potential models, or
 6115 ranking models, or something else to do with assessing the relative merits of a set of
 6116 models. A good thing about classical analysis based on likelihood is we can apply Akaike
 6117 Information Criterion (AIC) methods (Burnham and Anderson, 2002) without difficulty.
 6118 AIC is convenient for assessing the relative merits of these different models although if
 6119 there are only a few models it is not objectionable to use hypothesis tests or confidence
 6120 intervals to determine importance of effects. A second model selection context has to
 6121 do with choosing among various detection models, although, as a general rule, we don't

recommend this application of model selection. This is because there is hardly ever (if at all) a rational subject-matter based reason motivating specific distance functions. As a result, we believe that doing too much model selection will invariably lead to over-fitting and thus over-statement of precision. This is the main reason that we haven't loaded you down with a basket of models for detection probability so far, although we discuss many possibilities in Chapt. 7.

Goodness-of-fit or model-checking – For many standard capture-recapture models, it is possible to identify goodness-of-fit statistics based on the multinomial likelihood, (Cooch and White, 2006, Chapt. 5), and evaluate model adequacy using formal statistical tests. Similar strategies can be applied to SCR models using expected cell-frequencies based on the marginal distribution of the observations. Also, because computing MLEs is somewhat more efficient in many cases compared to Bayesian analysis, it is sometimes feasible to use bootstrap methods. At the present time, we don't know of any applications of goodness-of-fit testing for SCR models based on likelihood inference, although we discuss the use of Bayesian p-values for assessing model fit in Chapt. 8. An important practical problem in trying to evaluate goodness-of-fit is that, in realistic sample sizes, fit tests often lack the power to detect departures from the model under consideration and so they may not be generally useful in practice.

6.4 LIKELIHOOD ANALYSIS OF THE WOLVERINE CAMERA TRAPPING DATA

Here we compute the MLEs for the wolverine data using an expanded version of the function we developed in the previous section. To accommodate that each trap might be operational a variable number of nights, we provided an additional argument to the likelihood function (allowing for a vector $\mathbf{K} = (K_1, \dots, K_J)$), which requires also a modification to the construction of the likelihood. In addition, we accommodate the state-space is a general rectangle, and we included a line in the code to compute the state-space area which we apply below for computing density. The more general function (`intlik3`) is given in the **R** package `scrbook`. Incidentally, this function also returns the area of the state-space for a given set of parameter values, as an attribute to the function value, which will be used in converting \hat{N} to \hat{D} . To use this function to obtain the MLEs for the wolverine camera trap study, we execute the following commands (note: these are in the help file and will execute if you type `example(intlik3)`):

```
6152 > library(scrbook)
6153 > data(wolverine)
6154
6155 > traps <- wolverine$wtraps
6156 > traplocs <- traps[,2:3]/10000
6157 > K.wolv <- apply(traps[,4:ncol(traps)],1,sum)
6158
6159 > y3d <- SCR23darray(wolverine$wcaps,traps)
6160 > y2d <- apply(y3d,c(1,2),sum)
6161
6162 > starts <- c(-1.5,0,3)
6163
```

```

6164 > wolv <- nlm(intlik3,starts,hessian=TRUE,y=y2d,K=K.wolv,X=traplocs,
6165           delta=.2,ssbuffer=2)
6166
6167 > wolv
6168 $minimum
6169 [1] 220.4313
6170
6171 $estimate
6172 [1] -2.8176120 0.2269395 3.5836875
6173
6174 [.... output deleted ....]

```

6175 Of course we're interested in obtaining an estimate of population size for the prescribed
 6176 state-space, or density, and associated measures of uncertainty which we do using the delta
 6177 method (Williams et al., 2002, Appendix F4). To do all of that we need to manipulate the
 6178 output of `nlm` since we have our estimate in terms of $\log(n_0)$. We execute the following
 6179 commands:

```

6180 > wolv <- nlm(intlik3,starts,hessian=TRUE,y=y2d,K=K.wolv,X=traplocs,delta=.2,
6181           ssbuffer=2)
6182 > Nhat <- nrow(y2d)+exp(wolv$estimate[3])
6183 > area <- attr(intlik3(starts,y=y2d,K=K.wolv,X=traplocs,delta=.2,ssbuffer=2),
6184           "SSarea")
6185 > Dhat <- Nhat/area
6186
6187 > Dhat
6188 [1] 0.5494947
6189
6190 > SE <- (1/area)*exp(wolv$estimate[3])*sqrt(solve(wolv$hessian)[3,3])
6191
6192 > SE
6193 [1] 0.1087073

```

6194 Our estimate of density is 0.55 individuals per “standardized unit” which is 100 km^2 ,
 6195 because we divided UTM coordinates by 10000. So this is about 5.5 individuals per 1000 km^2 ,
 6196 with a SE of around 1.09 individuals. This compares closely with 5.77 reported in
 6197 Sec. 5.9 based on Bayesian analysis of the model.

6198 6.4.1 Sensitivity to integration grid and state-space buffer

6199 The effect of approximating the integral by a discrete mesh of points is that it induces
 6200 some numerical error in evaluation of the integral and, further, that error increases as the
 6201 coarseness of the mesh increases. To evaluate the effect (or sensitivity) of the integration
 6202 grid spacing, we obtained the MLEs for a state-space buffer of 2 (standardized units) and
 6203 for integration grid with spacing $\delta = .3, .2, .1, .05$. The MLEs for these 4 cases including
 6204 the relative runtime are given in Table 6.1. We see the results change only slightly as the
 6205 integration grid changes. Conversely, the runtime on the platform of the day for the 4 cases

6206 increases rapidly. These runtimes could be regarded in relative terms, across platforms,
 6207 for gaging the decrease in speed as the fineness of the integration grid increases.

Table 6.1. 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

6208 We studied the effect of the state-space buffer on the MLEs, using a fixed $\delta = .2$ for
 6209 all analyses. We used state-space buffers of 1 to 4 units stepped by .5. As we can see
 6210 (Table 6.2), the estimates of D stabilize rapidly and the incremental difference is within
 6211 the numerical error associated with approximating the integral.

Table 6.2. Results of the effect of the state-space buffer on the MLE. Given here are the state-space buffer, area of the state-space (area), the MLE of N (\hat{N}) for the prescribed state-space and the corresponding MLE of density (\hat{D}).

Buffer	Area	\hat{N}	\hat{D}
1.0	66.98212	37.73338	0.5633352
1.5	84.36242	46.21008	0.5477567
2.0	103.74272	57.00617	0.5494956
2.5	125.12302	69.03616	0.5517463
3.0	148.50332	82.17550	0.5533580
3.5	173.88362	96.44018	0.5546249
4.0	201.26392	111.83524	0.5556646

6212 6.4.2 Using a habitat mask (Restricted state-space)

6213 In Sec. 5.10 we used a discrete representation of the state-space in order to have control
 6214 over its extent and shape. This makes it easy to do things like clip out non-habitat, or
 6215 create a *habitat mask* which defines suitable habitat. Clearly that formulation of the model
 6216 is relevant to the calculation of the marginal likelihood in the sense that the discrete state-
 6217 space is equivalent to the integration grid. Thus, for example, we could easily compute
 6218 the MLE of parameters under some model with a restricted state-space merely by creating
 6219 the required state-space at whatever grid resolution is desired, and then inputting that
 6220 state-space into the likelihood function above, instead of computing it within the function.
 6221 We can easily create an explicit state-space grid for integration from arbitrary polygons or
 6222 GIS shapefiles which we demonstrate here. Our approach is to create the integration grid
 6223 (or state-space grid) outside of the likelihood evaluation, and then determine which points
 6224 of the grid lie in the polygon defined by the shapefile using functions in the **R** packages **sp**
 6225 and **maptools**. For each point in the state-space grid (object **G** in the code below which is

6226 assumed to exist), we determine whether it is inside the polygon³, identifying such points
 6227 with a value of `mask=1` and `mask=0` for points that are *not* in the polygon. We load the
 6228 shapefile which originates by an application of the `readShapeSpatial` function. We have
 6229 saved the result into an **R** data object called `SSp` which is in the `scrbook` package. Here
 6230 are the **R** commands for doing this (see the helpfile `?intlik4`):

```
6231 > library(maptools)
6232 > library(sp)
6233 > library(scrbook)
6234
6235 ##### If we have the .shp file in place, we would use this command:
6236 ##### SSp <- readShapeSpatial('Sim_Polygon.shp')
6237 ##### The object SSp is in data(fakeshapefile)
6238 > data(fakeshapefile)
6239 > Pcoord <- SpatialPoints(G)
6240 > PinPoly <- over(Pcoord,SSp)  ### determine if each point is in polygon
6241 > mask <- as.numeric(!is.na(PinPoly[,1])) ## convert to binary 0/1
6242 > G <- G[mask==1,]
```

6243 We created the function `intlik4` which accepts the integration grid as an explicit argument,
 6244 and this function is also available in the package `scrbook`.

6245 We apply this modification to the wolverine camera trapping study. Royle et al.
 6246 (2011b) created 2, 4 and 8 km state-space grids so as to remove “non-habitat” (mostly
 6247 ocean, bays, and large lakes). We previously analyzed the model using **JAGS** and **Win-**
BUGS in Chapt. 5. To set up the wolverine data and fit the model using maximum
 6249 likelihood we execute the following commands:

```
6250 > library(scrbook)
6251 > data(wolverine)
6252
6253 > traps <- wolverine$wtraps
6254 > traplocs <- traps[,2:3]/10000
6255 > K.wolv <- apply(traps[,4:ncol(traps)],1,sum)
6256
6257 > y3d <- SCR23darray(wolverine$wcaps,traps)
6258 > y2d <- apply(y3d,c(1,2),sum)
6259 > G <- wolverine$grid2/10000
6260
6261 > starts <- c(-1.5,0,3)
6262 > 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. 17.5 in Chapt. 17). If you use the `over` function, make sure you know the class of your second argument so that when processing the function output you index it correctly.

Table 6.3. 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

```

6263
6264 > wolv
6265
6266 $minimum
6267 [1] 225.8355
6268
6269 $estimate
6270 [1] -2.9955424 0.2350885 4.1104757
6271
6272 [... some output deleted ...]

```

6273 Next we convert the parameter estimates to estimates of total population size for the
 6274 prescribed state-space, and then obtain an estimate of density (per 1000 km²) using the
 6275 area computed as the number of pixels in the state-space grid, G, multiplied by the area
 6276 per grid cell. In the present case (the calculation above) we used a state-space grid with 2
 6277 km × 2 km pixels. Finally, we compute a standard errors using the delta approximation:

```

6278 > area <- nrow(G)*4
6279 # Nhat = n (observed) + MLE of n0 (not observed)
6280 > Nhat <- 21 + exp(wolv$estimate[3])
6281 > SE <- exp(wolv$estimate[3])*sqrt(solve(wolv$hessian)[3,3])
6282 > D <- (Nhat/(nrow(G)*area))*1000
6283 > SE.D <- (SE/(nrow(G)*area))*1000

```

6284 We did this for each the 2 km, 4 km and 8 km state-space grids which produced the
 6285 estimates summarized in Table 6.3. These estimates compare with the 8.6 (2 km grid)
 6286 and 8.2 (8 km grid) reported in Royle et al. (2011b) based on a clipped state-space as
 6287 described in Sec. 5.10.

6.5 DENSITY AND THE R PACKAGE SECR

6288 **DENSITY** is a software program developed by Efford (2004) for fitting spatial capture-
 6289 recapture models based mostly on classical maximum likelihood estimation and related
 6290 inference methods. Efford (2011) has also released an **R** package called **secr**, that contains
 6291 much of the functionality of **DENSITY** but also incorporates new models and features.
 6292 Here, we briefly introduce the **secr** package which we prefer to use over **DENSITY**, be-
 6293 cause it allows us to remain in the **R** environment for data processing and summarization.
 6294 We provide a brief introduction to **secr** and some of its capabilities here, and we also use

6295 it for doing some analysis in other parts of this book. We believe that **secr** will be sufficient
 6296 for many (if not most) of the SCR problems that one might encounter. It provides
 6297 a flexible analysis platform, with a large number of summary features, and “publication
 6298 ready” output. Its user-interface is clean and intuitive to **R** users, and it has been stable,
 6299 efficient and reliable in the (fairly extensive) evaluations that we have done.

6300 To install and run models in **secr**, you must download the package and load it in **R**.

```
6301 > install.packages("secr")
6302 > library(secr)
```

6303 **secr** allows the user to simulate data and fit a suite of models with various detection functions
 6304 and covariate responses. It also contains a number of helpful constructor functions
 6305 for creating objects of the proper class that are recognized by other **secr** functions. We
 6306 provide a brief overview of the capabilities here, but the **secr** help manual can be accessed
 6307 with the command:

```
6308 > RShowDoc("secr-manual", package = "secr")
```

6309 We note that **secr** has many capabilities that we will not cover or do so only sparingly.
 6310 We encourage you to read through the manual, the extensive documentation, and the
 6311 vignettes, in order to get a better understanding of what the package is capable of. We
 6312 also cover certain capabilities of **secr** in other chapters.

6313 The main model-fitting function in **secr** is called **secr.fit**, which makes use of the
 6314 standard **R** model specification framework with tildes. As an example, the equivalent of
 6315 the basic model SCR0 is fitted as follows:

```
6316 > secr.fit(capturedata, model = list(D ~ 1, g0 ~ 1, sigma ~ 1),
6317   buffer = 20000)
```

6318 where **capturedata** is the object created by **secr** containing the encounter history data
 6319 and the trap information, and the model expression $g0 \sim 1$ indicates the intercept-only (i.e.,
 6320 constant) model. Note that we use p_0 for the baseline encounter probability parameter,
 6321 which is g_0 in **secr** notation. A number of possible models for encounter probability can
 6322 be fitted including both pre-defined variables (e.g., **t** and **b** corresponding to “time” and
 6323 “behavior”), and user-defined covariates of several kinds. For example, to include a global
 6324 behavioral response, this would be written as $g0 \sim b$. The discussion of this (global versus
 6325 local trap-specific behavioral response) and other covariates is developed more in Chapt.
 6326 7. We can also model covariates on density in **secr**, which we discuss in Chapt. 11. It
 6327 is important to note that **secr** requires the buffer distance to be defined in meters and
 6328 density will be returned as number of animals per hectare. Thus to make comparisons
 6329 between **secr** and output from other programs, we will often have to convert the density
 6330 to the same units.

6331 Before we can fit the models, the data must first be packaged properly for **secr**.
 6332 We require data files that contain two types of information: trap layout (location and
 6333 identification information for each trap), which is equivalent to the trap deployment file
 6334 (TDF) described in Sec. 5.9 and the capture data file containing sampling *session*, animal
 6335 identification, trap occasion, and trap location, equivalent in information content to the
 6336 encounter data file (EDF). Sample session can be thought of as primary period identifier

6337 in a robust design like framework – it could represent a yearly sample or multiple sample
 6338 periods within a year, each of them producing data on a closed population. We discuss
 6339 “multi-session” models in more detail below, in Sec. 6.5.4 and Chapt. 14.

6340 There are three important constructor functions that help package-up your data for
 6341 use in **secr**: **read.traps**, **make.capthist** and **read.mask**. We provide a brief description
 6342 of each here, but apply them to our wolverine camera trapping data in the next section:

6343 (1) **read.traps**: This function points to an external file *or* **R** data object containing the
 6344 trap coordinates, and other information, and also requires specification of the type of
 6345 encounter devices (described in the next section). A typical application of this function
 6346 looks like the following, invoking the **data=** option when there is an existing **R** object
 6347 containing the trap information:

```
6348 > trapfile <- read.traps(data=traps, detector="proximity")
```

6349 (2) **make.capthist**: This function takes the EDF and combines it with trap information,
 6350 and the number of sampling occasions. A typical application looks like this:

```
6351 > capturedata <- make.capthist(enc.data, trapfile, fmt="trapID",  

  6352   noccasions=165)
```

6353 See **?make.capthist** for definition of distinct file formats. Specifying **fmt = trapID** is
 6354 equivalent to our EDF format.

6355 (3) **read.mask**: If there is a habitat mask available (as described in sec. 6.4.2), then this
 6356 function will organize it so that **secr.fit** knows what to do with it. The function
 6357 accepts either an external file name (see **?read.mask** for details of the structure) or a
 6358 $nG \times 2$ **R** object, say **mask.coords**, containing the coordinates of the mask. A typical
 6359 application looks like the following:

```
6360 > grid <- read.mask(data=mask.coords)
```

6361 These constructor functions produce output that can then be used in the fitting of models
 6362 using **secr.fit**.

6363 6.5.1 Encounter device types and detection models

6364 The **secr** package requires that you specify the type of encounter device. Instead of
 6365 describing models by their statistical distribution (Bernoulli, Poisson, etc..), **secr** uses
 6366 certain operational classifications of detector types including ‘proximity’, ‘multi’, ‘single’,
 6367 ‘polygon’ and ‘signal’. For camera trapping/hair snares we might consider ‘proximity’
 6368 detectors or ‘count’ detectors. The ‘proximity’ detector type allows, at most, one detection
 6369 of each individual at a particular detector on any occasion (i.e., it is equivalent to what
 6370 we call the Bernoulli or binomial encounter process model, or model SCR0). The ‘count’
 6371 detector designation allows repeat encounters of each individual at a particular detector
 6372 on any occasion. There are other detector types that one can select such as: ‘polygon’
 6373 detector type which allows for a trap to be a sampled polygon (Royle and Young, 2008)
 6374 which we discuss further in Chapt. 15, and ‘signal’ detector which allows for traps that
 6375 have a strength indicator, e.g., acoustic arrays (Dawson and Efford, 2009). The detector
 6376 types ‘single’ and ‘multi’ refer to traps that retain individuals, thus precluding the ability
 6377 for animals to be captured in other traps during the sampling occasion. The ‘single’ type

6378 indicates trap that can only catch one animal at a time (single-catch traps), while 'multi'
 6379 indicates traps that may catch more than one animal at a time (multi-catch). These are
 6380 both variations of the multinomial encounter models described in Chapt. 9.

6381 As with all SCR models, **secr** fits an encounter probability model ("detection function"
 6382 in **secr** terminology relating the probability of encounter to the distance of a detector from
 6383 an individual activity center. **secr** allows the user to specify one of a variety of detection
 6384 functions including the commonly used half-normal ("Gaussian"), hazard rate ("Gaussian
 6385 hazard"), and (negative) exponential models. There are 12 different functions as of version
 6386 2.3.1 (see Table 7.1 in Chapt. 7), but some are only available for simulating data. The
 6387 different detection functions are defined in the **secr** manual and can be found by calling
 6388 the help function for the detection function:

6389 > ?detectfn

6390 Most of the detection functions available in **secr** contain some kind of a scale parameter
 6391 which is usually labeled σ . The units of this parameter default to meters in the **secr**
 6392 output. We caution that the meaning of this parameter depends on the specific detection
 6393 model being used, and it should not be directly compared as a measure of home-range size
 6394 across models. Instead, as we noted in Sec. 5.4 most encounter probability models imply
 6395 a model of space-usage and fitted encounter models should be converted to a common
 6396 currency such as "area used."

6397 6.5.2 Analysis using the **secr** package

6398 To demonstrate the use of the **secr** package, we will show how to do the same analysis on
 6399 the wolverine study as shown in Sec. 5.9. To use the **secr** package, the data need to be
 6400 formatted in a similar but slightly different manner than we use in **WinBUGS**.

6401 For example, in Sec. 5.9 we introduced a standard data format for the encounter data
 6402 file (EDF) and trap deployment file (TDF). The EDF shares the same format as that used
 6403 by the **secr** package with 1 row for every encounter observation and 4 columns representing
 6404 trap session ('Session'), individual identity ('ID'), sample occasion ('Occasion'), and trap
 6405 identity ('trapID'). For a standard closed population study that takes place during a single
 6406 season, the 'Session' column in our case is all 1's, to indicate a single primary sampling
 6407 occasion. In addition to providing the encounter data file (EDF), we must tell **secr** infor-
 6408 mation about the traps, which is formated as a matrix with column labels 'trapID', 'x' and
 6409 'y', the last two being the coordinates of each trap, with additional columns representing
 6410 the operational state of each trap during each occasion (1=operational, 0=not).

6411 We demonstrate these differences now by walking through an analysis of the wolverine
 6412 camera trapping data using **secr**. To read in the trap locations and other related infor-
 6413 mation, we make use of the constructor function **read.traps** which also requires that we
 6414 specify the detector type. The detector type is important because it will determine the
 6415 likelihood that **secr** will use to fit the model. Here, we have selected "proximity" which
 6416 corresponds to the Bernoulli encounter model in which individuals are captured at most
 6417 once in each trap during each sampling occasion:

6418 > library(secr)
 6419 > library(scrbook)

```

6420 > data(wolverine)
6421
6422 > traps <- as.matrix(wolverine$wtraps)
6423 > dimnames(traps) <- list(NULL,c("trapID","x","y",paste("day",1:165,sep="")))
6424 > traps1 <- as.data.frame(traps[,1:3])
6425 > trapfile1 <- read.traps(data=traps1,detector="proximity")

```

6426 Here we note that trap coordinates are extracted from the wolverine data but we do
6427 not scale them. This is because **secr** defaults to coordinate scaling of meters which is
6428 the extant scaling of the wolverine trap coordinates. Note that we add a 'trapID' column
6429 to the trap coordinates and provide appropriate column labels to the 'traps' matrix. An
6430 important aspect of the wolverine study is that while the camera traps were operated over
6431 a 165 day period, each trap was operational during only a portion of that period. We need
6432 to provide the trap operation information which is contained in the columns to the right
6433 of the trap coordinates in our standard trap deployment file (TDF). Unfortunately, this is
6434 less easy to do in **secr**⁴, which requires an external file with a single long string of 1's and
6435 0's indicating the days in which each trap was operational (1) or not (0). The **read.traps**
6436 function will not allow for this information on trap operation if the data exists as an **R**
6437 object – instead, we can create this external file and then read it back in with **read.traps**
6438 using these commands:

```

6439 > hold <- rep(NA,nrow(traps))
6440 > for(i in 1:nrow(traps)){
6441 >   hold[i] <- paste(traps[i,4:ncol(traps)],collapse="")
6442 > }
6443 > traps1 <- cbind(traps[,1:3],"usage"=hold)
6444
6445 > write.table(traps1, "traps.txt", row.names=FALSE, col.names=FALSE)
6446 > trapfile2 <- read.traps("traps.txt",detector="proximity")

```

6447 These operations can be accomplished using the function **scr2secr** which is provided in
6448 the **R** package **scrbook**.

6449 After reading in the trap data, we now need to create the encounter matrix or array
6450 using the **make.capthist** command, where we provide the capture histories in EDF format,
6451 which is the existing format of the data input file **wcaps**. In creating the capture history,
6452 we provide also the trapfile created previously, the format (e.g., here EDF format is
6453 **fmt= "trapID"**), and finally, we provide the number of occasions.

```

6454 #
6455 # Grab the encounter data file and format it:
6456 #
6457 wolv.dat <- wolverine$wcaps
6458 dimnames(wolv.dat) <- list(NULL,c("Session","ID","Occasion","trapID"))
6459 wolv.dat <- as.data.frame(wolv.dat)
6460 wolvcapt2 <- make.capthist(wolv.dat,trapfile2,fmt="trapID",noccaasions=165)

```

⁴as of v. 2.3.1

6461 We also set up a habitat mask using the 2×2 km grid which we used previously in the
 6462 analysis of the wolverine data and then pass the relevant objects to `secr.fit` as follows:

```

6463 #
6464 # Grab the habitat mask (2 x 2 km) and format it:
6465 #
6466 gr2 <- (as.matrix(wolverine$grid2))
6467 dimnames(gr2) <- list(NULL,c("x","y"))
6468 gr2 <- read.mask(data=gr2)
6469 #
6470 # To fit the model we use secr.fit:
6471 #
6472 wolv.secr2 <- secr.fit(wolvcapt2,model=list(D ~ 1, g0 ~ 1, sigma ~ 1),
6473                         buffer=20000,mask=gr2)
```

6474 We are using the “proximity detector” model (SCR0), so we do not need to make any
 6475 specifications in the command line because we have specified the detector type using the
 6476 constructor function `read.traps`, except to provide the buffer size (in meters). To specify
 6477 different models, you can change the default model $D \sim 1$, $g_0 \sim 1$, $\sigma \sim 1$. We provide all
 6478 of these commands and additional analyses in the `scrbook` package with the function called
 6479 `secr_wolverine`. Printing the output object produces the following (slightly edited):

```

6480 > wolv.secr2
6481
6482 secr 2.3.1, 15:52:45 29 Aug 2012
6483
6484 Detector type      proximity
6485 Detector number    37
6486 Average spacing     4415.693 m
6487 x-range             593498 652294 m
6488 y-range             6296796 6361803 m
6489 N animals          : 21
6490 N detections        : 115
6491 N occasions         : 165
6492 Mask area           : 987828.1 ha
6493
6494 Model               : D ~ 1 g0 ~ 1 sigma ~ 1
6495 Fixed (real)        : none
6496 Detection fn        : halfnormal
6497 Distribution         : poisson
6498 N parameters        : 3
6499 Log likelihood      : -602.9207
6500 AIC                 : 1211.841
6501 AICc                : 1213.253
6502
6503 Beta parameters (coefficients)
6504          beta      SE.beta       lcl       ucl
```

```

6505 D      -9.390124 0.22636698 -9.833795 -8.946452
6506 g0     -2.995611 0.16891982 -3.326688 -2.664535
6507 sigma   8.745547 0.07664648  8.595323  8.895772
6508
6509 Variance-covariance matrix of beta parameters
6510          D           g0           sigma
6511 D      0.0512420110 -0.0004113326 -0.003945371
6512 g0     -0.0004113326  0.0285339045 -0.006269477
6513 sigma  -0.0039453711 -0.0062694767  0.005874683
6514
6515 Fitted (real) parameters evaluated at base levels of covariates
6516      link   estimate    SE.estimate      lcl      ucl
6517 D      log 8.354513e-05 1.915674e-05 5.360894e-05 1.301982e-04
6518 g0     logit 4.762453e-02 7.661601e-03 3.466689e-02 6.509881e-02
6519 sigma  log 6.282651e+03 4.822512e+02 5.406315e+03 7.301037e+03

```

6520 The object returned by `secr.fit` provides extensive default output when printed.
6521 Much of this is basic descriptive information about the model, the traps, or the encounter
6522 data. We focus here on the parameter estimates. Under the fitted (real) parameters, we
6523 find D , the density, given in units of individuals/hectare (1 hectare = 10000 m^2). To
6524 convert this into individuals/1000 km², we multiply by 100000, thus our density estimate
6525 is 8.35 individuals/1000 km². The parameter σ is given in units of meters, and so this
6526 corresponds to 6.283 km. Both of these estimates are very similar to those obtained in
6527 our likelihood analysis summarized in Table 6.3 which, for the 2 × 2 km grid, we obtained
6528 $\hat{D} = 8.31$ with a SE of $100000 \times 1.915674e - 05 = 1.9156$ and, accounting for the scale
6529 difference (1 unit = 10000 m in the previous analysis), $\hat{\sigma} = \sqrt{1/(2\hat{\alpha}_1)} * 10000 = 6.289$
6530 km. The difference in the MLE between Table 6.3 and those produced by `secr` could be
6531 due to subtle differences in internal tuning of optimization algorithms, starting values or
6532 other numerical settings. In addition, the likelihood is based on a Poisson prior for N (see
6533 the next section). On the other hand, the SE is slightly larger based on `secr` which is due
6534 to a subtle difference in the interpretation of D under the `secr` model (See below).

6535 6.5.3 Likelihood analysis in the `secr` package

6536 The `secr` package does likelihood analysis of SCR models for most classes of models
6537 as developed by Borchers and Efford (2008). Their formulation deviates slightly from
6538 the binomial form we presented in Sec. 6.2 above (though Borchers and Efford (2008)
6539 also mention the binomial form). Specifically, the likelihood that `secr` implements is that
6540 based on removing N from the likelihood by integrating the binomial likelihood (Eq. 6.2.1
6541 above) over a Poisson prior for N – what we will call the *Poisson-integrated likelihood* as
6542 opposed to the conditional-on- N (*binomial-form*) considered previously.

6543 To develop the Poisson-integrated likelihood we compute the marginal probability of
6544 each \mathbf{y}_i and the probability of an all-0 encounter history, π_0 , as before, to arrive at the
6545 marginal likelihood in the binomial-form:

$$\mathcal{L}(\boldsymbol{\alpha}, n_0 | \mathbf{y}) = \frac{N!}{n! n_0!} \left\{ \prod_i [\mathbf{y}_i | \boldsymbol{\alpha}] \right\} \pi_0^{n_0}$$

6546 Now, what Borchers and Efford (2008) do is assume that $N \sim \text{Poisson}(\Lambda)$ and they do a
 6547 further level of marginalization over this prior distribution:

$$\sum_{n_0=0}^{\infty} \frac{N!}{n_0! n_0!} \left\{ \prod_i [\mathbf{y}_i | \boldsymbol{\alpha}] \right\} \pi_0^{n_0} \frac{\exp(-\Lambda) \Lambda^N}{N!}$$

6548 In Chapt. 11 we write $\Lambda = \mu ||\mathcal{S}||$ where $||\mathcal{S}||$ is the area of the state-space, and μ is the
 6549 density (“intensity”) of the point process. Carrying out the summation above produces
 6550 exactly this marginal likelihood:

$$\mathcal{L}_2(\boldsymbol{\alpha}, \Lambda | \mathbf{y}) = \left\{ \prod_i [\mathbf{y}_i | \boldsymbol{\alpha}] \right\} \Lambda^n \exp(-\Lambda(1 - \pi_0))$$

6551 which is Eq. 2 of Borchers and Efford (2008) except for notational differences. It also
 6552 resembles the binomial-form of the likelihood in Eq. 6.2.1 with $\Lambda^n \exp(-\Lambda\pi_0)$ replacing
 6553 the combinatorial term and the $\pi_0^{n_0}$ term. We emphasize there are two marginalizations
 6554 going on here: (1) the integration to remove the latent variables \mathbf{s} ; and, (2) summation
 6555 to remove the parameter N . We provide a function for computing this in the **scrbook**
 6556 package called **intlik3Poisson**. The help file for that function shows how to conduct a
 6557 small simulation study to compare the MLE under the Poisson-integrated likelihood with
 6558 that from the binomial form.

6559 The essential distinction between our MLE and Borchers and Efford as implemented in
 6560 **secr** is whether you keep N in the model or remove it by integration over a Poisson prior.
 6561 If you have prescribed a state-space explicitly with a sufficiently large buffer, then we
 6562 imagine there should be hardly any difference at all between the MLEs obtained by either
 6563 the Poisson-integrated likelihood or the binomial-form of the likelihood which retains N .
 6564 There is a subtle distinction in the sense that under the binomial form, we estimate the
 6565 realized population size N for the state-space whereas, for the Poisson-integrated form we
 6566 estimate the *prior* expected value which would apply to a hypothetical new study of a
 6567 similar population (see Sec. 5.7.3).

6568 Both models (likelihoods) assume \mathbf{s} is uniformly distributed over space, but for the
 6569 binomial model we make no additional assumption about N whereas we assume N is
 6570 Poisson using the formulation in **secr** from (Borchers and Efford, 2008). Using data
 6571 augmentation we could do a similar kind of integration but integrate N over a binomial
 6572 (M, ψ) prior – which we referred to as the binomial-integrated likelihood in Sec. 4.2.4.
 6573 So obviously the two approaches (data augmentation and Poisson-integrated likelihood)
 6574 are approximately the same as M gets large. However, doing a Bayesian analysis by
 6575 MCMC, we obtain an estimate of both N , the *realized population size*, and the parameter
 6576 controlling its expected value ψ which are, in fact, both identifiable from the data even
 6577 using likelihood analysis (Royle et al., 2007). That said we can integrate N out completely
 6578 and just estimate ψ as we noted in Sec. 6.2.1 above.

6579 6.5.4 Multi-session models in **secr**

6580 In practice we will often deal with SCR data that have some meaningful stratification or
 6581 group structure. For example, we might conduct mist-netting of birds on K consecutive
 6582 days, repeated, say, T times during a year, or perhaps over T years. Or we might collect

6583 data from R distinct trapping grids. In these cases, we have T or R groups which we might
 6584 reasonably regard as being samples of independent populations. While the groups might
 6585 be distinct sites, year, or periods within years, they could also be other biological groups
 6586 such as sex or age. Conveniently, **secr** fits a specific model for stratified populations –
 6587 referred to as *multi-session* models. These models build on the Poisson assumption which
 6588 underlies the integrated likelihood used in **secr** (as described in the previous section). To
 6589 understand the technical framework, let N_g be the population size of group g and *assume*

$$N_g \sim \text{Poisson}(\Lambda_g).$$

6590 Naturally, we model group-specific covariates on Λ_g :

$$\log(\Lambda_g) = \beta_0 + \beta_1 z_g$$

6591 where z_g is some group-specific covariate such as a categorical index to the group, or a
 6592 trend variable, or a spatial covariate, such as treatment effect or habitat structure, if the
 6593 groups represent spatial units. Under this model, we can marginalize *all* N_g parameters
 6594 out of the likelihood to concentrate the likelihood on the parameters β_0 and β_1 precisely
 6595 as discussed in the previous section. This Poisson hierarchical model is the basis of the
 6596 multi-session models in **secr**.

6597 To implement a multi-session model (or stratified population model) in **secr**, we pro-
 6598 vide the relevant stratification information in the ‘Session’ variable of the input encounter
 6599 data file (EDF). If ‘Session’ has multiple values then a “multi-session” object is created
 6600 by default and session-specific variables can be described in the model. For example, if
 6601 the session has 2 values for males and females then we have sex-specific densities , and
 6602 baseline encounter probability p_0 (g_0 in **secr**) by just doing this (see Chapt. 8 for the **R**
 6603 code to set this up):

```
6604 > out <- secr.fit(capdata, model=list(D ~ session, g0 ~ session, sigma^~ 1),  

  6605   buffer=20000)
```

6606 More detailed analysis is given in Sec. 8.1 where we fit a number of different models and
 6607 apply methods of model selection to obtain model-averaged estimates of density.

6608 We can also easily implement stratified population models in the various **BUGS** en-
 6609 gines using data augmentation (Converse and Royle, 2012; Royle and Converse, in review)
 6610 which we discuss, with examples, in Chapt. 14.

6611 6.5.5 Some additional capabilities of **secr**

6612 The **secr** package has capabilities to do a complete analysis of SCR data sets, including
 6613 model fitting, selection, and many summary analyses. In the previous sections, we’ve
 6614 given a basic overview, and we do more in later chapters of this book. Here we mention a
 6615 few of these other capabilities that you should know about as you use **secr**. Of course, you
 6616 should skim through the associated documentation (`?secr`) to see more of what’s available.

6617 Alternative observation models

6618 **secr** fits a wide range of alternative observation models besides the Bernoulli encounter
 6619 model, including multinomial encounter models for “multi-catch” and “single catch” traps,
 6620 models for sound attenuation from acoustic detection devices, and many others. We
 6621 discuss many of these other methods in Chapt. 9 and elsewhere in the book.

Summary statistics

6623 **secr** provides a useful default summary of the data, but it also has summary statistics
 6624 about animal movement including mean-maximum distance moved (the function **MMDM**).
 6625 For example, see the help page **?MMDM** which lists a number of other summary functions
 6626 which take a **capthist** object:

```
6627 > moves(capthist)
6628 > dbar(capthist)
6629 > RPSV(capthist)
6630 > MMDM(capthist, min.recapt = 1, full = FALSE)
6631 > ARL(capthist, min.recapt = 1, plt = FALSE, full = FALSE)
```

6632 The function **moves** returns the observed distances moved, **dbar** returns the average dis-
 6633 tance moved, **RPSV** produces a measure of dispersion about the home-range center, and
 6634 **ARL** gives the *Asymptotic Range Length* which is the asymptote of an exponential model
 6635 fit to the observed range length vs. the number of detections of each individual (Jett and
 6636 Nichols, 1987).

State-space buffer

6637 **secr** will produce a warning if the state-space buffer is chosen too small. For example,
 6638 in fitting the wolverine data as in Sec. 6.5.2 but with a 1000 m buffer, and we see the
 6639 following warning message:

```
6641 Warning message:
6642 In secr.fit(wolvcapt2, model=list(D ~ 1, g0 ~ 1, sigma ~ 1), buffer=1000):
6643   predicted relative bias exceeds 0.01 with buffer = 1000
```

6644 This should cause you to contemplate modifying the state-space buffer if that is a reason-
 6645 able thing to do in the specific application.

Model selection and averaging

6646 **secr** does likelihood ratio tests to compare nested models using the function **LR.test**.
 6647 You can create model selection tables based on AIC or AICc, using the function **AIC**,
 6648 and obtain model-averaged parameter estimates using the function **model.average** (See
 6649 Chapt. 8 for examples).

Population closure test

6650 **secr** has a population closure test with the function **closure.test** which implements the
 6651 tests of Stanley and Burnham (1999) or Otis et al. (1978). The function is used like this:
 6652 **closure.test(object, SB = FALSE)**. Here **object** is a **capthist** object and **SB** is a logical
 6653 variable that, if TRUE, produces the Stanley and Burnham (1999) test.

Density mapping and effective sample area

6654 **secr** produces likelihood versions of the various summaries of posterior density and effec-
 6655 tive sample area that we discussed in Chapt. 5. For example, while **secr** reports estimates
 6656 of the expected value of N or density directly in the summary output from fitting a model,
 6657 you can use the function **region.N** to produce estimates of N for any given region. In
 6658 addition, **secr** has functions for creating maps of detection contours for individuals traps,
 6659 or for the entire trap array. See the function **pdot.contour**, and also **fxi.contour** for

6663 computing the 2-dimensional pdf of the locations of one or more individual activity cen-
 6664 ters (as in Sec. 5.11.3). In the context of likelihood analysis, estimation of a random effect
 6665 **s** is based on a plug-in application of Bayes' Rule. When **s** has a uniform distribution, and
 6666 we use a discrete evaluation of the integral, it can be computed simply by renormalizing
 6667 the likelihood:

$$[s|y, \theta] = \frac{[y|s, \theta]}{\sum_s [y|s, \theta]}.$$

6668 Any of the **intlik** functions given previously in this chapter can be easily modified to
 6669 return the posterior distribution of **s** for any, or all, individuals, or an individual that is
 6670 not encountered.

6671 Effective sample area (see Sec. 5.12) can be calculated in **secr** using the functions **esa**
 6672 and **esa.plot**).

6673 Covariate models

6674 **secr** has many capabilities for modeling covariates. It has a number of built-in models
 6675 that allow certain covariates on encounter probability, which we cover to a large extent
 6676 in Chapt. 7, and also see Chapt. 8 for more examples. **secr** also allows covariates to be
 6677 built into the density model (see Chapt. 11). It has some built in response surface models,
 6678 allowing for the fitting of linear or quadratic response surfaces. This is done by modifying
 6679 the density model in **secr.fit**. For example, $D \sim 1$ is a constant density surface, and
 6680 $D \sim x + y$ fits a linear response surface, etc.. See the manual **secr-densitysurfaces.pdf**
 6681 for the details.

6682 There are a number of ways to model your own "custom" covariates (as opposed to
 6683 pre-specified models). One way is to use the **addCovariates** function and supply it a
 6684 **mask** or **traps** object along with some "spatialdata." Or, if you have covariates at each
 6685 trap location then it will extrapolate to all points on the habitat mask. There's also a
 6686 method by which the user can create a function of geographic coordinates, **userDfn**, which
 6687 seems to provide additional flexibility, although we haven't used this method. There is a
 6688 handy function **predictDsurface** for producing density maps under the specified model
 6689 for density.

6.6 SUMMARY AND OUTLOOK

6690 In this chapter, we discussed basic concepts related to classical analysis of SCR models
 6691 based on likelihood methods. Analysis is based on the so-called integrated or marginal
 6692 likelihood in which the individual activity centers (random effects) are removed from the
 6693 conditional-on-**s** likelihood by integration. We showed how to construct the integrated
 6694 likelihood and fit some simple models in the **R** programming language. In addition,
 6695 likelihood analysis for some broad classes of SCR models can be accomplished using the
 6696 **R** library **secr** (Efford, 2011) which we provided a brief introduction to. In later chapters
 6697 we provide more detailed analyses of SCR data using likelihood methods and the **secr**
 6698 package.

6699 Why or why not use likelihood inference exclusively? For certain specific models, it
 6700 is may be more computationally efficient to produce MLEs (for an example see Chapt.
 6701 12). And, likelihood analysis makes it easy to do model-selection by AIC and compute
 6702 standard errors or confidence intervals. However, **BUGS** is extremely flexible in terms
 6703 of describing models and we can devise models in the **BUGS** language easily that we

6704 cannot fit in **secr**. For example, in Chapt 16 we consider open population models which
6705 are straightforward to develop in **BUGS** but, so far, there is no available platform for
6706 doing MLE of such models. We can also fit models in **BUGS** that accommodate missing
6707 covariates in complete generality (e.g., unobserved sex of individuals), and we can adopt
6708 SCR models to include auxiliary data types. For example, we might have camera trapping
6709 and genetic data and we can describe the models directly in **BUGS** and fit a joint model
6710 (Gopalaswamy et al., 2012b). To do maximum likelihood estimation, we have to write a
6711 custom new piece of code for each model⁵ or hope someone has done it for us. You should
6712 have some capability to develop your own MLE routines with the tools we provided in
6713 this chapter.

⁵Although we may be able to handle multiple survey methods together in **secr** using the multi-session models.

6714
6715

7

MODELING ENCOUNTER PROBABILITY

6717 In previous chapters we showed how to fit basic spatial capture-recapture models using
6718 Bayesian analysis (in **WinBUGS** or **JAGS**; Chapt. 5) or by classical likelihood methods
6719 (Chapt. 6 or using **secr**). We covered a suite of possible encounter models (e.g., the
6720 Binomial, Poisson, and Multinomial) for dealing with different types of sampling. We have
6721 not, however, described a general framework for modeling covariates that might influence
6722 encounter probability of individuals, traps or over time. In practice, investigators are
6723 invariably concerned with explicit factors or covariates that might influence variation in
6724 parameters. Such covariates include time (e.g., day of year, or season), behavior (e.g.,
6725 is there an effect of trapping on subsequent capture probabilities), sex of the individual,
6726 and trap type (e.g., various camera types, or different constructions for hair snares).
6727 Traditionally, in the non-spatial capture recapture literature, such models were called
6728 “model M_t ”, “model M_h ”, or “model M_b ”, identifying models that account for variation
6729 in detection probability as a function of time, “individual heterogeneity” or “behavior”,
6730 where behavior describes whether or not an individual had been previously captured. In
6731 SCR models, more complex covariate models are possible because we might also have trap-
6732 specific covariates, or covariates that vary spatially over the landscape, and because we
6733 generally have more than one parameter describing the detection function: Most detection
6734 functions include a baseline encounter rate (λ_0) or probability (p_0) parameter, and a shape
6735 parameter (σ), which takes on different interpretations depending on the specific encounter
6736 probability function under consideration.

6737 In this chapter, we generalize the basic SCR model to accommodate both alternative
6738 detection functions as well as many different kinds of covariates. We focus on the binomial
6739 observation model used throughout Chaps. 5 and 6 and the Gaussian encounter model
6740 (also called the “half-normal” model in the distance sampling literature), but the extension
6741 to other observation models is straightforward (and other encounter probability models
6742 with different functions of distance are considered in Sec. 7.1). Specifically, we consider
6743 three distinct types of covariates – those which are fixed, partially observed or completely
6744 unobserved (latent). Fixed covariates are those that are fully observed; for example, the
6745 date of all sampling occasions. Partially observed covariates are those which are not known

for all observations; for example, the sex of an individual cannot always be determined from photos taken during camera trapping. Even if we are able to observe the sex of all individuals sampled, we cannot know it for those individuals never observed during the study. And finally, unobserved covariates are those which we cannot observe at all, for example, the home range size of individuals, or unstructured random “individual effects”.

We will see that models containing these different types of covariates are relatively easy to describe in **WinBUGS** or **JAGS**, and therefore to analyze using Bayesian analysis of the joint likelihood based on data augmentation thus providing a coherent and flexible framework for inference for all classes of SCR models. Throughout the chapter, we will continue to develop the analysis of the black bear study introduced in Chapt. 4, using the software **JAGS**. We also consider the likelihood analysis of many of these models; to do so, we will demonstrate the use of the **R** package **secr** and how to do model comparison with AIC (Sec. 7.4 at the end of the chapter). There are other types of covariates that we do *not* cover in this chapter; for example, covariates that vary across the landscape might affect density, and we consider these covariates in Chapt. 11. Alternatively, these landscape covariates might affect the way individuals use space. There are probably very few circumstances under which animals use all space uniformly and we develop more realistic models of encounter probability in which covariates affect space usage in Chapt. 12.

7.1 ENCOUNTER PROBABILITY MODELS

In Chapt. 5, we developed a basic spatial capture recapture model using a standard detection probability function based on the kernel of a normal (Gaussian) probability distribution:

$$p_{ij} = p_0 \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

where $||\mathbf{x}_j - \mathbf{s}_i||$ is the distance between \mathbf{x}_j and \mathbf{s}_i and $\alpha_1 = 1/(2 * \sigma^2)$. We argued (see Sec. 5.4) that this model corresponds to an explicit model of space usage – namely, that individual locations are draws from a bivariate normal distribution. We also mentioned that other detection models are possible, including a logit model of the form:

$$\text{logit}(p_{ij}) = \alpha_0 + \alpha_1 ||\mathbf{x}_j - \mathbf{s}_i||. \quad (7.1.1)$$

However, there's nothing preventing us from constructing a myriad of other models for detection probability as a function of distance. The most commonly used detection probability models are also those used in the distance sampling literature: the half-normal (Gaussian), the hazard, and the negative exponential. The negative exponential model is:

$$p_{ij} = p_0 * \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||)$$

where we define $\alpha_1 = 1/\sigma$. We could use the general power model (Russell et al., 2012):

$$p_{ij} = p_0 * \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^\theta)$$

of which the Gaussian and exponential models are special cases. Another model that could be considered is the Gaussian hazard rate model (Hayes and Buckland, 1983):

$$p_{ij} = 1 - \exp(-\lambda_0 * \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2))$$

6779 which was previously discussed in Sec. .

6780 In each of the cases, the relationship of α_1 to σ varies and must be properly specified.
 6781 The **R** package **secr** allows the user to access 12 different detection models, of which some
 6782 are only used for simulating data (see Table 7.1). These detection functions can also be
 6783 implemented in **R**, **WinBUGS**, **JAGS** etc..

Table 7.1. Basic encounter probability models (“distance functions”) available in **secr**.
 (Table taken from the **secr** help files). Notation deviates from that used in the text. In
 this table g_0 is the baseline encounter rate or probability parameter used in **secr** 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	$g(d) = 1 - F[\{c - (\beta_0 + \beta_1(d-1) - 10 * \log_{10}(d^2))\}/S]$

6784 Insofar as all these encounter probability models are symmetric and stationary, they
 6785 are pretty crude descriptions of space usage by real animals. This is not to say they are
 6786 inadequate descriptions of the data and, as we discuss in Chaps. 13 and 12, we can use
 6787 them as the basis for producing more realistic models of space usage.

6788 By changing the encounter probability model and the specification of α_1 , we can
 6789 basically create any function of distance for the data. It is important to note that σ is not
 6790 comparable under these different encounter probability models and should not be regarded
 6791 as “home range radius” in general. While there is generally a relationship between σ and
 6792 home range size, that relationship varies depending on the model under consideration. We
 6793 demonstrate how to fit different encounter probability models in the Bayesian framework
 6794 here, and then provide a section on the likelihood analysis (in **secr**) in a separate section
 6795 below.

6796 **7.1.1 Bayesian analysis with bear.JAGS**

6797 To demonstrate how to incorporate various types of covariates into models for encounter
 6798 probability using **JAGS**, we return to the data collected during the Fort Drum bear study.
 6799 This data set was first introduced in Chapt. 4, but, to refresh your memory, there were
 6800 38 baited hair snares that were operated between June and July 2006. The snares were
 6801 checked each week for a total for $K = 8$ sample occasions and $n = 47$ individual bears
 6802 were encountered at least once. The data are provided in the **R** package **scrbook** and an
 6803 **R** function called **bear.JAGS** allows the user to easily pick which model to analyze. The
 6804 function **bear.JAGS** will set up the data, write the model, define the MCMC specifications
 6805 (e.g., initial values, etc.) and, finally, run the selected model in **JAGS**. In addition to
 6806 choosing which model to run, the user can also specify the number of chains, iterations and
 6807 length of the burn-in phase. Calling the function will provide all the code to implement
 6808 the models independently as well. In the following sections we will present the model code
 6809 and output for the most commonly employed models; for all analyses we ran 3 chains with
 6810 a burn-in of 500 iterations and 20000 saved iterations.

6811 **7.1.2 Bayesian analysis of encounter probability models**

6812 In Panel 7.1, we present the basic SCR model and show how to specify the negative ex-
 6813ponential encounter probability model. To call each of these from the function **bear.JAGS**
 6814 set **model='SCR0'** or **model='SCRexp'** in the function call, respectively. To reduce repe-
 6815 tition of the R coding, we include the basic code here and then only show modifications
 6816 when necessary throughout the chapter. All of the R coding can be found within the
 6817 **bear.JAGS** function as well. To begin, the required R libraries are installed and then we
 6818 attached the Ft. Drum bear data set. The bear data set includes a 3-d data array (called
 6819 **bearArray** in our code), with dimensions **nind** \times **ntraps** \times **nreps** representing the capture
 6820 histories of **nind** captured individuals at **ntraps** trap locations. In the Bayesian analysis,
 6821 data augmentation is used to estimate N and therefore the **bearArray** data must be aug-
 6822 mented with $M - nind$ all zero encounter histories. In models without time dependence,
 6823 the augmented **bearArray** (called **Yaug** in the code) will be reduced to a 2 dimensional
 6824 array (denoted **y** in the code) that has dimensions **M** \times **ntraps**.

```
6825 > library(rjags) #load the necessary libraries
6826 > library(scrbook)
6827
6828 > data(beardata) #attach the bear data for Ft. Drum
6829 > ymat <- beardata$bearArray
6830 > trapmat <- beardata$trapmat
6831 > nind <- dim(beardata$bearArray)[1]
6832 > K <- dim(beardata$bearArray)[3]
6833 > ntraps <- dim(beardata$bearArray)[2]
6834 > M <- 650
6835 > nz <- M-nind
6836
6837 #create augmented array
6838 > Yaug <- array(0, dim=c(M,ntraps,K))
```

```

6839 > Yaug[1:nind,,] <- ymat
6840 > y <- apply(Yaug,1:2, sum)

```

6841 The function `bear.JAGS` also establishes the upper and lower limits on the state space
 6842 by centering the trap array coordinates (which are imported with the `beardata` and saved
 6843 in the code above as `trapmat`) and then buffering by 20km.

```

model{
  alpha0 ~ dnorm(0,.1)                               # Prior distributions
  logit(p0) <- alpha0
  alpha1 <- 1/(2*sigma*sigma)
  sigma ~ dunif(0, 15)
  psi ~ dunif(0,1)

  for(i in 1:M){
    z[i] ~ dbern(psi)
    s[i,1] ~ dunif(xlim[1],xlim[2])
    s[i,2] ~ dunif(ylim[1],ylim[2])
    for(j in 1:J){
      d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
      y[i,j] ~ dbin(p[i,j],K)
      p[i,j] <- z[i]*p0*exp(-alpha1*d[i,j]*d[i,j]) # Gaussian model
      #p[i,j] <- z[i]*p0*exp(-alpha1*d[i,j])        # exponential model
    }
  }
  N <- sum(z[])
  D <- N/area
}

```

Panel 7.1: **JAGS** model specification for a basic SCR model with Gaussian encounter probability function and the alternative exponential encounter probability function.

6844 Applying the SCR model with Gaussian encounter probability model provides an
 6845 estimate (posterior mean) of $D = 0.167$ bears per km^2 and with the negative exponential
 6846 encounter probability model the posterior mean is virtually the same $D = 0.167$. In
 6847 distance sampling, the use of different encounter probability models often results in very
 6848 different estimates of density (especially when using the negative exponential model).
 6849 There are two main reasons why the different models may have less of an impact on the
 6850 density estimates under the SCR models. First, we can estimate the baseline encounter
 6851 probability parameter (p_0). In most distance sampling models, detection at distance 0 is
 6852 set to 1. In Table 7.2, the posterior mean of p_0 is 0.11 under the Gaussian model and 0.34
 6853 under the negative exponential model. The larger baseline encounter probability under

6854 the negative exponential model reduces the impact of the having “no shoulder”. Secondly,
 6855 the detection probability function here is governing ‘movement’ of individuals (which we
 6856 have more information on than in distance sampling), not the whole detection process, so
 6857 the shape of the detection probability function does not impact the density estimation as
 6858 much.

6859 In all analyses it is important to check that the size of the augmented data set (M) is
 6860 sufficiently large and does not impact the estimate of N . Here, the 97.5% percentile for N
 6861 is 628 (Table 7.2), thus not reaching our $M = 650$ value. We could also increase M and
 6862 compare the posterior of N under the different scenarios as another check that the data
 6863 augmentation is sufficient.

Table 7.2. Posterior summaries of SCR model parameters having different encounter probability models, for the Fort Drum black bear data.

Parameter	Mean	SD	2.5	97.5
Gaussian				
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

6864 A very important consideration when using different detection probability functions
 6865 is the interpretation of σ . The estimate of σ under the negative exponential model is
 6866 1.12, which is distinct from our estimate of σ under the Gaussian model, $\sigma = 1.996$. The
 6867 interpretation of σ in the two models is really quite distinct. In the normal model it can be
 6868 interpreted as the standard deviation of a bivariate normal movement model whereas the
 6869 manner in which σ relates to “area used” for the negative exponential model has nothing
 6870 to do with a bivariate normal model of movement. This highlights that it is important for
 6871 the user to know what detection probability function is used and what the interpretation
 6872 of σ might be in relation to the home range size. This relationship was discussed in Sec.
 6873 5.4.

6874 We now move onto incorporating covariates into the model using the **JAGS** language.
 6875 For this part, we will stick with the Gaussian encounter probability model shown in the
 6876 Panel 7.1 above.

7.2 MODELING COVARIATE EFFECTS

6877 The basic strategy for modeling covariate effects is to include them on the baseline en-
 6878 counter rate or probability parameter, p_0 (or λ_0), or the scale parameter of the encounter
 6879 model, σ , or in some cases, both parameters.

6880 Broadly speaking, we recognize (here) 3 types of covariates. Fixed covariates that are
 6881 fully observable and might vary by trap alone (e.g., type of trap, baited or not, disturbance
 6882 regime, even habitat), sample occasion (e.g., day of season or weather conditions), or both
 6883 (e.g., behavior, weather - if over a large region). Another class of covariates are those
 6884 which vary at the level of the individual (and possibly also over time). As a technical
 6885 matter, and as noted before, these are different from fixed covariates because we cannot
 6886 see all of the individuals and the covariates are almost always incompletely observed (if
 6887 at all). The lone exception is the behavioral response to capture which is known for all
 6888 individuals, captured or not (an animal never captured/observed has never been captured
 6889 before). We noted many times before that space itself (i.e., the activity centers) is a
 6890 type of individual covariate and this notion actually helped us derive the fully spatial
 6891 capture-recapture model from the traditional, non-spatial model (Chapt. 4). We do
 6892 not get to observe the activity center for any individuals, but for individuals that are
 6893 encountered we get to observe some information about it in the form of which traps the
 6894 individual was encountered in. And finally, we have completely unobserved covariates
 6895 such as heterogeneity in home range size. We consider heterogeneity in a separate section
 6896 below since alone there are a suite of models for describing latent heterogeneity.

Table 7.3. Examples of different types of covariates in SCR models.

Covariate type	Examples
individual	sex, age, home range
trap	baited/not, habitat (see also Chapter 13)
time	season, shedding, weather
individual x time	global behavioral response
trap x time	trap failures
individual x trap x time	local behavioral response

6897 To develop covariate models, we assume a standard sampling design in which an array
 6898 of J traps is operated for K sample occasions, which produces encounter histories for n
 6899 individuals. For the null model, there are no time-varying covariates that influence en-
 6900 counter, there are no explicit individual-specific covariates, and there are no covariates
 6901 that influence density. For fixed effects, those which we observe fully, we can easily incor-
 6902 porate these into the encounter probability model, just as we would do in any standard
 6903 GLM or GLMM, on some suitable scale for the encounter probability, p_{ijk} . For example,

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * C_{ijk}$$

$$p_{ijk} = p_{0,ijk} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

6904 where C_{ijk} is some covariate that varies (potentially) by individual (i), trap (j) and
 6905 occasions (k), and α_2 is the coefficient to be estimated. How we define specific covariates
 6906 (e.g., trap specific versus individual specific) will influence exactly how we include them
 6907 in the model. Table 7.3 shows examples of covariates by type – trap, individual, and time
 6908 – and also gives examples of some combined types. These are the types of covariates we
 6909 will specifically address in this chapter demonstrating how to analyze the different types
 6910 in the following sections.

6911 **7.2.1 Date and time**

6912 Often, researchers are interested in modeling the effect of date or chronological time on
 6913 encounter probability. For example, in a long term hair snare study, we may expect that
 6914 seasonal shedding (Wegan et al., 2012) will influence encounter probabilities directly. Or,
 6915 we may expect behaviors such as denning, mating, etc., to influence the encounter of
 6916 certain species at certain times of year (Kéry et al., 2011). There are two common ways
 6917 to incorporate date or time information into a model for encounter probability. For cases
 6918 with a small number of sampling occasions we can fit a time-specific intercept (analogous
 6919 to “model M_t ” in classical capture-recapture (Otis et al., 1978)). In this model, there are
 6920 K sampling occasion-specific parameters to reflect potential variation in sampling effort
 6921 or other factors that might vary across samples. Alternatively, we can model parametric
 6922 functions of date or time such as polynomial or sinusoidal functions.

6923 In the first case, we allow each sampling occasion, k , to have its own baseline encounter
 6924 probability, e.g.,

$$\text{logit}(p_{0,k}) = \alpha_{0,k}$$

6925 so that

$$p_{ijk} = p_{0,k} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2).$$

6926 This description of the model includes k occasion-specific baseline encounter probabilities.
 6927 Thus, if there were 4 sampling occasions, then there are 4 different baseline encounter
 6928 probabilities. We imagine that complete time-specificity of p_0 (i.e., one distinct value
 6929 for each sample occasion) would be most useful in situations where there are just a few
 6930 sampling occasions (if there are many, this formulation will dramatically increase the
 6931 number of parameters to be estimated) or we do not expect systematic patterns over time
 6932 (e.g., explainable by a polynomial function).

6933 To implement this in **JAGS**, α_0 has to be estimated for each time period k either
 6934 using an index vector or dummy variables (as described in Chapt. 2 and Sec. 4.3) and this
 6935 can be done by only changing only a few lines in Panel 7.1:

```
6936 alpha0[k] ~ dnorm(0, 1)
6937 logit(p0[k]) <- alpha0[k]
6938 .....
6939 .....
6940 y[i,j,k] ~ dbin(p[i,j,k], K)
6941 p[i,j,k] <- z[i]*p0[k]*exp(- alpha1*d[i,j]*d[i,j])
```

6942 Since the model contains a parameter for each time period, the encounter histories
 6943 must be time-dependent. Thus, a 3-d data array (called **bearArray** in our code), with
 6944 dimensions **nind** × **ntraps** × **nreps** is required (recall that we use the 3-d augmented array
 6945 called **Yaug** with dimensions **M** × **ntraps** × **nreps** for the Bayesian analysis). In addition
 6946 to using the 3-d data array, the initial values must be updated so that there are K values
 6947 generated for α_0 . And finally, this means that another nested for loop is needed in the
 6948 code to account for the K sample occasions. A side note: the computation time will
 6949 increase quite a bit (this model for the bear data may take up to 15 hours or more on
 6950 your machine to obtain a sufficient posterior sample).

6951 Running this model with the function **bear.JAGS** by setting **model=SCRt**, returns esti-
 6952 mates of density similar to those from the model without covariates (see Table 7.4), but

now we have a characterization of variation in encounter probability over time. Encounter probability seems to increase for the first few time periods before stabilizing around 0.14, dropping off again at the end of the study. The differences in encounter probability from the first time periods to the others might actually be due to something like a behavioral response (see below) or possibly seasonal differences in the efficiency of the sampling technique. Researchers have found that hair snares are more effective at different times of the year (even within season) due to shedding (Wegan et al., 2012). In this particular example, our density estimates are similar to the base model, likely because the differences in encounter probability between occasion were not that large. In a longer term study or in one with greater variation in the encounter probability, the implication of such differences might have a bigger impact on the estimates of density and σ .

Table 7.4. Posterior summaries of parameter estimates from a SCR model with time-dependent baseline encounter probability for the Ft. Drum black bear data set.

Parameter	Mean	SD	2.5	97.5
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. 5.9 where there were 165 daily sampling occasions). Particularly in such a case, variation in the encounter process over time is to be expected. For example, if a camera trap study is conducted for an entire year, it is expected that there would be behavioral patterns in individuals due to mating or denning. Instead of fitting a model with K baseline encounter probabilities, we can include date as a linear (or quadratic, ...) effect. An example can be found in Kéry et al. (2011) who incorporated a day-of-year covariate, both as a linear and a quadratic effect, into their SCR model of European wildcats; the data had been collected over a year-long period and cat behavior was expected to vary seasonally thus influencing the probability of encounter. In these cases, we would specifically incorporate day of year (variable “Date”) as a numeric covariate as:

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * \text{Date}_k$$

$$p_{ijk} = p_{0,ijk} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

6977 or a quadratic effect of day-of-year:

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * \text{Date}_k + \alpha_3 * \text{Date}_k^2 \\ p_{ijk} = p_{0,ijk} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

6978 where the variable **Date** is an integer coding of day-of-year, indexed to some arbitrary
6979 start point in time.

6980 7.2.2 Trap-specific covariates

6981 In some studies it makes sense to model encounter probability as a function of local or trap-
6982 specific covariates. These can be one of two types: genuine trap covariates that describe
6983 the trap or encounter site, such as whether a trap is baited or not, or how many traps were
6984 set at a sampling location, or what kind of bait was used, etc., or local covariates that
6985 describe the likelihood that an animal would use the habitat in the vicinity of the trap
6986 (see Chapt. 13 for more on this situation). We imagine that these covariates, of either
6987 type, should affect baseline encounter probability. For example, Sollmann et al. (2011)
6988 found a large difference in the encounter probability of jaguars due to traps being located
6989 on roads, which the animals were using to travel along, as opposed to traps placed off
6990 of roads. In this case, the trap type is a binary variable – on/off road, (another binary
6991 variable could be baited/non-baited). We can write this as:

$$\text{logit}(p_{0,j}) = \alpha_{0,type_j} \\ p_{ijk} = p_{0,j} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2).$$

6992 Here, we use an index variable, “type”, an integer value for the trap-specific covariate.
6993 Thus for our example of on/off road, we would have $type_j = 1$ if trap j is on a road
6994 and $type_j = 2$ otherwise, and we would estimate two separate α_0 parameters – one for
6995 on-road and one for off-road cameras. This general set up also allows for more than 2
6996 categories, say if 4 different camera models were used in a study, we would use a set of
6997 3 binary dummy variables to allow for estimation of the different encounter rates (i.e.,
6998 the intercept). To express the model in terms of dummy variables using the 2-category
6999 example above, we would specify our “type” vector as $Type_j = 0$ if trap j is on a road and
7000 $Type_j = 1$ otherwise, and write our model as

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * Type_j$$

7001 Now, α_0 is the baseline encounter probability (on the logit scale) for traps on a road
7002 ($Type_j = 0$) and α_2 is the effect on baseline encounter probability of a trap being of
7003 $Type = 1$. While these models are equivalent, and should yield identical results, sometimes
7004 one parameterization might work better than the other in **WinBUGS** or **JAGS** (Kéry,
7005 2010).

7006 7.2.3 Behavior or trap response by individual

7007 One of the most basic of encounter models is that which accommodates a change in
7008 encounter probability as a result of initial encounter. This is colloquially referred to as

7009 “trap happiness” or “trap shyness”, or in other words, a behavioral response of individuals
 7010 to being captured (Otis et al., 1978). If a trap is baited with a food source, an individual
 7011 might come back for more. On the other hand, if being captured is traumatic then an
 7012 individual might learn to avoid traps. Both of these types of responses can occur in
 7013 most species depending on the type of encounter mechanisms being employed. Moreover,
 7014 behavioral response can be either global (Gardner et al., 2010b) or local (Royle et al.,
 7015 2011b). The local response is a trap-specific response while a global response suggests that
 7016 initial capture provides a net increase or decrease in subsequent probabilities of capture
 7017 (across all traps). A behavioral response does not need to be enduring (i.e., persist for
 7018 the entire study after the individual has been captured/observed for the first time) but
 7019 can also be ephemeral, if, for example, an animal only avoids a trap on the occasion
 7020 immediately after it was captured (Yang and Chao, 2005; Royle, 2008). While we will
 7021 focus the examples in this chapter on enduring behavioral effects, extending such a model
 7022 to the case of an ephemeral response should not pose any difficulties.

7023 To describe these behavioral models we need to create a binary matrix that indicates
 7024 if an individual has been captured previously. For the global behavioral response, define
 7025 the $n \times K$ matrix, \mathbf{C} , where $C_{ik} = 1$ if individual i was captured at least once prior to
 7026 session k , otherwise $C_{ik} = 0$.

$$\text{logit}(p_{0,ik}) = \alpha_0 + \alpha_2 * C_{ik}$$

$$p_{ijk} = p_{0,ik} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

7027 For the local behavioral response, which is trap specific, we create an array, C_{ijk} , that
 7028 indicates if an individual i has been previously captured in trap j at time k . We then
 7029 include this in the model in the exact same form as above (with the sole difference that
 7030 both C and p are now also indexed by k):

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * C_{i,j,k}$$

$$p_{ijk} = p_{0,ijk} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

7031 Since the behavioral response is occasion specific, to implement either the local or
 7032 global response model in **JAGS**, we will have to use the 3-d array of the augmented
 7033 capture histories ($M \times ntraps \times nreps$) as we did for the time-varying encounter probability
 7034 model above. The code must loop over each sampling occasion, but otherwise, the model
 7035 varies only a little from the basic SCR model shown in Panel 7.1. Here is the specification
 7036 of the the occasion specific (k) loop:

```
7037 for(k in 1:K){
  7038   logit(p0[i,j,k]) <- alpha0 + alpha2*C[i,j,k]
  7039   y[i,j,k] ~ dbin(p[i,j,k],1)
  7040   p[i,j,k] <- z[i]*p0[i,j,k]*exp(- alpha1*d[i,j]*d[i,j])
  7041 }
```

7042 Despite the minor changes to the **BUGS** code, this model can require quite a bit of
 7043 time and computational effort to carry out the behavior response models. Implementing
 7044 the behavioral models with the function **bear.JAGS** by setting **model=SCRb** or **model=SCRB**
 7045 for the local or global model respectively, returns the results, shown in Table 7.5. There
 7046 is a strong global behavior response suggested by the posterior mean of $\alpha_2 = 0.90$. The

7047 estimate of N and subsequently D are larger than under the model without a behavioral
 7048 response, here we estimate $N = 577.56$ and in the SCR0 model, we estimated $N = 500$.
 7049 This makes sense given the large estimate of α_2 , which suggests that bears are trap happy.
 7050 In situations where animals are trap happy, the model tends to over estimate encounter
 7051 probability (i.e., the bears that are never observed have a lower encounter probability than
 7052 those that have been captured in the study) and thereby reduce the estimate of N . We
 7053 do not include the results here, but the estimates were similar under the local behavioral
 7054 response model.

Table 7.5. 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

7055 7.2.4 Individual covariates

7056 Individual covariates are those which are measured (or measurable) on individuals, so
 7057 we get to observe them only for the captured individuals. Sex is a simple example of
 7058 an individual covariate, but one of the most commonly used in capture-recapture studies.
 7059 The sex of an individual can influence many aspects of its ecology and behavior, including,
 7060 for example, its home range size, frequency of movement, and seasonal behavior. This is
 7061 common in studies of carnivores where females often have smaller home ranges than males
 7062 (Gardner et al., 2010b; Sollmann et al., 2011). Additionally, we may find differences in
 7063 the baseline encounter probability between males and females because females may move
 7064 around less frequently, or possibly because they are less likely to use landscape structures
 7065 that researchers may target with sampling devices in order to increase sample size, such
 7066 as roads (e.g. Salom-Pérez et al., 2007). Therefore, we can imagine that sex may impact
 7067 both the baseline encounter probability α_0 and the typical home range size, so that α_1
 7068 might also be sex-specific also. The fully sex-specific model is:

$$\text{logit}(p_{0,i}) = \alpha_{0,sex_i}$$

$$p_{ijk} = p_{0,i} \exp(-\alpha_{1,sex_i} * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

7069 where sex_i is a vector indicating the sex of each individual (1 = male, 2 = female). While
 7070 we might know the sex of all individuals observed in the study, we will never know the sex
 7071 of individuals that are not observed, resulting in missing values (Gardner et al., 2010b).
 7072 It is also possible that we may not be able to determine the sex of individuals that are
 7073 observed during the study. For example photographic captures do not necessarily result
 7074 in pictures that allow the sex to be absolutely determined, thus sometimes resulting in
 7075 missing values of this covariate for animals captured in the study. We deal with this slightly

7076 differently depending on the inference framework that we adopt (Bayesian or likelihood).
 7077 Here we demonstrate the Bayesian implementation and we discuss the likelihood approach
 7078 using `secr` in detail below in Sec. 7.4.2. Before proceeding with that, we note that it
 7079 would be possible also to model covariates directly on the parameter σ (or its logarithm),
 7080 e.g., $\log(\sigma_i) = \theta_1 + \theta_2 \text{sex}_i$ (see Sec. 8.1). One or the other (or perhaps *some* other)
 7081 parameterization may yield a better performing MCMC algorithm or provide a more
 7082 natural or preferred interpretation. In the context of Bayesian analysis, given that priors
 7083 are not invariant to transformation of the parameters, this may be a consideration in
 7084 choosing the particular parameterization.

7085 Specifying a fully sex-specific model for **JAGS** is similar to the time-specific model
 7086 shown above. We need to use an index or dummy variable to let α_0 and/or α_1 be defined
 7087 separately for males and females. The main difference in this specification is that we do
 7088 not observe sex for the augmented individuals. Therefore, we have missing observations
 7089 of the covariate for those individuals. As a result, sex is regarded as a random variable
 7090 and so the missing values can be estimated along with the other structural parameters of
 7091 the model.

7092 Because we are regarding sex as a random variable, we have to specify a distribution for
 7093 it. With only two possible outcomes, it is natural to suppose that $\text{Sex}_i \sim \text{Bernoulli}(\psi_{\text{sex}})$
 7094 where the parameter ψ_{sex} is the sex ratio of the population. We assume our default non-
 7095 informative prior for this parameter: $\psi_{\text{sex}} \sim \text{Uniform}(0, 1)$. The model specification in
 7096 Panel 7.2 demonstrates how to incorporate a partially observed covariate (i.e., “sex”). It
 7097 is important to note that in the previous equation, sex_i is a vector with two categories
 7098 indicating the sex of each individual (e.g., 1 = male, 2 = female). This corresponds
 7099 directly to having a binary indicator of sex (e.g., $\text{Sex}_i = 1$ if individual i is female, and 0
 7100 otherwise). In the Bayesian formulation of the model, we use both the binary indicator
 7101 (`Sex`) and a categorical indicator (`Sex2 = Sex + 1`). The former (termed `Sex` in Panel
 7102 7.2) allows us to specify the Bernoulli distribution for the random variable, and the latter
 7103 (termed `Sex2`) allows us to use the dummy or indicator variable specification in the model.

7104 In both **JAGS** or **BUGS** missing data are indicated by `NA` in the data objects passed
 7105 to the program through the `bugs` or `jags` functions in **R**. To set up the data, we need to
 7106 create a vector of length M with the first n elements being 0 if individual i is a female, or
 7107 1 if i is a male (for the Fort Drum black bear data the function `bear.JAGS` extracts this
 7108 information automatically from the `beardata` object), and the subsequent $M - n$ elements
 7109 being `NA`. It is generally a good idea to provide starting values for the missing data, but we
 7110 cannot provide starting values for observed data; in this case where one vector (or other
 7111 object) contains both observed and missing data, initial values for the observed data have
 7112 to be specified as `NA`. The code snippet below shows you how to set up the data including
 7113 the `Sex` vector and the initial values function (the remainder of the code is identical to
 7114 what we've shown before).

```
7115 > sex <- beardata$sex #the sex data for captured individual
7116 > Sex <- c(sex-1, rep(NA, nz)) #sex enters as 1/2, this recodes it to 0/1
7117                                         #so we can use Bernoulli distribution
7118
7119 > data <- list(y=y,Sex=Sex, M=M,K=K, J=ntraps, xlim=xlim, ylim=ylim,area=areaX)
7120 > params <- c('psi','p0','N', 'D', 'sigma', 'psi.sex')
7121 > inits <- function() { list(z=c(rep(1,nind), rbinom(nz,1,0.5)),psi=runif(1),
```

```

7122      s=cbind(runif(M, xlim[1],xlim[2]), runif(M,ylim[1],ylim[2])),
7123      psi.sex=runif(1),Sex=c(rep(NA, nind), rbinom(nz,1,0.5)),
7124      sigma=runif(2,2,3),alpha0=runif(2)) }
```

7125 The **BUGS** model specification is shown in Panel ??.

```

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]
  alpha1[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(-alpha1[Sex2[i]]*d[i,j]*d[i,j])
  }
}
N <- sum(z[])
D <- N/area
}
```

Panel 7.2: **JAGS** model specification for an SCR model with sex-specific encounter probability parameters.

7126 Our estimate of density under the fully sex-specific model is still very similar to the
 7127 previous models (Table 7.6), and while the baseline detection was not very different be-
 7128 tween males and females, we can see that they had very different σ estimates (note that
 7129 the BCIs do not overlap). As usual, you can reproduce this analysis by calling the function
 7130 **bear.JAGS** and set **model='SCRsex'**.

Table 7.6. Posterior summaries of parameter estimates from sex-specific SCR models for the Fort Drum black bear data set.

Parameter	Mean	SD	2.5	97.5
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

7.3 INDIVIDUAL HETEROGENEITY

7131 Here we consider SCR models with individual heterogeneity. Capture-recapture models
 7132 with individual heterogeneity in detection probability, so-called model M_h , have a long
 7133 history in classical capture recapture models and they have special relevance to SCR (Sec.
 7134 4.4). While the advent of SCR models may appear to have rendered the use of classical
 7135 model M_h obsolete (because the heterogeneity is being accounted for explicitly) we may
 7136 still wish to consider heterogeneity models for other biological reasons. It is reasonable to
 7137 expect in real populations that there exists heterogeneity in home range size and so we
 7138 think that α_1 could exhibit heterogeneity among individuals. As we noted previously, it
 7139 may be advantageous or desirable in some cases to model heterogeneity directly in terms
 7140 of the scale parameter of the distance function σ or some other transformation of the
 7141 “distance coefficient”, perhaps even 95% home range area.

7142 In this section, we describe a class of spatial capture-recapture models to allow for
 7143 individual heterogeneity in encounter probability. In particular, one class of models we
 7144 propose explicitly admits individual heterogeneity in home range size. In addition, we con-
 7145 sider a standard representation for heterogeneity in which an additive individual-specific
 7146 random effect is included in the linear predictor for baseline encounter probability.

7147 7.3.1 Models of heterogeneity

7148 An obvious extension to the SCR model is to include an additive individual effect, analo-
 7149 gous to classical “model M_h ”. We’ll call this model “SCR+Mh”:

$$\text{logit}(p_{0,i}) = \alpha_0 + \eta_i$$

$$p_{ijk} = p_{0,i} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

7150 where η_i is an individual random effect having distribution $[\eta|\sigma_p]$. A popular class of
 7151 models arises by assuming $\eta_i \sim \text{Normal}(0, \sigma_p^2)$ (Coull and Agresti, 1999; Dorazio and
 7152 Royle, 2003). We show how to implement this specific SCR + Mh model in Panel 7.3,
 7153 although many other random effects distributions are possible. A popular one is the finite-
 7154 mixture of point masses (Norris and Pollock, 1996; Pledger, 2004) which we demonstrate
 7155 how to fit using `secr` in Sec. 7.4.3.

```

model{

alpha0 ~ dnorm(0,.1)                                # Prior distributions
alpha1 <- 1/(2*sigma*sigma)
sigma ~ dunif(0, 15)
psi ~ dunif(0,1)
tau_p ~ dgamma(.001,.001)

for(i in 1:M){
  eta[i] ~ dnorm(0, tau_p)                         # Individual level variables
  z[i] ~ dbern(psi)
  s[i,1] ~ dunif(xlim[1],xlim[2])
  s[i,2] ~ dunif(ylim[1],ylim[2])

  for(j in 1:J){                                    # The "likelihood" etc..
    d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
    y[i,j] ~ dbin(p[i,j],K)
    logit(p0[i,j]) <- alpha0 + eta[i]
    p[i,j] <- z[i]*p0[i,j]*exp(- alpha1*d[i,j]*d[i,j])
  }
}
N <- sum(z[])
D <- N/area                                         # N, D are derived
}

```

Panel 7.3: **JAGS** model specification for the SCR + Mh model with Gaussian encounter probability model and additive normal random effect.

7156 **7.3.2 Heterogeneity induced by variation in home range size**

7157 An alternative heterogeneity model, one that has more of a direct biological motivation and
 7158 interpretation, describes heterogeneity in home range size among individuals. To model
 7159 heterogeneity in home range area, we can assume a distribution for a transformation of
 7160 the scale parameter of the encounter probability model such as σ^2 , or $\log(\sigma^2)$, etc.. We
 7161 call this “model SCR + Ah” (Ah here for area-induced heterogeneity).

7162 Consider the following log-normal model for individual scale parameter of the Gaussian
 7163 encounter probability model, σ_i^2 :

$$\log(\sigma_i^2) \sim \text{Normal}(\mu_{hra}, \tau_{hra}^2)$$

7164 then the 95% home range area has a scaled log-normal distribution with mean

$$6\pi \exp(\mu_{hra} + \tau_{hra}^2/2).$$

7165 The variance is slightly more complicated, but you can look-up the variance of a log-normal
 7166 distribution and combine it with the 95% home range area calculation in Sec. 5.4 to work
 7167 out the implied variance of home range area under this model. We show two examples of
 7168 the implied *population* distribution of home range area under this log-normal model that
 7169 implies a mean home range area of about 6.9 area units (Figure 7.1). The left panel shows
 7170 a standard deviation in home range area of 2.88 units and the right panel shows a standard
 7171 deviation in home range area of 0.70 units. The two cases were generated by tweaking the
 7172 μ_{hra} and τ_{hra}^2 parameters of the log-normal distribution to achieve a constant expected
 7173 value of home range area, but modify the standard deviation.

7.4 LIKELIHOOD ANALYSIS IN SECR

7174 Previously, in Chapt. 6, we introduced the **R** package **secr** and described the likelihood
 7175 based inference approach taken by that package (see Sec. 6.5.3). Here we discuss how
 7176 to implement some standard covariate models in **secr** and provide an example of model
 7177 selection using AIC. As we saw in Chapt. 6, **secr** uses the standard **R** model specifi-
 7178 cation syntax, defining the dependent and independent variable relationship using tildes
 7179 (e.g., $y \sim x$). Thus, in **secr** we might have $g0 \sim \text{behavior}$ or $\text{sigma} \sim \text{time}$; when left
 7180 unspecified or set to 1 (e.g., $g0 \sim 1$), this will default to a model with no covariates (i.e.,
 7181 constant parameter values). A number of default model formulas for the baseline and
 7182 scale parameter of the encounter probability model are available in **secr**. Additionally,
 7183 **secr** allows us to specify covariates on density (we cover this in Chapt. 11), which are
 7184 set for example as $D \sim \text{habitat}$.

7185 To demonstrate models with various types of covariates using **secr**, we continue using
 7186 the Fort Drum black bear data. We include in the **scrbook** package a function called
 7187 **secr.bear** that will format the data (see Chapt. 6 for the **secrdata** format) and then
 7188 fit and compare 8 models (details shown in Panel 7.4). We have described all of these
 7189 models in the previous sections, so we only briefly comment here on how to fit certain
 7190 models in **secr** and compare them using AIC, and give a few helpful notes.

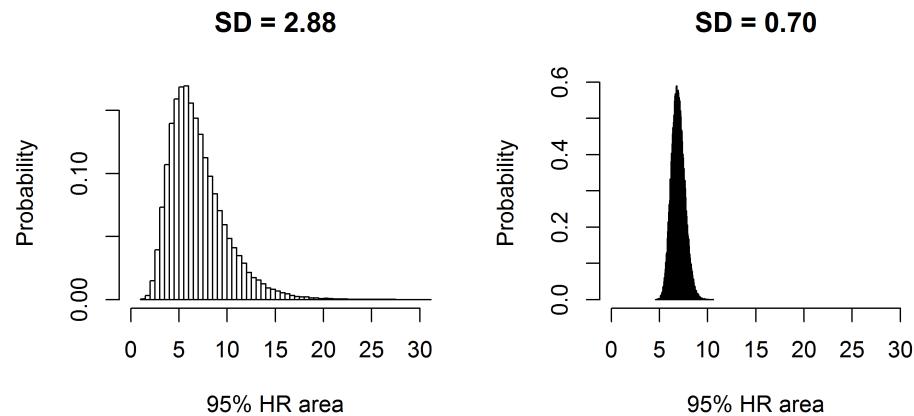


Figure 7.1. Population distribution of home range area for a model in which $\log(\sigma^2)$ has a normal distribution with mean μ_{hra} and variance τ_{hra}^2 . The parameters were chosen to yield a constant expected value of about 6.9 units of area, but to produce two different levels of heterogeneity: A population standard deviation of 2.88 units (left panel) and 0.70 units (right panel).

7.4.1 Notes for fitting standard models

In the **secr** package, the encounter probability model is called the “detection function” and it is specified by changing the “`detectfn`” option (an integer code) within the `secr.fit` command. Table 7.1 shows the possible encounter probability models that **secr** allows; the default is that based on the kernel of a bivariate normal probability distribution function (hence we call this the Gaussian model, but it is referred to as “half-normal” in **secr**) and the (negative) exponential is `detectfn = 2`. See model 2 in Panel 7.4 for how to fit the exponential model to the Fort Drum bear data set.

The **secr** package easily fits a range of SCR equivalents of standard capture-recapture models. The package has pre-defined versions of the classic model M_t where each occasion has its own encounter probability, as well as a linear trend in baseline encounter probability over occasions (in a spatial modeling framework 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 7.4); whereas, for a linear trend over occasions **secr** uses ‘T’.

The global trap response model (what we called model M_B), or a local trap-specific behavioral response (model M_b) can be fitted in **secr** using formulae with “b” for the global response model and “bk” for the local trap response model (see models 4 and 5 in Panel 7.4; note that to fit the trap specific behavioral response model you need version

7211 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 7.4: Models called from `secr.bear` function. All models use `buffer = 20000`

7212 **7.4.2 Sex effects**

7213 Incorporating sex effects into models with `secr` can be done a few different ways, but
 7214 there are not pre-defined models for this. A limitation of fitting models with sex effects
 7215 in `secr` is that it does not accommodate missing values of the sex variable. Thus, in all
 7216 cases, individuals that are of unknown sex must be removed from the data set (recall that
 7217 in a Bayesian framework we can keep these individuals in the data set by specifying a
 7218 distribution for the individual covariate “sex”). In `secr`, the easiest way to include sex
 7219 effects is to code sex as a “session” variable using the multi-session models (see Sec. 6.5.4
 7220 for a description of the multi-session models), providing two sessions, one representing
 7221 males and one for females (see model 7 in Panel 7.4). This method provides two separate
 7222 density estimates, which can then be combined into a total density.

7.4.3 Individual heterogeneity

7224 To incorporate heterogeneity, **secr** fits a set of finite mixture models (Norris and Pollock,
 7225 1996; Pledger, 2004). These are expensive in terms of parameters but they have been
 7226 widely adopted because they are easy to analyze using likelihood methods, as the marginal
 7227 distribution of the data is just a sum of a small number of components. Using **secr**,
 7228 individual heterogeneity can be incorporated into the encounter probability model using
 7229 default models for either a 2- or 3-component finite mixture model using the “h2” or “h3”
 7230 model terms. The 2-part mixture is shown in model 8 of panel 7.4 and the 3-part mixture
 7231 can easily be fit by substituting h3 for h2. The finite-mixture model can be fit in **JAGS** or
 7232 **BUGS**, but we only showed the SCR + Mh logit-normal mixture in the version above
 7233 (see Sec. 7.3.1).

7.4.4 Model selection in secr using AIC

7234 One practical advantage to using the **secr** package, or likelihood inference in general, is
 7235 the convenience of automatic model selection using AIC (Burnham and Anderson, 2002).
 7236 The **secr** package has a number of convenient functions for computing AIC and producing
 7237 model selection tables, or doing model-averaging (as described in Chapt. 8). Running the
 7238 function **secr.bear**, which calls all of the models we have described, will return, in addition
 7239 to all model results, an AIC table with all of the summarized results including the AIC
 7240 values, delta AIC, and model weights (see Table 7.7 or reproduce results in R using **out<-**
 7241 **secr.bear()**; **out\$AIC.tab**).

7242 It is important to note that AIC is not comparable between a multi-session model and
 7243 a model that is not a multi-session model. Therefore, to compare the sex-specific model
 7244 (which uses “sessions”) with all the other models including the null, time, and behavioral
 7245 models, we coded the data set as a multi-session design when first loading it to **secr**. This
 7246 results in all the model outputs listing separate parameter estimates for each session, even
 7247 the null model with no covariates; however, the estimates are the same for both “sessions”
 7248 in all but the sex-specific model.

Table 7.7. Log-likelihood, AIC, deltaAIC and AIC weight for several models run in **secr** for
 the Fort Drum black bear data set.

model	logLik	AIC	AICc	dAICc	AICwt
bear.b	-641.7215	1291.443	1292.395	0.000	1
bear.h2	-653.8382	1319.676	1321.776	29.381	0
bear.0exp	-663.9152	1333.830	1334.389	41.994	0
bear.B	-677.6175	1363.235	1364.187	71.792	0
bear.bt	-668.3044	1358.609	1366.152	73.757	0
bear.sex	-677.7151	1367.430	1369.530	77.135	0
bear.t	-674.4134	1368.827	1374.938	82.543	0
bear.0	-686.2455	1378.491	1379.049	86.654	0

7250 The results from this AIC analysis are straightforward to interpret; the model with
 7251 a local trap response of encounter probability, “bk”, has a model weight of 1 and thus,
 7252 according to AIC, 100% support. The 2-part finite mixture model for g_0 and σ has the

7253 second lowest AIC, but considering the large dAICc compared to the local trap response
 7254 model we would probably not consider it any further.

7.5 SUMMARY AND OUTLOOK

7255 There are endless covariates and encounter probability models that can be defined and our
 7256 goal in this chapter was to introduce basic types of covariate models and demonstrate how
 7257 to implement them in **BUGS** and **secr**. Essentially, SCR's are GLMMs and therefore
 7258 we develop covariate models in much the same way, using a suitable transformation (link
 7259 function) of the parameter(s). In SCR models, we typically have 2 parameters of the
 7260 encounter probability model for which we might specify covariate models – the **baseline**
 7261 encounter probability (or rate) parameter, and a scale parameter that is related in many
 7262 cases to the home range size of the species. A few examples of different covariate models
 7263 are given in Table 7.3. We can also consider covariates by their classification as fixed,
 7264 partially observed, or unobserved (see Table 7.8). This classification of covariate types
 7265 can be important because the MLE and Bayesian approaches to dealing with partially
 7266 and unobserved covariates is often different. This was seen above in how the covariate **Sex**
 7267 was handled in the two frameworks.

Table 7.8. Examples of different covariate classifications.

Covariate class	Examples
Fixed	baited, weather, habitat
Partially observed	sex, age,
Unobserved	home range size, ind. effects

7268 While the move to spatially explicit models in capture-recapture studies has largely
 7269 rendered the basic CR models (Otis et al., 1978) obsolete, we continue to find this clas-
 7270 sification useful for categorizing the *spatial* extensions of these standard CR models. The
 7271 extended models include the standard M_0 , M_t , M_b , and M_h , but also new models that
 7272 allow for trap-specific information such as "baited/not-baited" or "on/off road". In addi-
 7273 tion, in Chaps. 12, 13 and 11, we explore additional models for explaining variation in
 7274 encounter probability and density based on spatial covariates that describe variation in
 7275 landscape or habitat conditions.

7276
7277

8

MODEL SELECTION AND ASSESSMENT

7279 Our purpose in life is to analyze models. By that, we mean one or more of the following
7280 basic 4 tasks: (1) estimate parameters, (2) make predictions of unobserved random vari-
7281 ables, (3) evaluate the relative merits of different models or choosing a best model (model
7282 selection), and (4) checking whether a specific model appears to provide a reasonable de-
7283 scription of the data or not (model checking, assessment, or “goodness-of-fit”). In previous
7284 chapters we addressed the problems of estimation of model parameters, and also making
7285 predictions of latent variables, s or z , or functions of these variables such as density or
7286 population size. In this chapter, we focus on the last two of these basic inference tasks:
7287 model selection (which model or models should be favored), and model assessment (do
7288 the data appear to be consistent with a particular model).

7289 In this chapter we review basic strategies of model selection using both likelihood
7290 methods (as implemented in the `secr` package) and Bayesian analysis. Specifically, we
7291 review a number of standard methods of model selection that apply to “variable selection”
7292 problems, when our set of models consists of distinct covariate effects and they represent
7293 constraints of some larger model. For classical analysis based on likelihood, model selection
7294 by Akaike Information Criterion (AIC) is the standard approach (Burnham and Anderson,
7295 2002). For Bayesian analysis we rely on a number of different methods. We demonstrate
7296 the use of the deviance information criterion (DIC) (Spiegelhalter et al., 2002) for variable
7297 selection problems although it has deficiencies when applied to hierarchical models in some
7298 cases (Millar, 2009). We use the Kuo and Mallick indicator variable selection approach
7299 (Kuo and Mallick, 1998) which produces direct statements of posterior model probabilities
7300 which we think are the most useful, and leads directly to model-averaged estimates of
7301 density. There is a good review paper recently by O’Hara and Sillanpää (2009) that
7302 discusses these and many other related ideas for variable selection. In addition to O’Hara
7303 and Sillanpää (2009) we also recommend Link and Barker (2010, Chapt. 7) for general
7304 information on model selection and assessment.

7305 To check model adequacy in a Bayesian framework, or whether a specific model pro-
7306 vides a satisfactory description of our data set, we rely exclusively on the Bayesian p-value
7307 framework (Gelman et al., 1996). For assessing fit of SCR models, part of the challenge

7308 is coming up with good measures of model fit, and there does not appear much definitive
 7309 guidance in the literature on this point. Following Royle et al. (2011a), we break the prob-
 7310 lem up into 2 components which we attack separately: (1) Conditional on the underlying
 7311 point process, does the encounter model fit? (2) Do the uniformity and independence
 7312 assumptions appear adequate for the point process model of activity centers? The latter
 7313 component of model fit has a considerable precedence in the ecological literature as it
 7314 is analogous to the classical problem of testing “complete spatial randomness” (Cressie,
 7315 1991; Illian et al., 2008).

7316 We apply some of these methods to the wolverine camera trapping data first introduced
 7317 in Chapt. 5 to investigate sex specificity of model parameters and whether there is a
 7318 behavioral response to encounter. We note that individuals are drawn to the camera
 7319 trap devices by bait and therefore it stands to reason that once an individual discovers a
 7320 trap, it might be more likely to return subsequently, a response termed “trap happiness”.
 7321 We evaluate whether certain models for encounter probability appear to be adequate
 7322 descriptions of the data, and we evaluate the uniformity assumption for the underlying
 7323 point process.

8.1 MODEL SELECTION BY AIC

7324 Using classical analysis based on likelihood, model selection is easily accomplished using
 7325 AIC (Burnham and Anderson, 2002) which we demonstrate below. The AIC of a model is
 7326 simply twice the negative log-likelihood evaluated at the MLE, penalized by the number
 7327 of parameters (np) in the model:

$$\text{AIC} = -2\log L(\hat{\theta}|\mathbf{y}) + 2np$$

7328 Models with small values of AIC are preferred. It is common to use a modified (“cor-
 7329 rected”) AIC referred to as AIC_c for small sample sizes which is

$$\text{AIC}_c = -2\log L(\hat{\theta}|\mathbf{y}) + \frac{2np(np+1)}{n-np-1}$$

7330 where n is the sample size. Two important problems with the use of AIC and AIC_c are
 7331 that they don’t apply directly to hierarchical models that contain random effects, unless
 7332 they are computed directly from the marginal likelihood (for SCR models we can do this,
 7333 see Chapt. 6). Moreover, it is not clear what should be the effective sample size n in
 7334 calculation of AIC_c , as there can be covariates that affect individuals, that vary over
 7335 time, or space. We do not offer strict guidelines as to when to use a small sample size
 7336 adjustment.

7337 The R package **secr** computes and outputs AIC automatically for each model fitted
 7338 and it provides some capabilities for producing a model selection table (function **AIC**) and
 7339 also doing model-averaging (function **model.average**), which we recommend for obtaining
 7340 estimates of density from multiple models.

8.1.1 AIC analysis of the wolverine data

7342 We provide an example of model selection for the wolverine camera trapping data using
 7343 **secr**. We consider a model set with distinct models to accommodate various types of sex
 7344 specificity of model parameters:

7345 Model 0: model SCR0 with constant density and constant encounter model parameters;
 7346 Model 1: model SCR0 with constant parameter values for both male and female wolverines but with sex-specific density only;
 7347 Model 2: Sex-specific density, sex-specific p_0 but constant σ ;
 7348 Model 3: Sex-specific density, sex-specific σ but constant p_0 ;
 7349 Model 4: Sex-specific density, sex-specific p_0 and sex-specific σ .

7351 To model sex-specific abundance (density), we use the multi-session models provided
 7352 by **secr** (introduced in Sec. 6.5.4), which allow one to model session-specific effects on
 7353 density, baseline encounter probability, p_0 (labeled g_0 in **secr**), and also the scale parameter
 7354 σ of the encounter probability model. Using this formulation, we define the “Session”
 7355 variable to be a *categorical* sex code having value 1 or 2 (demonstrated below) and thus
 7356 *session*-specific parameters represent *sex*-specific parameters. For example, if we model
 7357 session-specific density, D , then this corresponds to Model 1 in our list above. We note
 7358 that “Model 0” in our list corresponds to a model where all of the encounter histories
 7359 have the same session ID. This model is one of constant density, which implies that the
 7360 population sex ratio is fixed at 0.5, i.e., $\psi_{\text{sex}} = 0.5$.

7361 Although **secr** also uses the logit/log linear predictors as the default for modeling
 7362 covariates on baseline encounter probability and the scale parameter, respectively, **secr**
 7363 does something different with the multi-session models. It reports estimates in a *session*
 7364 *mean* parameterization (equivalent to, in **BUGS**, using an index variable instead of a set
 7365 of dummy variables), and not the *session effect* (i.e., deviation from the intercept) which
 7366 arises from the use of dummy variables. We show this **BUGS** model description in Sec.
 7367 8.2.2.

7368 To fit these models using **secr**, we load the wolverine data and do a slight bit of
 7369 formatting to prepare the data objects for analysis by **secr**. The key difference from our
 7370 analysis in Chapt. 6 is, here, we use the wolverine sex information (**wolverine\$wsex**)
 7371 which is a binary 0/1 variable (1=male) and we add 1 so that we can define a categorical
 7372 “Session” variable (having values 1 or 2). We also have a function **scr2secr** which converts
 7373 a standard trap-deployment file (TDF) matrix into a **secr** object of class “traps.” The
 7374 R commands are as follows (contained in the help file **?secr_wolverine**):
 7375

```

7375
7376 > library(secr)
7377 > library(scrbook)
7378 > data(wolverine)
7379 > traps <- as.matrix(wolverine$wtraps)

7380 ## Name variables as required by secr
7381 > dimnames(traps) <- list(NULL,c("trapID","x","y",paste("day",1:165,sep="")))
7382 ## Convert trap information to a secr "traps" object
7383 > trapfile <- scr2secr(scrtraps=traps,type="proximity")

7385 ## Grab the wolverine state-space grid (2km here)
7386 > gr <- as.matrix(wolverine$grid2)
7387 > dimnames(gr) <- list(NULL,c("x","y"))
7388 > gr2 <- read.mask(data=gr)
  
```

```

7390
7391 ## Grab the encounter data, and re-name variables
7392 > wolv.dat <- wolverine$wcaps
7393 > dimnames(wolv.dat) <- list(NULL,c("Session","ID","Occasion","trapID"))
7394
7395 ## Convert binary 0/1 sex variable to categorical 1/2 for "session"
7396 > wolv.dat[,1] <- wolverine$wsex[wolv.dat[,2]]+1
7397 > wolv.dat <- as.data.frame(wolv.dat)
7398
7399 ## Convert to capthist object
7400 > wolvcapt <- make.capthist(wolv.dat,trapfile,fmt="trapID",noccasions=165)

```

7401 Once the data have been prepared in this way, we use the `secr` model fitting function
 7402 `secr.fit` to fit the different models, and then the function `AIC` to package the models
 7403 together and summarize them in the form of an AIC table, with rows of the table ordered
 7404 from best to worst. The function `model.average` performs AIC-based model-averaging of
 7405 the parameters specified by the `realnames` variable (below this is demonstrated for the
 7406 parameter density, D). Because this function defaults to averaging by AIC_c , we slightly
 7407 modified this function (called `model.average2`) to do model averaging by either AIC or
 7408 AIC_c as specified by the user. The model fitting commands look like this (for Model 0
 7409 and Model 1):

```

7410 > model0 <- secr.fit(wolvcapt, model=list(D~1, g0~1, sigma~1),
7411                 buffer=20000)
7412 > model1 <- secr.fit(wolvcapt, model=list(D~session, g0~1, sigma~1),
7413                 buffer=20000)

```

7414 Next we use the function `AIC`, passing the fit objects from all 5 models, and that
 7415 produces the following output (abbreviated horizontally to fit on the page):

```

7416 > AIC (model0,model1,model2,model3,model4)
7417           model      ... npar logLik   AIC    AICc dAICc  AICwt
7418 model0  D~1 g0~1 sigma~1  ...  3 -627.2603 1260.521 1261.932 0.000 0.5831
7419 model2      ..      ...  5 -624.9051 1259.810 1263.810 1.878 0.2280
7420 model1      ..      ...  4 -627.2365 1262.473 1264.973 3.041 0.1275
7421 model4      ..      ...  6 -624.6632 1261.326 1267.326 5.394 0.0393
7422 model3      ..      ...  5 -627.2358 1264.472 1268.472 6.540 0.0222

```

7423 Model averaging the results is done as follows:

```

7424 > model.average (model0,model1,model2,model3,model4,realnames="D")
7425           estimate  SE.estimate      lcl      ucl
7426 session=1 2.707190e-05 7.913577e-06 1.544474e-05 4.745224e-05
7427 session=2 2.927423e-05 8.270402e-06 1.700631e-05 5.039193e-05

```

7428 As usual, estimates and standard errors of the individual model parameters can be
 7429 obtained from the `secr.fit` summary output of any of the `modelX` objects shown above.
 7430 The default output of estimated density is in individuals per ha, so we have to scale this
 7431 up to something more reasonable. To get into units of per 1000 km², we need to first

7432 multiply by 100 to get to units of km^2 and then multiply by 1000. This produces an
 7433 estimated density of about 2.71 for `session=1` (females) and 2.93 for `session=2` (males).
 7434 We can use the generic **R** function `predict` applied to the `secr.fit` output to obtain
 7435 specific information about the MLEs on the natural scale.

7436 We don't necessarily agree with the use of AIC_c here and think its better to use AIC,
 7437 in general. This is because, as noted previously, it is not clear what the effective sample
 7438 size is for most capture-recapture problems. While we have 21 individuals in the data
 7440 set, most of the model structure has to do with encounter probability samples and for
 7441 that there are hundreds of observations. We do note that the AIC and AIC_c results are
 7442 not entirely consistent. By looking at the best model by AIC (Table 8.1), we find that
 7443 the model with sex specific density and sex-specific baseline encounter probability, p_0 , is
 7444 preferred (Model 2). This is just slightly better than the null model (Model 0) with no
 sex effects at all and hence an implied fixed sex ratio of $\psi_{\text{sex}} = 0.50$.

Table 8.1. Model selection results for the wolverine models of sex specificity, with/without habitat mask. Fitting was done using `secr` with a half-normal (Gaussian) encounter probability model. Models are ordered by *AIC*. Density, *D*, is reported in units of individuals per 1000 km^2 . Model abbreviations indicate which parameters are sex-specific in order $D/p_0/\sigma$.

NO HABITAT MASK										
model	npar	Female			Male			D	p_0	σ
		AIC	AIC_c	D	p_0	σ				
2: sex/sex/1	5	1259.8	1263.8	2.45	0.08	6435.51	3.16	0.04	6435.51	
0: 1/1/1	3	1260.5	1261.9	2.83	0.06	6298.66	2.83	0.06	6298.66	
4: sex/sex/sex	6	1261.3	1267.3	2.59	0.08	6080.70	2.99	0.04	6833.16	
1: sex/1/1	4	1262.5	1265.0	2.69	0.06	6298.69	2.96	0.06	6298.69	
3: sex/1/sex	5	1264.5	1268.5	2.70	0.06	6280.49	2.95	0.06	6319.03	
WITH HABITAT MASK										
model	npar	Female			Male			D	p_0	σ
		AIC	AIC_c	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	

7445 We fit the same models but now using a modified state-space which excludes the ocean
 7446 (this is a habitat mask in `secr`). Results are shown in Table 8.1 along with the previous
 7447 models without a mask. We see AIC values are smaller for the model without the mask.
 7448 It is probably acceptable to compare these different fits (with and without habitat mask)
 7449 by AIC because we recognize the mask as having the effect of modifying the random
 7450 effects distribution (i.e., of the activity centers, *s*) and the results should be sensitive to
 7451 choice of the distribution for *s*. That said, we tend to prefer the mask model because it
 7452 makes sense to exclude the areas of open water from the state-space of *s*. For females the
 7453 model-averaged density is 3.88 individuals per 1000 km^2 and for males the model-averaged
 7454 density estimate is 4.46 individuals per 1000 km^2 as we see here:

```
7455 > model.average (model0b,model1b,model2b,model3b,model4b,realnames="D")
```

```

7456
7457      estimate SE.estimate      lcl      ucl
7458 session=1 3.876615e-05 1.189102e-05 2.153795e-05 6.977518e-05
7459 session=2 4.459658e-05 1.323696e-05 2.523280e-05 7.882022e-05

```

7460 This is quite a bit higher than that based on the rectangular state-space (i.e., not
 7461 specifying a habitat mask). This is not surprising given that **the state-space is part**
 7462 **of the model** and the specific state-space modification we made here, which reduces the
 7463 area from the rectangular state-space, should be extremely important from a biological
 7464 standpoint (i.e., wolverines are not actively using open ocean).

8.2 BAYESIAN MODEL SELECTION

7465 Model selection is somewhat less straightforward as a Bayesian, and there is no canned
 7466 all-purpose method like AIC. As such we recommend a pragmatic approach, in general,
 7467 for all problems, based on a number of basic considerations:

- 7468 (1) For a small number of fixed effects we think it is reasonable to adopt a conventional
 7469 “hypothesis testing” approach – i.e., if the posterior for a parameter overlaps zero
 7470 substantially, then it is probably reasonable to discard that effect from the model.
- 7471 (2) Calculation of posterior model probabilities: In some cases we can implement methods
 7472 which allow calculation of posterior model probabilities. One such idea is the indicator
 7473 variable selection method from Kuo and Mallick (1998). For this, we introduce a latent
 7474 variable $w \sim \text{Bern}(.5)$ and expand the model to include the variable w as follows:

$$\text{logit}(p_{ijk}) = \alpha_0 + w * \alpha_1 * C_{ijk}.$$

7475 The importance of the covariate C is then measured by the posterior probability that
 7476 $w = 1$.

7477 (3) The Deviance Information Criterion (DIC): Bayesian model selection is now routinely
 7478 carried out using DIC ((Spiegelhalter et al., 2002)), although its effectiveness in hier-
 7479 archical models depends very much on the manner in which it is constructed (Millar,
 7480 2009). We recommend using it if it leads to sensible results, but we think it should be
 7481 calibrated to the extent possible for specific classes of models. This has not yet been
 7482 done in the literature for SCR models, to our knowledge.

7483 (4) Logical argument: For something like sex specificity of certain parameters, it seems
 7484 to make sense to leave an extra parameter in the model no matter what because, bio-
 7485 logically, we might expect a difference (e.g., home range size). In some cases failure to
 7486 apply logical argument leads to meaningless tests of gratuitous hypotheses (Johnson,
 7487 1999).

7488 In all modeling activities, as in life itself, the use of logical argument should not be under-
 7489 utilized.

8.2.1 Model selection by DIC

7490 The availability of AIC makes the use of likelihood methods convenient for problems where
 7491 likelihood estimation is achievable. For Bayesian analysis, DIC seemed like a general-
 7492 purpose equivalent, at least for a brief period of time after its invention. However, there

7494 seem to be many variations of DIC, and a consistent version is not always reported across
 7495 computing platforms. Even statisticians don't have general agreement on practical issues
 7496 related to the use of DIC (Millar, 2009). Despite this, it is still widely reported. We think
 7497 DIC is probably reasonable for certain classes of models that contain only fixed effects,
 7498 or for which the latent variable structure is the same across models so that only the fixed
 7499 effects are varied (this covers many SCR model selection problems). However, it would be
 7500 useful to see some calibration of DIC for some standardized model selection problems.

7501 Model deviance is defined as negative twice the log-likelihood; i.e., for a given model
 7502 with parameters θ : $\text{Dev}(\theta) = -2 * \log L(\theta|\mathbf{y})$. The DIC is defined as the posterior mean
 7503 of the deviance, $\overline{\text{Dev}}(\theta)$, plus a measure of model complexity, p_D :

$$\text{DIC} = \overline{\text{Dev}}(\theta) + p_D$$

7504 The standard definition of p_D is

$$p_D = \overline{\text{Dev}}(\theta) - \text{Dev}(\bar{\theta})$$

7505 where the 2nd term is the deviance evaluated at the posterior mean of the model parameter(s), $\bar{\theta}$. The p_D here is interpreted as the effective number of parameters in the model.
 7506 Gelman et al. (2004) suggest a different version of p_D based on one-half the posterior
 7507 variance of the deviance:

$$p_V = \text{Var}(\text{Dev}(\theta)|\mathbf{y})/2.$$

7509 This is what is produced from **WinBUGS** and **JAGS** if they are run from **R2WinBUGS** or
 7510 **R2jags**, respectively. It is less easy to get DIC summaries from **rjags**, so we used **R2jags**
 7511 in our analyses below.

7512 8.2.2 DIC analysis of the wolverine data

7513 We repeated the analysis of the wolverine models with sex specificity, but this time doing
 7514 a Bayesian analysis paralleling the likelihood analysis we did above in **secr**, using the
 7515 logit/log parameterization of the model parameters. To do so in **BUGS**, we used dummy
 7516 variables. Thus, we can express models allowing for sex specificity using a dummy variable
 7517 **Sex** and new parameters (α_{sex} , β_{sex}) which represent the effect of **Sex** at level 1:

$$\text{logit}(p_{0,i}) = \alpha_0 + \alpha_{sex} \mathbf{Sex}_i$$

7518 and

$$\log(\sigma_i) = \log(\sigma_0) + \beta_{sex} \mathbf{Sex}_i.$$

7519 In these expressions, the sex variable \mathbf{Sex}_i is a binary variable where $\mathbf{Sex}_i = 0$ corresponds to female, and $\mathbf{Sex}_i = 1$ corresponds to male.

7521 Unlike the multi-session model in **secr**, we carry out the analysis of the sex-specific
 7522 model here by putting all of the data into a single data set, and explicitly accounting for
 7523 the covariate 'sex' in the model by assigning it a Bernoulli prior distribution with ψ_{sex}
 7524 being the proportion of males in the population. In this case, we produce "Model 0" above,
 7525 the model with no sex effect on density, by setting the population proportion of males at
 7526 one-half: $\psi_{sex} = 0.5$ (see also Sec. 7.2.4). As usual, handling of missing values of the
 7527 sex variable is done seamlessly which might be a practical advantage of Bayesian analysis

in situations where sex is difficult to record in the field which may lead to individuals of unknown sex (i.e., missing values).

The **BUGS** model specification for the most complex model, Model 4, is shown in Panel 8.1. This model has sex-specific intercept, scale parameter, σ , and density. We provide an **R** script named `wolvSCR0ms` in the `scrbook` package which will fit each model. The function uses **JAGS** by default for the fitting, using the `R2jags` package. The kernel of this function is the model specification in Panel 8.1, which gets modified depending on the model we wish to fit using a command line option `model`. For example, `model = 1` fits the model with constant parameter values for males and females, but sex-specific population sizes (`model = 0` constrains the male probability parameter, ψ_{sex} , to be 0.5). The **R** function fits each of the 5 models using a binary indicator variable to turn ‘on’ or ‘off’ each effect. Here is how we obtain the MCMC output for each of the 5 models:

```
7540 > wolv0 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=0)
7541 > wolv1 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=1)
7542 > wolv2 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=2)
7543 > wolv3 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=3)
7544 > wolv4 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=4)
```

We fitted the 5 models to the wolverine data and summarize the DIC computation results in Table 8.2. The model rank has model 0, model 2, model 1, model 4, model 3. Interestingly, this is the same order as the models based on AIC_c which we found above (see Table 8.1). The posterior mean and SD of model parameters under the 5 models are given in Table 8.3.

Table 8.2. DIC results for the 5 models of sex specificity fitted to the wolverine camera trapping data, using the function `wolvSCR0ms`. Results are based on 3 chains of length 61000 yielding 180000 posterior samples.

	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

7550 8.2.3 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 prescribed 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

```

alpha.sex ~ dunif(-3,3)                      ## Prior distributions
beta.sex ~ dunif(-3,3)
sigma0 ~ dunif(0,50)
alpha0 ~ dnorm(0,.1)
psi ~ dunif(0,1)                             ## Data augmentation parameter
psi.sex ~ dunif(0,1)                          ## Probability of 'male'

for(i in 1:M){                                ## DA loop
  wsex[i] ~ dbern(psi.sex)                   ## Latent sex state (male = 1)
  z[i] ~ dbern(psi)                         ## DA variables
  s[i,1] ~ dunif(Xl,Xu)
  s[i,2] ~ dunif(Yl,Yu)
  logit(p0[i]) <- alpha0 + alpha.sex*wsex[i]
  log(sigma.vec[i]) <- log(sigma0) + beta.sex*wsex[i]
  alpha1[i] <- 1/(2*sigma.vec[i]*sigma.vec[i])
  for(j in 1:ntraps){
    mu[i,j] <- z[i]*p[i,j]
    y[i,j] ~ dbin(mu[i,j],K[j])
    dd[i,j] <- pow(s[i,1] - traplocs[j,1],2) + pow(s[i,2] - traplocs[j,2],2)
    p[i,j] <- p0[i]*exp(-alpha1[i]*dd[i,j])
  }
}

```

Panel 8.1: Part of the **BUGS** specification for a complete sex specificity of model parameters. This is a simplified version of the model contained in the **wolvSCR0ms** script, because it does not contain the on/off switches for creating the various sub-models.

Table 8.3. Posterior summaries of model parameters for models with varying sex specificity of model parameters. Model 0 = no sex specificity, model 4 = fully sex-specific (see text). Models are based on the Gaussian encounter probability model, each with 21000 iterations, 1000 burn-in, 3 chains for a total of 60000 posterior samples.

Parameter	model 0		model 1		model 2		model 3		model 4	
	Mean	SD								
N	60.02	11.91	60.24	11.93	59.37	11.97	59.67	11.97	58.77	11.75
D	5.79	1.15	5.81	1.15	5.72	1.15	5.75	1.15	5.66	1.13
α_0	-2.81	0.18	-2.82	0.17	-2.44	0.25	-2.82	0.18	-2.43	0.25
α_{sex}	0.00	1.73	0.00	1.73	-0.75	0.34	0.00	1.73	-0.79	0.36
σ_0	0.64	0.06	0.64	0.05	0.66	0.06	0.65	0.08	0.63	0.09
β_{sex}	0.00	1.73	-0.01	1.73	0.01	1.74	-0.01	0.17	0.10	0.18
ψ	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	

the different models because they are meaningless (averaging apples and oranges....). For example, if a regression parameter is in a specific model then the posterior is informed by the data and a specific MCMC draw is from the appropriate posterior distribution. On the other hand, if the regression parameter is not in the model then the MCMC draw is obtained directly from the prior distribution, and so we need to think carefully about whether it makes sense to report an average of such a thing (in the vast majority of cases the answer is no). But some parameters like N or density, D , do have a consistent interpretation and we support producing model-averaged results of those parameters.

To implement the Kuo and Mallick approach, we expand the model to include the latent indicator variables, say w_m , for variable m in the model, such that

$$w_m = \begin{cases} 1 & \text{linear predictor includes covariate } m \\ 0 & \text{linear predictor does not include covariate } m \end{cases}$$

We assume that the indicator variables w_m are mutually independent with

$$w_m \sim \text{Bernoulli}(0.5)$$

for each variable $m = 1, 2, \dots$, in the model. For example, with 2 variables, the expanded model has the linear predictor:

$$\text{logit}(p_{ijk}) = \alpha_0 + \alpha_1 w_1 C_{1,i} + \alpha_2 w_2 C_{2,ijk}$$

where, let's suppose, $C_{1,i}$ is an individual covariate such as sex, and $C_{2,ijk}$ is a behavioral response covariate which is individual-, trap-, and occasion-specific. We can assume a parallel model specification on the parameter σ which is liable to vary by individual level covariates such as sex:

$$\log(\sigma_i) = \beta_0 + \beta_1 w_3 C_{1,i}.$$

Using this indicator variable formulation of the model selection problem we can characterize unique models by the sequence of w variables. In this case, each unique sequence (w_1, w_2, w_3) represents a model, and we can tabulate the posterior frequencies of each model by post-processing the MCMC histories of (w_1, w_2, w_3) , as we demonstrate shortly. This method then evaluates all possible combinations of covariates or 2^m models.

Conceptually, analysis of this expanded model within the data augmentation framework does not pose any additional difficulty. One broader, technical consideration is that posterior model probabilities are well known to be sensitive to priors on parameters (Aitkin, 1991; Link and Barker, 2006). See also Royle and Dorazio (2008, Sec. 3.4.3) and Link and Barker (2010, Sec. 7.2.5). What might normally be viewed as vague or non-informative priors, are not usually innocuous or uninformative when evaluating posterior model probabilities. The use of AIC seems to avoid this problem largely by imposing a specific and perhaps undesirable prior that is a function of the sample size (Kadane and Lazar, 2004). One solution is to compute posterior model probabilities under a model in which the prior for parameters is fixed at the posterior distribution under the full model (Aitkin, 1991). At a minimum, one should evaluate the sensitivity of posterior model probabilities to different prior specifications.

Analysis of the wolverine data

The **R** script `wolvSCR0ms` in the package `scrbook` provides the model indicator variable implementation for the fully sex-specific SCR model. It is run by setting `model=5` in the function call. We note again that it is not very useful to report most parameter estimates from this model because their marginal posterior is a mixture from the prior (when a value of the indicator variable of 0 is sampled) and draws informed by the data (i.e., from the posterior, when a 1 is drawn for the indicator variable w). On the other hand, the parameters N and density D should be reported and they represent marginal posteriors over all models in the model set. In effect, model averaging is done as part of the MCMC sampling. The variable ‘mod’ contains the two binary indicator variables (w above) which pre-multiply the ‘sex’ term in each of the p_0 and 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:

```

7610 > mod <- wolv5$BUGSoutput$sims.list$mod
7611 > mod <- paste(mod[,1],mod[,2],mod[,3],sep="")
7612 >
7613 > table(mod)
7614 mod
7615   000   001   010   011   100   101   110   111
7616 17181 4935 1057 296 25211 8337 2275 708
7617
7618 > round( table(mod)/length(mod) , 3)
7619 mod
7620   000   001   010   011   100   101   110   111
7621 0.286 0.082 0.018 0.005 0.420 0.139 0.038 0.012

```

7622 This results in a comparison of all 8 possible models (based on $m = 3$ covariates) instead
 7623 of just the 5 models we originally proposed. We see that the best model is that labeled
 7624 100 which, according to our construction above, has `mod[1]=1, mod[2]=0` and `mod[3]=0`.
 7625 This is the model having sex-specific baseline encounter probability p_0 , and $\psi_{sex} = 0.5$.
 7626 This model has posterior model probability 0.420. The model with no sex specificity at
 7627 all (the model with label 000) has posterior probability 0.286 and the remaining posterior
 7628 mass is distributed over the other six models. We could arrive at a qualitatively similar
 7629 conclusion using a more ad hoc approach based on looking at the posterior mass for each
 7630 parameter under the full model (model 4; see Table 8.3, in part). Considering the sex-
 7631 specific intercept, it appears to be very important as its posterior mass is mostly away
 7632 from 0. On the other hand, the coefficient on log-sigma is concentrated around 0, and
 7633 the estimated ψ_{sex} (probability that an individual is a male) is 0.54 with a large posterior
 7634 standard deviation. We might therefore be inclined to discard the sex effect on $\log(\sigma)$
 7635 based on classical thinking-like-a-hypothesis-testing-person and settle for the model with
 7636 a sex-specific intercept in the encounter probability model. This is consistent with our
 7637 indicator variable approach which found that model (1,0,0) has posterior probability of
 7638 0.420. Looking at the posteriors for each parameter to thin the model down is consistent
 7639 with these results. We can obtain model-averaged estimates from the indicator variable
 7640 approach, which produces direct model-averaged estimates of N and D :

```
7641     mu.vect sd.vect   2.5%    25%    50%    75%  97.5% Rhat n.eff
7642 D      5.695   1.133  3.759  4.916  5.591  6.362  8.193 1.002 3600
7643 N     59.077  11.758 39.000 51.000 58.000 66.000 85.000 1.002 3600
```

7644 We obtain a model-averaged estimate (posterior mean) for density of $D = 5.695$ which
 7645 is hardly any different from our model specific estimates (Table 8.3) and, in particular,
 7646 from model 2 which has only a sex-specific intercept.

7647 8.2.4 Choosing among detection functions

7648 Another approach to implementing model indicator variables is to introduce a categorical
 7649 “model identity” variable which is itself a parameter of the model. Using this approach,
 7650 then each distinct model is associated with a unique set of covariates or other set of model
 7651 features. This is convenient especially when we cannot specify the linear predictor as
 7652 some general model that reduces to various alternative sub-models simply by switching
 7653 binary variables on or off. In the context of SCR models, choosing among different en-
 7654 counter probability models would be an example. For this case we do something like this
 7655 `mod ~ dcat(probs[])` where `probs` is a vector with elements $1/(\#models)$, and the en-
 7656 counter probability matrix is filled in depending on the value of `mod`. In particular, instead
 7657 of a 2-dimensional array `p[i,j]`, we build `p[i,j,m]` for each of $m = 1, 2, \dots, M$ models.
 7658 An example with 3 distinct models is:

```
7659     mod ~ dcat(probs[])
7660     ##
7661     ## Using a double loop construction fill-in p[,] for each model:
7662     ##
7663     p[i,j,1] <- p0[1]*exp( - alpha1[1]*dist2[i,j] )
```

```

7664 p[i,j,2] <- 1-exp(-p0[2]*exp( - alpha1[2]*dist2[i,j] ) )
7665 logit(p[i,j,3]) <- p0[3] - alpha1[3]*dist2[i,j]
7666
7667 mu[i,j] <- z[i]*p[i,j,mod]
7668 y[i,j] ~ dbin(mu[i,j],K[j])

```

7669 As before the posterior probabilities can be highly sensitive to priors on the different
 7670 model parameters and sometimes mixing is really poor and, in general, we've experienced
 7671 mixed success trying to carry out model selection using this construction. We do provide
 7672 a template **R/JAGS** script (`wolvSCR0ms2`) in the `scrbook` package which has an example
 7673 of choosing among 3 different encounter probability models: The Gaussian encounter
 7674 probability, Gaussian hazard, and logistic model with the square of distance (defined
 7675 in Sec. 7.1). The key things to note are that there are 3 intercepts and 3 different
 7676 '`alpha1`' parameters (the coefficient on distance). The parameters should not be regarded
 7677 as equivalent across the models, so it is important to have them separately defined (and
 7678 estimated) for each model. In our analysis we used a vague normal prior (precision = 0.1)
 7679 for the intercept parameter (either log or logit-scale of baseline encounter probability p_0)
 7680 and a `Uniform(0,5)` prior for one-half the inverse of the coefficient on distance-squared. In
 7681 the **BUGS** model specification the priors look like this:

```

7682 for(i in 1:3){
7683   alpha0[i] ~ dnorm(0,.1)
7684   sigma[i] ~ dunif(0,5)
7685   alpha1[i] <- 1/(2*sigma[i]*sigma[i])
7686 }

```

7687 Then, we create a probability of encounter for each individual, trap *and* model so that
 7688 the holder object "p" in the model description is a 3-dimensional array (sometimes this
 7689 would have to be a 4 or 5-d array in more complex models with time effects, etc..), so that
 7690 construction of the encounter probability models look like this:

```

7691 p[i,j,1] <- p0[1]*exp( - alpha1[1]*dist2[i,j] )
7692 p[i,j,2] <- 1-exp(-p0[2]*exp( - alpha1[2]*dist2[i,j] ) )
7693 logit(p[i,j,3]) <- p0[3] - alpha1[3]*dist2[i,j]

```

7694 where

```

7695 logit(p0[1]) <- alpha0[1]
7696 log(p0[2]) <- alpha0[2]
7697 p0[3] <- alpha0[3]

```

7698 You can experiment with the `wolvSCR0ms2` script to investigate the importance of different
 7699 models of encounter probability and whether they have an affect on the inferences.

8.3 EVALUATING GOODNESS-OF-FIT

7700 In practical settings, we estimate parameters of a desirable model, or maybe fit a bunch
 7701 of models and report estimates from all of them or a model-averaged summary of density.

7702 An important question is: Is our model worth anything? In other words, does the model
7703 appear to be an adequate description of our data? Formal assessment of model adequacy or
7704 goodness-of-fit is a challenging problem and there are no all-purpose algorithms for doing
7705 this in either frequentist or Bayesian paradigms. Moreover, there are some philosophical
7706 challenges to evaluating model fit, such as, if we do model averaging then should all of
7707 the models have to fit? Or should the averaged model have to fit? What if none of the
7708 models fit? We don't know the answers to these questions and we won't try to answer
7709 them. Instead, we will provide what guidance we can on taking the first steps to evaluating
7710 fit, of a single model, as if it were a cherished family heirloom of great importance. We
7711 suggest that if you have a model that you really like, a single model, then it is a sensible
7712 thing to check that the model is a good fit to your data. If it is not, we do not imagine
7713 that the model is useless but just that some thought should be put into why the model
7714 doesn't fit so that, perhaps, some remediation might happen as future data are collected.
7715 After all, you may have spent 2, 3 or many more years of your life collecting that data set,
7716 perhaps thousands of hours, and therefore it seems a reasonable proposition to expect to
7717 do some estimation and analysis of the model regardless of model fit. You can still learn
7718 something from a model that does not pass some technical litmus test of model fit.

7719 Conceptually, we can think of evaluation of model fit as follows: if we simulate data
7720 under the model in question, do the simulated realizations resemble the data set that we
7721 actually have? For either Bayesian or classical inference, the basic strategy to assessing
7722 model fit is to come up with a fit statistic that depends on the parameters and the data
7723 set, which we denote by $T(\mathbf{y}, \theta)$, and then we compute this for the observed data set, and
7724 compare its value to that computed for perfect data sets simulated under the correct model.
7725 In the case of classical inference, we will often rely on the standard practice of parametric
7726 bootstrapping (Dixon, 2002), where we simulate data sets conditional on the MLE $\hat{\theta}$ and
7727 compare realizations with what we've observed. The R package **unmarked** (Fiske and
7728 Chandler, 2011) contains generic bootstrapping methods for certain hierarchical models,
7729 including distance sampling (e.g., see Sillett et al., 2012, for an application). In simple
7730 cases, using classical inference methods, it is sometimes possible to identify a test statistic
7731 of theoretical merit, perhaps with a known asymptotic distribution. For examples from
7732 capture-recapture see Burnham et al. (1987), Lebreton et al. (1992), and Chapt. 5 of
7733 Cooch and White (2006). For Bayesian analysis we use the Bayesian p-value method
7734 (Gelman et al., 1996) (we introduced the Bayesian p-value in sec. 3.9.1). Using this
7735 approach, data sets are simulated based on a posterior sample of the model parameters
7736 θ and some fit statistic for the simulated data sets, usually based on the discrepancy of
7737 the observed data from its expected values, is compared to that for the actual data. In
7738 most cases, whether Bayesian or frequentist, the main idea for assessing model fit is the
7739 same: We compare data sets from the model we're interested in with the data set we have
7740 in hand. If they appear to be consistent with one another, then our faith in the model
7741 increases, at least to some extent, and we say "the model fits."

7742 To date, we are unaware of any goodness-of-fit applications based on likelihood analysis
7743 of SCR models. For Bayesian analysis of SCR models, there has not been a definitive or
7744 general proposal for a fit statistic or even a class of fit statistics, although a few specialized
7745 implementations of Bayesian p-values have been provided (Royle, 2009; Gardner et al.,
7746 2010a; Royle et al., 2011a; Gopalaswamy et al., 2012a,b; Russell et al., 2012). While
7747 we universally adopt the Bayesian p-value approach, and suggest some fit statistics in

7748 the following text, we caution that there is no general expectation to support how well
 7749 they should do. As such, one might consider doing some kind of custom evaluation or
 7750 calibration when using such methods, if the power of the test (ability to reject under
 7751 specific departures from the model) is of paramount interest. We note that this uncertain
 7752 power or performance of the Bayesian p-value is not a weakness of the Bayesian approach
 7753 because the same issue applies in using bootstrap approaches applied to classical analysis
 7754 of models, if we were to devise such methods.

8.4 THE TWO COMPONENTS OF MODEL FIT

7755 For most SCR models, there are at least two distinct components of model fit, and we
 7756 propose to evaluate these two distinct components individually. First, we can ask, are the
 7757 data consistent with the *observation* model, conditional on the underlying point process?
 7758 We can evaluate this based on the encounter frequencies of individuals *conditional* on
 7759 (posterior samples of) the underlying point process $\mathbf{s}_1, \dots, \mathbf{s}_N$. We discuss some potential
 7760 fit statistics for addressing this in the next section. Second, we can evaluate whether the
 7761 data appear consistent with the *state* process model (i.e., the “uniformity” assumption of
 7762 the point process). For the simple model of independence and uniformity, this is similar
 7763 to the assumption of *complete spatial randomness* (CSR) which we consider in Sec. 8.4.1
 7764 below. Actually, this is not strictly the assumption of CSR because of the binomial
 7765 assumption on N under data augmentation, so we instead use the term *spatial randomness*.

8.4.1 Testing uniformity or spatial randomness

7766 Historically, especially in ecology, there has been an extraordinary amount of interest in
 7767 whether a realization of a point process indicates “complete spatial randomness,” i.e., that
 7768 the points are distributed uniformly and independently in space. Two good references
 7769 for such things are Cressie (1991, Ch. 8) and Illian et al. (2008)¹. In the context of
 7770 animal capture-recapture studies, the spatial randomness hypothesis is manifestly false,
 7771 purely on biological grounds. Typically individuals will be clustered, or more regular (for
 7772 territorial species), than expected under spatial randomness and heterogeneous habitat
 7773 will generate the appearance of clustering even if individuals are distributed independently
 7774 of one another. While we recommend modeling spatial structure explicitly when possible
 7775 (Chaps. 11, 12, 13), the uniformity assumption may be an adequate description of data
 7776 sets in some situations. Further, we find that it is generally flexible enough to reflect
 7777 non-uniform patterns in the data, because we do observe some direct information about
 7778 some of the point locations.

7779 The basic technical framework for evaluating the spatial randomness hypothesis is
 7780 based on counts of activity centers in cells or bins. For that we use any standard goodness-
 7781 of-fit test statistic, based on gridding (i.e., binning) the state-space of the point process into
 7782 $g = 1, 2, \dots, G$ cells or bins, and we tabulate $N_g \equiv N(\mathbf{x}_g)$ the number of activity centers in
 7783 bin g , centered at coordinate \mathbf{x}_g . Specifically, let $B(\mathbf{x})$ indicate a bin centered at coordinate

¹We also like Tony Smith’s lecture notes (Univ. of Penn. ESE 502), which can be found at http://www.seas.upenn.edu/~ese502/NOTEBOOK/Part_I/3_Testing_Spatial_Randomness.pdf, accessed January 24, 2013.

7785 \mathbf{x} , then² $N(\mathbf{x}) = \sum_{i=1}^N I(\mathbf{s}_i \in B(\mathbf{x}))$ is the population size of bin $B(\mathbf{x})$. In Sec. 5.11.1,
 7786 we used the summaries $N(\mathbf{x})$ for producing density maps from MCMC output. Here, we
 7787 use them for constructing a fit statistic. We have used the Freeman-Tukey statistic of this
 7788 form:

$$T(\mathbf{N}, \theta) = \sum_g (\sqrt{N_g} - \sqrt{\mathbb{E}(N_g)})^2$$

7789 where $\mathbb{E}(N_g)$ is estimated by the mean bin count. An alternative conventional assessment
 7790 of fit is based on the following statistic: Conditional on N , the total number of activity
 7791 centers in the state-space \mathcal{S} , the bin counts N_g should have a binomial distribution. It will
 7792 usually suffice to approximate the binomial cell counts by Poisson cell counts, in which
 7793 case we can use the classical “index-of-dispersion” test (Illian et al., 2008, p. 87), based
 7794 on the variance-to-mean ratio:

$$I = (G - 1) * s^2 / \bar{N}$$

7795 where s^2 is the sample variance of the bin counts and \bar{N} is the sample mean. When the
 7796 point process realization is *observed*, as in classical point pattern modeling (but not in
 7797 SCR), this statistic has approximately a Chi-square distribution on $(G - 1)$ degrees-of-
 7798 freedom under the spatial randomness hypothesis. If $s^2/\bar{N} > 1$, clustering is suggested
 7799 whereas, $s^2/\bar{N} < 1$ suggests the point process is too regular.

7800 Whatever statistic we choose as our basis for assessing spatial randomness, *the im-*
 7801 *portant technical issue is that we don’t observe the point process and so the standard*
 7802 *statistics for evaluating spatial randomness cannot be computed directly. However, using*
 7803 *Bayesian analysis, we do have a posterior sample of the underlying point process and*
 7804 *so we suggest computing the posterior distribution of any statistic in a Bayesian p-value*
 7805 *framework. For a given posterior draw of all model parameters, N is known, based on the*
 7806 *value of the data augmentation variables z_i , and so we can obtain a posterior sample of*
 7807 *$N(\mathbf{x})$ by taking all of the output for MCMC iterations $m = 1, 2, \dots$, and doing this:*

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

7808 Thus, $N(\mathbf{x})^{(1)}, N(\mathbf{x})^{(2)}, \dots$, is the Markov chain for the derived parameter $N(\mathbf{x})$.

7809 In addition to computing the bin counts for each iteration of the MCMC algorithm,
 7810 at the same time we generate a realization of the activity centers \mathbf{s}_i under the spatial
 7811 randomness model, and we obtain bin counts for these “new” data, $\tilde{N}(\mathbf{x})$. For each of
 7812 the posterior samples – that of the real data, and that of the posterior simulated data, we
 7813 compute the fit-statistic. The fit statistic based on the actual data is:

$$T(\mathbf{N}, \theta) = \sum_x (\sqrt{N(x)} - \sqrt{\tilde{N}(x)})^2$$

7814 whereas the fit statistic based on a simulated realization of points under the spatial ran-
 7815 domness hypothesis is:

$$T(\tilde{\mathbf{N}}, \theta) = \sum_x (\sqrt{\tilde{N}(x)} - \sqrt{\tilde{N}(x)})^2$$

² $I(arg)$ is the indicator function which evaluates to 1 if *arg* is true, otherwise 0

7816 And we compute the Bayesian p-value by tallying up the proportion of times that $T(\tilde{\mathbf{N}}, \theta)$
 7817 is larger than $T(\mathbf{N}, \theta)$, as an estimate of: $p = \Pr(T(\tilde{\mathbf{N}}, \theta) > T(\mathbf{N}, \theta))$. The **R** function
 7818 **SCRgof** in our package **scrbook** will do this, given the output from **JAGS** (see below).

7819 Sensitivity to bin size

7820 Evaluating fit based on bin counts in point process models are sensitive to the number of
 7821 bins (Illian et al., 2008, p. 87-88). This is related to the classical problem of fit testing
 7822 for binary regression because in a point process model, as the number of grid cells gets
 7823 small, the grid cell counts go to 0 or 1 and standard fit statistics (e.g., based on deviance
 7824 or Pearson residuals) are known not to be very useful. There is some good discussion of
 7825 this in McCullagh and Nelder (1989, Sec. 4.4.5). What it boils down to is, using the
 7826 example of the Pearson residual statistic considered by McCullagh and Nelder (1989), the
 7827 fit statistic is exactly a deterministic function of the sample size only, which clearly should
 7828 not be regarded as useful for model fit. This is why, in order to do a check of model fit
 7829 when you have a binary response, one must always aggregate the data in some fashion. In
 7830 the context of testing spatial randomness, computing the test statistic we described above
 7831 has us chop up the region \mathcal{S} into bins, and tally up N_g , the frequency of activity centers
 7832 in each bin g . Suppose that we choose the bin size to be extremely small such that $\mathbb{E}(N_g)$
 7833 tends to N/G (N being the number of activity centers). Further, N_g tends to a binary
 7834 outcome. Therefore the fit statistic has N components that have value $N_g = 1$, and it has
 7835 $G - N$ components that have value $N_g = 0$. Therefore, the fit statistic resembles:

$$T(\mathbf{N}, \theta) = \sum_{g \ni N_g=1}^N (1 - \sqrt{N/G})^2 + \sum_{g \ni N_g=0}^{G-N} (N/G)^2 = N(1 + (G - N)/G)$$

7836 (here \ni means “such that”). If G is huge relative to N , then we see that this tends to
 7837 about $2 * N$, which does not provide any meaningful assessment of model fit. So if you
 7838 look at this in the limit in which the bin counts become binary, the fit statistic loses all
 7839 its variability to the specific model used and is just a deterministic function of N . As a
 7840 practical matter, it probably makes sense to restrict the number of bins to *fewer* than the
 7841 number of observed individuals in the sample size. In typical SCR applications this will
 7842 therefore result, usually, in very large (and few) bins, and presumably not much power.

7843 There are some extensions that help resolve the issue of sensitivity to bin size. We can
 7844 construct fit statistics based not just on quadrat counts but also the neighboring quadrat
 7845 counts – this is the Greig-Smith method (Greig-Smith, 1964). In addition, there are a
 7846 myriad of “distance methods” for evaluating point process models, and we believe that
 7847 many of these can (and will) be adapted to SCR models. Again the main feature is that
 7848 the point process on which inference is focused is completely latent in SCR models – so
 7849 this makes the fit assessment slightly different than in classical point processes. That said,
 7850 the methods should be adaptable, e.g., in a Bayesian p-value kind of way.

7851 Sensitivity to state-space extent

7852 An issue that we have not investigated is that any model assessment that applies to a *latent*
 7853 point process is probably sensitive to the size of the state-space. As the size of the state-
 7854 space increases then the cell counts (far away from the data) *are* independent binomial
 7855 counts with constant density, and so we can overwhelm the fit statistic with extraneous
 7856 “data” simulated from the posterior, which is equal to the prior as we move away from the

7857 data, and therefore uninformed by the observed data that live in the vicinity of the trap
 7858 array. Therefore we recommend computing these goodness-of-fit statistics in the vicinity
 7859 of the trap array only. Perhaps, as an ad hoc rule-of-thumb, less than the average trap
 7860 spacing from the rectangle enclosing the trap array. For example, if the average trap
 7861 spacing is, say, 10 km, then the bins used to obtain the observed and predicted activity
 7862 centers should not extend any further from the traps than 5 km. This should be a matter
 7863 of future research.

7864 **8.4.2 Assessing fit of the observation model**

7865 In evaluating the spatial randomness hypothesis, we could draw on well-established ideas
 7866 from point process modeling. On the other hand, it is less clear how to approach goodness-
 7867 of-fit evaluation of the observation model. For most SCR problems, we have a 3-dimensional
 7868 data array of *binary* observations, y_{ijk} for individual i , trap j and sample occasion k . As
 7869 discussed in the previous section, we need to construct fit statistics based on observed and
 7870 expected frequencies that are aggregated in some fashion. In practice, the data will be
 7871 too sparse to have much power, unless the data are highly aggregated. We recommend
 7872 focusing on summary statistics that represent aggregated versions of y_{ijk} over 1 or 2 of
 7873 the dimensions. We describe 3 such fit statistics below. We recognize that, depending on
 7874 the model, some information about model fit will be lost by summarizing the data in this
 7875 way. For example if there is a behavioral response and we aggregate over time to focus
 7876 on the individual and trap level summaries then some information about lack of fit due
 7877 to temporal structure in the data is lost.

7878 **Fit statistic 1: individual x trap frequencies** We summarize the data by individual
 7879 and trap-specific counts y_{ijk} aggregated over all sample occasions. Using standard
 7880 “dot notation” to represent summed quantities, we express that as: $y_{ij\cdot} = \sum_{k=1}^K y_{ijk}$.
 7881 Conditional on \mathbf{s}_i , the expected value under any encounter model is:

$$\mathbb{E}(y_{ij\cdot}) = p_{ij} K$$

7882 (or K_j if the traps are operational for variable periods). If there is time-varying structure
 7883 to the model, then expected values would have to be computed according to $\mathbb{E}(y_{ij\cdot}) =$
 7884 $\sum_k p_{ijk}$. Then we can define a fit statistic from the Freeman-Tukey residuals according
 7885 to:

$$T_1(\mathbf{y}, \theta) = \sum_i \sum_j (\sqrt{y_{ij\cdot}} - \sqrt{\mathbb{E}(y_{ij\cdot})})^2$$

7886 where we use θ here to represent the collection of all parameters in the model. This is
 7887 conditional on \mathbf{s} as well as on the data augmentation variables \mathbf{z} . We compute this statistic
 7888 for *each* iteration of the MCMC algorithm for the observed data set and also for a new
 7889 data set simulated from the posterior distribution, say $\tilde{\mathbf{y}}$.

7890 We could also use a similar fit statistic derived from summarizing over traps to obtain
 7891 an $n_{ind} \times K$ matrix of count statistics. We imagine that either summary of the data will
 7892 probably be too disaggregated (have mostly values of 0) in most practical settings to have
 7893 much power.

7894 **Fit statistic 2: Individual encounter frequencies.** SCR models represent a
 7895 type of model for heterogeneous encounter probability, like model M_h , but with an ex-
 7896 plicit factor (space) that explains part of the heterogeneity. For model M_h , the individual

7897 encounter frequencies are the sufficient statistic for model parameters, and so it makes in-
 7898 tuitive sense to provide some kind of omnibus fit assessment of the core heuristic that SCR
 7899 model is adequately explaining the heterogeneity using a model M_h -like statistic based
 7900 on individual encounter frequencies. So, we build a fit statistic based on the individual
 7901 total encounters (Russell et al., 2012), $y_{i..} = \sum_j \sum_k y_{ijk}$. In addition, the expected value
 7902 is a similar summary over traps and occasions: $\mathbb{E}(y_{i..}) = \sum_j \sum_k p_{ijk}$. Then, we define
 7903 statistic T_2 according to:

$$T_2(\mathbf{y}, \theta) = \sum_i (\sqrt{y_{i..}} - \sqrt{\mathbb{E}(y_{i..})})^2$$

7904 We imagine this test statistic should provide an omnibus test of extra-binomial variation
 7905 and should therefore capture some effect of variable exposure to encounter of individuals,
 7906 although we have not carried out any evaluations of power under specific alternatives.
 7907 Obviously, in using this statistic, we lose information on departures from the model that
 7908 might only be trap- or time-specific.

7909 **Fit Statistic 3: Trap frequencies.** We construct an analogous statistic based
 7910 on aggregating over individuals and replicates to form trap encounter frequencies: $y_{.j} =$
 7911 $\sum_i \sum_k y_{ijk}$ (Gopalaswamy et al., 2012b) and the expected value is a similar summary
 7912 over individuals and occasions: $\mathbb{E}(y_{.j}) = \sum_i \sum_k p_{ijk}$. Then statistic T_3 is:

$$T_3(\mathbf{y}, \theta) = \sum_j (\sqrt{y_{.j}} - \sqrt{\mathbb{E}(y_{.j})})^2$$

7913 This seems like a sensible fit statistic because we can think of SCR models as spatial
 7914 models for counts (Chandler and Royle, In press). Therefore, we should seek models that
 7915 provide good predictions of the observable spatial data, which are the trap totals. In this
 7916 context, it might even make sense to pursue cross-validation based methods for model
 7917 selection. Cross-validation is a standard method of evaluating models such as in kriging
 7918 or spline smoothing, so we could as well develop such ideas based on the trap-specific
 7919 frequencies.

7920 8.4.3 Does the SCR model fit the wolverine data?

7921 We use the ideas described in the previous section to evaluate goodness-of-fit of the SCR
 7922 model to the wolverine camera trapping data.

7923 We consider first whether the simple model of spatial randomness of the activity
 7924 centers is adequate. We think that the encounter model shouldn't have a large effect
 7925 on whether the spatial randomness assumption is adequate or not, so we fit "Model 0"
 7926 (in which parameters are *not* sex-specific) using an **R** script provided in the function
 7927 **wolvSCR0gof** which will default to fitting the model in **JAGS**. This is the same script as
 7928 **wolvSCR0ms** except that it saves the MCMC output for the activity centers **s** and the data
 7929 augmentation variables **z**, which are required in order to compute the Bayesian p-value
 7930 test of spatial randomness.

7931 The MCMC output is processed with the **R** function **SCRgof** which computes the test
 7932 of spatial randomness based on bin counts, using the Bayesian p-value calculation. The
 7933 function **SCRgof** requires a few things as inputs: (1) the output from a **BUGS** run (in
 7934 particular, the activity center coordinates and the data augmentation variables); (2) the

7935 number of bins to create for computing spatial frequencies of activity centers; (3) the trap
 7936 locations and, (4) the buffer around the trap array to use in computing the bin counts.
 7937 This buffer could be that used in defining the state-space for the model fitting, but we
 7938 think it should be relatively tighter to the trap array than the state-space used in model-
 7939 fitting. For the wolverine analysis, where we're using 10-km grid cells (1 unit = 10 km)
 7940 and a 20 km buffer for model fitting, we'll use a state-space buffer of 0.4 units (4 km) for
 7941 computing the fit statistic. The **R** code to fit the model and obtain the goodness-of-fit
 7942 result is as follows:

```
7943 > wolv1 <- wolvSCR0gof(nb=1000,ni=6000,buffer=2,M=200,model=0)
7944
7945 > bugsout <- wolv1$BUGSoutput$sims.list
7946
7947 > traplocs <- wolverine$wtraps[,2:3]
7948 > traplocs[,1] <- traplocs[,1] - min(traplocs[,1])
7949 > traplocs[,2] <- traplocs[,2] - min(traplocs[,2])
7950 > traplocs <- traplocs/10000
7951
7952 > set.seed(2013) # set seed so Bayesian p-value is the same each time
7953
7954 > SCRgof(bugsout,5,5,traplocs=traplocs,buffer=.4)
7955
7956 Cluster index observed: 1.099822
7957 Cluster index simulated: 1.000453
7958 P-value index of dispersion: 0.408
7959 P-value2 freeman-tukey: 0.6842667
```

7960 The output produced by **SCRgof** is the cluster index based on the ratio of the variance
 7961 to the mean (see above), which is computed as the posterior mean index of dispersion for
 7962 the latent point process, and also the average value for simulated data. If this value is
 7963 > 1 then clustering is suggested, which we see a (very) minor amount of evidence for here.
 7964 Two Bayesian p-values are produced: the first is based on the cluster index, and the 2nd
 7965 is based on the Freeman-Tukey statistic calculated as described in Sec. 8.4.1. Because our
 7966 p-values aren't close to 0 or 1, we judge that the model of spatial randomness provides
 7967 an adequate fit to the data. You can verify that a similar result is obtained if we use the
 7968 model with fully sex-specific parameters (Model 4).

7969 Next, we did a Bayesian p-value analysis of the observation component of the model,
 7970 using the 3 fit statistics described in Sec. 8.4.2. These statistics can be calculated as
 7971 part of the **BUGS** model specification or by post-processing the MCMC output returned
 7972 from a **BUGS** run. The **R** script **wolvSCR0gof** contains the relevant calculations. For
 7973 example, to compute fit statistic 1, we have to add some commands to the **BUGS** model
 7974 specification such as this (note: this is only a fraction of the model specification):

```
7975 .....
7976 for(j in 1:ntraps){
7977   mu[i,j] <- w[i]*p[i,j]
7978   y[i,j] ~ dbin(mu[i,j],K[j])
```

```

7980   ynew[i,j] ~ dbin(mu[i,j],K[j])
7981
7982   err[i,j] <- pow(pow(y[i,j],.5) - pow(K[j]*mu[i,j],.5),2)
7983   errnew[i,j] <- pow(pow(ynew[i,j],.5) - pow(K[j]*mu[i,j],.5),2)
7984 }
7985
7986 Tlobs <- sum(err[,])
7987 Tnew <- sum(errnew[,])
7988 .....

```

7989 Similar calculations are carried out to obtain the posterior samples of test statistics 2
7990 (individual totals) and 3 (trap totals). For the wolverine data, the Bayesian p-value
7991 calculations produce:

```

7992 > mean(wolv1$BUGSoutput$sims.list$T1new>wolv1$BUGSoutput$sims.list$T1obs)
7993 [1] 0
7994
7995 > mean(wolv1$BUGSoutput$sims.list$T2new>wolv1$BUGSoutput$sims.list$T2obs)
7996 [1] 0.17
7997
7998 > mean(wolv1$BUGSoutput$sims.list$T3new>wolv1$BUGSoutput$sims.list$T3obs)
7999 [1] 0.02066667

```

8000 Based on statistic T_2 , we might conclude that the model is adequate for explaining
8001 individual heterogeneity although the other two statistics suggest a general lack of fit of
8002 the observation model. A similar result is obtained using the fully sex-specific model. We
8003 note that one individual was captured 8 times in one trap, which is pretty extreme under
8004 a model which assumes independent Bernoulli trials. We summarize that the trap-counts
8005 simply are not well-explained by this model.

8006 In attempt to resolve this problem, we extended the model to include a local (trap-
8007 specific) behavioral response (following Royle et al. (2011b)) which can be fitted using
8008 the sample **R** script **wolvSCRMb**. To fit a model using **WinBUGS**, and then compute the
8009 Bayesian p-values we do this:

```

8010 > wolv.Mb <- wolvSCRMb(nb=1000,ni=6000,buffer=2,M=200)
8011
8012 > mean(wolv.Mb$sims.list$T1new>wolv.Mb$sims.list$T1obs)
8013 [1] 0.9666667
8014
8015 > mean(wolv.Mb$sims.list$T2new>wolv.Mb$sims.list$T2obs)
8016 [1] 0.3644667
8017
8018 > mean(wolv.Mb$sims.list$T3new>wolv.Mb$sims.list$T3obs)
8019 [1] 0.4990667

```

8020 Given that this model seems to fit better, we might prefer reporting estimates under
8021 this model, which we do in Table 8.4. (the behavioral response parameter is labeled α_2
8022 in the table). Estimated density is about 1 individual higher per 1000 km² compared

with the various models that lack a behavioral response. It might be useful to try these fit assessment exercises using the habitat mask as described in Sec. 5.10. That takes an extremely long time to run in **BUGS** though, especially for the behavioral response model.

Table 8.4. Posterior summary statistics for local (trap-specific) behavioral response model M_b fitted to the wolverine camera trapping data using **WinBUGS**. The parameter α_2 is the local (trap-specific) behavioral response parameter. $T_x()$ are the posterior summaries of fit statistics $x = 1, 2, 3$ used in the Bayesian p-value analysis (See text for definitions). Results are based on 3 chains, each with 6000 iterations (first 1000 discarded) for a total of 15000 posterior samples.

Parameter	Mean	SD	2.5%	50%	97.5%	Rhat	n.eff
N	71.32	19.07	42.00	69.00	114.02	1.00	2100
D	6.87	1.84	4.05	6.65	10.99	1.00	2100
σ	0.88	0.13	0.68	0.86	1.17	1.00	730
p_0	0.01	0.00	0.01	0.01	0.02	1.01	530
α_1	0.69	0.19	0.37	0.67	1.10	1.00	730
α_2	2.50	0.27	1.99	2.50	3.04	1.00	700
ψ	0.36	0.10	0.20	0.35	0.58	1.00	2600
T_1^{obs}	54.71	6.12	43.69	54.39	67.47	1.00	3900
T_1^{new}	64.73	7.62	50.93	64.39	80.96	1.00	3900
T_2^{obs}	13.93	4.07	7.25	13.53	23.04	1.00	5700
T_2^{new}	12.65	3.35	6.93	12.36	20.07	1.00	2000
T_3^{obs}	12.80	1.74	9.80	12.64	16.61	1.00	2400
T_3^{new}	12.94	3.05	7.77	12.67	19.58	1.00	15000

8.5 QUANTIFYING LACK-OF-FIT AND REMEDIATION

Molinari-Jobin et al. (2013) used a strategy for assessing model fit in dynamic occupancy models (Royle and Kéry, 2007) similar to that which we suggested above. They constructed a fit statistic based on aggregating the data over replicate samples (k), to obtain the total detections per site i and year j . They used a Bayesian p-value analysis based on a Chi-squared test statistic (also see Kéry and Schaub, 2012, Chapt. 12). Their analysis suggested a model that didn't fit, and, so they computed the "lack-of-fit ratio" (see Kéry and Schaub, 2012, Sec. 12.3) – the ratio of the fit statistic computed for the actual data to that of the replicate data sets. They interpret this analogous to the over-dispersion coefficient in generalized linear models (McCullagh and Nelder, 1989), usually called the c-hat statistic in capture-recapture literature (see Cooch and White, 2006, Chapt. 5). Molinari-Jobin et al. (2013) reported the lack-of-fit ratio for their model to be 1.14 which suggests a minor lack-of-fit, compared to perfect data having a value of 1, because the posterior standard deviations will be too small by a factor of $\sqrt{1.14} = 1.07$. In classical capture-recapture applications of goodness-of-fit assessment, inference for non-fitting models is dealt with by inflating the resulting SEs (of the non-fitting model), by the square-root of c-hat. We believe that these ideas related to quantifying lack-of-fit and understanding its effect could also be applied to SCR models, although we have not yet explored this.

8.6 SUMMARY AND OUTLOOK

8044 In this chapter, we offered some general strategies for model selection and model checking,
8045 or assessment of model fit. We think the strategies we outlined for model selection are fairly
8046 standard and can be effectively applied to many SCR modeling problems. Some technical
8047 issues of Bayesian analysis need to be addressed (in general) before Bayesian methods
8048 are more generally useful and accessible. For one thing, Bayesian model selection based
8049 on the indicator variable approach of Kuo and Mallick (1998) can be tediously slow even
8050 for small data sets, and so improved computation will improve our ability to do Bayesian
8051 model selection in practical situations. Also, and most importantly, sensitivity to prior
8052 distributions is an important issue. Further research and practice might identify preferred
8053 prior configurations for SCR that provide a good calibration in relevant model selection
8054 problems. Finally, we believe that cross-validation should prove to be a useful method
8055 in model assessment and selection, as SCR models are a form of spatial model of counts,
8056 and so it is natural to pick models that predict the observable spatial counts (i.e., at trap
8057 locations) well.

8058 For Bayesian model assessment, or goodness-of-fit checking, we suggested a framework
8059 based on independent testing of the spatial model of independence and uniformity, and
8060 testing fit of the observation model conditional on the underlying point process. These
8061 ideas are based on mostly *ad hoc* attempts in a number of published applications (Royle
8062 et al., 2009a, 2011a; Gopalaswamy et al., 2012b; Russell et al., 2012, e.g.). While we think
8063 this general strategy should be fruitful, we know of no studies on the power to detect
8064 various model departures, and so the ideas should be viewed as experimental. We have
8065 not discussed assessment of model fit for SCR models using likelihood methods, although
8066 we imagine that standard bootstrapping ideas should be effective, perhaps based on the
8067 fit statistics (or similar ones) we suggested here for computing Bayesian p-values.

8068 Clearly there is much research to be done on assessment of model fit in SCR models.
8069 For testing the spatial randomness hypothesis, we used a classical approach based on
8070 count frequencies, in which point locations are put into spatial bins. Other approaches
8071 from spatial point process modeling should be pursued including nearest-neighbor methods
8072 or distance-based methods. In addition, studies to evaluate the power to detect relevant
8073 departures from the standard assumptions, and the robustness of inferences about N or
8074 density, need to be conducted. If the spatial randomness model appears inadequate, it
8075 is possible to fit models that allow for a non-uniform distribution of points (see Chapt.
8076 11) and even point process models that allow for interactions among points (Reich et al.,
8077 2012). On the other hand, we expect that most of these Bayesian p-value tests will have
8078 low power in typical data sets consisting of a few to a few dozen individuals. As such,
8079 failure to detect a lack of fit may not be that meaningful. But, on the other hand, it
8080 may not make a difference in terms of density estimates either. We think inference about
8081 density should be relatively insensitive to departures from spatial randomness, because
8082 we get to observe direct information on some component of the population, component
8083 of density is *observed*. For those activity centers, the assumed model of the point process
8084 should exert little influence on the placement of the activity centers. Conversely, as is
8085 the case with classical closed population models (Otis et al., 1978; Dorazio and Royle,
8086 2003; Link, 2003), inferences may be somewhat more sensitive to bad-fitting models for
8087 the observation process.

8088

9

8089

8090

ALTERNATIVE OBSERVATION MODELS

8091 In previous chapters we considered various models of *encounter probability*, both in terms
8092 of parametric functions of distance and also a myriad of covariate models (Chapt. 7 and
8093 elsewhere). However, we have so far only considered a specific probability model for the
8094 observations (we'll call this the "encounter process") – the Bernoulli encounter process
8095 model which, in **secr**, is the *proximity detector* model. This assumes that individual and
8096 trap-specific encounters are independent Bernoulli trials. Here, we focus on developing
8097 additional models for the encounter process. The encounter process could be thought of as
8098 being determined by the type of device – or the type of "detector" using the terminology
8099 of **secr** (Efford, 2011).

8100 In this chapter, we consider alternative observation models that accommodate ob-
8101 servations that are not binary, and do not require independence of the observations. In
8102 particular, we consider models for encounter *frequencies*, and encounter process models
8103 based on the multinomial distribution. For example, if sampling devices can detect an
8104 individual some arbitrary number of times during an interval, then it is natural to consider
8105 observation models for encounter frequencies, such as the Poisson model. Another type
8106 of encounter device is the "multi-catch" device (Efford et al., 2009a) which is a physical
8107 device that can capture and hold an arbitrary number of individuals. A typical example
8108 is a mist-net for birds (Borchers and Efford, 2008). It is natural to regard observations
8109 from these kinds of studies as independent multinomial observations. A related type of
8110 device that produces *dependent* multinomial observations are the so-called *single-catch*
8111 traps (Efford, 2004; Efford et al., 2009a). The canonical example are small-mammal live
8112 traps which catch and hold a single individual. Competition among individuals for traps
8113 induces a complex dependence structure among individual encounters. To date, no formal
8114 inference framework has been devised for this method although it stands to reason that
8115 the independent multinomial model should be a good approximation in some situations
8116 (Efford et al., 2009a). We analyze a number of examples of these different observation
8117 models using **JAGS** and also the **R** package **secr** (Efford, 2011).

9.1 POISSON OBSERVATION MODEL

8118 The models we analyze in Chapt. 5 assumed binary observations – i.e., standard encounter
 8119 history data – so that individuals are captured at most one time in a trap on any given
 8120 sample occasion. This makes sense for many types of DNA sampling (e.g., based on hair
 8121 snares) because distinct visits to sampled locations or devices cannot be differentiated.
 8122 However, for some encounter devices, or methods, the potential number of encounters is
 8123 *not* fixed, and so it is possible to encounter an individual some arbitrary number of times
 8124 during any particular sampling episode. That is, we might observe encounter frequencies
 8125 $y_{ijk} > 1$ for individual i , trap j and sampling interval k . As an example, if a camera
 8126 device is functioning properly it may be programmed to take photos every few seconds if
 8127 triggered. For a second example, suppose we are searching a quadrat or length of trail
 8128 for scat, we may find multiple samples from the same individual. Therefore, we seek
 8129 observation models that accommodate such encounter frequency data. In general, any
 8130 discrete probability mass function could be used for this purpose, including the standard
 8131 models for count data used throughout ecology, the Poisson and negative binomial. Here
 8132 we focus on using the Poisson model only although other count frequency models are
 8133 possible for SCR models (Efford et al., 2009b).

8134 Let y_{ijk} be the frequency of encounter for individual i , in trap j , during occasion k ,
 8135 then assume:

$$y_{ijk} \sim \text{Poisson}(\lambda_{ij})$$

8136 where the expected encounter frequency λ_{ij} depends on both individual and trap. As we
 8137 did in the binary model of Chapt. 5, we now seek to model the expected value of the
 8138 observation (which was p_{ij} in Chapt 5) as a function of the individual activity center \mathbf{s}_i .
 8139 We propose

$$\lambda_{ij} = \lambda_0 k(\mathbf{x}_j, \mathbf{s}_i)$$

8140 Where $k(\mathbf{x}, \mathbf{s})$ is any positive valued function, such as the negative exponential or the
 8141 bivariate Gaussian kernel, and λ_0 is the baseline encounter rate – the expected number
 8142 of encounters if a trap is placed precisely at an individuals home range center (note: in
 8143 `secr` the notation for this is g_0). Then, $\lambda_0 k(\mathbf{x}_j, \mathbf{s}_i)$ is the expected encounter rate in trap
 8144 \mathbf{x}_j for an individual having activity center \mathbf{s}_i . Note that

$$\log(\lambda_{ij}) = \log(\lambda_0) + \log(k(\mathbf{x}_j, \mathbf{s}_i)).$$

8145 Equating $\alpha_0 \equiv \log(\lambda_0)$, and, if $k(\mathbf{x}, \mathbf{s}) \equiv \exp(-d(\mathbf{x}, \mathbf{s})^2/(2\sigma^2))$ (i.e., the Gaussian model),
 8146 then:

$$\log(\lambda_{ij}) = \alpha_0 - \alpha_1 d(\mathbf{x}_j, \mathbf{s}_i)^2 \quad (9.1.1)$$

8147 where $\alpha_1 = 1/(2\sigma^2)$, which is the same linear predictor as we have seen for the Bernoulli
 8148 model in Chapt. 5. This Poisson SCR model is therefore a type of Poisson generalized
 8149 linear mixed model (GLMM).

8150 We can accommodate covariates at the level of individual-, trap- or sample occasion
 8151 by including them on the baseline encounter rate parameter λ_0 . For example, if C_j is
 8152 some covariate that depends on trap only, then we express the relationship between λ_0
 8153 and C_j as:

$$\log(\lambda_{0,ijk}) = \alpha_0 + \alpha_2 C_j$$

8154 and therefore covariates on the logarithm of baseline encounter probability appear also as
 8155 linear effects on λ_{ij} . In general, covariates might also affect the coefficient on the distance

8156 term (α_1) (e.g., sex of individual). We don't get into too much discussion of general
 8157 covariate models here, but we covered them in some detail in both Chaps. 7 and 8.

8158 For models in which we do not have covariates that vary across the sample occasions
 8159 k , we can aggregate the observed data by the property of compound additivity of the
 8160 Poisson distribution (if x and y are *iid* Poisson with mean λ then $x + y$ is Poisson with
 8161 mean 2λ). Therefore,

$$y_{ij} = \left(\sum_{k=1}^K y_{ijk} \right) = \text{Poisson}(K\lambda_0 k(\mathbf{x}_j, \mathbf{s}_i))$$

8162 We see that K and λ_0 serve the same role as affecting the base encounter rate. Since the
 8163 observation model is the same, probabilistically speaking, for all values of K , evidently
 8164 we need only $K = 1$ "survey" from which to estimate model parameters (Efford et al.,
 8165 2009b). We know this intuitively, as sampling by multiple traps serves as replication
 8166 in SCR models. This has great practical relevance to the conduct of capture-recapture
 8167 studies and the use of SCR models. For example, if individuality is obtained by genetic
 8168 information from scat sampling, one should only have to carry out a single spatial sampling
 8169 of the study area. However, one must be certain that sufficient spatial recaptures will be
 8170 obtained so that effective estimation is possible.

8171 9.1.1 Poisson model of space usage

8172 It is natural to interpret the Poisson encounter model as a model of space usage resulting
 8173 from movement of individuals about their home range (Sec. 5.4). Imagine we have perfect
 8174 samplers in every pixel of the landscape so that whenever an individual moves from one
 8175 pixel to another, we can record it. Let m_{ij} be the number of times individual i was
 8176 recorded in pixel j (i.e., it selected or used pixel j). Then, we might think of the Poisson
 8177 model for the observed *use* frequencies:

$$m_{ij} \sim \text{Poisson}(\lambda_0 k(\mathbf{x}_j, \mathbf{s}_i))$$

8178 where λ_0 is related to the baseline movement rate of the animal (how often it moves). This
 8179 model of space usage gives rise to the standard resource selection function (RSF) models
 8180 (see Chapt. 13). But now suppose our samplers are not perfect but, rather, record only
 8181 a fraction of the resulting visits. A sensible model is

$$y_{ij}|m_{ij} \sim \text{Binomial}(m_{ij}, p).$$

8182 The marginal distribution of y_{ij} is:

$$y_{ij} \sim \text{Poisson}(p_0 k(\mathbf{x}_j, \mathbf{s}_i)).$$

8183 where p_0 is a composite of the movement rate and conditional detection probability p .
 8184 Therefore, we see that encounters accumulate in proportion to the frequency of outcomes
 8185 of an individual using space (or "selecting resources").

8186 We introduced an interpretation of SCR models in terms of movement and space usage
 8187 in Sec. 5.4, and it is one of the main underlying concepts of SCR models that is not present
 8188 in ordinary capture-recapture models. As we noted there, the underlying model of space
 8189 usage is only as complex as the encounter probability model which has been, so far in this
 8190 book, only symmetric and stationary (does not vary in space). We generalize this model
 8191 of space usage substantially in Chapt. 13.

9.1.2 Poisson relationship to the Bernoulli model

8192 There is a sense in which the Poisson and Bernoulli models can be viewed as consistent with
 8193 one another. Note that under the Poisson model, the relationship between the expected
 8194 count and the probability of counting “at least 1”, is given by
 8195

$$\Pr(y > 0) = 1 - \exp(-\lambda) \quad (9.1.2)$$

8196 where $\mathbb{E}(y) = \lambda$. Therefore, if we equate the event “encountered” with the event that the
 8197 individual was captured at least 1 time under the Poisson model, i.e., $y > 0$, then it would
 8198 be natural to set $p_{ij} = \Pr(y > 0)$ according to Eq. 9.1.2. That is, we can use Eq. 9.1.2
 8199 as the model for encounter probability for binary observations. This is the “hazard rate”
 8200 model in distance sampling.

8201 In fact, as λ gets small, the Poisson model is a close approximation to the Bernoulli
 8202 model in the sense that outcomes concentrate on $\{0, 1\}$, i.e., $\Pr(y \in \{0, 1\}) \rightarrow 1$ as $\lambda \rightarrow 0$.
 8203 Indeed, under the Poisson model, $\Pr(y > 0) \rightarrow \lambda$ for small values of λ . This phenomenon
 8204 is shown in Fig. 9.1 where the left panel shows a plot of $\lambda_{ij} = \lambda_0 k(\mathbf{x}_j, \mathbf{s}_i)$ vs. distance and
 8205 superimposed on that is a plot of $p_{ij} = 1 - \exp(-\lambda_{ij})$ vs. distance, for values $\lambda_0 = 0.1$
 8206 and $\sigma = 1$, and the right panel shows a plot of $\Pr(y > 0)$ vs. $\mathbb{E}(y)$. We see that the two
 8207 quantities are practically indistinguishable. This is convenient in some cases because the
 8208 Poisson model might be more tractable to fit (or even vice versa). For an example, see
 8209 the models described in Chapt. 18, and we also consider another case in Sec. 9.3 below.
 8210 To evaluate the closeness of the approximation, you can use the following R commands
 8211 which we used to produce Fig. 9.1:

```
8212 > x <- seq(0.001, 5, , 200)
8213 > lam0 <- .1
8214 > sigma <- 1
8215 > lam <- lam0*exp(-x**/(2*sigma*sigma))
8216
8217 > par(mfrow=c(1,2))
8218 > p1 <- 1-exp(-lam)
8219 > plot(x, lam, ylab="E[y] or Pr(y>0)", xlab="distance", type="l", lwd=2)
8220 > lines(x,p1,lwd=2,col="red")
8221 > plot(lam, p1, xlab="E[y]", ylab="Pr(y>0)", type="l", lwd=2)
8222 > abline(0,1,col="red")
```

8223 To summarize, if y is Poisson then, as λ gets small,

$$\begin{aligned} \Pr(y > 0) &\approx \mathbb{E}(y) \\ 1 - \exp(-\lambda_0 k(\mathbf{x}, \mathbf{s})) &\approx \lambda_0 k(\mathbf{x}, \mathbf{s}) \end{aligned} \quad (9.1.3)$$

8224 What all of this suggests it that if we have very few observations > 1 in our SCR data
 8225 set, then we won’t lose much information by using the Bernoulli model. On the other
 8226 hand, the Poisson model may have some advantages in terms of analytic or numerical
 8227 tractability in some cases. Further, this approximation explains the close correspondence
 8228 we have found between these two versions of the Gaussian encounter probability model
 8229 (Sec. 5.4). Namely, the Gaussian hazard model and the Gaussian encounter probability
 8230 model are close approximations because $1 - \exp(-\lambda) \approx \lambda$ if λ is small.

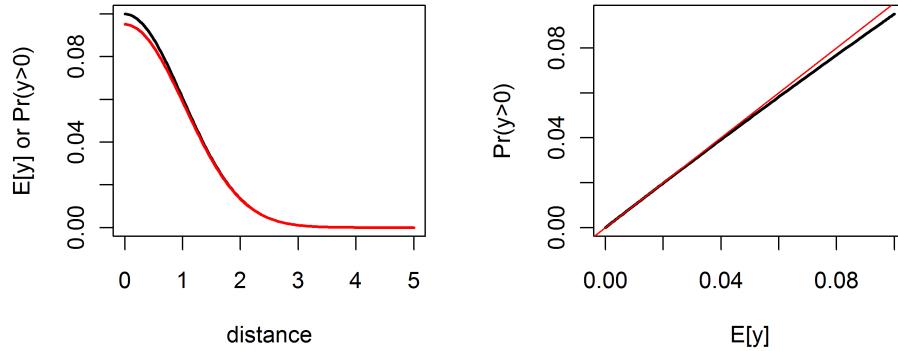


Figure 9.1. Poisson approximation to the binomial. As the Poisson mean approaches 0, then $\Pr(y > 0)$ under the Poisson model approaches λ and therefore $y \sim \text{Poisson}(\lambda)$ is well-approximated by a Bernoulli model with parameter λ .

Even in such cases where the Poisson and Bernoulli models are not quite equivalent, we might choose to truncate individual encounter frequencies to binary observations anyhow (transforming counts to 0/1 is called “quantizing”). We might do this intentionally in some cases, such as when the distinct encounter events are highly dependent as often happens in camera trap studies when the same individual moves back-and-forth in front of a camera during a short period of time. But sometimes, truncation is a feature of the sampling. For example, in the case of bear hair snares, the number of encounters might be well approximated by a Poisson distribution but we cannot determine unique visits and so only get to observe the binary event “ $y > 0$ ”. In this case, we might choose to model the encounter probability for the binary encounter using Eq. 9.1.4. This is equivalent to the complementary log-log link model, or the “Gaussian hazard” as we called it in Chapt. 5:

$$\text{cloglog}(p_{ij}) = \log(\lambda_0) + \log(k(\mathbf{x}, \mathbf{s}))$$

where $\text{cloglog}(u) = \log(-\log(1 - u))$.

9.1.3 A cautionary note on modeling encounter frequencies

Other models for counts might be appropriate. For example, ecologists are especially fond of negative binomial models for count data (Ver Hoef and Boveng, 2007; White and Bennetts, 1996; Kéry et al., 2005) but other models for excess-Poisson variation are possible. For example, we might add a normally distributed random effect to the linear predictor (Coull and Agresti, 1999).

As a general rule we favor the Bernoulli observation model even if our sampling scheme

8251 produces encounter frequencies. The main reason is that, with frequency data, we are
 8252 forced to confront a model choice problem (i.e., Poisson, negative binomial, log-normal
 8253 mixture) that is wholly unrelated to the fundamental space usage process that underlies
 8254 the genesis of many types of SCR data. Repeated encounters over short time intervals are
 8255 not likely to be the result of independent encounter events. E.g., an individual moving back
 8256 and forth in front of a camera yields a cluster of observations that is not informative about
 8257 the underlying spatial structure of the population. Similarly in scat surveys dogs are used
 8258 to locate scats which are processed in the lab for individuality (Kohn et al., 1999; MacKay
 8259 et al., 2008; Thompson et al., 2012). The process of local scat deposition is not strictly
 8260 the outcome of movement or space usage but rather the outcome of complex behavioral
 8261 considerations as well as dependence in detection of scat by dogs. For example, dogs find
 8262 (or smell) one scat and then are more likely to find one or more nearby ones, if present, or
 8263 they get into a den or latrine area and find many scats. The additional assumption required
 8264 to model variation in observed frequencies (i.e., conditional on location) provides relatively
 8265 no information about space usage and density, and we feel that the model selection issue
 8266 should therefore be avoided.

8267 To elaborate on this, we suppose that an individual with activity center \mathbf{s} visits
 8268 a particular pixel \mathbf{x} with some probability $p(\mathbf{x}, \mathbf{s})$, and then, once there, deposits a
 8269 number of scat, or visits a camera some number of times with frequency $y(\mathbf{x}, \mathbf{s}) \geq 0$.
 8270 We describe the outcome of this movement/usage process with a two-level hierarchical
 8271 model of the form: $[y|w][w|p(\mathbf{x}, \mathbf{s})]$ where $w(\mathbf{x}, \mathbf{s})$ is a binary variable that indicates
 8272 whether the individual with activity center \mathbf{s} used pixel \mathbf{x} during some interval, and let
 8273 $w(\mathbf{x}, \mathbf{s}) \sim \text{Bernoulli}(p(\mathbf{x}, \mathbf{s}))$. If we suppose encounter frequency y is independent of \mathbf{x} and
 8274 \mathbf{s} conditional on the use variable w , then we see that the model for y (amount of use) does
 8275 not depend on \mathbf{s} .

8276 9.1.4 Analysis of the Poisson SCR model in BUGS

8277 We consider the simplest possible model here in which we have no covariates that vary
 8278 over sample occasions $k = 1, 2, \dots, K$ so that we work with the aggregated individual-
 8279 and trap-specific encounters:

$$y_{ij} = (\sum_{k=1}^K y_{ijk}) = \text{Poisson}(K\lambda_0 k(\mathbf{x}_j, \mathbf{s}_i))$$

8280 and we consider the bivariate normal form of $k(\mathbf{x}, \mathbf{s})$:

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

8281 so that

$$\log(\lambda_{ij}) = \alpha_0 - \alpha_1 d(\mathbf{x}_j, \mathbf{s}_i)^2$$

8282 where $\alpha_0 = \log(\lambda_0)$ and $\alpha_1 = 1/(2\sigma^2)$.

8283 As usual, we approach Bayesian analysis of these models using data augmentation
 8284 (Sec. 4.2). Under data augmentation, we introduce a collection of all-zero encounter
 8285 histories to bring the total size of the data set up to M , and a corresponding set of data
 8286 augmentation variables $z_i \sim \text{Bern}(\psi)$. Then the observation model is specified conditional
 8287 on z according to:

$$y_{ij} \sim \text{Poisson}(z_i K \lambda_{ij})$$

which evaluates to a point mass at $y = 0$ if $z = 0$. In other words, the observation model under data augmentation is a zero-inflated Poisson model which is easily analyzed by Bayesian methods, e.g., in one of the **BUGS** dialects or, alternatively, using likelihood methods, which we neglect here although the same principles as in Chapt. 6 apply.

9.1.5 Simulating data and fitting the model

Simulating a sample SCR data set under the Poisson model requires only a couple minor modifications to the procedure we used in Chapt. 5 (see the function `simSCR0`). In particular, we modify the block of code which defines the model to be that of $E(y)$ and not $\Pr(y = 1)$, and we change the random variable generator from `rbinom` to `rpois`:

```
8297 ##  
8298 ## S =activity centers and traplocs defined as in simSCR0()  
8299 ##  
8300 ## Compute distance between activity centers and traps:  
8301 > D <- e2dist(S,traplocs)  
8302  
8303 ## Define parameter values:  
8304 > alpha0 <- -2.5  
8305 > sigma <- 0.5  
8306 > alpha1 <- 1/(2*sigma*sigma)  
8307  
8308 ## Encounter probability model:  
8309 > muy <- exp(alpha0)*exp(-alpha1*D*D)  
8310  
8311 ## Now generate the encounters of every individual in every trap  
8312 > Y <-matrix(NA,nrow=N,ncol=ntraps)  
8313 > for(i in 1:nrow(Y)){  
8314   Y[i,] <- rpois(ntraps,K*muy[i,])  
8315 }
```

We modified our simulation code from Chapt. 5 to simulate Poisson encounter frequencies for each trap and then we analyze an ideal data set using **BUGS**. This Poisson simulator function `simPoissonSCR` is available in the `scrbook` package (it can produce 3-d encounter history data too, although we don't do that here). Here is an example of simulating a data set and harvesting the required data objects, and doing the data augmentation:

```
8322 ## Simulate data and extract data elemements  
8323 ##  
8324 > data <- simPoissonSCR(discard0=TRUE,rnd=2013)  
8325 > y <- data$Y  
8326 > nind <- nrow(y)  
8327 > X <- data$traplocs  
8328 > K <- data$K  
8329 > J <- nrow(X)
```

```

8330 > xlim <- data$xlim
8331 > ylim <- data$ylim
8332
8333 ## Data augmentation
8334 > M <- 200
8335 > y <- rbind(y,matrix(0,nrow=M-nind,ncol=ncol(y)))
8336 > z <- c(rep(1,nind),rep(0,M-nind))

```

8337 The process for fitting the model in **WinBUGS** or **JAGS** is identical to what we've
 8338 done previously in Chapt. 5. In particular, we set up some starting values, package
 8339 the data and inits, identify the parameters to be monitored, and then send everything
 8340 off to our MCMC engine. Here it all is for fitting the Poisson observation model (these
 8341 commands are shown in the help file for `simPoissonSCR`):

```

8342 ## Starting values for activity centers
8343 ##
8344 > sst <- X[sample(1:J,M,replace=TRUE),]
8345 > for(i in 1:nind){
8346   if(sum(y[i,])==0) next
8347   sst[i,1] <- mean( X[y[i,>0,1] )
8348   sst[i,2] <- mean( X[y[i,>0,2] )
8349 }
8350 ## Dithered a little bit from trap locations
8351 > sst <- sst + runif(nrow(sst)*2,0,1)/8
8352 > data <- list (y=y,X=X,K=K,M=M,J=J,xlim=xlim,ylim=ylim)
8353 > inits <- function(){
8354   list (alpha0=rnorm(1,-2,.4),alpha1=runif(1,1,2),s=sst,z=z,psi=.5)
8355 }
8356 > parameters <- c("alpha0","alpha1","N","D")

```

8357 Next, we write the **BUGS** model to an external file:

```

8358 > cat("
8359 model{
8360   alpha0 ~ dnorm(0,.1)
8361   alpha1 ~ dnorm(0,.1)
8362   psi ~ dunif(0,1)
8363
8364   for(i in 1:M){
8365     z[i] ~ dbern(psi)
8366     s[i,1] ~ dunif(xlim[1],xlim[2])
8367     s[i,2] ~ dunif(ylim[1],ylim[2])
8368     for(j in 1:J){
8369       d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
8370       y[i,j] ~ dpois(lam[i,j])
8371       lam[i,j] <- z[i]*K*exp(alpha0)*exp(- alpha1*d[i,j]*d[i,j])
8372     }
8373   }

```

```

8374   N <- sum(z[])
8375   D <- N/64
8376 }
8377 ",file = "SCR-Poisson.txt")

```

8378 To fit the model we execute **bugs** in the usual way:

```

8379 > library(R2WinBUGS)
8380 > out1 <- bugs (data, inits, parameters, "SCR-Poisson.txt", n.thin=1,
8381           n.chains=3,n.burnin=1000,n.iter=2000,working.dir=getwd(),
8382           debug=TRUE)

```

8383 Or, using **JAGS** via **rjags** we would do something like this:

```

8384 > library(rjags)
8385 > jm <- jags.model("SCR-Poisson.txt", data=data, inits=inits,
8386   n.chains=3, n.adapt=1000)
8387 > out2 <- coda.samples(jm, parameters, n.iter=1000, thin=1)

```

8388 Summarizing the output from the **WinBUGS** run produces the following:

```

8389 > print(out1,digits=2)
8390 Inference for Bugs model at "SCR-Poisson.txt", fit using WinBUGS,
8391 3 chains, each with 2000 iterations (first 1000 discarded)
8392 n.sims = 3000 iterations saved
8393        mean    sd   2.5%   25%   50%   75% 97.5% Rhat n.eff
8394 alpha0   -2.57  0.19  -2.95  -2.69  -2.57  -2.44  -2.19 1.00  2600
8395 alpha1    2.34  0.36   1.69   2.08   2.32   2.57   3.12 1.00  3000
8396 N       114.13 15.25  87.97 103.00 113.00 124.00 147.00 1.01  370
8397 D       1.78  0.24   1.37   1.61   1.77   1.94   2.30 1.01  370
8398 deviance 329.95 21.92 290.00 314.20 329.50 344.40 375.80 1.00 1700
8399 ...
8400 [...some output deleted..]
8401 ...

```

8402 9.1.6 Analysis of the wolverine study data

8403 We reanalyzed the data from the wolverine camera trapping study that were first introduced in Sec. 5.9. We modified the **R** script from the function **wolvSCR0** to fit the Poisson model (see the help file for **wolvSCR0pois**). Executing this function produces the results shown in Table 9.1. The results are almost indistinguishable from the Bernoulli model fitted previously, where we had a posterior mean for N of 59.84 and σ was 0.64. You can edit the script **wolvSCR0pois** to obtain more posterior samples, or modify the model in some way.

Table 9.1. Results of fitting the SCR model with Poisson encounter frequencies to the wolverine camera trapping data. Posterior summaries were obtained using **WinBUGS** with 3 chains, each with 6000 iterations, discarding the first 1000 as burn-in, to yield a total of 15000 posterior samples.

Parameter	Mean	SD	2.5%	50%	97.5%	Rhat	n.eff
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

8410 9.1.7 Count detector models in the secr package

8411 The **R** package **secr** will fit Poisson or negative binomial encounter frequency models.
 8412 The formatting of data and structure of the analysis proceeds in a similar fashion to the
 8413 Bernoulli model described in Sec. 6.5, except that we specify the `detector='count'`
 8414 option when the traps object is created. The set-up proceeds as follows:

```
8415 > library(secr)
8416 > library(scrbook)
8417 > data(wolverine)
8418
8419 > traps <- as.matrix(wolverine$wtraps)
8420 > dimnames(traps) <- list(NULL,c("trapID","x","y",paste("day",1:165,sep="")))
8421 > traps1 <- as.data.frame(traps[,1:3])
8422 > trapfile1 <- read.traps(data=traps1,detector="count")
```

8423 You can proceed with analysis of these data and compare/contrast with the Bayesian
 8424 analysis given above, or the results of the Bernoulli model fitted in Chapt. 6.

9.2 INDEPENDENT MULTINOMIAL OBSERVATIONS

8425 Several types of encounter devices yield multinomial observations in which an individual
 8426 can be caught in a single trap during a particular encounter occasion, but traps might
 8427 catch any number of individuals. Mist netting is the canonical example of such a “multi-
 8428 catch” device (Efford et al., 2009a). Also some kinds of bird or mammal cage-traps
 8429 hold multiple animals, as do pit-fall traps which are commonly used for many species of
 8430 herptiles. Another type of sample method that might be viewed (in some cases) as a
 8431 multi-catch device are area-searches of, for example, reptiles where we think of a small
 8432 polygon as the “trap” – we could get multiple individuals (turtles, lizards) in the same
 8433 plot but not, in the same sample occasion, at different plots. The key features of this
 8434 independent multinomial or multi-catch model are: (1) capture of an individual in a trap
 8435 is *not* independent of its capture in other traps, because initial capture precludes capture
 8436 in any other trap and (2) individuals behave independently of one another, so whether a
 8437 trap captures some individual doesn’t have an affect on whether it captures another. A

8438 type of model in which the 2nd assumption is violated are the “single catch” trap systems
 8439 which we address in Sec. 9.3 below.

8440 In this case we assume the observation \mathbf{y}_{ik} for individual i during sample occasion k is
 8441 a multinomial observation which consists of a sequence of 0’s and a single 1 indicating the
 8442 trap of capture, or “not captured”. For the “not captured” event we define an additional
 8443 outcome, by convention element $J + 1$ of the vector. As an example, if we capture an
 8444 individual in trap 2 during some occasion of a study involving $J = 6$ traps. Then, the
 8445 multinomial observation has length $J+1 = 7$, and the observation is $\mathbf{y}_i = (0, 1, 0, 0, 0, 0, 0)$.
 8446 An individual not captured at all would have the observation vector $(0, 0, 0, 0, 0, 0, 1)$. If
 8447 we sample for 5 occasions in all and the individual is also caught in trap 4 during occasion
 8448 3, but otherwise uncaptured, then the 5 encounter observations for that individual are as
 8449 follows:

8450	occassion	trap						"not captured"
		1	2	3	4	5	6	
8451		-----	-----	-----	-----	-----	-----	-----
8452		-----	-----	-----	-----	-----	-----	-----
8453	1	0	1	0	0	0	0	0
8454	2	0	0	0	0	0	0	1
8455	3	0	0	0	1	0	0	0
8456	4	0	0	0	0	0	0	1
8457	5	0	0	0	0	0	0	1

8458 Statistically we regard the *rows* of this data matrix as *independent* multinomial trials.

8459 Analogous to our previous Bernoulli and Poisson models, we seek to construct the
 8460 multinomial cell probabilities for each individual, as a function of *where* that individual
 8461 lives, through its center of activity \mathbf{s} . Thus we suppose that

$$\mathbf{y}_{ik} | \mathbf{s}_i \sim \text{Multinomial}(1, \boldsymbol{\pi}(\mathbf{s}_i)) \quad (9.2.1)$$

8462 where $\boldsymbol{\pi}(\mathbf{s}_i)$ is a vector of length $J + 1$, where $\pi_{i,J+1}$, the last cell, corresponds to the
 8463 probability of the event “not captured”. Now we have to construct these cell probabili-
 8464 ties in some meaningful way that depends on each individual’s \mathbf{s} . We use the standard
 8465 multinomial logit with distance as a covariate:

$$\pi_{ij} = \frac{\exp(\alpha_0 - \alpha_1 d_{ij})}{1 + \sum_j \exp(\alpha_0 - \alpha_1 d_{ij})}$$

8466 for $j = 1, 2, \dots, J$ and, for $J + 1$, i.e., “not captured”,

$$\pi_{i,(J+1)} = \frac{\exp(0)}{1 + \sum_j \exp(\alpha_0 - \alpha_1 d_{ij})}$$

8467 or, more commonly, we use d_{ij}^2 to correspond to our Gaussian kernel model for encounter
 8468 probability. Whatever function of distance we use in the construction of multinomial prob-
 8469 abilities will have a direct correspondence to the standard encounter probability models
 8470 we used in the Bernoulli or Poisson models as well (see Sec. 5.4).

8471 It is convenient to express these multinomial models short-hand as follows, e.g., for
 8472 the Gaussian encounter probability model:

$$\text{mlogit}(\pi_{ij}) = \alpha_0 - \alpha_1 d_{ij}^2$$

8473 In this way we can refer to models with covariates in a more concise way. For example, a
 8474 model with a trap-specific covariate, say C_j , is:

$$\text{mlogit}(\pi_{ij}) = \alpha_0 - \alpha_1 d_{ij}^2 + \alpha_2 C_j$$

8475 or we could include occasion-specific covariates too, such as behavioral response.

8476 A statistically equivalent distribution to the multinomial is the *categorical* distribution.

8477 If \mathbf{y} is a multinomial trial with probabilities $\boldsymbol{\pi}$ than the *position* of the non-zero element of
 8478 \mathbf{y} is a categorical random variable with probabilities $\boldsymbol{\pi}$. We express this for SCR models
 8479 as

$$\mathbf{y}|\mathbf{s} \sim \text{Categorical}(\boldsymbol{\pi}(\mathbf{s}))$$

8480 In the SCR context, the categorical version of the multinomial trial corresponds to the
 8481 *trap of capture*. Using our example above with 6 traps then we could as well say y_{ik} is a
 8482 categorical random variable with possible outcomes $(1, 2, 3, 4, 5, 6, 7)$ where outcome $y = 7$
 8483 corresponds to “not captured.” Obviously, how this is organized or labeled is completely
 8484 irrelevant, although it is convenient to use the integers 1 to $(J + 1)$ where $J + 1$ is the
 8485 event not captured. Therefore, for our illustration in the previous table, $y_{i1} = 2$, $y_{i2} = 7$,
 8486 $y_{i3} = 4$ and so on.

8487 For simulating and fitting data in the **BUGS** engines we will typically use the cat-
 8488 egorical representation of the model because it is somewhat more convenient. We have
 8489 found that fitting multinomial models in **WinBUGS** is less efficient than **JAGS** (Royle
 8490 and Converse, in review), which we use in the subsequent examples involving multinomial
 8491 observation models.

8492 9.2.1 Multinomial resource selection models

8493 The multinomial probabilities in Eq. 9.2.2 look similar to the multinomial resource selec-
 8494 tion function (RSF) model for telemetry data (Manly et al., 2002; Lele and Keim, 2006).
 8495 This suggests how we might model landscape or habitat covariates using such methods
 8496 – i.e., by including them as explicit covariates in a larger multinomial model for “use” –
 8497 which, if we take the product of use with encounter, produces a model for the observable
 8498 encounter data. This leads naturally to the development of models that integrate RSF
 8499 data from telemetry studies with SCR data (Royle et al., 2012b), which is the topic of
 8500 Chapt. 13.

8501 9.2.2 Simulating data and analysis using JAGS

8502 We’re going to show the nugget of a simulation function which is used in the function
 8503 **simMnSCR** found in the **R** package **scrbook**. The first lines of the following **R** code make
 8504 use of some things that you need to define, but we omit them here (e.g., **xlim**, **ylim** are
 8505 the boundaries of the state-space, **N** is the population size, etc.):

```
8506 ##
8507 ## Simulate random activity centers:
8508 ##      (first define N, xlim, ylim, etc..)
8509 ##
8510 > S <- cbind(runif(N,xlim[1],xlim[2]),runif(N,ylim[1],ylim[2]))
```

```

8511
8512 ## Distance from each individual to each trap
8513 > D <- e2dist(S,traplocs)
8514
8515 ## Set parameter values
8516 > sigma <- 0.5
8517 > alpha0 <- -1
8518 > alpha1 <- -1/(2*sigma*sigma)
8519
8520 ## make an empty data matrix and fill it up with data
8521 > Ycat <- matrix(NA,nrow=N,ncol=K)
8522 > for(i in 1:N){
8523   for(k in 1:K){
8524     lp <- alpha0 + alpha1*D[i,]*D[i,]
8525     cp <- exp(c(lp,0))
8526     cp <- cp/sum(cp)
8527     Ycat[i,k] <- sample(1:(ntraps+1),1,prob=cp)
8528   }
8529 }
```

8530 We save the data in the matrix `Ycat` to clarify that it is the categorical observation
 8531 representing “trap of capture”. The matrix `Ycat` here has the maximal dimension N
 8532 and so, to do an analysis that mimics a real situation, we would have to discard the
 8533 uncaptured individuals. The function `simMnSCR` in the package `scrbook` will also simulate
 8534 data that includes a behavioral response which will be the typical situation in small-
 8535 mammal trapping problems (see Converse and Royle, 2012, for details).

8536 Here we use our function `simMnSCR` to simulate a data set with $K = 7$ occasions. We’ll
 8537 run the model using `JAGS` which we have found is much more effective for this class of
 8538 models. We get the data set-up for analysis by augmenting the size of the data set to
 8539 $M = 200$. In addition we choose starting values for s and the data augmentation variables
 8540 z . For starting values of s we cheat a little bit here and use the true values for the observed
 8541 individuals and then augment the $M \times 2$ matrix \mathbf{S} with $M - n$ randomly selected activity
 8542 centers. Our function `spiderplot` returns the mean observed location of individuals for
 8543 use as starting values for the `nind` encountered individuals. The parameters input to
 8544 `simMnSCR` are the intercept α_0 , $\sigma = \sqrt{1/(2\alpha_1)}$ for the Gaussian encounter probability
 8545 model, and α_2 is the behavioral response parameter. The data simulation and set-up
 8546 proceeds as follows:

```

8547 > set.seed(2013)
8548 > parms <- list(N=100,alpha0= -.40, sigma=0.5, alpha2= 0)
8549 > data <- simMnSCR(parms, K=7, ssbuff=2)
8550 > nind <- nrow(data$Ycat)
8551
8552 > M <- 200
8553 > Ycat <- rbind(data$Ycat,matrix(nrow(data$X)+1,nrow=(M-nind),ncol=data$K))
8554 > Sst <- rbind(data$S,cbind(runif(M-nind,data$xlim[1],data$xlim[2]),
8555                           runif(M-nind,data$ylim[1],data$ylim[2])))

```

```
8556 > zst <- c(rep(1,160),rep(0,40))
```

8557 The model specification is not much more complicated than the binomial or Poisson
 8558 models given previously. The main consideration is that we define the cell probabilities for
 8559 each trap $j = 1, 2, \dots, J$ and then define the last cell probability, $J+1$, for “not captured”,
 8560 to be the complement of the sum of the others. The code is shown in Panel 9.1. In the
 8561 last lines of code here we specify N and density, D , as derived parameters.

8562 To fit the model, we need to package everything up (inits, parameters, data) and send
 8563 it off to **JAGS** to build an MCMC simulator for us (these commands are executed in
 8564 the help file for **simMnSCR**). In addition to the usual data objects, we also pass the limits
 8565 of the assumed rectangular state-space (**ylim**, **xlim**, both 1×2 vectors) and the scale of
 8566 the standardized units, called **trap.space** here because we typically will define the trap
 8567 coordinates to be an integer grid. If the trap spacing is 10 m and we want units of density
 8568 computed in terms of individuals per meter-squared, then we input **trap.space=10**. The
 8569 analysis is carried out as follows:

```
8570 > inits <- function(){ list (z=zst,sigma=rnorm(1,.5,1) ,S=Sst) }  

8571  

8572 # Parameters to monitor  

8573 > parameters <- c("psi","alpha0","alpha1","sigma","N","D")  

8574  

8575 # Bundle the data. Note this reuses "data"  

8576 > data <- list (X=data$X,K=data$K, trap.space=1,Ycat=Ycat,M=M,  

8577   ntraps=nrow(data$X),ylim=data$ylim,xlim=data$xlim)  

8578  

8579 > library(R2jags)  

8580 > out <- jags (data, inits, parameters, "model.txt", n.thin=1,  

8581   n.chains=3, n.burnin=1000, n.iter=2000)
```

8582 The posterior summaries are provided in the following **R** output (recall that $N = 100$,
 8583 $\alpha_0 = -0.40$, and $\sigma = 0.5$):

```
8584 > out  

8585 Inference for Bugs model at "model.txt", fit using jags,  

8586 3 chains, each with 2000 iterations (first 1000 discarded)  

8587 n.sims = 3000 iterations saved  

8588    mu.vect sd.vect 2.5%   25%   50%   75% 97.5% Rhat n.eff  

8589 D        1.873  0.189  1.531  1.750  1.859  2.000  2.250 1.006 1300  

8590 N       119.867 12.107 98.000 112.000 119.000 128.000 144.000 1.006 1300  

8591 alpha0   -0.435  0.151 -0.738 -0.535 -0.439 -0.331 -0.146 1.004  580  

8592 alpha1    2.195  0.286  1.658  2.004  2.180  2.372  2.785 1.003 2400  

8593 psi      0.599  0.069  0.465  0.552  0.599  0.645  0.739 1.006 1400  

8594 sigma     0.480  0.032  0.424  0.459  0.479  0.500  0.549 1.003 2400  

8595 deviance 892.164 21.988 850.922 877.417 891.561 906.246 937.728 1.003  950  

8596  

8597 [... output deleted ....]
```

```

model{
psi ~ dunif(0,1)
alpha0 ~ dnorm(0,10)
sigma ~ dunif(0,10)
alpha1 <- 1/(2*sigma*sigma)

for(i in 1:M){
  z[i] ~ dbern(psi)
  S[i,1] ~ dunif(xlim[1],xlim[2])
  S[i,2] ~ dunif(ylim[1],ylim[2])
  for(j in 1:ntraps){
    #distance from capture to the center of the home range
    d[i,j] <- pow(pow(S[i,1]-X[j,1],2) + pow(S[i,2]-X[j,2],2),1)
  }
  for(k in 1:K){
    for(j in 1:ntraps){
      lp[i,k,j] <- exp(alpha0 - alpha1*d[i,j])*z[i]
      cp[i,k,j] <- lp[i,k,j]/(1+sum(lp[i,k,]))
    }
    cp[i,k,ntraps+1] <- 1-sum(cp[i,k,1:ntraps]) # last cell = not captured
    Ycat[i,k] ~ dcat(cp[i,k,])
  }
}
N <- sum(z[1:M])
A <- ((xlim[2]-xlim[1])*trap.space)*((ylim[2]-ylim[1])*trap.space)
D <- N/A
}

```

Panel 9.1: **BUGS** model specification for the independent multinomial observation model. For data simulation and model fitting see the help file `?simMnSCR` in the **R** package `scrbook`.

9.2.3 Multinomial relationship to the Poisson

8598 The multinomial is related to the Poisson encounter rate model by a conditioning argument.
 8599 Let y_{ij} be the number of encounters for individual i in trap j . If $y_{ij} \sim \text{Poisson}(\lambda_{ij})$,
 8600 then, conditional on the *total* number of captures (i.e., across all traps), $y_i = \sum_j y_{ij}$, the
 8601 trap encounter frequencies are multinomial with probabilities
 8602

$$\pi_{ij} = \frac{\lambda_{ij}}{\sum_j \lambda_{ij}}$$

8603 for $j = 1, 2, \dots, J$. Or equivalently the *trap of capture* is categorical with probabilities π_{ij}
 8604 as given above. Under the Gaussian kernel model, these probabilities are:

$$\pi_{ij} = \frac{\exp(-\alpha_1 d(\mathbf{x}_i, \mathbf{s}_i)^2)}{\sum_j \exp(-\alpha_1 d(\mathbf{x}_i, \mathbf{s}_j)^2)} \quad (9.2.2)$$

8605 where, we note, the intercept α_0 has canceled from both the numerator and denominator.
 8606 This makes sense because, here, these probabilities describe the trap-specific capture prob-
 8607 abilities *conditional on capture*. Therefore, the model is not completely specified, absent
 8608 a model for the “overall” probability of encounter or the expected frequency of captures,
 8609 say ϕ_i . Depending on how we specify a model for this quantity ϕ_i , we can reconcile it
 8610 directly with the Poisson model. Let y_i be the total number of encounters for individual
 8611 i and suppose y_i has a Poisson distribution with mean ϕ_i . Then, marginalizing Eq. 9.2.1
 8612 over the Poisson distribution for y_i produces the original set of *iid* Poisson frequencies
 8613 with probabilities:

$$\lambda_{ij} = \phi_i \pi_{ij}$$

8614 for $j = 1, 2, \dots, J$. In particular, if we suppose that $\phi_i = \sum_j \exp(\alpha_0 - \alpha_1 d(\mathbf{x}, \mathbf{s})^2)$ then
 8615 the marginal distribution of y_{ij} is Poisson with mean $\exp(\alpha_0 - \alpha_1 d(\mathbf{x}, \mathbf{s})^2)$, equivalent to
 8616 Eq. 9.1.1.

8617 In summary, the Poisson and multinomial models are equivalent in how they model
 8618 the distribution of captures among traps. It stands to reason that, if the encounter
 8619 rate of individuals is low, we could use the Poisson and multinomial models interchange-
 8620 ably. In fact, based on our discussion in Sec. 9.1.2 above we could use any of the bino-
 8621 mial/Poisson/multinomial models with little ill-effect when encounter rate is low.

9.2.4 Avian mist-netting example

8622 We analyze data from a mist-netting study of ovenbirds, conducted at the Patuxent
 8623 Wildlife Research Center, Laurel MD, by D.K. Dawson and M.G. Efford. The data from
 8624 this study are available in the **secr** package, and have been analyzed previously by Efford
 8625 et al. (2004), see also Borchers and Efford (2008). Forty-four mist nets spaced 30 m apart
 8626 on the perimeter of a 600-m x 100-m rectangle were operated on 9 or 10 non-consecutive
 8627 days in late May and June for 5 years from 2005-2009. The ovenbird data can be loaded
 8628 as follows:

```
8630 > library(secr)
8631 > data(ovenbird)
```

8632 The data set consists of adult ovenbirds caught during sampling in each of 5 years, 2005-
8633 2009. (one ovenbird was killed in 2009, indicated by a negative net number in the encounter
8634 data file). As with most mist-netting studies, nets are checked multiple times during a
8635 day (e.g., every hour during a morning session). However, for this data set, the within-day
8636 recaptures are not included so each bird has at most a single capture per day. Therefore
8637 the multinomial model (detector type ‘multi’ in **secr**) is appropriate. Although several
8638 individuals were captured in more than one year, this information is not used in the models
8639 presently offered in **secr**, but we do make use of it in the development of open models in
8640 Chapt. 16.

8641 **Multiple sample sessions**

8642 Up to this point we have only dealt with a basic closed population sampling situation
8643 consisting of repeated sample occasions on a single population of individuals using a single
8644 array of traps. In practice, many studies produce repeated samples over longer periods
8645 of time over which demographic closure isn’t valid, or at different locations where the
8646 populations are completely distinct. We adopt the **secr** terminology of *session* for such
8647 replication by groups of time or space, and the models are *multi-session* models, although
8648 we think of such models as being relevant to any stratified population (see Chapt. 14).
8649 We introduced **secr**’s multi-session models in Sec. 6.5.4. In the case of the ovenbird data,
8650 sampling was carried out in multiple years, with a number of sample occasions within
8651 each year (9 or 10), a type of data structure commonly referred to as “the robust design”
8652 (Pollock, 1982). In this context, it stands to reason that there is recruitment and mortality
8653 happening across years. In Chapt. 16 we model these processes explicitly but, here, we
8654 provide an analysis of the data that does not require explicit models for recruitment and
8655 survival, regarding the yearly populations as independent strata, and fitting a multi-session
8656 model.

8657 When the sessions represent explicit time periods, the multi-session model of **secr** can
8658 be thought of as a type of open population model. In particular, a special case of open
8659 models arises when we assume N_t (time-specific population sizes) are independent from
8660 one time period or session to the next – this can be thought of as a “random temporary
8661 emigration” model of the Kendall et al. (1997) variety, and this is the multi-session model
8662 implemented in **secr**. In particular, by assuming that N_t is Poisson with mean Λ_t , one can
8663 model variation in abundance among sessions based on the Poisson-integrated likelihood
8664 in which parameters of Λ_t appear directly in the likelihood as we noted in Sec. 6.5.4.
8665 We provide an analysis (below) of the ovenbird data here using the multi-session models
8666 in **secr**. We formalize the multi-session model approach from a Bayesian perspective
8667 using data augmentation in Chapt. 14 (Converse and Royle, 2012; Royle and Converse,
8668 in review).

8669 A 3rd way to develop models for stratified or grouped populations, not based on
8670 multi-session models, but that is convenient in **BUGS**, is to regard the data from each
8671 session as an independent data set with its own N_t parameter, and do T distinct data
8672 augmentations. Because each N_t is regarded as a free parameter, independent of the
8673 other parameters, we’ll call this the nonparametric multi-session model to distinguish it
8674 from the multi-session model which assumes the N_t are related to one another by having
8675 been generated from a common Poisson distribution. We can analyze this model in the
8676 normal context of data augmentation by augmenting each year separately in the same
8677 **BUGS** model specification. This approach avoids making explicit model assumptions

about the N_t parameters. This is distinct from the model implemented in **secr** in that **secr** is removing the N_t parameters by integrating the conditional-on- N_t likelihood over the Poisson prior for N_t ¹

We demonstrate these 3 approaches to analyzing grouped/stratified data using the ovenbird data: (1) In the following section, we provide the nonparametric multi-session model with unconstrained N_t ; (2) we demonstrate the Poisson model-based multi-session models from **secr** both here (following section) and in Chapt. 14 from a Bayesian standpoint; (3) later, in Chapt. 16, we provide a fully dynamic “spatial Jolly-Seber” model and apply it to the ovenbird data.

Analysis in JAGS

The ovenbird data are provided as a multi-session **capthist** object **ovenCH** which, by regarding years as independent strata, or sessions, allows for the fitting of the multi-session model. For doing a Bayesian analysis in one of the **BUGS** engines (we use **JAGS** here) there are a number of ways to structure the data and describe the model. We can analyze either a 2-d data set with all years (data augmented) “stacked” into a data set of dimension $(5 * M) \times 10$ (5 years, M = size of the augmented data set, K = 10 replicate sample occasions). Or, we could produce a 3-d array $(M \times J \times K)$. We adopted the former approach, analyzing the data as a 2-d array and creating an additional categorical variable for “year” to indicate which stratum (year) each record goes with.

Data on individual sex is included with **secr**, but we provide an analysis of a single model for all adults, constant σ across years, constant p_0 , and year-specific values of N_t (and hence D_t). There is a habitat mask provided with the data but the mask appears to just be a modified rectangle around the net locations, clipped to have rounded corners, and so we don’t use it here. Instead, we used a rectangular state-space buffer of 200 meters for our analysis. There was a single loss-on-capture which we accounted for by fixing $p = 0$ for all subsequent encounters of that individual (indicated by the binary variable **dead**, as shown in Panel 9.2). We have an **R** script in **scrbook** package called **SCRovenbird**, so you can see how to set-up the data and run the model. Executing the script **SCRovenbird** produces the posterior summaries given in Table 9.2. Here, density is in units of birds per ha. The posterior mean of σ is about 76 meters, and there is considerable variability in density over the 5 year period with density peaking at 1.2 birds/ha in year 3, although there is considerable posterior uncertainty. The R-hat’s look a little bit peaked and so we might consider running the MCMC analysis longer.

Analysis in secr

Included with the ovenbird data are a number of models fitted as examples. Those include:

```
8713 ovenbird.model.1    fitted secr model -- null
8714 ovenbird.model.1b   fitted secr model -- g0 net shyness
8715 ovenbird.model.1T   fitted secr model -- g0 time trend within years
8716 ovenbird.model.h2   fitted secr model -- g0 finite mixture
8717 ovenbird.model.D    fitted secr model -- trend in density across years
```

¹We do not know of **secr** documentation that states this (or contradicts it). We think this is what is being done, based partially on conversations or emails with M.G. Efford, D.L. Borchers, the various publications on **secr**, and our own thinking about it.

```

model{
  alpha0 ~ dnorm(0,.1)
  sigma ~ dunif(0,200)
  alpha1 <- 1/(2*sigma*sigma)

  A <- ((xlim[2]-xlim[1]))*((ylim[2]-ylim[1]))
  for(t in 1:5){
    N[t] <- inprod(z[1:bigM],yrdummy[,t])
    D[t] <- (N[t]/A)*10000 # Put in units of per ha
    psi[t] ~ dunif(0,1)
  }

  for(i in 1:bigM){ # bigM = total size of jointly augmented data set
    z[i] ~ dbern(psi[year[i]])
    S[i,1] ~ dunif(xlim[1],xlim[2])
    S[i,2] ~ dunif(ylim[1],ylim[2])

    for(j in 1:ntraps){ # X = trap locations, S = activity centers
      d2[i,j] <- pow(pow(S[i,1]-X[j,1],2) + pow(S[i,2]-X[j,2],2),1)
    }
    for(k in 1:K){
      Ycat[i,k] ~ dcat(cp[i,k,])
      for(j in 1:ntraps){
        lp[i,k,j] <- exp(alpha0 - alpha1*d2[i,j])*z[i]*(1-dead[i,k])
        cp[i,k,j] <- lp[i,k,j]/(1+sum(lp[i,k,1:ntraps]))
      }
      cp[i,k,ntraps+1] <- 1-sum(cp[i,k,1:ntraps]) # Last cell = not captured
    }
  }
}

```

Panel 9.2: **BUGS** model specification for the non-parametric multi-session model in which each N_t is independent of the other. The implied prior (by data augmentation) is that $N_t \sim \text{Uniform}(0, 100)$. To fit this model to the ovenbird data, see `?SCRovenbird` in the **R** package `scrbook`.

Table 9.2. Posterior summary statistics for the ovenbird mist-netting data based on the independent multinomial (“multi-catch”) encounter process model. Parameters ψ , N and D are indexed by year. MCMC was done using jags with 3 chains, each with 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

8718 The model fit objects provided in `secr` are based on the use of the habitat mask.
 8719 To make the analyses consistent with our previous analysis in **JAGS**, we refit all of the
 8720 models here without the habitat mask. The re-analysis proceeds as follows, changing the
 8721 “trend in density across years” model to allow for year-specific density:

```
8722 ## Fit constant-density model
8723 > ovenbird.model.1 <- secr.fit(ovenCH)
8724 ## Fit net avoidance model
8725 > ovenbird.model.1b <- secr.fit(ovenCH, model = list(g0 ~ b))
8726 ## Fit model with time trend in detection
8727 > ovenbird.model.1T <- secr.fit(ovenCH, model = list(g0 ~ T))
8728 ## Fit model with 2-class mixture for g0
8729 > ovenbird.model.h2 <- secr.fit(ovenCH, model = list(g0 ~ h2))
8730 ## Fit a model with session (year)-specific Density
8731 > ovenbird.model.DT <- secr.fit(ovenCH, model = list(D ~ session))
```

8732 All of these can be fitted easily in **JAGS** but the model we fitted previously is roughly
 8733 equivalent to the last model, `ovenbird.model.DT`, because we allowed for year-specific
 8734 population sizes (and hence density). So, we’ll compare our results from **JAGS** to that
 8735 model. The `secr` output is extensive and so we do not reproduce it completely here. By

8736 default, it summarizes the trap information for each year, encounter information, and then
 8737 output for each year. Here is an abbreviated version for `ovenbird.model.DT`:

```

8738 > print(ovenbird.model.DT,digits=2)
8739
8740 secr.fit( capthist = ovenCH, model = list(D ~ session), buffer = 300 )
8741 secr 2.3.1, 14:46:52 23 Jan 2013
8742
8743 $`2005`
8744 Object class      traps
8745 Detector type    multi
8746 Detector number   44
8747 Average spacing   30.27273 m
8748 x-range          -50 49 m
8749 y-range          -285 285 m
8750
8751 [... deleted ...]
8752
8753      2005 2006 2007 2008 2009
8754 Occasions     9   10   10   10   10
8755 Detections    35   42   52   30   33
8756 Animals       20   22   26   19   16
8757 Detectors     44   44   44   44   44
8758
8759 Model          : D~session g0~1 sigma~1
8760 Fixed (real)   : none
8761 Detection fn   : halfnormal
8762 Distribution    : poisson
8763 N parameters   : 7
8764 Log likelihood : -1119.845
8765 AIC            : 2253.689
8766 AICc           : 2254.868
8767
8768 [... deleted ...]
```

8769 To do model selection we use the handy helper-function `AIC` as follows (output edited
 8770 to fit on the page):

```

8771 AIC (ovenbird.model.1, ovenbird.model.1b, ovenbird.model.1T,
8772      ovenbird.model.h2, ovenbird.model.DT)
8773
8774      model detectfn npar logLik   AIC   AICc   dAICc
8775 ovenbird.model.1T [edited output] 4 -1111.850 2231.700 2232.109 0.000
8776 ovenbird.model.1b      ....      4 -1117.615 2243.229 2243.637 11.528
8777 ovenbird.model.h2      ....      3 -1121.164 2248.327 2248.570 16.461
8778 ovenbird.model.1       ....      5 -1119.762 2249.524 2250.143 18.034
8779 ovenbird.model.DT     ....      7 -1119.845 2253.689 2254.868 22.759
```

8780 We see that our DT model is way down at the bottom of the list. Instead, the model with
 8781 a time-trend (within-season) in detection probability is preferred, followed by a behavioral
 8782 response. We encourage you to adapt the **JAGS** model specification for such models which
 8783 is easily done (see Chapt. 7 for many examples). We provide the summary results for the
 8784 model having $D \sim \text{session}$ as follows:

```

8785 > print(ovenbird.model.DT,digits=2)
8786
8787 secr.fit( capthist = ovenCH, model = list(D ~ session), buffer = 300 )
8788 secr 2.3.1, 14:46:52 23 Jan 2013
8789
8790 [...deleted....]
8791
8792 Fitted (real) parameters evaluated at base levels of covariates
8793
8794 session = 2005
8795      link estimate SE.estimate    lcl    ucl
8796 D      log     0.920       0.228  0.571  1.484
8797 g0     logit    0.028       0.004  0.021  0.037
8798 sigma   log    78.566      6.379 67.025 92.095
8799
8800 session = 2006
8801      link estimate SE.estimate    lcl    ucl
8802 D      log     0.963       0.238  0.598  1.553
8803 g0     logit    0.028       0.004  0.021  0.037
8804 sigma   log    78.566      6.379 67.025 92.095
8805
8806 session = 2007
8807      link estimate SE.estimate    lcl    ucl
8808 D      log     1.139       0.282  0.706  1.836
8809 g0     logit    0.028       0.004  0.021  0.037
8810 sigma   log    78.566      6.379 67.025 92.095
8811
8812 session = 2008
8813      link estimate SE.estimate    lcl    ucl
8814 D      log     0.832       0.206  0.516  1.341
8815 g0     logit    0.028       0.004  0.021  0.037
8816 sigma   log    78.566      6.379 67.025 92.095
8817
8818 session = 2009
8819      link estimate SE.estimate    lcl    ucl
8820 D      log     0.701       0.173  0.435  1.130
8821 g0     logit    0.028       0.004  0.021  0.037
8822 sigma   log    78.566      6.379 67.025 92.095

```

8823 The point estimates (MLEs) of density are uniformly lower than the Bayesian estimates
 8824 (posterior means) shown in Table 9.2. We expect some difference in this direction due

8825 to small-sample skew of the posterior. In addition, there may be slight differences due
 8826 to the fact that **secr** multi-session model assumes that the N_t have a Poisson prior, but
 8827 the implementation in **JAGS** using data augmentation is based on a binomial prior. The
 8828 estimated σ is very similar between the **JAGS** analysis and **secr**.

9.3 SINGLE-CATCH TRAPS

8829 The classical animal trapping experiment is based on a physical trap which captures a
 8830 single animal and holds that individual until subsequent molestation by a biologist. This
 8831 type of observation model – the “single-catch” trap – was the original situation considered
 8832 in the context of spatial capture-recapture by Efford (2004). Nowadays, capture-recapture
 8833 data are more often obtained by other methods (DNA from hair snares, or scat sampling,
 8834 camera traps etc...) but nevertheless the single-catch traps are still widely used in small
 8835 mammal studies (Converse et al., 2006; Converse and Royle, 2012) and other situations.

8836 The single-catch model is basically a multinomial model but one in which the number
 8837 of available traps is reduced as each individual is captured. As such, the constraints on the
 8838 joint likelihood for the sample of n encounter histories are very complicated. As a result,
 8839 at the time of this writing, there has not been a formal development of either likelihood or
 8840 Bayesian analysis of this model and applications of SCR models to single-catch systems
 8841 have used the independent multinomial model as an approximation (see below).

8842 Nevertheless, we can make some progress to describing the basic observation model
 8843 formally. In particular, if we imagine that all of the individuals captured queued up at
 8844 the beginning of the capture session to draw a number indicating their order of capture,
 8845 then there is a nice conditional structure resulting from a “removal process” operating on
 8846 the traps. The first individual captured has the multinomial observation model:

$$\mathbf{y}_1 \sim \text{Multinomial}(\boldsymbol{\pi}_1)$$

8847 whereas the 2nd individual captured also has a multinomial encounter probability model
 8848 but with the trap which captured the first individual removed. We might express this as:

$$\mathbf{y}_2 \sim \text{Multinomial}(\boldsymbol{\pi}_2)$$

8849 where

$$\pi_{2j} = \frac{(1 - y_{1j}) * \exp(\alpha_0 - \alpha_1 d_{ij}^2)}{\sum_j (1 - y_{1j}) * \exp(\alpha_0 - \alpha_1 d_{ij}^2)}$$

8850 and so on for $i = 3, 4, \dots, n$. In a certain way, this model is a type of local behavioral
 8851 response model but where the response is to other individuals being captured. Evidently,
 8852 the **order of capture** is relevant to the construction of these multinomial cell probabil-
 8853 ities. More generally, the *time* of capture of an individual in any trapping interval will
 8854 affect the encounter probability of subsequently captured individuals, but we think that
 8855 order of capture might lead to a practical approximation to the single-catch process (this
 8856 is how we simulate the data in our function **simScSCR**). In the simulation of single catch
 8857 data, we randomly ordered the population of individuals for each sample occasion, and
 8858 then cycled through them, turning off each trap if an individual was captured in it.

8859 **9.3.1 Inference for single-catch systems**

8860 For the single-catch model, we argued that the observations have a multinomial type of
 8861 observation model, but the multinomial observations have a unique conditional dependence
 8862 structure among them owing to the “removal” of traps as they fill-up with individuals.
 8863 Thus, competition for single-catch traps renders the independence assumptions for the
 8864 independent multinomial model invalid. However, as Efford et al. (2009a) noted, we
 8865 expect “bias to be small when trap saturation (the proportion of traps occupied) is low.
 8866 Trap saturation will be higher when population density is high...” relative to trap density,
 8867 or when net encounter probability is high. Efford et al. (2009a) did a limited simulation
 8868 study and found essentially no effective bias and concluded that estimators of density
 8869 from the misspecified independent multinomial model are robust to the mild dependence
 8870 induced when trap saturation is low. Naturally then, we expect that the Poisson model
 8871 could also be an effective approximation under the same set of circumstances.

8872 In the **R** package **scrbook** we provide a function for simulating data from a single-catch
 8873 system (function **simScSCR**) and fitting the misspecified model (**example(simScSCR)**) in
 8874 **JAGS** so that you can evaluate the effectiveness of this misspecified model for situations
 8875 that interest you.

8876 **9.3.2 Analysis of Efford's possum trapping data**

8877 We provide an analysis here of data from a study of brushtail possums in New Zealand.
 8878 The data are available with the **R** package **secr** (Efford et al., 2009a); see the help file
 8879 **?possum** after loading the **secr** package. Originally the data were analyzed by Efford et al.
 8880 (2005), and a detailed description of the data set is available in the help file, from which
 8881 we summarize:

8882 *Brushtail possums (*Trichosurus vulpecula*) are an unwanted invasive species in New
 8883 Zealand. Although most abundant in forests, where they occasionally exceed densities
 8884 of 15/ha, possums live wherever there are palatable food plants and shelter.*

8885 To load the possum data, execute the following commands:

```
8886 > library(secr)
8887 > data(possum)
```

8888 The study area encompasses approximately 300 ha, and 180 live traps were organized in 5
 8889 distinct grids, shown in Fig. 9.2. Each square arrangement of traps consisted of 36 traps
 8890 with a spacing of 20 m. Thus the squares are 180 m on a side. Individuals were captured,
 8891 tagged, and released over 5 days during April, 2002. A noteworthy aspect of this study is
 8892 that it involves replicated grids selected in some fashion from within a prescribed region.
 8893 From an analysis standpoint, we could adopt the use of the multi-session models which we
 8894 used previously to analyze the ovenbird data. This would be useful if we had covariates
 8895 at the trapping grid level that we wanted to model. Alternatively, we could pool the data
 8896 from all of the grids and analyze them jointly as if they were based on a single trapping
 8897 grid (with 180 traps) which is clearly a reasonable view in this case. In doing this sort of
 8898 pooling, there is an implicit assumption that N_t (t indexing trapping grid in this case) is
 8899 Poisson distributed, with constant mean (Royle, 2004a; Royle et al., 2012c) which we also
 8900 address in Chapt. 14.

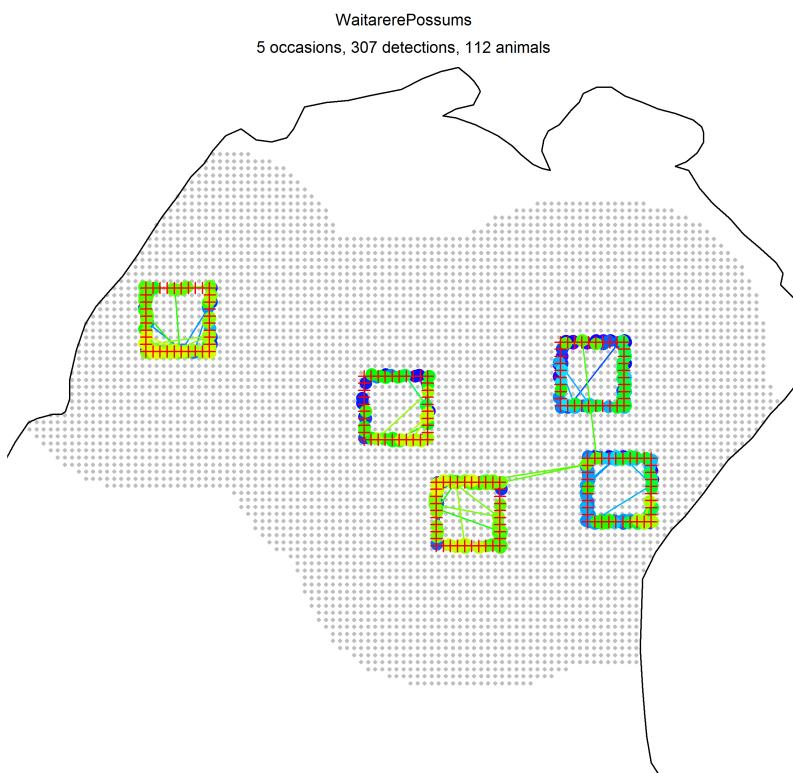


Figure 9.2. Trapping grids used in possum study from Efford et al. (2005), data are contained in the R package `secr` (Efford, 2011), refer to the help file `?possum` for additional details of this study.

8901 The data file **possumCH** contains 112 encounter histories, and we analyze those here
 8902 although the last 8 of those are recaptures treated as new individuals². The encounter
 8903 process is not strictly a single-catch multinomial process because, as noted in the **possum**
 8904 help file “One female possum was twice captured at two sites on one day, having entered
 8905 a second trap after being released; one record in each pair was selected arbitrarily and
 8906 discarded.” which is a similar situation to what might happen in bird mist net studies, as
 8907 a bird might fly into a net upon release from another. By discarding the two extra-capture
 8908 events, we can satisfactorily view these data as single-catch data, for which **secr** uses the
 8909 independent multinomial likelihood (M. Efford, pers. comm.). If multiple, same-session
 8910 captures were common, then it might be worth developing a model for n_{ik} = the number
 8911 of captures of individual i during sample occasion k , in order to make use of all captures.
 8912

8913 For our Bayesian analysis here, we used a rectangular state-space which doesn’t ac-
 8914 count for any geographic boundaries of the survey region, but we note that a habitat mask
 8915 is included in **secr** and it could be used in a Bayesian analysis. Whether or not we use the
 8916 mask is probably immaterial as long as we understand the predictions of N or D over the
 8917 water don’t mean anything biological and we probably wouldn’t report such predictions.
 8918 The **JAGS** model specification is based on that of the ovenbird analysis given previously,
 8919 and so we don’t reproduce the model here. The **R/JAGS** script is called **SCRpossum**,
 8919 which is in the **scrbook** package. The results are summarized in Table 9.3.

Table 9.3. Results of fitting the independent multinomial observation model to the possum trapping data. Strictly speaking, the trapping device is a “single-catch” trap, and the model represents an intentional misspecification. Density is reported in individuals per ha (D_{ha}). Posterior summaries were obtained using **JAGS** with 3 chains, each with 2000 iterations, discarding the first 1000 as burn-in, to yield a total of 3000 posterior samples.

Parameter	Mean	SD	2.5%	50%	97.5%	Rhat	n.eff
N	235.407	17.435	204.000	235.000	270.000	1.009	340
D_{ha}	1.549	0.115	1.343	1.547	1.777	1.009	340
α_0	-0.935	0.167	-1.270	-0.934	-0.605	1.007	870
α_1	0.000	0.000	0.000	0.000	0.000	1.001	2800
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

8920 The estimated density (posterior mean) is about 1.53 possums/ha. To obtain the **secr**
 8921 results for the equivalent null model, we execute the following command

```
8922 > secr.fit( capthist = possumCH, trace = F )
8923 which produces (edited) summary output:
8924 [... some output deleted ...]
8925
8926 Fitted (real) parameters evaluated at base levels of covariates
8927   link estimate SE.estimate      lcl      ucl
8928 D      log    1.6988930  0.17352645  1.3913904  2.0743547
```

²M. Efford, personal communication

```

8929 g0    logit  0.1968542  0.02256272  0.1563319  0.2448321
8930 sigma  log 51.4689114  2.59981905  46.6204139  56.8216500
8931
8932 [... some output deleted ...]

```

8933 As we've discussed previously, there are many reasons for why there might be differences
8934 between Bayesian and likelihood estimates. But even among likelihood estimates – any
8935 time you run a model there is some numerical integration going on which requires some
8936 specific choices of how to do the integration (see Chapt. 6). For now we just observe that
8937 the estimated density is certainly in the ballpark (compared to those in Table. 9.3), and
8938 so too is the estimated σ .

9.4 ACOUSTIC SAMPLING

8939 The last decade has seen an explosion of technology that benefits the study of animal
8940 populations. This includes DNA sampling methods that allow for identification from
8941 hair or scat, camera trapping and identification software that allow efficient sampling
8942 of many mammals, and the resulting statistical technology that helps us to make sense
8943 of such data (Borchers and Efford, 2008; Royle and Young, 2008; Efford et al., 2009b;
8944 Gopalaswamy et al., 2012b; Sollmann et al., 2012; Chandler and Royle, In press). One
8945 other extremely promising technology area is that of acoustic sampling using microphones
8946 or recording devices. That is, instead of having cameras record encounters, or humans pick
8947 up scat, we can establish an array of (usually) electronic recording devices which, instead of
8948 establishing a visual identity of individuals, record a vocal expression of each individual. In
8949 this context, Efford et al. (2009b) referred to audio recorders as “signal strength proximity
8950 detectors” to distinguish them from other types of proximity detections, including camera
8951 traps, which are *visual* proximity detector. Using audio records, the spatial pattern of the
8952 *signal strength* at the different audio recorders or microphones can be used for inference
8953 about density (Dawson and Efford, 2009; Efford et al., 2009b) in the same way as the
8954 spatial pattern of detections is used in the types of SCR models we have discussed so far.
8955 The basic technical formulation of these models comes from Efford et al. (2009b), and it
8956 was applied to field study of birds by Dawson and Efford (2009). In that study, recording
8957 devices were organized in groups of 4 (in a square pattern), with an array of 5×15 such
8958 clusters of 4, separated by 100 m (300 total recorder locations). This data set, called
8959 **signalCH**, is provided with the **secr** package along with some sample analyses and help
8960 files. See Efford and Dawson (2010), a version of the document **secr-sound.pdf** (that
8961 also comes with the **secr** package) which you can access directly from the main help file
8962 (**?secr**).

8963 Our development here mostly follows Efford et al. (2009b), but we change some nota-
8964 tion to be consistent with our previous material. Let $S(\mathbf{x}, \mathbf{u})$ be the strength of a signal
8965 emanating from signal location \mathbf{u} , as recorded by a device at location \mathbf{x} . Just as ordinary
8966 SCR models represent a model of *encounter frequency* as a function of distance, in acoustic
8967 models, the acoustic SCR model is a model of sound attenuation as a function of distance.
8968 In particular, the acoustic models assumes that S (or a suitable transformation) declines
8969 with distance d from the origin of the sound, to the recording device. In the context of
8970 spatial sampling of animals, the origin is the actual location of some individual animal,

8971 and the recording device is something we nailed to a tree, or mounted on a post. For ex-
 8972 ample, a model of sound attenuation used by Dawson and Efford (2009) is the following:

$$8973 \quad S(\mathbf{x}, \mathbf{u}) = \alpha_0 + \alpha_1 d(\mathbf{x}, \mathbf{u}) + \epsilon \quad (9.4.1)$$

8974 where $\epsilon \sim \text{Normal}(0, \sigma_s^2)$. In many standard situations, S will be measured in decibels,
 8975 which can be any value on the real line. In the conduct of acoustic sampling and the
 8976 development of custom models for your own situation, it would probably be helpful to know
 8977 something about sound dynamics and signal processing. In this model, the parameters
 8978 α_0 , α_1 and σ_s^2 are to be estimated. We abbreviate the set of parameters by $\boldsymbol{\theta}$ for short.

8979 The basic structure of an acoustic SCR study is not really much different from ordinary
 8980 SCR studies. Just as ordinary SCR models require that individuals be encountered at > 1
 8981 trap, these acoustic models require that individuals be heard at > 1 recorder. Therefore,
 8982 the acoustic signals (calls or vocalizations) must be reconcilable and, in fact, reconciled
 8983 successfully by the investigator. In practice, this would require associating signals that
 8984 occur at the same instant with the same individual (or making a decision one way or the
 8985 other). Further, if individuals are actively moving during the sample period (that recorders
 8986 are functioning) then individuals might be double-counted, thereby biasing estimates of
 8987 density. In general, the models produce an estimate of density of sources, and how that is
 8988 interpreted depends on whether individuals are stationary or mobile, and other things. In
 8989 particular, if multiple survey occasions are used (e.g., on different days), then modeling
 8990 movement of individuals would be essential in order to interpret estimates of density
 8991 meaningfully. Models that allow some movement should be possible (see Sec. 9.4.3 below,
 8992 and Chaps. 15 and 16).

8993 9.4.1 The signal strength model

8994 We assert that an individual is detected if S exceeds a threshold, c . The reason for intro-
 8995 ducing this threshold c is that sound recorders will always record some background sound,
 8996 and so effective use of the acoustic SCR models requires specification of the threshold of
 8997 measured signal below which the record is censored (non-detection occurs) because the
 8998 recorded sound is assumed to be background noise. So we assert that an individual is
 8999 detected if $S > c$ which occurs with probability $\Pr(S > c)$, the encounter probability. To
 9000 expand on and formalize this, let S_{ij} be the observed value of S for animal i at detector
 9001 j . The encounter probability is $\Pr(S_{ij} > c)$ which is $\Pr(S_{ij} > c) = 1 - \Pr(S_{ij} < c)$, so
 9002 that, if we standardize the variate we have

$$1 - \Pr\left(\frac{(S_{ij} - \mathbb{E}(S))}{\sigma_s} < \frac{(c - \mathbb{E}(S))}{\sigma_s}\right)$$

9003 This probability calculation requires evaluation of the CDF of a standard normal variate
 9004 say, $\eta = (S_{ij} - \mathbb{E}(S))/\sigma_s$, being less than $\gamma(\boldsymbol{\theta}) = (c - \mathbb{E}(S))/\sigma_s$, which is a function of all
 9005 the parameters α_0 , α_1 , σ_s^2 and also the individual location \mathbf{u} and trap location \mathbf{x} . We'll
 9006 identify it by $\gamma(\boldsymbol{\theta}, \mathbf{x}, \mathbf{u})$ when we need to be explicit about those things. We can compute
 9007 $\Pr(S_{ij} > c) = 1 - \Pr(\eta < \gamma(\boldsymbol{\theta}, \mathbf{x}, \mathbf{u}))$ easily using any software package including **R** which
 9008 has a standard function, **pnorm**, for computing the normal cdf. To be more precise, we'll
 9009 use the **Phi()** to represent the normal cdf. Therefore, an individual is encountered whenever
 9010 $S_{ij} > c$ which happens with probability $\Pr(S_{ij} > c) = 1 - \Phi(\gamma(\boldsymbol{\theta}, \mathbf{x}, \mathbf{u}))$.

9011 Naturally this quantity should depend on *where* an individual is located at the time
 9012 of recording – what we call it’s instantaneous location, say \mathbf{u} , to distinguish it from it’s
 9013 home-range center \mathbf{s} (but we outline a model below that contains both \mathbf{u} and \mathbf{s}), and
 9014 also the trap \mathbf{x} , so we index the quantity γ by those two quantities, in addition to the
 9015 parameters α_0 , α_1 and σ_s . The probability of detection is therefore

$$p_{ij} = p(\alpha_0, \alpha_1, \sigma | \mathbf{x}_j, \mathbf{u}_i) = 1 - \Phi(\gamma(\cdot))$$

9016 where \mathbf{u}_i is the instantaneous location of individual i and \mathbf{x}_j is the location of trap j .
 9017 We’ll suppose here that the random variables \mathbf{u}_i have state-space \mathcal{U} ³.

9018 How do we interpret this probability? Well, two things have to happen for an individual
 9019 to be encountered by a trap: (1) it has to vocalize; (2) the microphone has to record a
 9020 signal $> c$. These two things together are a product of biological and environmental factors
 9021 which could include time of day, wind direction and speed, or maybe rain, humidity and
 9022 other things. The bottom line is a lot of factors are balled up in whether or not the
 9023 microphone records a sound greater than the threshold.

9024 The observations from an acoustic survey are the signal strength measurements, and
 9025 the likelihood of the observed signal strength from individual i at detection device j can
 9026 be specified by noting that the likelihood is the normal pdf for the observed signal *if* the
 9027 signal strength is $> c$ and, otherwise, the contribution to the likelihood is $\Phi(\gamma(\cdot))$ (see Eq.
 9028 8 of Efford et al. (2009b)):

$$\Pr(S_{ij} | \mathbf{u}_i) = \Phi(\gamma(\cdot))^{1-I(S_{ij}>c)} \text{Normal}(S_{ij}; \alpha_0, \alpha_1, \sigma_s, \mathbf{x}_j, \mathbf{u}_i)^{I(S_{ij}>c)}$$

9029 We can use this as the basis for constructing the binomial-form of the likelihood as
 9030 we did in Chapt. 6, which involves the number of individuals not encountered, n_0 . The
 9031 probability that an individual is *not* captured is equal to the probability that its signal
 9032 strength doesn’t exceed c at any microphone. The probability of not being captured at a
 9033 microphone \mathbf{x}_j is:

$$1 - p_{\mathbf{u},j} = \Phi(\gamma(\cdot))$$

9034 and therefore the probability of not being captured at any microphone is:

$$\Pr(\text{all } S_{\mathbf{u},j} < c | \mathbf{u}) = \prod_{j=1}^J (1 - p_{\mathbf{u},j}) = \prod_{j=1}^J \Phi(\gamma(\cdot, \mathbf{x}_j, \mathbf{u}))$$

9035 and therefore the marginal probability of not being captured is

$$\pi_0 = [\text{all } S_{\mathbf{u},j} < c | \boldsymbol{\alpha}] = \int_{\mathcal{U}} \left\{ \prod_{j=1}^J \Phi(\gamma(\boldsymbol{\theta}, \mathbf{x}_j, \mathbf{u})) \right\} d\mathbf{u}$$

9036 which can be used to construct the binomial form of the likelihood as we did in Chapt. 6
 9037 (see Eq. 6.2.1).

³We use \mathcal{U} here to avoid confusion with definition of signal strength, S . However, \mathcal{U} is the same state-space as \mathcal{S} in the rest of the book

9038 **9.4.2 Implementation in secr**

9039 Fitting acoustic encounter models in **secr** is no more difficult than other SCR models.
 9040 There is a handy manual (**secr-sound.pdf**) with examples (Efford and Dawson, 2010)
 9041 which comes with the **secr** package. The basic process is that **make.capthist** will make a
 9042 **capthist** object from a 3-dimensional encounter array – which is a binary array indicating
 9043 whether each individual was detected or not at each recorder/microphone. In the case
 9044 of signal strength data, **secr** handles the case where # occasions = 1, i.e., the recorders
 9045 obtained data for a single sample occasion, but this is not a general requirement of the
 9046 model for signal strength data (see next section). The “signal” attribute of the **capthist**
 9047 object contains the signal strength in decibels. The best way to include the signal attribute
 9048 is to use **make.capthist** in the usual way, providing it with the encounter data and
 9049 trap data and, in addition, the variable “*c*utval” (which is *c* in our notation above) and
 9050 then provide the signal strength data as an extra column of the **capthist** object. See
 9051 **?make.capthist** for details.

9052 **9.4.3 Implementation in BUGS**

9053 We don’t know of any Bayesian applications of acoustic SCR models, although we imagine
 9054 that implementation of such models in the **BUGS** engines should be achievable. It seems
 9055 easy enough to write down a general hierarchical model that would accommodate sampling
 9056 on repeated occasions. Let \mathbf{s}_i be the home range center, and let \mathbf{u}_{ik} the instantaneous
 9057 location of individual i during sample occasion k (see Chapt. 15 for similar models). The
 9058 model for \mathbf{u}_{ik} can be specified conditional on \mathbf{s}_i . For example, we could assume that \mathbf{u}_{ik}
 9059 are bivariate normal draws with mean \mathbf{s}_i and some variance σ_u^2 . Then, conditional on \mathbf{u}_{ik}
 9060 an individual produces a signal according to the signal attenuation model (Eq. 9.4.1), or
 9061 perhaps some other model. Then we generate the binary encounter data by truncating the
 9062 observed signal at c . This general model then is an example of an SCR model in which
 9063 parameters of a movement model are identifiable (see Sec. 2.6) because there is direct
 9064 information about movement outcomes from the sampling method, unlike other types of
 9065 encounter methods (e.g., camera traps) for which animal locations are restricted to a set of
 9066 fixed, pre-determined points where traps are located. Other types of SCR methods allow
 9067 for movement information too, including some of the search-encounter models (Chapt.
 9068 15).

9069 Instead of developing a Bayesian version of this model here, we leave it to the reader
 9070 to explore simulating data and devising a Bayesian implementation of the acoustic model
 9071 in one of the **BUGS** engines. Note that for a single occasion, you can simulate the data
 9072 using the two stage model (having both \mathbf{s} and \mathbf{u}) or you can simulate \mathbf{u} uniformly without
 9073 dealing with \mathbf{s} in the model. The kernel of the **BUGS** model specification should resemble
 9074 the following snippet:

```
9075 model {
  9076   # Ignoring loops and data augmentation
  9077   u[i,1] ~ dunif(xlim[1], xlim[2])
  9078   u[i,2] ~ dunif(ylim[1], ylim[2])
  9079   mu[i,j] <- alpha0 + alpha1*d[i,j]
  9080   ####
```

```

9081  ##### JAGS has this T() truncation feature
9082  S[i,j] ~ dnorm(mu[i,j], 1/sigma^2)T(c,Inf)
9083  #####
9084  gamma[i,j] <- (c - mu[i,j])/sigma
9085  p[i,j] <- 1 - pnorm(gamma[i,j], 0, 1) # JAGS has pnorm() function
9086  y[i,j] ~ dbern(p[i,j])
9087 }
```

9088 **9.4.4 Other types of acoustic data**

9089 Efford and Dawson (2010) noted that various other types of acoustic data might arise
 9090 for which SCR-like models would be useful⁴. For example, we could measure the *time of*
 9091 *arrival* of a vocal queue of some sort at multiple recorders to estimate the number and
 9092 origin of N queues. Another example is that where we measure *direction* to a queue from
 9093 multiple devices and do, effectively, a type of statistical triangulation to the multiple but
 9094 unknown number of sources. This has direct relevance to types of double or multiple-
 9095 observer sampling that people do in field studies of birds. Normally 2 observers stand
 9096 in close proximity and record birds, reconciling their detections after data collection.
 9097 An SCR-based formulation of the double-observer method has two observers (or more)
 9098 standing some distance apart, e.g., 50 or 100 meters, and marking individual birds on a
 9099 map (or at least a direction) and a time of detection. The SCR/double-observer method
 9100 could be applied to such data.

9.5 SUMMARY AND OUTLOOK

9101 In this chapter we extended SCR models to accommodate alternative models for the
 9102 observation process, including Poisson and multinomial models. Along with the binomial
 9103 model described in Chapt. 5, this sequence of models will accommodate a substantial
 9104 majority of contemporary spatial capture-recapture problems, including the 4 main types
 9105 of encounter data: binary encounters, multinomial trials from “multi-catch” and “single-
 9106 catch” (Efford, 2004, 2011; Royle and Gardner, 2011) trap systems, and Poisson encounter
 9107 frequency data from devices that can record multiple encounters of the same individual
 9108 at a device. We summarize the standard observation models and the corresponding **secr**
 9109 terminology in Table 9.4. What we refer to as search-encounter (or area-search) models
 9110 (see Chapt. 15) are distinct from most of the other classes in that the observation location
 9111 can also be random (in contrast to traps, where the location is fixed by design). This
 9112 auxiliary data is informative about an intermediate process related to movement (Royle
 9113 and Young, 2008).

9114 There is a need for other types of encounter models that arise in practice. We identify
 9115 a few of them here, although we neglect a detailed development of them at the present
 9116 time or, in some cases, put that off until later chapters: (1) Removal systems – Sometimes
 9117 traps kill individuals and SCR models can handle that. This can be viewed as a kind of
 9118 open model, with mortality only, and we handle such models (in part) in Chapt. 16; (2)
 9119 There are models for which only specific summary statistics are observable (Chandler and

⁴Some of the following is also related to material presented by D.L. Borchers at the ISEC 2012 conference in Norway.

Table 9.4. Different observation models, where we discuss them in this book, and what the corresponding **secr** terminology is

observation model	Where in this book?	secr name
Bernoulli	Chapt. 5	proximity
Poisson	Sec. 9.1	count
Multinomial (ind)	Sec. 9.2	multi-catch
Multinomial (dep)	Sec. 9.3	single-catch
Acoustic	Sec. 9.4	signal
Search-encounter	Chapt. 15	polygon (in part)

9120 Royle, In press; Sollmann et al., 2012) which we cover in Chaps. 18 - 19; (3) We can have
 9121 multiple observation methods working together as in Gopalaswamy et al. (2012b).

9122 There remains much research to be done to formalize models for certain observation
 9123 systems. For example, while we think one will usually be able to analyze single-catch
 9124 systems using the multi-catch model, or even the Bernoulli model if encounter probability
 9125 is sufficiently low, a formalization of the single-catch model would be a useful development
 9126 and, we believe, it should be achievable using one or another of the **BUGS** engines. In
 9127 addition, classical “trapping webs” (Anderson et al., 1983; Wilson and Anderson, 1985a;
 9128 Jett and Nichols, 1987; Parmenter and MacMahon, 1989; Link and Barker, 1994) have
 9129 been around for quite some time and it seems like they are amenable to formulation as
 9130 a type of SCR model although we have not pursued that development simply because
 9131 trapping webs are rarely used in practice.

9132

9133

9134

10

SAMPLING DESIGN

10.1 GENERAL CONSIDERATIONS

9135 10.1.1 Model-based not design-based

9136 10.1.2 Sampling space or sampling individuals?

9137 10.1.3 Scope of inference vs. state-space

10.2 STUDY DESIGN FOR (SPATIAL) CAPTURE-RECAPTURE

10.3 TRAP SPACING AND ARRAY SIZE RELATIVE TO ANIMAL MOVEMENT

9138 10.3.1 Example: Black bears from Pictured Rocks National Lakeshore:

9139 **10.3.2 Final musings: SCR models, trap spacing and array size**

10.4 SPACING OF TRAPS WITH TELEMETERED INDIVIDUALS

10.5 SAMPLING OVER LARGE SCALES

10.6 MODEL-BASED SPATIAL DESIGN

9140 **10.6.1 Formalization of the Design Problem for SCR Studies**

9141 **10.6.2 An Optimal Design Criterion for SCR**

(10.6.1)

(10.6.2)

(10.6.3)

9142 **10.6.3 Optimization of the criterion**

9143 **10.6.4 Illustration**

10.7 COVARIATE MODELS

10.8 SUMMARY AND OUTLOOK

9144

Part III

9145

9146

Advanced SCR Models

9147

9148

9149

9150

11

MODELING SPATIAL VARIATION IN DENSITY

11.1 HOMOGENEOUS POINT PROCESS REVISITED

11.2 INHOMOGENEOUS POINT PROCESSES

(11.2.1)

(11.2.2)

(11.2.3)

(11.2.4)

(11.2.5)

(11.2.6)

11.3 OBSERVED POINT PROCESSES**11.4 FITTING INHOMOGENEOUS POINT PROCESS SCR MODELS**

9151 **11.4.1 Continuous space**

9152 **11.4.2 Discrete space**

11.5 ECOLOGICAL DISTANCE AND DENSITY COVARIATES**11.6 THE JAGUAR DATA****11.7 SUMMARY AND OUTLOOK**

9153
9154

12

9155

MODELING LANDSCAPE CONNECTIVITY

12.1 SHORTCOMINGS OF EUCLIDEAN DISTANCE MODELS

(12.2.1)

(12.2.2)

(12.2.3)

9156 12.2.1 Example of Computing Cost-weighted distance

12.3 SIMULATING SCR DATA USING ECOLOGICAL DISTANCE

12.4 LIKELIHOOD ANALYSIS OF ECOLOGICAL DISTANCE MODELS

(12.4.1)

9157 **12.4.1 Example of SCR with Least-Cost Path**

12.5 BAYESIAN ANALYSIS

12.6 SIMULATION EVALUATION OF THE MLE

9158 **12.6.1 Simulation Results**

12.7 DISTANCE IN AN IRREGULAR PATCH

9159 **12.7.1 Basic Geographic Analysis in R**

12.8 SUMMARY AND OUTLOOK

9160
9161

13

9162
9163
9164

INTEGRATING RESOURCE SELECTION WITH SPATIAL CAPTURE-RECAPTURE MODELS

9165
9166
9167

13.1 A SIMPLE MODEL OF SPACE USAGE

13.1.1 Poisson use model

13.1.2 Thinning

13.1.3 Capture-recapture Data

13.2 THE JOINT RSF/SCR LIKELIHOOD

13.3 APPLICATION: NEW YORK BLACK BEAR STUDY

13.4 SIMULATION STUDY

13.5 SUMMARY AND OUTLOOK

9168
9169

14

9170
9171

STRATIFIED POPULATIONS: MULTI-SESSION AND MULTI-SITE DATA

14.1 DATA STRUCTURE

14.2 MULTINOMIAL ABUNDANCE MODELS

(14.2.1)

(14.2.2)

(14.2.3)

(14.2.4)

(14.2.5)

9172
9173
9174
9175

14.2.1 Observation Models

14.2.2 Simulating group structured capture-recapture data

14.2.3 Fitting in BUGS

14.2.4 Approach B modeling ψ

14.3 SPATIAL CAPTURE-RECAPTURE

(14.3.1)

14.4 APPLICATION**9176 14.4.1 Results**

14.5 TOPICS IN MULTI-SESSION MODELS

9177 **14.5.1 Temporal models**

9178 **14.5.2 Dependence – is it a problem?**

14.6 MULTI-SESSION MODELS IN SECR

9179 **14.6.1 Ovenbird data in WinBUGS?**

9180 **14.6.2 Converse data in secr?**

14.7 SUMMARY AND OUTLOOK

9181
9182

15

9183

MODELS FOR SEARCH-ENCOUNTER DATA

9184

15.1 SEARCH-ENCOUNTER SAMPLING DESIGNS

15.2 A MODEL FOR SEARCH-ENCOUNTER DATA

(15.2.1)

(15.2.2)

(15.2.3)

9185 **15.2.1 Ecological process model**

9186 **15.2.2 Other stuff**

15.3 EXAMPLES

9187 **15.3.1 Hard plot boundaries**

9188 **15.3.2 Analysis of other protocols**

15.4 DESIGN 3: AD HOC IMPLEMENTATION OF DESIGN 1.

15.5 CAPRICAILLIE CRAP

9189 **15.5.1 model**

15.6 DESIGN 4 – NO LOCATION INFO

15.7 SUMMARY AND OUTLOOK

9190
9191
9192

16

OPEN POPULATION MODELS

16.1 INTRODUCTION

- 9193 **16.1.1 Overview of Population Dynamics**
- 9194 **16.1.2 Animal movement related to population demography**
- 9195 **16.1.3 Basic assumptions of JS and CJS models**

16.2 TRADITIONAL JOLLY-SEBER MODELS

- 9196 **16.2.1 Data Augmentation for the Jolly-Seber Model**
-
-

- 9197 **16.2.2 Mist-netting example**
- 9198 **16.2.3 Shortcomings of the traditional JS models**

16.3 SPATIAL JOLLY-SEBER MODELS

(16.3.1)

9199 **16.3.1 Mist-netting example**

16.4 TRADITIONAL CJS MODELS

9200 **16.4.1 Migratory fish example**

16.5 MULTI-STATE CJS MODELS

9201 **16.5.1 Migratory fish example**

16.6 SPATIAL CJS MODELS

9202 **16.6.1 Migratory fish example**

16.7 MOVING ACTIVITY CENTERS

9203 **16.7.1 Migratory Fish Example Notes**

16.8 SUMMARY AND OUTLOOK



9204

Part IV

9205

9206

Super-Advanced SCR Models

9207
9208

9209
9210

17

DEVELOPING MARKOV CHAIN MONTE CARLO SAMPLERS

9211 17.0.1 Why build your own MCMC algorithm?

17.1 MCMC AND POSTERIOR DISTRIBUTIONS

(17.1.1)

(17.1.2)

17.2 TYPES OF MCMC SAMPLING

9212 17.2.1 Gibbs sampling

(17.2.1)

9213 **17.2.2 Metropolis-Hastings sampling**

9214 **17.2.3 Metropolis-within-Gibbs**

9215 **17.2.4 Rejection sampling and slice sampling**

17.3 MCMC FOR CLOSED CAPTURE-RECAPTURE MODEL MH

17.4 MCMC ALGORITHM FOR MODEL SCR0

9216 **17.4.1 SCR model with binomial encounter process**

9217 **17.4.2 Looking at model output**

9218 **Markov chain time series plots**

9219 **17.4.3 Posterior density plots**

9220 **17.4.4 Serial autocorrelation and effective sample size**

9221 **17.4.5 Summary results**

9222 **17.4.6 Other useful commands**

17.5 MANIPULATING THE STATE-SPACE

17.6 INCREASING COMPUTATIONAL SPEED

9223 **17.6.1 Parallel computing**

9224 **17.6.2 Using C++**

17.7 SUMMARY AND OUTLOOK

9225
9226

9227
9228

18

SPATIAL CAPTURE-RECAPTURE FOR UNMARKED POPULATIONS

**18.1 EXISTING MODELS FOR INFERENCE ABOUT DENSITY IN
UNMARKED POPULATIONS**

18.2 SPATIAL CORRELATION AS INFORMATION

18.3 DATA

18.4 MODEL

(18.4.1)

(18.4.2)

18.5 NORTHERN PARULA EXAMPLE

18.6 IMPROVING PRECISION WITH PRIOR INFORMATION

18.7 DESIGN ISSUES

9229 **18.7.1 How Much Correlation Is Enough?**

9230 **18.7.2 Linear Designs**

9231 **18.7.3 Quadrat counts**

18.8 ALTERNATIVE OBSERVATION MODELS

9232 **18.8.1 Spatial point process models**

18.9 CONCLUSION

19

9233

9234

9235

9236

9237

SPATIAL MARK-RESIGHT MODELS FOR PARTIALLY IDENTIFIABLE POPULATIONS

9238 So far, we have dealt with the situation where all detected individuals are identifiable
9239 upon encounter, and in Chapt. 18 we introduced and developed an SCR model for non-
9240 identifiable populations, a spatial *non*-capture-recapture model, if you will. These two
9241 extremes are common in the study of animal populations with non-invasive sampling meth-
9242 ods. However, there is also an intermediate situation, where a part of the population is
9243 tagged or otherwise marked and can thus be identified upon recapture, while the untagged
9244 portion remains unidentified. In this situation so-called mark-resight models (Bartmann
9245 et al., 1987; Arnason et al., 1991; Neal et al., 1993) can be used to estimate population
9246 size and density combining data from both the marked and unmarked individuals.

9247 Traditionally, capture-recapture studies involve physical capture of individuals through-
9248 out the study; new individuals are marked on every re-capture occasion. This methodology
9249 is still widely applied in the study of species that are relatively easy to capture, such as
9250 small mammals, but can be very costly, logically challenging and risky when dealing
9251 with larger species. In contrast, in mark-resight studies a sample of individuals is captured
9252 and tagged (or otherwise marked) during a single marking event. Marking is followed by
9253 resighting surveys, upon which both the detection of marked and recognizable individuals
9254 and unmarked animals is recorded. Resighting surveys are usually non-invasive (hence the
9255 name resighting), so that they don't involve handling of animals. As such, mark-resight
9256 models have a major advantage over traditional capture-recapture models in that they
9257 only require individuals to be captured and handled once, during the initial marking.
9258 This reduces field costs and risks for the animals (and potentially the researchers).

9259 Mark-resight models have a set of underlying assumptions, most of which are analogous
9260 to those of capture-recapture models, e.g. demographic population closure (violation
9261 of geographic population closure can be accommodated by some models) and no loss

9262 or misidentification of marks (see also 5). Just like regular capture-recapture models,
9263 there are means to incorporate heterogeneity in capture probability. However, a new and
9264 essential assumption of mark-resight models is that the tagged (or otherwise identifiable)
9265 individuals are a representative sample of the study population, so that inference about
9266 detection can be made for the whole population from the tagged sample. This issue is
9267 usually addressed by using a different method for marking than for resighting, and by
9268 marking a random sample of the population.

9269 Owing to the advantages of mark-resight over capture-recapture, especially when dealing
9270 with hard-to-trap species, mark-resight is a popular tool in wildlife population studies.
9271 The method has been applied for decades and to a suite of species and survey techniques,
9272 ranging from banding and resighting Canada geese (Hestbeck and Malecki, 1989) to ear-
9273 tagging and camera-trapping grizzly bears (Mace et al., 1994) to paintball marking and
9274 areal resightings of large ungulates (Skalski et al., 2005).

9275 In this chapter we consider mark-resight in spatial context and develop a spatial mark-
9276 resight (SMR) model. To motivate this model development, imagine you conduct a live-
9277 trapping study during which you capture and mark a number of animals with individually
9278 recognizable tags. Subsequently, you go back out to the field and conduct resighting
9279 surveys on an array of locations, and during these resighting surveys you see some of your
9280 tagged individuals as well as new, untagged ones. Then, for the tagged animals you obtain
9281 the same form of spatially explicit individual encounter histories as you would in a regular
9282 SCR study. On top of that you obtain site (and occasion) specific counts of individuals you
9283 did not tag. Thus, spatial mark-resight is an SCR framework for populations where only
9284 part of the individuals can be identified and the major difference between SCR and SMR
9285 is how we include those counts of unmarked individuals in the model. In the following
9286 sections we first provide some background information on mark-resight and the types of
9287 data such surveys can provide. We then move on to the formal development of SMR
9288 models, which, as we will see, are hybrids of regular SCR models and the models for data
9289 without individual identity presented in Chapt. 18.

19.1 BACKGROUND

9290 19.1.1 Types of partial ID data

9291 Before we start exploring mark-resight approaches in more detail, we need a clear under-
9292 standing of what types of mark-resight data we can have, in order to appreciate and
9293 understand the different flavors of mark-resight models. In general, we have (at least) two
9294 sets of data: encounter histories for identifiable individuals i at trap j and occasion k ,
9295 y_{ijk} , and counts of unidentified records for each j and k , n_{jk} . Depending on the sampling
9296 technique, we can conceive of three slightly different types of partial ID data.

9297 **(1) Known number of tagged individuals** If you implement your resighting survey
9298 shortly after the marking session, you may be confident that none of the marked individuals
9299 has died or lost its mark. Under these circumstances you know that the number of marked
9300 individuals available for resighting, m , is equal to the number of individuals you tagged.
9301 Alternatively, tags might be radio-transmitters, allowing you to confirm the presence or
9302 absence of marked individuals in the resighting survey area using radio-telemetry (White
9303 and Shenk, 2001). In both cases, you know the number of marked individuals in the
9304 population you survey. In this situation, even though you may fail to resight some of the

9305 tagged individuals, since you know how many there are, you can simply assign those you
9306 never resighted all-zero encounter histories - in other words, contrary to regular capture-
9307 recapture models, in mark-resight models with a known number of tagged individuals, we
9308 can observe all-zero encounter histories. Under these circumstances, estimating N reduces
9309 to estimating the number of unmarked individuals, U .

9310 **(2) Unknown number of tagged individuals** If we suspect that some of the
9311 marks may have been lost between tagging and conducting the resighting samples, we
9312 obtain a slightly different type of mark-resight data. Here, we do not accurately know
9313 the number of marked individuals available for resighting. As a consequence, individuals
9314 have to be resighted at least once for us to know they are still tagged and alive and thus
9315 available for resighting. So, contrary to the situation where we know m and analogous to
9316 regular capture-recapture models, we cannot observe all-zero encounter histories of marked
9317 individual. Here, estimating N involves estimating both m and U .

9318 A special case of this kind of data can arise from camera trapping. Even when dealing
9319 with a species that has no spots or stripes, some individuals in the study population can
9320 have natural marks that make them identifiable on pictures, such as scars or some distinct
9321 coloration. Again, in this scenario an individual has to be photographed at least once to
9322 be known. Here, the fact that both the “marking” method and the subsequent resighting
9323 method are the same (although marking in this case does not involve any actual physical
9324 marking) can be cause for concern: our sample of “marked” individuals may not be a
9325 random sample of the population but consist of individuals that for some reason are more
9326 likely to be photographed. In that case, a basic assumption of the mark-resight model is
9327 violated.

9328 **(3) Unknown marked status** Finally, consider a scat or hair snare survey, where
9329 only a part of the sample is analyzed genetically (or DNA can only be extracted from
9330 a subset of samples due to sample quality). In this scenario, your n_{jk} can contain both
9331 completely unknown individuals that are not represented at all in Y , but it can also contain
9332 samples from individuals that we previously identified. The difference is that in the first
9333 two scenarios, part of the population of individuals is identifiable, while in the second
9334 scenario, part of the samples is identifiable. This type of data violates one of the basic
9335 assumptions of mark-resight models, namely, that tagged individuals are always correctly
9336 identified as such.

9337 To our knowledge there are currently no mark-resight models available that account for
9338 possible misidentification of the marking status of individuals (although some literature is
9339 available on misidentification of individuals in capture-recapture studies, e.g., Yoshizaki
9340 et al., 2009; Lukacs and Burnham, 2005; Link et al., 2010). In this chapter we will ignore
9341 this kind of data and focus instead on the two types of typical mark-resight data:

- 9342 (1) Known number of tagged individuals
9343 (2) Unknown number of tagged individuals,

9344 For both types of data a slightly different situation arises when in some instances we
9345 can only tell that an individual is tagged, but not who it is. You may be able to see that
9346 an individual is tagged but the identifying feature of the tag (a number or coloration)
9347 may have become unreadable, or may be hidden from view. In this case, in addition to
9348 your y_{ijk} and your n_{jk} you also have a number of sightings of tagged but unidentified
9349 individuals, say r_{jk} .

9350 **19.1.2 A short history of mark-resight models**

9351 Initially, mark-resight methods focused on radio-tagged individuals to estimate popula-
 9352 tion size (White and Shenk, 2001). Radio-collars provide a means of determining which
 9353 of the animals were in the study area and available for sampling, i.e. determining the
 9354 number of marked individuals in the population. Knowing this number was a prerequisite
 9355 for most earlier mark-resight approaches (White, 1996). The oldest mark-resight model
 9356 is the good old Lincoln-Petersen estimator, where individuals are marked and a single
 9357 resight/recapture occasion is carried out (Krebs, 1999). We need not identify individuals,
 9358 but only tell apart marked from unmarked individuals. Let m be the number of marked
 9359 individuals in the population, $m_{(R)}$ the number of marked individuals seen on the resight-
 9360 ing occasion, and $n_{(R)}$ the total number of marked and unmarked individuals observed
 9361 during resighting. Population size N is then estimated as

$$N = m \times n_{(R)} / m_{(R)}$$

9362 A suite of more elaborate models using individual capture histories over several re-
 9363 sighting occasions were developed in the 1980s and 90s and compiled into the program
 9364 NOREMARK (White, 1996). Apart from the basic model with known number of marked
 9365 individuals and no individual variation in resighting probabilities (joint hypergeometric
 9366 maximum likelihood estimator) (Bartmann et al., 1987; White and Garrot, 1990; Neal,
 9367 1990; Neal et al., 1993), NOREMARK contains models that account for lack of geographic
 9368 population closure (Neal et al., 1993), individual heterogeneity in resighting rates and
 9369 sampling with replacement (i.e. individuals can be seen more than once on any occasion,
 9370 (Minta and Mangel, 1989; Bowden, 1993)). A first mark-resight model allowing for an
 9371 unknown number of marked individuals was developed by Arnason et al. (1991).

9372 While many of these models perform well under certain situations, they are somewhat
 9373 limited: they do not allow for combining data across several surveys (McClintock et al.,
 9374 2006) and not all of them are likelihood-based or allow for different parameterizations
 9375 (e.g., including a time effect on detection), so that selection of the most appropriate
 9376 model cannot be based on standard approaches such as AIC, but is largely left up to
 9377 educated guesswork (McClintock et al., 2006). Recently, more flexible and generalized
 9378 likelihood-based mark-resight models have been developed. These models can account
 9379 for individual heterogeneity in detection, unknown number of marked individuals and
 9380 lack of geographical closure, as well as a less than 100% individual identification rate of
 9381 tagged individuals; they can be applied to sampling with and without replacement and
 9382 can combine data across several primary sampling occasions in a robust design type of
 9383 analysis (McClintock et al., 2009a,b). Since they are all likelihood-based, model selection
 9384 among different parameterizations and model averaging based on AIC is an option. Most
 9385 of these models have also been incorporated into the program **MARK** (McClintock and
 9386 White, 2012).

9387 For a detailed treatment of these different non-spatial mark-resight models, we refer
 9388 you to the original papers cited in the preceding paragraph. In short, these models are
 9389 based on the joint likelihood of two major model components: one describing the resight-
 9390 ing process of marked individuals (either using a Poisson or a Bernoulli observation model,
 9391 depending on whether sampling is with or without replacement), where resighting proba-
 9392 bilities can have both fixed effects to model individual and environmental covariates, and

9393 a random-effect component to accommodate variation in detection due to individual heterogeneity; and one describing the number of unmarked individuals observed (or, under a
 9394 Poisson observation model, the number of times unmarked individuals are observed), n_t
 9395 (t here and in the following description denotes a primary sampling occasion, for example,
 9396 a year or a season; for a single-season study we could easily drop this subscript) which are
 9397 approximated as a normal distribution (McClintock et al., 2006), or a normal distribution
 9398 left-truncated at 0 (McClintock et al., 2009a):
 9399

$$n_t \sim \text{Normal}(E(n_t), V(n_t))$$

9400 Although this is a simplification of the actual sampling process, McClintock et al. (2006)
 9401 found this normal distribution to be a satisfactory approximation, which allows N to enter
 9402 the model likelihood via $E(n_t)$ and $V(n_t)$.

9403 In the simplest model case without any variation in detection, the expected number
 9404 of resightings of unmarked individuals, $E(n_t)$, can be written as the number of unmarked
 9405 individuals times the expected number of detections of a single individual, which is the
 9406 mean or expected value of the underlying observation model:

$$E(n_t) = (N - m) * \theta \quad (19.1.1)$$

9407 where $\theta = K \times p$ for a Binomial observation model with K replicates and individual
 9408 detection probability p , or $\theta = \text{expected}/\text{average individual encounter rate } \lambda$ for a Poisson
 9409 observation model. Similarly, $V(n_t)$ depends on the underlying observation model and is
 9410 based on the parameters that determine the individual detection probability/encounter
 9411 rate. Combining these two components, N is directly incorporated into the joint likelihood
 9412 of the model.

9413 While these mark-resight models are very flexible, they share the shortcomings of
 9414 regular capture-recapture models when it comes to estimating population density (e.g.,
 9415 Chaps. 1 and 4). As long as resightings are collected across a network or array of locations,
 9416 however, they come with the same spatial information as recaptures in a regular
 9417 SCR study. In the following sections we will consider mark-resight sampling in the framework
 9418 of spatial capture-recapture. We will look at models for both known and unknown
 9419 numbers of marked individuals, and for imperfect individual identification of marks. In the
 9420 spatial framework, most of the information on model parameters comes from the marked
 9421 individuals. But in sec. 19.5 we will see that, analogous to the models we developed in
 9422 the previous Chapt. 18, the spatial correlation in counts of unmarked individuals also
 9423 contributes information about detection and movement.

19.2 KNOWN NUMBER OF MARKED INDIVIDUALS

9424 We begin with the easiest situation: a known number of individuals constituting a random,
 9425 representative sample from the population are marked and a series of resight samples are
 9426 conducted following marking. No marks (or marked animals) are lost between marking
 9427 and resighting, all individuals are correctly identified as marked or unmarked, and marked
 9428 individuals are 100 % correctly identified to individual level.

9429 Recall from Chapt. 18 that without any individual identity, the observed counts at
 9430 trap j and occasion k , n_{jk} , represent the sum of all latent individual detections at j and

9431 $k, \sum_{i=1}^N y_{ijk}$, where y_{ijk} are the latent individual encounter histories which we include as
 9432 variables (or missing data) in our MCMC scheme. We can model these counts as

$$n_{jk} \sim \text{Poisson}(\Lambda_j)$$

9433 where

$$\Lambda_j = \sum_{i=1}^M (\lambda_{ij})$$

9434 Under this formulation we do not need to update the individual y_{ijk} in our model, which
 9435 is more efficient in terms of computing. However, we can also formulate the model as
 9436 conditional on the latent y_{ijk} . This is useful because if we have m individually known
 9437 animals in our study population, than those m y_{ijk} are no longer latent, but fully observed
 9438 and can easily be included in the analysis to provide information on detection parameters.

9439 The formulation conditional on y_{ijk} basically brings us back to the original SCR model,
 9440 where individual site and occasion specific counts, y_{ijk} , are modeled as

$$y_{ijk} \sim \text{Poisson}(\lambda_{ij})$$

9441 and

$$\lambda_{ij} = \lambda_0 \exp(-d_{ij}^2 / (2\sigma^2))$$

9442 Unobserved y_{ijk} are essentially missing data and have to be updated as part of the
 9443 MCMC procedure. We can do that by using their full conditional distribution, which is
 9444 multinomial with sample size n_{jk} :

$$y_{ujk} \sim \text{Multinomial}(n_{jk}, \lambda_{uj})$$

9445 where u is an index vector of the $M - m$ hypothetical unmarked individuals.

9446 While in the non-spatial mark-resight analysis known individuals provide direct infor-
 9447 mation about individual detection probability (or rate), in the spatial setting they also
 9448 inform σ . Including known individuals into the analysis helps estimate model parameters
 9449 more accurately and precisely. We will address the relationship between the number of
 9450 marked individuals and accuracy of the estimated parameters in sec. 19.5.

9451 19.2.1 MCMC for a spatial mark-resight model

9452 Implementing a spatial mark-resight model in **JAGS** is not trivial, since the program
 9453 does not accept partially observed multivariate nodes (in this case the partially observed
 9454 individual encounter histories which we model as coming from a multinomial distribution).
 9455 Therefore, knowing how to write your own MCMC algorithm comes in extremely handy.
 9456 You will find that we only have to make relatively simple modifications to the MCMC
 9457 code for the model without any individual identification presented in Chapt. 18, which,
 9458 in turn, has much in common with the algorithms we developed for regular SCR models
 9459 in Chapt. 17. Essentially, since we observe individual detections for the marked part
 9460 of the population, we have to update only the unobserved part of \mathbf{Y} , and modify the
 9461 updating steps for z_i and ψ , the parameters introduced by data augmentation, to reflect
 9462 some contribution to our knowledge of these parameters from the m marked individuals.

9463 First, we set up an array to hold \mathbf{Y} , fill the first m rows of the array with the m
 9464 observed individual encounter histories, then update \mathbf{Y} for the unknown individuals only
 9465 (note that the code is set up so that n_{jk} contains both pictures of marked **and** unmarked
 9466 individuals at j and k):

```
9467 # set up placeholders and create vectors for marked and unmarked
9468 Y <- array(NA, c(M, J, K))
9469   nMarked <- nrow(y)
9470   marked <- rep(FALSE, M)
9471     marked[1:nMarked] <- TRUE
9472     Y[1:nMarked, , ] <- y
9473   z[marked] <- 1
9474   Ydata <- !is.na(Y)
9475   for (j in 1:J) {
9476     for (k in 1:K) {
9477       if (y[j, k] == 0) {
9478         Y[, j, k] <- 0
9479         next
9480       }
9481       unmarked <- !Ydata[, j, k]
9482       nUnknown <- n[j, k] - sum(Y[!unmarked, j,k])
9483       if (nUnknown < 0)
9484         browser()
9485       probs <- lam[, j] * z
9486       probs <- probs[unmarked]
9487       probs <- probs/sum(probs)
9488       Y[unmarked, j, k] <- rmultinom(1, nUnknown, probs)
9489     }
9490   }
```

9491 When we know the number of marked individuals in the population estimating N
 9492 is reduced to estimating u . Thus, we only need to estimate the z_i for $M - m$ unknown
 9493 individuals and the updater for z_i becomes:

```
9494 zUps <- 0
9495 seen <- apply(Y > 0, 1, any)
9496 for (i in 1:M) {
9497   if (seen[i] | marked[i])
9498     next
9499   zcand <- ifelse(z[i] == 0, 1, 0)
9500   ll <- sum(dpois(Y[i, , ], lam[i, ] * z[i], log = TRUE))
9501   llcand <- sum(dpois(Y[i, , ], lam[i, ] * zcand,
9502     log = TRUE))
9503   prior <- dbinom(z[i], 1, psi, log = TRUE)
9504   prior.cand <- dbinom(zcand, 1, psi, log = TRUE)
9505   if (runif(1) < exp((llcand + prior.cand) - (ll +
9506     prior))) {
```

```

9507     z[i] <- zcand
9508     zUps <- zUps + 1
9509   }
9510 }

```

Observe that while we skip the update of z_i for the “seen” individuals (where `seen=TRUE` for any individual observed at least once and `seen=FALSE` otherwise), `seen` is defined based on \mathbf{Y} and \mathbf{Y} is updated at each iteration, so the z_i for the observed but unmarked individuals are still updated.

Finally, our update for ψ needs to reflect that we are effectively only estimating U . In the full conditional beta distribution we have to replace M with $M - m$ and $\sum z$ with $\sum z - m$:

```

9518 psi<-rbeta(1,1+sum(w[!marked]),1+sum(!marked)-sum(w[!marked]))

```

The remainder of the code is essentially identical to the MCMC code for regular SCR models we developed in Chapt. 17. You can find the full MCMC code (including the modeling options we’ll discuss in the following sections) in the accompanying **R** package `scrbook` by invoking `scrPID`.

9523 19.2.2 Binomial encounter model

So far, we have only worked with Poisson encounter models for partially identifiable or unmarked populations. When we use a Bernoulli model instead, we have to make some changes to how we update the latent y_{ijk} , to ensure that a hypothetical individual receives at most a single observation at a given trap and occasion from the pool of n_{jk} pictures. Effectively, we move from a multinomial situation where the same individual could be drawn repeatedly, to a sampling without replacement situation (an individual drawn once at j and k cannot be drawn again); here is how we implement this in our MCMC algorithm:

```

9531 Y <- array(NA, c(M, J, K))
9532 #[...]
9533 for (j in 1:J) {
9534   for (k in 1:K) {
9535     if (y[j, k] == 0) {
9536       Y[, j, k] <- 0
9537     next
9538   }
9539   unmarked <- !Ydata[, j, k]
9540   nUnknown <- n[j, k] - sum(Y[!unmarked, j,k])
9541   if (nUnknown < 0)
9542     browser()
9543   probs <- lam[, j] * z
9544   probs <- probs[unmarked]
9545   probs <- probs/sum(probs)
9546   Y[unmarked, j, k] <- 0
9547   guys <- sample(which(unmarked), nUnknown, prob = probs)
9548   Y[guy, j, k] <- 1

```

Table 19.1. Posterior summaries of the spatial mark-resight model for Canada geese in North Carolina.

	Mean	SD	2.5%	50%	97.5%
σ , females	1.06	0.02	1.02	1.06	1.10
σ , males	1.13	0.02	1.09	1.13	1.18
λ_0	0.32	0.01	0.31	0.32	0.34
ψ	0.79	<0.01	0.73	0.79	0.86
ϕ	0.43	0.02	0.40	0.43	0.47
N	3720.81	121	3492	3717	3961
D	6.68	0.22	6.27	6.68	7.11

9549 }

9550 }

9551 Example: Canada geese in North Carolina

9552 We applied the spatial mark-resight model with a binomial encounter process to a
 9553 dataset of Canada goose resightings (Rutledge, 2012) XXXget full citation with LizXXX.
 9554 During the molt of 2008, 751 individual geese were captured and tagged with neck and
 9555 leg bands in Greensboro, North Carolina (Fig. 19.1). Geese were resighted at 87 different
 9556 locations on 81 resighting events over a period of 18 months. In addition to the banded
 9557 geese, the number of unmarked geese was recorded during each resighting event. Here,
 9558 we only looked at a subset of the data, from mid July to the end of October 2008, which
 9559 corresponds to the first part of the post-molt season, before migratory Canada geese arrive
 9560 in North Carolina. During this time frame, 746 of the 751 marked geese were known to
 9561 be alive. Of those, 654 were resighted 3994 times at 40 different sites. In addition, 7944
 9562 sightings of unmarked geese were recorded at 48 sites.

9563 In this model, we also allowed σ to vary between males and females. We augmented
 9564 the data set with $4500 - m$ all-zero encounter histories, ran 50000 MCMC iterations and
 9565 removed a burn-in of 1000 iterations. We provide all the data (`data('canadageese')`)
 9566 and functions (`pIDgeese`) for you to repeat this analysis but be aware that given the large
 9567 data set it will take days to do so. The **R** code to set up the data and run 5000 iterations
 9568 of the goose model is given as an example on the help page for `pIDgeese`. The model
 9569 results, including the derived parameter density (D) in individuals per km^2 are shown in
 9570 Tab. 19.2.2.

9571 We see that credible intervals of estimates are pretty narrow. Take, for example, σ for
 9572 males and females: Although they differ only by 0.08, there is barely any overlap between
 9573 the respective credible intervals, surely an effect of the large data set. The parameter ϕ in
 9574 this model is the probability of being a male, a measure of the sex ratio of the population,
 9575 which is close to 1:1.

19.3 UNKNOWN NUMBER OF MARKED INDIVIDUALS

9576 Now let us consider the case where we do not know the exact number of tagged individuals
 9577 available for resighting so that we have to capture an individual at least once to be sure
 9578 that it is available. Unless we have a direct means of confirming the number of marked

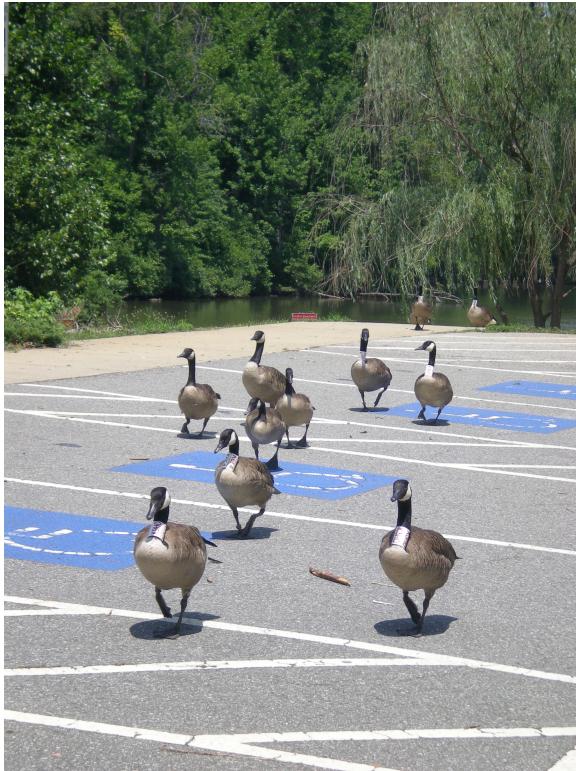


Figure 19.1. Banded and unbanded Canada geese in a parking lot in Greensboro, North Carolina. (Photo credit: M.E. Rutledge, NCSU Canada goose project)

9579 animals available for resighting, treating this number as unmarked is probably more realistic
9580 in most circumstances. As a consequence of not knowing the exact number of marked
9581 individuals, we cannot observe all-zero encounter histories. When using maximum likelihood
9582 inference, this situation requires a model where detection rates of known individuals
9583 are modeled using a zero-truncated distribution (McClintock et al., 2009a). If we did not
9584 account for the fact that zeros are unobservable, our estimates of detection rates would
9585 be artificially inflated and estimates of population size would be negatively biased.

9586 Working with zero-truncated distributions in a spatial mark-resight setting is less
9587 straight-forward than for non-spatial mark-resight. A marked individual only has to show
9588 up once, anywhere on the resighting array, for us to know that it is there. When resightings
9589 are pooled across the entire sampling grid, then the total individual counts $\sum_j y_{ij}$ have
9590 to be > 0 for all resighted individuals and a zero-truncated distribution can be used to
9591 model these counts. However, we are concerned with trap-specific encounters, y_{ij} , which
9592 can easily be 0 for a resighted individual, as long as a single y_{ij} is > 0 . Thus, the zero-
9593 truncation does not apply to the individual and trap specific counts we observe, but only

9594 to the sum of these counts over all traps.

9595 As an alternative to a zero-truncated distribution, in a Bayesian framework, we can
 9596 make use of data augmentation to estimate the number of marked individuals¹. In the
 9597 previous example, where we knew the number of marked individuals, we separate those
 9598 individuals from the augmented population by fixing their z_i at 1 and letting ψ refer only
 9599 to the unmarked population, $M - m$. All we have to do in the spatial mark-resight model
 9600 with unknown number of marked individuals is to let our marked individuals be part of
 9601 the augmented population again, analogous to the situation in regular SCR models:

```
9602     psi <- rbeta(1, 1 + sum(z), 1 + M - sum(z))
```

9603 Whether you have a known or an unknown number of marked individuals is included
 9604 as an option in **scrPID**.

9605 A simulation example

9606 For illustration purposes we simulated a data set with $N = 80$ individuals randomly
 9607 distributed across a state space of 10x10 units. Of those, we randomly choose 40 to be
 9608 marked and identifiable, and then simulate encounter data for both marked and unmarked
 9609 individuals on an 8x8 grid with unit spacing over $K = 5$ occasions, with $\sigma = 0.5$ and $\lambda_0 =$
 9610 0.5, adopting a Poisson encounter process. To do so we use the **scrbook** function **sim.data**,
 9611 which also allows you to create data sets from a Binomial observation process, known
 9612 number of marked individuals, and with telemetry locations (sec. 19.6) or individual
 9613 identification rate < 100 % (sec. 19.4). We analyzed the simulated data both assuming
 9614 we do not know the total number of marked animals in our state space, and assuming we
 9615 do know this number, using the **scrPID** function and running 20000 iterations. You can
 9616 repeat the analysis by executing the R code below.

```
9617 set.seed(2501)
9618
9619 #set input values
9620 N=80
9621 lam0=0.5
9622 knownID=40
9623 rat=0.8
9624 sigma=0.5
9625 K=5
9626
9627 #create grid and state space
9628 coords<-seq(0,7, 1)
9629 grid<-expand.grid(coords, coords)
9630 trapmat<-as.matrix(grid)
9631 buff<- 3*sigma
9632 xl<-min(trapmat[,1])-buff
9633 xu<-max(trapmat[,1])+buff
9634 yl<-min(trapmat[,2])-buff
```

¹For the interested reader, McClintock and Hoeting (2010) implement a non-spatial mark-resight model with a binomial observation model in a Bayesian framework using data augmentation

```

9635 yu<-max(trapmat[,2])+buff
9636 xlims=c(xl, xu)
9637 ylims=c(yl,yu)
9638 area<-(xu-xl)*(yu-yl)
9639
9640 #simulate data
9641 dat<-sim.pID.data(N=N, K=K, sigma=sigma, lam0=lam0, knownID=knownID,
9642 X=trapmat, xlims=xlims, ylims=ylims, obsmod= "pois",
9643 nmarked="unknown", rat=1, tel =0, nlocs=0)
9644
9645 #create initial values function for scrPID, set M and tuning parameters
9646 inits<-function(){list(S=cbind(runif(M, xlims[1], xlims[2]),
9647 runif(M, ylims[1], ylims[2])), lam0=runif(1, 0.4, 0.6),
9648 sigma=runif(1, 0.4, 0.6), psi=runif(1, 0.4, 0.6))}
9649 M<-160
9650 delta=c(0.1, 0.01, 2)
9651
9652 #run model, first m=unknown, then m=known
9653 mod<-scrPID(n=dat$n, X=trapmat, y=dat$Yobs, M=M, obsmod = "pois",
9654 nmarked="unknown", niters=20000, xlims=xlims, ylims=ylims,
9655 inits=inits(), delta=delta ) )
9656 mod2<-scrPID(n=dat$n, X=trapmat, y=dat$Yobs, M=M, obsmod = "pois",
9657 nmarked="known", niters=20000, xlims=xlims, ylims=ylims,
9658 inits=inits(), delta=delta ) )
9659

```

9660 Looking at the data, we see that of the 40 marked animals, 26 were recorded at least
9661 once. In terms of data that means that in the second model, where we know m , we have
9662 14 observed all-zero encounter histories that we cannot use in the model where we assume
9663 m is not known. This reduction in data is reflected in the model results (Tab. 19.3). The
9664 estimate of N for the unknown- m model shows some positive bias, although the 95 % BCI
9665 still includes the true value of 80. Thus, while we can formally account for the fact that we
9666 often do not know the number of marked individuals in the state space, we clearly loose
9667 quite a bit of accuracy and precision. It would be an interesting little project to quantify
9668 this loss in accuracy and precision in a small simulation study.

19.4 IMPERFECT IDENTIFICATION OF MARKED INDIVIDUALS

9669 Often during resighting, it may be possible to see that an individual is tagged but impos-
9670 sible to determine its individual identity. In such a situation in addition to the y_{ijk} and
9671 n_{jk} , we also have site and occasion specific counts of marked but unidentified individuals,
9672 r_{jk} . Here, the individual encounter histories of marked animals are incomplete, and if we
9673 used these incomplete data to inform the detection parameter of the model, we would run
9674 the risk of underestimating detection/trap encounter rate and overestimating abundance.
9675 Some non-spatial mark-resight models do not require that marked animals be identified
9676 individually, as long as the marking status can be observed unambiguously, but ignoring

Table 19.2. Posterior summaries of the spatial mark-resight model for a simulated data set analyzed with number of marked individuals m assumed to be unknown and known. First 500 iterations discarded as burn-in.

		Mean	SD	2.5%	97.5%
m unknown	σ	0.521	0.029	0.470	0.583
	λ_0	0.4679	0.069	0.346	0.602
	ψ	0.541	0.070	0.411	0.684
	N	86.612	9.386	70	107
m known	σ	0.514	0.0284	0.4638	0.5750
	λ_0	0.550	0.077	0.403	0.707
	ψ	0.332	0.066	0.212	0.468
	N	79.525	6.149	69	93

9677 individual level information means that we cannot accommodate heterogeneity in detection
 9678 (McClintock and White, 2012). In a spatial framework we could ignore marked and
 9679 unmarked status completely and apply the model by Chandler and Royle (In press) we
 9680 discussed in Chapt. 18. But, that would mean losing important information on individual
 9681 detection and movement. Therefore, being able to retain the individual identity of records
 9682 that can be identified while at the same time accounting for imperfect identification of
 9683 marked individuals is extremely useful.

9684 McClintock et al. (2009a,b) suggest an intuitive means of correcting for this bias in a
 9685 non-spatial model framework when dealing with a Poisson encounter model (or sampling
 9686 with replacement). When marked but unknown resightings are part of the data, the
 9687 expected number of records of unmarked individuals at time t , n_t , changes from Eq.
 9688 19.1.2 to:

$$E(n) = (N - m)\lambda + \eta/m$$

9689 Here, λ is the individual encounter rate estimated from the known resighted individuals
 9690 and η is the number of records of marked but unidentified individuals. So, because the
 9691 observed λ is known to be too low, the average number of unidentified pictures per known
 9692 individual is added as a correction factor. This procedure assumes that the inability to
 9693 identify a marked individual occurs at random throughout the population, which seems
 9694 to be a reasonable assumption under most circumstances.

9695 We can relatively easily translate this concept to our spatial mark-resight models. In
 9696 the spatial model framework we are interested in the individual and trap specific encounter
 9697 rate, λ_{ij} . Further, we do not look at the sum of all records of unmarked individuals, but
 9698 formulate the model conditional on the latent individual encounter histories. Thus, instead
 9699 of using η/m as a correction factor, we need something that applies at the individual and
 9700 trap level. If we take the sum of all correctly identified records of marked individuals,
 9701 $\sum y_c$ and divide it by the total number of records of marked individuals, $\sum y_m$, we get
 9702 the average rate of correct individual identification for marked individuals, say, c :

$$c = \sum y_c / \sum y_m$$

9703 We could then apply c as a correction factor for λ_0 for the marked individuals.

9704 A more formal, model-based way to specify c is by assuming that

$$\sum y_c \sim \text{Binomial}(\sum y_m, c)$$

9705 and estimating c as another model parameter, so that we account for the uncertainty about
 9706 it. If we choose an uninformative (and conjugate) beta(1, 1) prior for c , we can update it
 9707 directly from its full conditional distribution, which is beta($1 + \sum y_c, 1 + (\sum y_m - \sum y_c)$),
 9708 within our MCMC algorithm.

9709 For the marked individuals we can then multiply λ_0 with c to account for the fact that
 9710 we observe incomplete individual encounter histories. Since we don't have this identifica-
 9711 tion issue for unmarked individuals, their baseline trap encounter rate remains as before
 9712 simply λ_0 (or in other words, their c equals 1). Observe that now, in addition to assuming
 9713 that failure to identify tagged individuals occurs at random throughout the population,
 9714 we also assume that it occurs at random throughout space, i.e. our success of identifying
 9715 a tagged individual does not depend on the trap we encounter it in. Incomplete individual
 9716 identification of marked individuals is included as an option in the `scrPID` function and
 9717 we show an example of using c in an analysis in sec. 19.6.

9718 **Imperfect individual identification and unknown number of marks.** The ap-
 9719 proach described above works only if the number of marked individuals is known because,
 9720 in that case, we can observe the all-zero encounter histories of marked individuals and
 9721 know that all augmented individuals have to be unmarked individuals. If the number of
 9722 marked animals is unknown, on the other hand, some of the augmented individuals may
 9723 well be marked individuals we never observed. For those individuals we should multiply
 9724 λ_0 with c , but we don't know who (or how many) they are. As of this moment we have
 9725 not implemented a model with unknown number of marked individuals and imperfect
 9726 identification of marks. It seems like one strategy to tackle that problem would be to
 9727 estimate the number of marked and unmarked individuals separately, using two sets of
 9728 data augmentation (i.e., estimating ψ_{marked} and ψ_{unmarked}), but with shared detection
 9729 parameters, σ and λ_0 .

9730 As long as individuals are identified based on the same type of tags the assumption that
 9731 failure to identify marked individuals occurs at random throughout the population should
 9732 be valid. The assumption that failure to identify marked individuals occurs at random in
 9733 space could be violated, for example when spatially varying habitat conditions influence
 9734 the ability to recognize individual tags, or when an observer effect influences individual
 9735 identification rates. While we haven't ourselves experimented with it, we believe that the
 9736 approach described above could readily be extended to account for these differences. For
 9737 example, identification rates could be calculated separately for different observers, or be
 9738 modeled as functions of habitat covariates. As an alternative to the approach we present
 9739 here, model development could explore assigning records of marked but unidentified indi-
 9740 viduals to marked individuals in a fashion similar to how unmarked records are assigned
 9741 to hypothetical individuals in this model, namely, based on the location of the record and
 9742 the estimates of home range centers of marked individuals. While this is computationally
 9743 more advanced it would make full use of the spatial information of the unmarked records.

19.5 HOW MUCH INFORMATION DO MARKED AND UNMARKED INDIVIDUALS CONTRIBUTE?

9744 It is intuitive that having marked individuals in the study population should lead to more
 9745 accurate and precise parameter estimates than when no individuals are identifiable. To
 9746 evaluate how strongly adding marked individuals to a population improves parameter

estimates, Chandler and Royle (In press) performed a simulation study. They used a 15×15 trapping grid and simulated detection data of $N = 75$ individuals in a 20×20 units state-space over $k = 5$ occasions with $\sigma = 0.5$ and $\lambda_0 = 0.5$. They generated 100 datasets each for $m = (0, 5, 15, 25, 35)$ where m is the known number of marked individuals randomly sampled from the population.

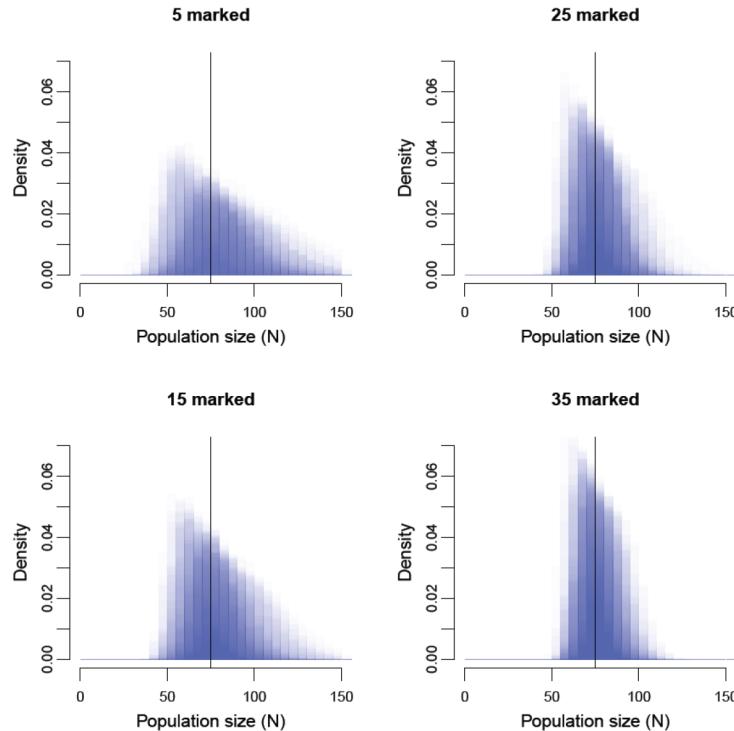


Figure 19.2. Overlaid posterior distributions of N from 100 simulations for four levels of marked individuals.

Without any marked individuals in the population, the posterior distribution of N turned out to be highly skewed, but its mode was still an approximately unbiased point estimator of N . As anticipated, posterior precision increased substantially with the proportion of marked individuals (Tab. 19.3 and Fig. 19.2). The posterior mode was approximately unbiased as a point estimator, and the relative root-mean squared error decreased from 0.246 when no individuals were marked to 0.085 when 35 individuals were marked (Tab. 19.3). Coverage was nominal for all values of m and posterior skew greatly diminished with increasing m (Tab. 19.3).

As we saw in the previous chapter, the spatial correlation in unmarked counts can be sufficient to obtain estimates of movement and detection parameters. However, only

Table 19.3. Posterior mean, mode, and associated relative RMSE for simulations in which m of $N=75$ individuals were marked. One hundred simulations of each case were conducted.

	Parameter	Mean	rRMSE	Mode	rRMSE	BCI
m=0	N	85.866	0.259	77.720	0.242	0.950
	λ_0	0.506	0.180	0.488	0.182	0.960
	σ	0.495	0.115	0.486	0.113	0.960
m=5	N	80.898	0.184	76.360	0.182	0.970
	λ_0	0.510	0.178	0.494	0.180	0.950
	σ	0.496	0.089	0.488	0.086	0.970
m=15	N	79.028	0.148	76.250	0.147	0.950
	λ_0	0.508	0.163	0.494	0.164	0.950
	σ	0.496	0.073	0.492	0.071	0.970
m=25	N	77.765	0.114	75.810	0.113	0.950
	λ_0	0.511	0.153	0.498	0.157	0.950
	σ	0.496	0.067	0.493	0.065	0.940
m=35	N	76.446	0.085	74.900	0.085	1.000
	λ_0	0.513	0.142	0.501	0.144	0.950
	σ	0.497	0.056	0.493	0.057	0.940

marked and thus identifiable individuals provide us with direct information about these parameters and may well dominate estimates. To single out the contribution of marked and unmarked individuals to parameter estimates, we re-ran the same simulations but let σ and λ_0 be updated based solely on the data of marked individuals. Results are summarized in Tab. 19.4. We see that if we update λ_0 and σ based on marked individuals only, estimates of these parameters are more biased and less precise. For estimates of N , especially for $m=5$ and $m=15$, we observe a stronger positive bias, lower accuracy and considerably lower BCI coverage as compared to when both marked and unmarked individuals contribute to parameter estimates (Tab. 19.4). Thus, unmarked individuals do actually contribute noticeably to estimating model parameters.

Table 19.4. Posterior mean, mode, and associated relative RMSE for simulations in which m of $N=75$ individuals were marked and unmarked individuals did not contribute to estimating λ_0 and σ . One hundred simulations of each case were conducted.

	Parameter	Mean	RMSE	Mode	RMSE	BCI
m=5	N	88.621	0.369	83.139	0.421	0.810
	λ_0	1.255	1.247	0.606	1.148	0.950
	σ	0.472	0.252	0.426	0.333	0.910
m=15	N	81.031	0.192	78.361	0.175	0.820
	λ_0	0.535	0.281	0.476	0.284	0.970
	σ	0.503	0.109	0.490	0.107	0.940
m=25	N	78.206	0.129	76.594	0.123	0.920
	λ_0	0.531	0.204	0.496	0.202	0.960
	σ	0.497	0.081	0.489	0.084	0.950
m=35	N	76.833	0.099	75.422	0.096	0.940
	λ_0	0.528	0.192	0.505	0.186	0.940
	σ	0.499	0.069	0.493	0.070	0.960

19.6 INCORPORATING TELEMETRY DATA

As we expected, parameter estimates of spatial mark-resight models get better the more marked individuals we have in our study population. While this is great advice in theory, it may not be very helpful in practice, especially when dealing with animals that are hard or somewhat dangerous to capture, such as large carnivores. Oftentimes, studies involving the physical capture of such animals will employ telemetry tags in order to learn about the study species' spatial ecology and behavior. In the context of spatial mark-resight models, the actual locational data collected by telemetry tags can provide detailed information on individual location and movement, and being able to incorporate this information directly into the SMR model should improve estimates of these parameters, especially when resighting information is sparse.

So how could we combine resighting data and telemetry data in a unified mark-resight model? Recall that the basic SCR model underlying all the SMR models we discuss here uses a half-normal detection function. By using this function, we can relate the parameters σ and \mathbf{s}_i directly to those from a bivariate normal model of space usage, with mean = \mathbf{s}_i , and variance-covariance matrix Σ , where the variance in both dimensions is σ^2 and the covariance is 0. Ordinarily, these parameters are estimated directly from the spatial distribution of individual recaptures/resightings. Telemetry data, however, provide more detailed information on individual location and movement, since the resolution and extent of the data are not limited by the trapping grid and potentially more locations can be accumulated through telemetry than resighting (depending on the monitoring frequency and resighting rates of individuals).

By assuming that the R locations of individual i , \mathbf{l}_i (consisting of a pair of x and y coordinates, l_{ix} and l_{iy}), are a bivariate normal random variable:

$$\mathbf{l}_i \sim \text{Normal}_2(\mathbf{s}_i, \Sigma)$$

we can estimate σ as well as \mathbf{s}_i for the collared individuals directly from telemetry locations, using their full conditional distributions:

$$[\sigma | \mathbf{l}, \mathbf{s}] \propto \left\{ \prod_{i=1}^m \prod_{r=1}^R \frac{1}{2\pi\sigma^2} \exp \left(-1/2 \left[\frac{(l_{irx} - s_{ix})^2}{\sigma^2} + \frac{(l_{iry} - s_{iy})^2}{\sigma^2} \right] \right) \right\} * [\sigma]$$

and

$$[\mathbf{s}_i | \mathbf{l}, \sigma] \propto \left\{ \prod_{r=1}^R \frac{1}{2\pi\sigma^2} \exp \left(-1/2 \left[\frac{(l_{irx} - s_{ix})^2}{\sigma^2} + \frac{(l_{iry} - s_{iy})^2}{\sigma^2} \right] \right) \right\} * [\mathbf{s}_i]$$

Under the standard mark-resight assumption that marked individuals are a representative sample of the population, the estimate of σ can be applied for the entire population. For the unmarked individuals \mathbf{s}_i are estimated as described before conditional on their latent encounter histories.

R makes it easy to implement the update of σ and \mathbf{s}_i based on telemetry data and the above described full conditionals within our existing MCMC algorithm. We replace the current updating step for σ with:

```
9805 #ntot = number of telemetry-tagged individuals
9806 #locs = list of length ntot; each element is a matrix
```

```

9807 #with telemetry locations
9808 #telID = vector with identifier for telemetry-tagged
9809 #individuals
9810
9811 sigma.cand <- rnorm(1, sigma, delta[1])
9812 if (sigma.cand > 0) {
9813
9814 llsig<-llsig.cand<-rep(NA, ntot)
9815
9816 for (x in 1:ntot) {
9817 lls[x]<-sum(dmvnorm(x=locs[[x]],mean=c(S[telID[x],1],S[telID[x],2]),
9818 sigma=cbind(c(sigma^2,0), c(0,sigma^2)), log=T))
9819 lls.cand[x]<-sum(dmvnorm(x=locs[[x]],mean=c(S[telID[x],1],S[telID[x],2]),
9820 sigma=cbind(c(sigma.cand^2,0), c(0,sigma.cand^2)), log=T))
9821 }
9822 if(runif(1) < exp( sum(lls.cand) - sum(lls) ) ){
9823 sigma<-sigma.cand
9824 lam <- lam0*exp(-(D*D)/(2*sigma.cand*sigma.cand))
9825 }
9826 }

```

9827 For the s_i we use an analogous updater for the telemetry-tagged individuals and the
9828 regular updater for individuals without associated telemetry location information. A full
9829 example can be found in the **R** package **scrbook**, by calling **scrPID.tel**. Note that not
9830 all marked individuals need to be telemetry-tagged, but telemetry data used on the model
9831 should correspond to the period over which resighting surveys were conducted (as we
9832 discussed in Chapt. 5, both the s_i and σ should only be interpreted against the specific
9833 sampling period). Further, this approach of incorporating telemetry data into a spatial
9834 mark-resight model can easily be extended to update σ and s conditional on both resighting
9835 and telemetry data and applies equally to regular SCR models where all individuals are
9836 identifiable.

9837 **Example: Raccoons on the Outer Banks of North Carolina**

9838 Solmann et al. (2012) applied a spatial mark-resight model with telemetry data to
9839 a camera-trap and radio-telemetry data set from the raccoon population on South Core
9840 Banks, a barrier island within Cape Lookout National Seashore, North Carolina. Between
9841 May and September 2007, 131 raccoons were marked with dog collars and large indi-
9842 vidual numbered cattle tags; 44 of these tagged individuals were equipped with radio
9843 collars. Collared individuals were located using a VHF receiver and antenna, and their
9844 locations were estimated approximately weekly. Twenty camera traps were set up along
9845 the length of South Core Banks and camera trapping data collected between October 1
9846 2007 to January 22 2008 constituted the resighting data in this analysis. During this
9847 period 104 marked individuals, 38 radio-collared, were alive and available for resighting
9848 with camera traps.

9849 The state-space \mathcal{S} was defined as the entire area of South Core Banks island. A
9850 change in the number of photocaptures over the course of the study suggested a variation
9851 of detection rate with time. Since date recording in cameras malfunctioned, photographic



Figure 19.3. Camera trap picture of a raccoon marked with a cattle tag that cannot be read to determine individual identity. Taken on South Core Banks, North Carolina. (*Photo credit: Arielle Parsons*)

records could only be assigned to the time interval between subsequent trap checks, and these intervals between checks are referred to as sampling occasions. These occasions ranged from 2 to 43 days; λ_0 was standardized to 7-day intervals and allowed to change with sampling occasion. Since not all pictures of marked raccoons could be identified to the individual level, the authors applied the correction factor c as described in sec. 19.4, estimated separately for each occasion.

Camera-traps recorded 117 pictures of unmarked raccoons, 33 pictures of 18 marked and identifiable raccoons, and 49 records of marked but not individually identifiable individuals (Fig. 19.3). An average of 16.32 telemetry locations (SD 4.91) were collected for each of the 38 collared individuals. Raccoon abundance on the island was estimated at 186.712 (SE 14.810) individuals, which translated to a density of 8.291 (SE 0.658) individuals per km^2 . Parameter estimates are listed in Tab. 19.5.

In this study, although a large number of raccoons were tagged, photographic data of these tagged individuals were surprisingly sparse. Analysis of the photographic data set without the telemetry data did not render usable estimates as parallel Markov chains did not converge. One reason for the relatively sparse data was the camera trap study design: traps were spaced on average 1.77 km apart, which is about 3.5 times σ . Consequently, very few individual raccoons were photographed at more than one trap. Under these

Table 19.5. Summary statistics of parameter estimates from spatial mark-resight model for raccoon camera trapping and telemetry data. Baseline trap encounter rate λ_0 was standardized to 7-day intervals; λ_0 and the probability of identifying a picture of a marked individual, c , were allowed to vary among the 6 sampling occasions (t); σ is estimated from telemetry data of 38 radio-collared individuals.

	Mean (SE)	2.5%	50%	97.5%
σ	0.491 (0.010)	0.472	0.491	0.512
λ_0 (t=1)	0.237 (0.045)	0.158	0.234	0.335
λ_0 (t=2)	0.397 (0.081)	0.257	0.391	0.573
λ_0 (t=3)	0.108 (0.028)	0.061	0.105	0.170
λ_0 (t=4)	0.296 (0.073)	0.174	0.289	0.459
λ_0 (t=5)	0.032 (0.011)	0.015	0.030	0.056
λ_0 (t=6)	0.031 (0.009)	0.016	0.030	0.052
c (t=1)	0.545 (0.085)	0.377	0.546	0.709
c (t=2)	0.389 (0.112)	0.184	0.385	0.616
c (t=3)	0.294 (0.107)	0.110	0.286	0.523
c (t=4)	0.375 (0.162)	0.099	0.364	0.710
c (t=5)	0.375 (0.161)	0.099	0.364	0.709
c (t=6)	0.300 (0.138)	0.075	0.287	0.600
N	186.712 (14.810)	162	185	220
D	8.291 (0.658)	7.194	8.215	9.769

circumstances, the telemetry data provide the necessary spatial information to estimate σ and the activity centers of individual animals and thus make other model parameter estimable. Similarly, in a camera-trapping study on Florida panthers (*Puma concolor coryi*), Sollmann et al. (in revision), including telemetry data from the 3 individuals that were collared and known to use the study area resulted in density estimates with considerably higher precision as compared to preliminary estimates *without* telemetry location data, reducing the width of the 95 % BCI by about 60 %. Such improvements in precision of estimates is especially important when we are interested in changes in the population over time.

19.7 SUMMARY AND OUTLOOK

In this chapter we combined SCR models and the spatial model for unmarked populations to derive a spatial mark-resight model, which accommodates that part of the population is individually identifiable, usually through artificial tags. The basic model with known number of marked individuals and 100 % individual identification of marked is easily modified for situations where the number of marked individuals is unknown, or where marked animals can sometimes not be identified to individual level. As expected, having marked individuals in the study population improved accuracy and precision of parameter estimates when compared to fully unmarked populations, but we also saw that the spatial counts of unmarked individuals still contribute information to parameter estimates. Finally, we present an approach of how to incorporate telemetry location data into the spatial mark-resight model to inform estimates of σ and activity centers. Especially for difficult-to-study, cryptic species where often only a small sample of the population can be

9891 tagged this enables researchers to make optimal use of all existing data and obtain robust
9892 density estimates without the need for additional invasive methods. Just as SCR, the
9893 spatial mark-resight model framework is flexible to account for a variety of factors that
9894 may influence individual movement and detection, as well as survey-related parameters,
9895 and we saw one example for the Canada geese, where σ was sex-specific.

9896 Spatial mark-resight models are a fairly new development and much remains to be ex-
9897 plored. We mentioned the assignment of marked but unidentified records to actual marked
9898 individuals based on their spatial location, which provides some (though imperfect) infor-
9899 mation of their identity (sec. 19.4). Similarly, records where the marked status cannot be
9900 determined could potentially be included in the model as some form of overall correction
9901 factor on detection. GPS telemetry devices and their ability to collect location data with
9902 much higher frequency offer the opportunity to assign records of collared animals to indi-
9903 viduals based on how close to a given camera the collared individuals were, both in space
9904 and time. In this scenario, individual identity itself could be expressed probabilistically,
9905 leading to an SMR model accounting for potential misidentification. All these possible
9906 extensions can tailor SMR models to specific survey techniques. As such, the approach is
9907 applicable to a wide range of population estimation problems when dealing with animals
9908 that cannot be identified based on natural marks.

20

9909

9910

9911 2012: A SPATIAL CAPTURE-RECAPTURE 9912 ODYSSEY

9913 Capture recapture methods have been a cornerstone of ecological modeling and analysis
9914 for decades. Yet there are essentially no real capture-recapture data sets that come *without*
9915 auxiliary spatial information about location of capture (but sometimes such information
9916 is thrown into the trashcan).

9917 The big point is that we provide a framework for spatial analysis of animal populations
9918 from individual encounter data: MOVEMENT, SPACE USAGE, SPATIAL VARIATION
9919 IN DENSITY – much to be done: how do individuals interact? how is space usage
9920 changing over time, etc...

9921 Topics to discuss here:

9922 (1) Strauss process model (2) Need for general purpose software.... all of the spatial
9923 stuff + open populations in one big model. (3) Efficient computation is still an issue. (4)
9924 Fit and model selection will continue to be important practical issues.

20.1 10 THESIS OR DISSERTATION TOPICS

9925 Future research directions:
9926 Modeling dynamics of the point process. Transient individuals. Dispersal. Things like
9927 that.
9928 Calibration of GoF under meaningful alternatives
9929 Calibration of AIC/DIC and efficacy study
9930 Models for non-uniform point processes that exhibit clustering or repulsion
9931 no-marking model + RSF
9932 occupancy and counts data + SCR data (AOAS and Sollmann et al.)
9933 Spatial genetics – can use SCR to study gene flow, related things....
9934 SCR on dendritic networks (streams and trails).

20.2 THREE DIMESIONAL SPACE

9935 Throughout this book we have treated space as two-dimensional, meaning that activity
9936 centers are assumed to occur on the real plane. This approximation of reality is reasonable
9937 for many terrestrial species, but aquatic organisms, especially marine animals move about
9938 in three-dimensional space. Treating space as three-dimensional could also conceivably
9939 be useful in studies of flying organisms or species that use multiple strata of tall forests;
9940 however, we suspect that two dimensional models of space should suffice in such contexts.
9941 Regardless, a three-dimensional view of space requires that activity centers s_i are indexed
9942 by x, y, z coordinates. In theory, this presents no problem whatsoever. In practice, estima-
9943 tion based on integrated likelihood methods must involve a three-dimensional integration.
9944 This will clearly be more computationally demanding, but it should be possible using
9945 packages such as `R2Cuba`.

20.3 GREGARIOUS SPECIES

9946 Many social species move about in large groups rather than as single individuals. Even
9947 species regarded as solitary often join family groups for some portion of their life cycle.
9948 The consequences of gregariousness?? are x-fold....
9949 To account for this, we change our definition of s_i from the location of an individual's
9950 activity center, to the location of a group's activity center. We then expand our model to
9951 include a submodel for group size, and we can estimate both the density of group activity
9952 centers and total population size.

9953

Part V

9954

9955

Appendices

9956 **APPENDIX I - USEFUL SOFTWARE AND**
9957 **R PACKAGES**

9959 Throughout this book we have used a suite of software and R packages, all of which are
9960 freely available online. To make life a little easier for you, here we provide you with a list
9961 of all software and R packages, download links and some (hopefully) helpful tips regarding
9962 their installation.

20.4 WINBUGS

9963 Although **WinBUGS** (Gilks et al., 1994) is becoming increasingly obsolete with the
9964 faster and more flexible **OpenBUGS** and **JAGS**, there are still situations in which
9965 the program comes in handy. The .exe file can be downloaded from <http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/contents.shtml>. On 32 bit machines you can just go ahead
9966 and double-click on the .exe file and follow the installation instructions on the screen. On
9967 64 bit machines, according to the BUGS project you should download a zip file (from the
9968 same page) and unzip it into a folder of your choice. There are a couple of additional
9969 steps to make BUGS run. First, you need to obtain a key (which is free and valid for
9970 life) here: 'http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/WinBUGS14_immortality_key.txt'. The key comes with instructions on how to activate it. Second, you need
9971 to update the basic **WinBUGS** version to the most current one (which is from August
9972 2007) following the instructions given here: 'http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/WinBUGS14_cumulative_patch_No3_06_08_07_RELEASE.txt'. **WinBUGS** is
9973 ready to use after quitting and re-opening it. Remember that **WinBUGS** only runs on
9974 Windows machines. Also, there appears to be a problem installing the program in Vista,
9975 although we have no personal experience with this.

9979 20.4.1 WinBUGS through R

9980 While you can run **WinBUGS** as a standalone application, we recommend you access
9981 it from within **R** using the package **R2WinBUGS** (Sturtz et al., 2005), so you can conve-
9982 niently process your output, make graphs etc. **R2WinBUGS** also allows you to run mod-
9983 els in **OpenBUGS** (see below). You can install the package from within **R** directly
9984 from a cran mirror. In addition to the usual package help document (<http://cran.r-project.org/web/packages/R2WinBUGS/R2WinBUGS.pdf>) you can also download a short
9985 manual with some examples ('http://voterview.com/bayes_beach/R2WinBUGS.pdf').

20.5 OPENBUGS

9987 **OpenBUGS** is the up-to-date version of **WinBUGS** and can be downloaded here:
 9988 ''<http://www.openbugs.info/w/Downloads>'' (Windows, Mac and Linux versions are
 9989 available). The name '**OpenBUGS**' refers to the software being open source, so users
 9990 do not need to download a license key, like they have to for **WinBUGS** (although the
 9991 license key for **WinBUGS** is free and valid for life). For Windows, install by double-
 9992 clicking on the .exe file and following the instructions on the installer screen. Compared
 9993 to **WinBUGS**, **OpenBUGS** has more built-in functions. The method of how to deter-
 9994 mine the right updater for each model parameter has changed and the user can manually
 9995 control the MCMC algorithm used to update model parameters. Several other changes
 9996 have been implemented in **OpenBUGS** and a detailed list of differences between the two
 9997 **BUGS** versions, can be found at <http://www.openbugs.info/w/OpenVsWin>. We have
 9998 encountered convergence problems with simple scr models in this program. There is an
 9999 extensive help archive for both **WinBUGS** and **OpenBUGS** and you can subscribe to
 10000 a mailing list, where people pose and answer questions of how to use these programs at
 10001 <http://www.mrc-bsu.cam.ac.uk/bugs/overview/list.shtml>

10002 20.5.1 OpenBUGS through R

10003 Like **WinBUGS**, **OpenBUGS** can be used as a standalone application or through **R**.
 10004 There are several packages that allow **R** to interface with **OpenBUGS**, all of which can
 10005 be installed directly from a cran mirror:

10006 **R2WinBUGS**: One of the options in the `bugs()` call is `program`, which lets you specify either
 10007 **WinBUGS** or **OpenBUGS**. This is a convenient option because after having worked
 10008 through some of this book you will likely be familiar with the format of `bugs()` output
 10009 and other functions of the **R2WinBUGS** package.

10010 **R2openBUGS**: **R2openBUGS** (Sturtz et al., 2005) is very similar to, and actually based on,
 10011 **R2WinBUGS** and it is unclear to us what can be gained by using the former over the latter.
 10012 Arguments of the `bugs()` call differ slightly between the two packages and given that
 10013 **R2WinBUGS** allows for the use of both **OpenBUGS** and **WinBUGS** it is probably easiest
 10014 to stick with it.

10015 **BRugs**: **BRugs** (Thomas et al., 2006) can be installed from within **R** directly from a cran
 10016 mirror. In addition to the help document at ''http://www.biostat.umn.edu/~brad/software/BRugs/BRugs_9_21_07.pdf'' there is a **WinBUGS** style manual you can ac-
 10017 cess at ''<http://www.rni.helsinki.fi/openbugs/OpenBUGS/Docu/BRugs%20Manual.html>''.
 10018 **BRugs** has the convenient feature that all pieces of a **BUGS** analysis can be run
 10019 from within **R**, including checking the model syntax, something that requires opening the
 10020 **BUGS** GUI with other packages.

20.6 JAGS

10022 **JAGS** (Just Another Gibbs Sampler) (Plummer, 2003) runs scr models considerably faster
 10023 than **WinBUGS**, does not have the convergence problem with simple scr models we have

10024 encountered in **OpenBUGS** but similar to the latter program, is flexible and constantly
10025 updated. Writing a **JAGS** model is virtually identical to writing a **WinBUGS** model.
10026 However, some functions may have slightly different names and you can look up available
10027 functions and their use in the **JAGS** manual. One potential downside is that **JAGS** can
10028 be very particular when it comes to initial values. These may have to be set as close to
10029 truth as possible for the model to start. Although **JAGS** lets you run several parallel
10030 Markov chains, this characteristic interferes with the idea of using overdispersed initial
10031 values for the different chains. Also, we have found that when running models, sometimes
10032 **JAGS** crashes for unclear reasons, taking **R** down with it. Oftentimes, in order to make
10033 it run again you'll have to go through downloading and installing it again (remove the
10034 non-functioning version first).

10035 **JAGS** has a variety of functions that are not available in **WinBUGS**. For example,
10036 **JAGS** allows you to supply observed data for some deterministic functions of unobserved
10037 variables. In **BUGS** we cannot supply data to logical nodes. Another useful feature is
10038 that the adaptive phase of the model (the burn-in) is run separately from the sampling
10039 from the stationary Markov chains. This allows you to easily add more iterations to the
10040 adaptive phase if necessary without the need to start from 0. There are other, more
10041 subtle differences and there is an entire manual section on differences between **JAGS** and
10042 **OpenBUGS**.

10043 **JAGS** is available for download at '<http://sourceforge.net/projects/mcmc-jags/files/>', together with the R package **rjags** (Plummer, 2011), which allows running
10044 **JAGS** through **R**, user and installation manuals and examples. At this site **JAGS** is
10045 available for Windows and Mac; Linux binaries are distributed separately and you can
10046 find links to various sources here: '<http://mcmc-jags.sourceforge.net/>'. **JAGS**
10047 comes with a 32 bit and a 64 bit version and can be installed by double-clicking on the
10048 .exe file and following the instructions on the installer screen. For questions and prob-
10049 lems concerning **JAGS** there is a forum online at <http://sourceforge.net/projects/>
10050 **mcmc-jags**/**forums**/**forum**/610037.

10052 20.6.1 JAGS through R

10053 Unlike the two **BUGS** programs, **JAGS** does not have a GUI interface but a command
10054 line interface that can be used to run the program as a standalone application. **JAGS**
10055 will solely perform the MCMC simulation; analyzing and summarizing the output has to
10056 be done outside of **JAGS**. To run **JAGS** through **R** you have two options.

10057 **rjags**: As mentioned above, **rjags** (Plummer, 2011) can be found together with **JAGS**
10058 and was developed/is being maintained by the inventor of **JAGS**, which means it is
10059 guaranteed to stay up to date when/as **JAGS** changes. The package can be installed from
10060 a cran mirror and the help document can be accessed at '<http://cran.r-project.org/web/packages/rjags/rjags.pdf>'

10062 **R2jags**: Alternatively, the package **R2jags** (Su and Yajima, 2011) provides a means of
10063 accessing **JAGS** through **R**. We prefer **rjags** for the reason named above, as well as be-
10064 cause it stores data in a more memory-efficient way and has better **plot()** and **summary()**
10065 methods.

20.7 R

10066 At the time of the preparation of this list, **R** for Windows is at version 2.15.0, which can be
 10067 downloaded at <http://cran.r-project.org/bin/windows/base/> This site also contains help-
 10068 ful tips on how to install **R** in Windows Vista, how to update **R** packages etc. Installation
 10069 of **R** in Windows is straightforward: download the .exe file, double-click on it and follow
 10070 the instructions of the Windows installer. The later versions of **R** come with versions for
 10071 both 64 bit and 32 bit machines. The **R** site (''<http://mirrors.softliste.de/cran/>'')
 10072 has an extensive FAQ section Hornik (2011), which includes instructions on how to install
 10073 **R** on Unix and Mac computers.

10074 20.7.1 R packages

10075 This section provides an alphabetical list of useful **R** packages. There is a large number
 10076 of **R** packages and by no means is this list intended to be complete in terms of what is
 10077 useful. Rather, we list packages that we are familiar with and that we employ at one point
 10078 or the other in this book. Unless explicitly stated otherwise, all packages can be installed
 10079 directly from within **R** trough a cran mirror.

10080 **adapt**: **adapt** (Genz et al., 2007) is a package for multidimensional numerical integration.
 10081 The package has been removed from the CRAN repository but can be obtained from
 10082 ''<http://cran.r-project.org/src/contrib/Archive/adapt/>''.

10083 **coda**: **coda** (Plummer et al., 2006) lets you summarize and perform diagnostics on mcmc
 10084 output. For a list and description of functions, see the manual at ''<http://cran.r-project.org/web/packages/coda/coda.pdf>''.

10086 **gdistance**: **gdistance** (van Etten, 2011) is a package for calculating distances and routes
 10087 on geographical grids and can be used to calculate least cost path surfaces. Manual at
 10088 ''<http://cran.r-project.org/web/packages/gdistance/gdistance.pdf>''.

10089 **igraph**: **igraph** (Csardi and Nepusz, 2006) provides routines for graphs and network anal-
 10090 ysis. Manual at ''<http://cran.r-project.org/web/packages/igraph/igraph.pdf>''.

10091 **inline**: **inline** (Sklyar et al., 2010) allows the user to define R functions with in-lined **C**,
 10092 **C++** or **Fortran** code. Manual at <http://cran.r-project.org/web/packages/inline/inline.pdf>.

10094 **maps**: **maptools** (?) is a library of maps. Manual at ''<http://cran.r-project.org/web/packages/maps/index.html>''.

10096 **maptools**: **maptools** (Lewin-Koh et al., 2011) provides a set of tools for reading and manip-
 10097 ulating spatial data, especially ESRI shapefiles. Manual at ''<http://cran.r-project.org/web/packages/maptools/maptools.pdf>''.

10099 **R2cuba**: **R2cuba** (Hahn et al., 2010) is another package for multidimensional integration.
 10100 Manual at ''<http://cran.r-project.org/web/packages/R2Cuba/R2Cuba.pdf>''.

10101 **raster**: **raster** (van Etten, 2012) provides functions for geographic analysis and modeling
10102 with raster data. Manual at '<http://cran.r-project.org/web/packages/raster/raster.pdf>'.

10104 **Rcpp**: **Rcpp** (Eddelbuettel and François, 2011) provides R functions as well as a C++ library
10105 which facilitate the integration of R and C++. Manual at <http://cran.r-project.org/web/packages/Rcpp/Rcpp.pdf>.

10107 **RcppArmadillo**: **RcppArmadillo** (François et al., 2011) is a templated C++ linear algebra
10108 library, integrating the **Armadillo** library and R. Manual at <http://cran.r-project.org/web/packages/RcppArmadillo/RcppArmadillo.pdf>.

10110 **reshape**: **reshape** (Wickham and Hadley, 2007) allows you to easily manipulate, summarize
10111 and reshape data. Manual at '<http://cran.r-project.org/web/packages/reshape/reshape.pdf>'.

10113 **rgeos**: **rgeos** (Bivand and Rundel, 2011) provides many useful functions for spatial operations
10114 such as intersecting or buffering spatial features. Manual at '<http://cran.r-project.org/web/packages/rgeos/rgeos.pdf>'.

10116 **SCRbayes**: (Russell et al., 2012)

10117 **secr**: **secr** (Efford et al., 2009a)

10118 **shapefiles**: **shapefiles** (Stabler, 2006) allows you to read and write ESRI shapefiles
10119 (i.e. shapefiles you would use in ArcGIS). Manual at '<http://cran.r-project.org/web/packages/shapefiles/shapefiles.pdf>'.

10121 **snow**, **snowfall**: **snow** (Tierney et al., 2011) and **snowfall** (Knaus, 2010) provide functionality
10122 for parallel computing. The latter is a more user-friendly wrapper around the former. Manuals at <http://cran.r-project.org/web/packages/snowfall/snowfall.pdf> and <http://cran.r-project.org/web/packages/snow/snow.pdf>.

10125 **sp**: **sp** (Pebesma and Bivand, 2011) is a package for plotting, selecting, subsetting etc.
10126 spatial data. **sp** and **spatstat** (see below) are complementary in many ways and data
10127 formats can be easily converted between the two packages. Manual at '<http://cran.r-project.org/web/packages/sp/sp.pdf>'.

10129 **SPACECAP**: (Gopalaswamy et al., 2012a)

10130 **spatstat**: **spatstat** (Baddeley and Turner, 2005) is an extensive package for analyzing
10131 spatial data. We use it, for example, to generate random points within a state space
10132 that cannot be described as a rectangle but consists of a (or several) arbitrary polygon(s).
10133 Manual at '<http://cran.r-project.org/web/packages/spatstat/spatstat.pdf>'.

10134 **unmarked**:

10135 #####

10136 #####

10137 **References**

10138

BIBLIOGRAPHY

10139

10140

- 10141 Aitkin, M. (1991), "Posterior bayes factors," *Journal of the Royal Statistical Society. Series B (Methodological)*, 53, 111–142.
- 10142 Alho, J. M. (1990), "Logistic regression in capture-recapture models," *Biometrics*, 46, 623–635.
- 10143 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.
- 10144 Amstrup, S. C., McDonald, T. L., and Manly, B. F. J. (2005), *Handbook of capture-recapture analysis*, Princeton Univ Pr.
- 10145 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.
- 10146 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.
- 10147 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.
- 10148 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.
- 10149 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.
- 10150 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.
- 10151 Berger, J., Liseo, B., and Wolpert, R. (1999), "Integrated likelihood methods for eliminating nuisance parameters," *Statistical Science*, 14, 1–28.
- 10152 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.
- 10153 Bibby, C. J. and Buckland, S. T. (1987), "Bias of bird census results due to detectability varying with habitat," *Acta Ecologica*, 8, 103–112.
- 10154 Bibby, C. J., Burgess, N. D., and Hill, D. A. (1992), "Bird census techniques: Academic Press," London, UK, 257.
- 10155 Bivand, R. and Rundel, C. (2011), *rgeos: Interface to Geometry Engine - Open Source (GEOS)*, r package version 0.1-8.

- 10178 Blair, W. (1940), "A study of prairie deer-mouse populations in southern Michigan,"
10179 *American Midland Naturalist*, 24, 273–305.
- 10180 Bolker, B. (2008), *Ecological models and data in R*, Princeton Univ Pr.
- 10181 Borchers, D. L. (2012), "A non-technical overview of spatially explicit capture–recapture
10182 models," *Journal of Ornithology*, 152, 1–10.
- 10183 Borchers, D. L., Buckland, S. T., and Zucchini, W. (2002), *Estimating animal abundance:
10184 closed populations*, vol. 13, Springer Verlag.
- 10185 Borchers, D. L. and Efford, M. G. (2008), "Spatially explicit maximum likelihood methods
10186 for capture–recapture studies," *Biometrics*, 64, 377–385.
- 10187 Borchers, D. L., Zucchini, W., and Fewster, R. M. (1998), "Mark-recapture models for
10188 line transect surveys," *Biometrics*, 54, 1207–1220.
- 10189 Boulanger, J. and McLellan, B. (2001), "Closure violation in DNA-based mark-recapture
10190 estimation of grizzly bear populations," *Canadian Journal of Zoology*, 79, 642–651.
- 10191 Bowden, D. C. (1993), "A simple technique for estimating population size. Technical Re-
10192 port 93/12." Tech. rep., Department of Statistics, Colorado State University, Fort
10193 Collins, Colorado, USA.
- 10194 Brooks, S. P., Catchpole, E. A., and Morgan, B. J. T. (2000), "Bayesian Animal Survival
10195 Estimation," *Statistical Science*, 15, 357–376.
- 10196 Buckland, S., Anderson, D., Burnham, K., Laake, J., Borcher, D., and L, T. (2001), *In-
10197 troduction to distance sampling: estimating abundance of biological populations*, Oxford,
10198 UK: Oxford University Press.
- 10199 Buckland, S. T. (2004), *Advanced distance sampling*, Oxford University Press, USA.
- 10200 Burnham, K., Anderson, D., and Laake, J. (1980), "Estimation of density from line tran-
10201 sect sampling of biological populations," *Wildlife monographs*, 3–202.
- 10202 Burnham, K., Anderson, D., White, G., Brownie, C., and Pollock, K. (1987), "Design and
10203 analysis methods for fish survival experiments based on release-recapture," *American
10204 Fisheries Society Monograph 5. American Fisheries Society, Bethesda Maryland.* 1987.
10205 737.
- 10206 Burnham, K. P. and Anderson, D. R. (2002), *Model selection and multimodel inference:
10207 a practical information-theoretic approach*, Springer Verlag.
- 10208 Burnham, K. P. and Overton, W. S. (1978), "Estimation of the size of a closed population
10209 when capture probabilities vary among animals," *Biometrika*, 65, 625.
- 10210 Burt, W. (1943), "Territoriality and home range concepts as applied to mammals," *Journal
10211 of mammalogy*, 24, 346–352.
- 10212 Casella, G. and Berger, R. L. (2002), *Statistical inference*, Duxbury Press.
- 10213 Chandler, R. B. and Royle, J. A. (In press), "Spatially-explicit models for inference about
10214 density in unmarked or partially marked populations," *Annals of Applied Statistics*.
- 10215 Chandler, R. B., Royle, J. A., and King, D. (2011), "Inference about density and tempo-
10216 rary emigration in unmarked populations," *Ecology*, 92, 1429–1435.
- 10217 Cloeber, J., Danchin, E., Dhondt, A., and Nichols, J. (2001), *Dispersal*, Oxford.
- 10218 Converse, S., White, G., Farris, K., and Zack, S. (2006), "Small mammals and forest fuel
10219 reduction: national-scale responses to fire and fire surrogates," *Ecological Applications*,
10220 16, 1717–1729.
- 10221 Converse, S. J. and Royle, J. A. (2012), "Dealing with incomplete and variable detectabil-
10222 ity in multi-year, multi-site monitoring of ecological populations," in *Design and Anal-
10223 ysis of Long-term Ecological Monitoring Studies*, eds. Gitzen, R. R., Millspaugh, J. J.,

- 10224 Cooper, A. B., and Licht, D. S., Cambridge University Press, pp. 426–442.
- 10225 Cooch, E. and White, G. (2006), “Program MARK: a gentle introduction,” *available online with the MARK programme*, 7.
- 10226 Coull, B. A. and Agresti, A. (1999), “The Use of Mixed Logit Models to Reflect Heterogeneity in Capture-Recapture Studies,” *Biometrics*, 55, 294–301.
- 10227 Cressie, N. (1991), *Statistics for spatial data*, Wiley Series in Probability and Mathematical Statistics.
- 10228 Csardi, G. and Nepusz, T. (2006), “The igraph software package for complex network research,” *InterJournal, Complex Systems*, 1695.
- 10229 Dawson, D. K. and Efford, M. G. (2009), “Bird population density estimated from acoustic signals,” *Journal of Applied Ecology*, 46, 1201–1209.
- 10230 DeGraaf, R. M. and Yamasaki, M. (2001), *New England wildlife: habitat, natural history, and distribution*, University Press of New England.
- 10231 Dice, L. R. (1938), “Some census methods for mammals,” *Journal of Wildlife Management*, 2, 119–130.
- 10232 Dixon, P. (2002), “Bootstrap resampling,” *Encyclopedia of environmetrics*.
- 10233 Dorazio, R. M. (2007), “On the choice of statistical models for estimating occurrence and extinction from animal surveys,” *Ecology*, 88, 2773–2782.
- 10234 Dorazio, R. M. and Royle, J. A. (2003), “Mixture models for estimating the size of a closed population when capture rates vary among individuals,” *Biometrics*, 59, 351–364.
- 10235 Durban, J. and Elston, D. (2005), “Mark-recapture with occasion and individual effects: abundance estimation through Bayesian model selection in a fixed dimensional parameter space,” *Journal of agricultural, biological, and environmental statistics*, 10, 291–305.
- 10236 Eddelbuettel, D. and François, R. (2011), “Rcpp: Seamless R and C++ Integration,” *Journal of Statistical Software*, 40, 1–18.
- 10237 Efford, M. (2011), *secr: Spatially explicit capture-recapture models*, R package version 2.3.1.
- 10238 Efford, M. G. (2004), “Density estimation in live-trapping studies,” *Oikos*, 106, 598–610.
- 10239 Efford, M. G., Borchers, D. L., and Byrom, A. E. (2009a), “Density estimation by spatially explicit capture-recapture: likelihood-based methods,” *Modeling demographic processes in marked populations*, 255–269.
- 10240 Efford, M. G. and Dawson, D. K. (2009), “Effect of distance-related heterogeneity on population size estimates from point counts,” *The Auk*, 126, 100–111.
- 10241 — (2010), “SECR for acoustic data,” .
- 10242 Efford, M. G., Dawson, D. K., and Borchers, D. L. (2009b), “Population density estimated from locations of individuals on a passive detector array,” *Ecology*, 90, 2676–2682.
- 10243 Efford, M. G., Dawson, D. K., and Robbins, C. S. (2004), “DENSITY: software for analysing capture-recapture data from passive detector arrays,” *Animal Biodiversity and Conservation*, 27, 217–228.
- 10244 Efford, M. G. and Fewster, R. M. (2012), “Estimating population size by spatially explicit capture-recapture,” *Oikos*.
- 10245 Efford, M. G., Warburton, B., Coleman, M. C., and Barker, R. J. (2005), “A field test of two methods for density estimation,” *Wildlife Society Bulletin*, 33, 731–738.
- 10246 F Dormann, C., M McPherson, J., B Araújo, M., Bivand, R., Bolliger, J., Carl, G., G Davies, R., Hirzel, A., Jetz, W., Daniel Kissling, W., et al. (2007), “Methods to account for spatial autocorrelation in the analysis of species distributional data: a

- 10270 review," *Ecography*, 30, 609–628.
- 10271 Farnsworth, G. L., Pollock, K. H., Nichols, J. D., Simons, T. R., E., H. J., and R.,
10272 S. J. (2002), "A removal model for estimating detection probabilities from point-count
10273 surveys," *The Auk*, 119, 414–425.
- 10274 Fienberg, S. E., Johnson, M. S., and Junker, B. W. (1999), "Classical multilevel and
10275 Bayesian approaches to population size estimation using multiple lists," *Journal of the
10276 Royal Statistical Society of London A*, 163, 383–405.
- 10277 Fiske, I. J. and Chandler, R. B. (2011), "unmarked: An R package for fitting hierarchical
10278 models of wildlife occurrence and abundance," *Journal Of Statistical Software*, 43, 1–23.
- 10279 Foster, R. J. and Harmsen, B. J. (2012), "A critique of density estimation from camera-
10280 trap data," *The Journal of Wildlife Management*, 76, 224–236.
- 10281 François, R., Eddelbuettel, D., and Bates, D. (2011), *RcppArmadillo: Rcpp integration
10282 for Armadillo templated linear algebra library*, r package version 0.2.25.
- 10283 García-Alaníz, N., Naranjo, E. J., and Mallory, F. F. (2010), "Hair-snares: A non-invasive
10284 method for monitoring felid populations in the Selva Lacandona, Mexico," *Tropical
10285 Conservation Science*, 3, 403–411.
- 10286 Gardner, B., Reppucci, J., Lucherini, M., and Royle, J. (2010a), "Spatially explicit in-
10287 ference for open populations: estimating demographic parameters from camera-trap
10288 studies," *Ecology*, 91, 3376–3383.
- 10289 Gardner, B., Royle, J. A., and Wegan, M. T. (2009), "Hierarchical models for estimating
10290 density from DNA mark-recapture studies," *Ecology*, 90, 1106–1115.
- 10291 Gardner, B., Royle, J. A., Wegan, M. T., Rainbolt, R. E., and Curtis, P. D. (2010b),
10292 "Estimating black bear density using DNA data from hair snares," *The Journal of
10293 Wildlife Management*, 74, 318–325.
- 10294 Garshelis, D. L. and Hristienko, H. (2006), "State and provincial estimates of American
10295 black bear numbers versus assessments of population trend," *Ursus*, 17, 1–7.
- 10296 Gelman, A. (2006), "Prior distributions for variance parameters in hierarchical models,"
10297 *Bayesian analysis*, 1, 515–533.
- 10298 Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004), *Bayesian data analysis,
10299 second edition.*, Bocan Raton, Florida, USA: CRC/Chapman & Hall.
- 10300 Gelman, A., Meng, X. L., and Stern, H. (1996), "Posterior predictive assessment of model
10301 fitness via realized discrepancies," *Statistica Sinica*, 6, 733–759.
- 10302 Gelman, A. and Rubin, D. B. (1992), "Inference from iterative simulation using multiple
10303 sequences," *Statistical Science*, 7, 457–511.
- 10304 Geman, S. and Geman, D. (1984), "Stochastic relaxation, Gibbs distributions, and the
10305 Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine
10306 Intelligence*, PAMI-6, 721–741.
- 10307 Genz, A. S., Meyer, M. R., Lumley, T., and Maechler, M. (2007), "The adapt Package. R
10308 package version 1.0-4," .
- 10309 Gilks, W. R., Thomas, A., and Spiegelhalter, D. J. (1994), "A Language and Program for
10310 Complex Bayesian Modelling," *Journal of the Royal Statistical Society. Series D (The
10311 Statistician)*, 43, 169–177, ArticleType: primary-article / Issue Title: Special Issue:
10312 Conference on Practical Bayesian Statistics, 1992 (3) / Full publication date: 1994 /
10313 Copyright 1994 Royal Statistical Society.
- 10314 Gopalswamy, A. M. (2012), "Capture-recapture models, spatially explicit," in *Encyclo-
10315 pedia of Environmentrics Second Edition*, eds. El-Shaarawi, A. H. and Piegorsch, W.,

- 10316 John Wiley and Sons Ltd, Chichester, UK.
10317 Gopalaswamy, A. M., Royle, A. J., Hines, J., Singh, P., Jathanna, D., Kumar, N. S., and
10318 Karanth, K. U. (2012a), "Program SPACECAP: software for estimating animal density
10319 using spatially explicit capture-recapture models," *Methods in Ecology and Evolution*,
10320 online early, r package version 1.0.4.
10321 Gopalaswamy, A. M., Royle, J. A., Delampady, M., Nichols, J. D., Karanth, K. U., and
10322 Macdonald, D. W. (2012b), "Density estimation in tiger populations: combining infor-
10323 mation for strong inference," *Ecology*, 93, 1741–1751.
10324 Greig-Smith, P. (1964), *Quantitative plant ecology*, Butterworths (Washington).
10325 Hahn, T., Bouvier, A., and Kiêu, K. (2010), *R2Cuba: Multidimensional Numerical Inte-
10326 gration*, r package version 1.0-6.
10327 Hall, R. J., Henry, P. F. P., and Bunck, C. M. (1999), "Fifty-year trends in a box turtle
10328 population in Maryland," *Biological Conservation*, 88, 165–172.
10329 Hanski, I. A. (1999), *Metapopulation Ecology*, Oxford Univiversity Press.
10330 Hawkins, C. E. and Racey, P. A. (2005), "Low population density of a tropical forest
10331 carnivore, Cryptoprocta ferox: implications for protected area management," *Oryx*, 39,
10332 35–43.
10333 Hayes, R. J. and Buckland, S. T. (1983), "Radial distance models for the line transect
10334 method," *Biometrics*, 39, 29–42.
10335 Hayne, D. (1950), "Apparent home range of Microtus in relation to distance between
10336 traps," *Journal of mammalogy*, 31, 26–39.
10337 Hayne, D. W. (1949), "An Examination of the Strip Census Method for Estimating Animal
10338 Populations," *The Journal of Wildlife Management*, 13, 145–157.
10339 Hedley, S. L., Buckland, S. T., and Borchers, D. L. (1999), "Spatial modelling from line
10340 transect data," *Journal of Cetacean Research and Management*, 1, 255–264.
10341 Hestbeck, J. B. and Malecki, R. A. (1989), "Mark-resight estimate of Canada Goose
10342 midwinter numbers," *Journal of Wildlife Management*, 53, 749–752.
10343 Higdon, D. (1998), "A process-convolution approach to modelling temperatures in the
10344 North Atlantic Ocean," *Environmental and Ecological Statistics*, 5, 173–190.
10345 Hobbs, N. T. (2011), "An Ecological Modeler's Primer on JAGS," .
10346 Holdenried, R. (1940), "A population study of the long-eared chipmunk (*Eutamias quadri-
10347 maculatus*) in the central Sierra Nevada," *Journal of Mammalogy*, 21, 405–411.
10348 Hornik, K. (2011), "The R FAQ," ISBN 3-900051-08-9.
10349 Huggins, R. M. (1989), "On the statistical analysis of capture experiments," *Biometrika*,
10350 76, 133.
10351 Hurlbert, S. H. (1984), "Pseudoreplication and the design of ecological field experiments,"
10352 *Ecological monographs*, 54, 187–211.
10353 Illian, J., Penttinen, A., Stoyan, H., and Stoyan, D. (2008), *Statistical analysis and mod-
10354 ellng of spatial point patterns*, Wiley.
10355 Ivan, J. (2012), "Density, demography, and seasonal movements of snowshoe hares in
10356 central Colorado," Ph.D. thesis, Colorado State University.
10357 Ivan, J., White, G., and Shenk, T. (2013a), "Using auxiliary telemetry information to
10358 estimate animal density from capture-recapture data," *Ecology*.
10359 — (2013b), "Using simulation to compare methods for estimating density from capture-
10360 recapture data," *Ecology*.
10361 Jackson, R., Roe, J., Wangchuk, R., and Hunter, D. (2006), "Estimating Snow Leop-

- ard Population Abundance Using Photography and Capture-Recapture Techniques," *Wildlife Society Bulletin*, 34, 772–781.
- Jett, D. A. and Nichols, J. D. (1987), "A Field Comparison of Nested Grid and Trapping Web Density Estimators," *Journal of Mammalogy*, 68, 888–892.
- Johnson (2010), "A Model-Based Approach for Making Ecological Inference from Distance Sampling Data," *Biometrics*, 66, 310318.
- Johnson, D. H. (1999), "The insignificance of statistical significance testing," *The journal of wildlife management*, 63, 763–772.
- Kadane, J. B. and Lazar, N. A. (2004), "Methods and criteria for model selection," *Journal of the American Statistical Association*, 99, 279–290.
- Karanth, K. U. (1995), "Estimating tiger *Panthera tigris* populations from camera-trap data using capture–recapture models," *Biological Conservation*, 71, 333–338.
- Karanth, K. U. and Nichols, J. D. (1998), "Estimation of Tiger Densities in India Using Photographic Captures and Recaptures," *Ecology*, 79, 2852–2862.
- (2000), "Ecological status and conservation of tigers in India," *WCS, US Fish and Wildlife Service. Centre for Wildlife Studies. Bangalore, India*, 123.
- (2002), *Monitoring tigers and their prey: a manual for researchers, managers and conservationists in {T}ropical {A}sia*, Bangalore, India: Centre for Wildlife Studies.
- Kass, R. and Wasserman, L. (1996), "The selection of prior distributions by formal rules," *Journal of the American Statistical Association*, 91, 1343–1370.
- Kays, R. W., Slauson, K. M., Long, R. A., MacKay, P., Zielinski, W. J., and Ray, J. C. (2008), "Remote cameras." *Noninvasive survey methods for carnivores*, 110–140.
- Kelly, M., Noss, A., Di Bitetti, M., Maffei, L., Arispe, R., Paviolo, A., De Angelo, C., and Di Blanco, Y. (2008), "Estimating puma densities from camera trapping across three study sites: Bolivia, Argentina, and Belize," *Journal of Mammalogy*, 89, 408–418.
- Kendall, K. C., Stetz, J. B., Boulanger, J., Macleod, A. C., Paetkau, D., and White, G. C. (2009), "Demography and genetic structure of a recovering grizzly bear population," *The Journal of Wildlife Management*, 73, 3–16.
- Kendall, W. L., Nichols, J. D., and Hines, J. E. (1997), "Estimating Temporary Emigration Using Capture-Recapture Data with Pollock's Robust Design," *Ecology*, 78, 563–578.
- Kéry, M. (2010), *Introduction to WinBUGS for Ecologists: Bayesian Approach to Regression, ANOVA, Mixed Models and Related Analyses*, Academic Press.
- Kéry, M. (2011), "Towards the modelling of true species distributions," *Journal of Biogeography*, 38, 617–618.
- Kéry, M., Gardner, B., and Monnerat, C. (2010), "Predicting species distributions from checklist data using site-occupancy models," *Journal of Biogeography*, 37, 1851–1862.
- Kéry, M., Gardner, B., Stoeckle, T., Weber, D., and Royle, J. A. (2011), "Use of Spatial Capture-Recapture Modeling and DNA Data to Estimate Densities of Elusive Animals," *Conservation Biology*, 25, 356–364.
- Kéry, M., Royle, J. A., and Schmid, H. (2005), "Modeling avian abundance from replicated counts using binomial mixture models," *Ecological Applications*, 15, 1450–1461.
- Kéry, M. and Schaub, M. (2012), *Bayesian Population Analysis Using WinBugs*, Academic Press.
- King, R. and Brooks, S. (2001), "On the Bayesian analysis of population size," *Biometrika*, 88, 317–336.
- King, R., Brooks, S., and Coulson, T. (2008), "Analyzing Complex Capture–Recapture

- 10408 Data in the Presence of Individual and Temporal Covariates and Model Uncertainty,"
10409 *Biometrics*, 64, 1187–1195.
- 10410 Knaus, J. (2010), *snowfall: Easier cluster computing (based on snow)*., r package version
10411 1.84.
- 10412 Kohn, M., York, E., Kamradt, D., Haught, G., Sauvajot, R., and Wayne, R. (1999),
10413 "Estimating population size by genotyping faeces," *Proceedings of the Royal Society of*
10414 *London. Series B: Biological Sciences*, 266, 657–663.
- 10415 Krebs, C. J. (1999), *Ecological methodology*, Menlo Park, CA: Benjamin/Cummings.
- 10416 Kucera and Barrett (2011), "missing," *missing*, missing.
- 10417 Kuo, L. and Mallick, B. (1998), "Variable selection for regression models," *Sankhy: The*
10418 *Indian Journal of Statistics, Series B*, 60, 65–81.
- 10419 Laird, N. M. and Ware, J. H. (1982), "Random-effects models for longitudinal data,"
10420 *Biometrics*, 38, 963–974.
- 10421 Langtimm, C. A., Dorazio, R. M., Stith, B. M., and Doyle, T. J. (2011), "New aerial
10422 survey and hierarchical model to estimate manatee abundance," *The Journal of Wildlife*
10423 *Management*, 75, 399–412.
- 10424 Le Cam, L. (1990), "Maximum likelihood: an introduction," *International Statistical Re-*
10425 *view/Revue Internationale de Statistique*, 58, 153–171.
- 10426 Lebreton, J. D., Burnham, K. P., Clobert, J., and Anderson, D. R. (1992), "Modeling
10427 Survival and Testing Biological Hypotheses Using Marked Animals: A Unified Approach
10428 with Case Studies," *Ecological Monographs*, 62, 67–118.
- 10429 Lele, S. R. and Keim, J. L. (2006), "Weighted distributions and estimation of resource
10430 selection probability functions," *Ecology*, 87, 3021–3028.
- 10431 Lele, S. R., Moreno, M., and Bayne, E. (2012), "Dealing with detection error in site
10432 occupancy surveys: What can we do with a single survey?" *Journal of Plant Ecology*.
- 10433 Lewin-Koh, N. J., Bivand, R., contributions by Edzer J. Pebesma, Archer, E., Baddeley,
10434 A., Bibiko, H.-J., Dray, S., Forrest, D., Friendly, M., Giraudoux, P., Golicher, D., Rubio,
10435 V. G., Hausmann, P., Hufthammer, K. O., Jagger, T., Luque, S. P., MacQueen, D.,
10436 Niccolai, A., Short, T., Stabler, B., and Turner, R. (2011), *maptools: Tools for reading*
10437 *and handling spatial objects*, r package version 0.8-10.
- 10438 Link, W. (2003), "Nonidentifiability of Population Size from Capture-Recapture Data
10439 with Heterogeneous Detection Probabilities," *Biometrics*, 59, 1123–1130.
- 10440 Link, W. A. and Barker, R. J. (1994), "Density Estimation Using the Trapping Web
10441 Design: A Geometric Analysis," *Biometrics*, 50, 733–745.
- 10442 —— (2006), "Model weights and the foundations of multimodel inference," *Ecology*, 87,
10443 2626–2635.
- 10444 —— (2010), *Bayesian Inference: With Ecological Applications*, London, UK: Academic
10445 Press.
- 10446 Link, W. A. and Eaton, M. J. (2011), "On thinning of chains in MCMC," *Methods in*
10447 *Ecology and Evolution*, 3, 112–115.
- 10448 Link, W. A., Yoshizaki, J., Bailey, L. L., and Pollock, K. H. (2010), "Uncovering a latent
10449 multinomial: analysis of markrecapture data with misidentification," *Biometrics*, 66,
10450 178–185.
- 10451 Liu and Wu (1999), "Parameter expansion for data augmentation," *J Am Stat Assoc*, 94,
10452 1264–1274.
- 10453 Lukacs, P. M. and Burnham, K. P. (2005), "Estimating population size from DNA-based

- 10454 closed capture-recapture data incorporating genotyping error," *Journal of Wildlife
10455 Management*, 69, 396–403.
- 10456 Lunn, D., Spiegelhalter, D., Thomas, A., and Best, N. (2009), "The BUGS project: Evo-
10457 lution, critique, and future directions," *Statistics in Medicine*, 28, 3049–3067.
- 10458 Mace, R., Minta, S., Manley, T., and Aune, K. (1994), "Estimating grizzly bear population
10459 size using camera sightings," *Wildlife Society Bulletin*, 22, 74–83.
- 10460 MacEachern, S. N. and Berliner, L. M. (1994), "Subsampling the Gibbs sampler," *Amer-
10461 ican Statistician*, 48, 188–190.
- 10462 MacKay, P., Smith, D. A., Long, R. A., Parker, M., Long, R. A., MacKay, P., Zielinski,
10463 W. J., and Ray, J. C. (2008), "Scat detection dogs," *Noninvasive survey methods for
10464 carnivores*, 183–222.
- 10465 MacKenzie, D. I., Nichols, J. D., Lachman, G. B., Droege, S., Royle, J. A., and Langtimm,
10466 C. A. (2002), "Estimating site occupancy rates when detection probabilities are less than
10467 one," *Ecology*, 83, 2248–2255.
- 10468 MacKenzie, D. I., Nichols, J. D., Royle, J. A., Pollock, K. H., Bailey, L. L., and Hines,
10469 J. E. (2006), *Occupancy estimation and modeling: inferring patterns and dynamics of
10470 species occurrence*, Academic Press.
- 10471 Magoun, A. J., Long, C. D., Schwartz, M. K., Pilgrim, K. L., Lowell, R. E., and Valken-
10472 burg, P. (2011), "Integrating motion-detection cameras and hair snags for wolverine
10473 identification," *The Journal of Wildlife Management*, 75, 731–739.
- 10474 Manly, B., McDonald, L., Thomas, D., McDonald, T., and Erickson, W. (2002), *Resource
10475 selection by animals: statistical design and analysis for field studies*, Springer, 2nd ed.
- 10476 Marques, T., Thomas, L., Ward, J., DiMarzio, N., and Tyack, P. (2009), "Estimating
10477 cetacean population density using fixed passive acoustic sensors: An example with
10478 Blainvilles beaked whales," *The Journal of the Acoustical Society of America*, 125,
10479 1982.
- 10480 McCarthy, M. A. (2007), *Bayesian Methods for Ecology*, Cambridge: Cambridge Univer-
10481 sity Press.
- 10482 McClintock, B. and Hoeting, J. (2010), "Bayesian analysis of abundance for binomial
10483 sighting data with unknown number of marked individuals," *Environmental and Eco-
10484 logical Statistics*, 17, 317–332.
- 10485 McClintock, B. and White, G. (2012), "From NOREMARK to MARK: software for esti-
10486 mating demographic parameters using mark-resight methodology," *Journal of Ornithol-
10487 ogy*, 152, 641–650.
- 10488 McClintock, B., White, G., Antolin, M., and Tripp, D. (2009a), "Estimating abundance
10489 using mark-resight when sampling is with replacement or the number of marked indi-
10490 viduals is unknown," *Biometrics*, 65, 237–246.
- 10491 McClintock, B., White, G., and Burnham, K. (2006), "A robust design mark-resight abun-
10492 dance estimator allowing heterogeneity in resighting probabilities," *Journal of agricultural,
10493 biological, and environmental statistics*, 11, 231–248.
- 10494 McClintock, B. T., White, G. C., Burnham, K. P., and Pryde, M. A. (2009b), "A general-
10495 ized mixed effects model of abundance for mark-resight data when sampling is without
10496 replacement," in *Modeling demographic processes in marked populations*, ed. Thomson,
10497 D., New York: Springer, pp. 271–289.
- 10498 McCullagh, P. and Nelder, J. A. (1989), *Generalized linear models*, Chapman & Hall/CRC.
- 10499 Millar, R. B. (2009), "Comparison of hierarchical Bayesian models for overdispersed count

- 10500 data using DIC and Bayes' Factors," *Biometrics*, 65, 962–969.
- 10501 Mills, L. S., Citta, J. J., Lair, K. P., Schwartz, M. K., and Tallmon, D. A. (2000), "Estimating animal abundance using noninvasive DNA sampling: promise and pitfalls," *Ecological Applications*, 10, 283–294.
- 10504 Minta, S. and Mangel, M. (1989), "A Simple Population Estimate Based on Simulation for Capture-Recapture and Capture-Resight Data," *Ecology*, 70, 1738–1751.
- 10506 Møller, J. and Waagepetersen, R. P. (2004), *Statistical inference and simulation for spatial point processes*, Chapman & Hall/CRC.
- 10508 Mohr, C. (1947), "Table of equivalent populations of North American small mammals," *American midland naturalist*, 37, 223–249.
- 10510 Moldenhauer, R. R. and Regelski, D. J. (1996), "Northern Parula (*Parula americana*)," in *The Birds of North America Online*, ed. Poole, A., Ithaca, NY: Cornell Lab of Ornithology.
- 10512 Molinari-Jobin, A., Kéry, M., Marboutin, E., Marucco, F., Zimmermann, F., Molinari, P., Frick, H., Wölfl, S., Bled, F., Breitenmoser-Würsten, C., Fuxjäger, C., Huber, T., I., K., Kos, I., Manfred Wölfl, M., and Breitenmoser, U. (2013), "Mapping range dynamics from opportunistic data: spatio-temporal distribution modeling of lynx *Lynx lynx* L. in the Alps," *Biological Conservation*, xx, xxxx–xxxx.
- 10518 Neal, A., White, G., Gill, R., Reed, D., and Olterman, J. (1993), "Evaluation of mark-resight model assumptions for estimating mountain sheep numbers," *Journal of Wildlife Management*, 57, 436–450.
- 10521 Neal, A. K. (1990), "Evaluation of mark-resight population estimates using simulations and field data from mountain sheep." M.S. thesis, Colorado State University, Fort Collins, Colorado, USA.
- 10524 Nelder, J. A. and Wedderburn, R. W. M. (1972), "Generalized linear models," *Journal of the Royal Statistical Society. Series A (General)*, 135, 370–384.
- 10526 Nichols, J. D., Hines, J. E., Sauer, J. R., Fallon, F. W., Fallon, J. E., and Heglund, P. J. (2000), "A double-observer approach for estimating detection probability and abundance from point counts," *The Auk*, 117, 393–408.
- 10529 Nichols, J. D. and Karanth, K. U. (2002), "Statistical concepts: assessing spatial distributions," in *Monitoring tigers and their prey: a manual for researchers, managers and conservationists in Tropical Asia*, eds. Karanth, K. U. and Nichols, J. D., Bangalore, India: Centre for Wildlife Studies, pp. 29–38.
- 10533 Niemi, A. and Fernández, C. (2010), "Bayesian Spatial Point Process Modeling of Line Transect Data," *Journal of Agricultural, Biological, and Environmental Statistics*, 15, 327–345.
- 10536 Norris, J. L. and Pollock, K. H. (1996), "Nonparametric MLE under two closed capture-recapture models with heterogeneity," *Biometrics*, 52, 639–649.
- 10538 O'Brien, T. (2011), "Abundance, density and relative abundance: A conceptual framework," in *Camera traps in animal ecology: methods and analyses*, eds. O'Connell, A. F. J., Nichols, J. D., and Karanth, U., Tokyo, Japan: Springer Verlag, pp. 71–96.
- 10541 O'Connell, A. F., Nichols, J. D., and Karanth, U. K. (2010), *Camera traps in animal ecology: Methods and analyses*, Springer.
- 10543 O'Hara, R. and Sillampää, M. (2009), "A review of Bayesian variable selection methods: what, how and which," *Bayesian Analysis*, 4, 85–118.
- 10545 Otis, D. L., Burnham, K. P., White, G. C., and Anderson, D. R. (1978), "Statistical

- 10546 inference from capture data on closed animal populations," *Wildlife monographs*, 3–
10547 135.
- 10548 Ovaskainen, O. (2004), "Habitat-specific movement parameters estimated using mark–
10549 recapture data and a diffusion model," *Ecology*, 85, 242–257.
- 10550 Ovaskainen, O., Rekola, H., Meyke, E., and Arjas, E. (2008), "Bayesian methods for
10551 analyzing movements in heterogeneous landscapes from mark-recapture data," *Ecology*,
10552 89, 542–554.
- 10553 Parmenter, R. R. and MacMahon, J. A. (1989), "Animal Density Estimation Using a
10554 Trapping Web Design: Field Validation Experiments," *Ecology*, 70, 169–179.
- 10555 Parmenter, R. R., Yates, T. L., Anderson, D. R., Burnham, K. P., Dunnum, J. L., Franklin,
10556 A. B., Friggens, M. T., Lubow, B. C., Miller, M., Olson, G. S., and Others (2003),
10557 "Small-mammal density estimation: A field comparison of grid-based vs. web-based
10558 density estimators," *Ecological Monographs*, 73, 1–26.
- 10559 Pebesma, E. and Bivand, R. (2011), *Package 'sp'*, r package version 0.9-91.
- 10560 Pledger, S. (2004), "Unified maximum likelihood estimates for closed capture–recapture
10561 models using mixtures," *Biometrics*, 56, 434–442.
- 10562 Plummer, M. (2003), "JAGS: A program for analysis of Bayesian graphical models us-
10563 ing Gibbs sampling," in *Proceedings of the 3rd International Workshop on Distributed
10564 Statistical Computing (DSC 2003)*. March, pp. 20–22.
- 10565 — (2009), "rjags: Bayesian graphical models using mcmc. R package version 1.0-3-12," .
- 10566 — (2011), "rjags: Bayesian graphical models using mcmc. R package version 3-5," .
- 10567 Plummer, M., Best, N., Cowles, K., and Vines, K. (2006), "CODA: Convergence Diagnosis
10568 and Output Analysis for MCMC," *R News*, 6, 7–11.
- 10569 Pollock, K. H. (1982), "A Capture-Recapture Design Robust to Unequal Probability of
10570 Capture." *Journal of Wildlife Management*, 46, 752–757.
- 10571 Pollock, K. H., Nichols, J. D., Brownie, C., and Hines, J. E. (1990), "Statistical inference
10572 for capture-recapture experiments," *Wildlife monographs*, 3–97.
- 10573 Reich, B. J., Gardner, B., and Wilting, A. (2012), "A spatial capture-recapture model for
10574 territorial species," *Biometrics in review*, xx, xx–xxx.
- 10575 Robert, C. P. and Casella, G. (2010), *Introducing Monte Carlo Methods with R*, New York,
10576 USA: Springer.
- 10577 Royle, J., Kéry, M., and Guélat, J. (2011a), "Spatial capture-recapture models for search-
10578 encounter data," *Methods in Ecology and Evolution*, 2, 602–611.
- 10579 Royle, J. A. (2004a), "Generalized estimators of avian abundance from count survey data,"
10580 *Animal Biodiversity and Conservation*, 27, 375–386.
- 10581 — (2004b), "N-Mixture Models for Estimating Population Size from Spatially Replicated
10582 Counts," *Biometrics*, 60, 108–115.
- 10583 — (2006), "Site occupancy models with heterogeneous detection probabilities," *Biomet-
10584 rics*, 62, 97–102.
- 10585 — (2008), "Modeling individual effects in the Cormack–Jolly–Seber model: a state–space
10586 formulation," *Biometrics*, 64, 364–370.
- 10587 — (2009), "Analysis of capture-recapture models with individual covariates using data
10588 augmentation," *Biometrics*, 65, 267–274.
- 10589 Royle, J. A. and Chandler, R. B. (2012), "Integrating Resource Selection Information with
10590 Spatial Capture-Recapture," *arXiv preprint arXiv:1207.3288*.
- 10591 Royle, J. A., Chandler, R. B., Gazenski, K. D., and Graves, T. A. (2012a), "Spatial

- 10592 capture-recapture for jointly estimating population density and landscape connectivity,"
10593 *Ecology*, to appear.
- 10594 Royle, J. A., Chandler, R. B., Sun, C., and Fuller, A. (2012b), "Integrating Resource
10595 Selection Information with Spatial Capture-Recapture," *needs updated MEE*.
- 10596 Royle, J. A. and Converse, S. J. (in review), "Hierarchical spatial capture-recapture mod-
10597 els: Modeling population density from replicated capture-recapture experiments," *Ecol-
10598 ogy*.
- 10599 Royle, J. A., Converse, S. J., and Link, W. A. (2012c), "Data Augmentation for Hierar-
10600 chical Capture-recapture Models," *arXiv preprint arXiv:1211.5706*.
- 10601 Royle, J. A., Dawson, D. K., and Bates, S. (2004), "Modeling abundance effects in distance
10602 sampling," *Ecology*, 85, 1591–1597.
- 10603 Royle, J. A. and Dorazio, R. M. (2006), "Hierarchical models of animal abundance and
10604 occurrence," *Journal of Agricultural, Biological, and Environmental Statistics*, 11, 249–
10605 263.
- 10606 — (2008), *Hierarchical modeling and inference in ecology: the analysis of data from pop-
10607ulations, metapopulations and communities*, Academic Press.
- 10608 — (2012), "Parameter-expanded data augmentation for Bayesian analysis of capture-
10609 recapture models," *Journal of Ornithology*, 152, S521–S537.
- 10610 Royle, J. A., Dorazio, R. M., and Link, W. A. (2007), "Analysis of multinomial models with
10611 unknown index using data augmentation," *Journal of Computational and Graphical
10612 Statistics*, 16, 67–85.
- 10613 Royle, J. A. and Dubovský, J. A. (2001), "Modeling spatial variation in waterfowl band-
10614 recovery data," *The Journal of wildlife management*, 65, 726–737.
- 10615 Royle, J. A. and Gardner, B. (2011), "Hierarchical models for estimating density from trap-
10616 ping arrays," in *Camera traps in animal ecology: methods and analyses*, eds. O'Connel,
10617 A. F. J., Nichols, J. D., and Karanth, U., Tokyo, Japan: Springer Verlag, pp. 163–190.
- 10618 Royle, J. A., Karanth, K. U., Gopalaswamy, A. M., and Kumar, N. S. (2009a), "Bayesian
10619 inference in camera trapping studies for a class of spatial capture-recapture models,"
Ecology, 90, 3233–3244.
- 10620 Royle, J. A. and Kéry, M. (2007), "A Bayesian state-space formulation of dynamic occu-
10621 pancy models," *Ecology*, 88, 1813–1823.
- 10622 Royle, J. A. and Link, W. A. (2006), "Generalized site occupancy models allowing for
10623 false positive and false negative errors," *Ecology*, 87, 835–841.
- 10624 Royle, J. A., Magoun, A. J., Gardner, B., Valkenburg, P., and Lowell, R. E. (2011b),
10625 "Density estimation in a wolverine population using spatial capture–recapture models,"
10626 *The Journal of Wildlife Management*, 75, 604–611.
- 10627 Royle, J. A. and Nichols, J. D. (2003), "Estimating abundance from repeated presence-
10628 absence data or point counts," *Ecology*, 84, 777–790.
- 10629 Royle, J. A., Nichols, J. D., Karanth, K. U., and Gopalaswamy, A. M. (2009b), "A
10630 hierarchical model for estimating density in camera-trap studies," *Journal of Applied
10631 Ecology*, 46, 118–127.
- 10632 Royle, J. A. and Young, K. V. (2008), "A Hierarchical Model For Spatial Capture-
10633 Recapture Data," *Ecology*, 89, 2281–2289.
- 10634 Russell, R. E., Royle, J. A., Desimone, R., Schwartz, M. K., Edwards, V. L., Pilgrim,
10635 K. P., and McKelvey, K. S. (2012), "Estimating abundance of mountain lions from
10636 unstructured spatial samples," *Journal of Wildlife Management*.

- 10638 Rutledge, M. (2012), "XXX to be determined XXX," Ph.D. thesis, North Carolina State
10639 University.
- 10640 Salom-Pérez, R., Carrillo, E., Sáenz, J., and Mora, J. (2007), "Critical condition of the
10641 jaguar *Panthera onca* population in Corcovado National Park, Costa Rica," *Oryx*, 41,
10642 51.
- 10643 Sanathanan, L. (1972), "Estimating the size of a multinomial population," *The Annals of
10644 Mathematical Statistics*, 43, 142–152.
- 10645 Schofield, M. and Barker, R. (2008), "A unified capture-recapture framework," *Journal of
10646 agricultural, biological, and environmental statistics*, 13, 458–477.
- 10647 Schwartz, M. K. and Monfort, S. L. (2008), *Genetic and endocrine tools for carnivore
10648 surveys*, Island Press Washington, DC, USA.
- 10649 Seber, G. A. F. (1982), *The estimation of animal abundance and related parameters*,
10650 Macmillan Publishing Co.
- 10651 Sepúlveda, M. A., Bartheld, J. L., Monsalve, R., Gómez, V., and Medina-Vogel, G. (2007),
10652 "Habitat use and spatial behaviour of the endangered Southern river otter (*Lontra
10653 provocax*) in riparian habitats of Chile: conservation implications," *Biological Conservation*,
10654 140, 329–338.
- 10655 Sillett, S., Chandler, R. B., Royle, J. A., Kéry, M., and Morrison, S. (2012), "Hierarchical
10656 distance sampling models to estimate population size and habitat-specific abundance
10657 of an island endemic," *Ecological Applications*.
- 10658 Simons, T. R., Pollock, K. H., Wetrotth, J. M., Alldredge, M. W., Pacifici, K., and
10659 Brewster, J. (2009), "Sources of Measurement Error, Misclassification Error, and Bias
10660 in Auditory Avian Point Count Data," in *Modeling Demographic Processes In Marked
10661 Populations*, eds. Thomson, D. L., Cooch, E. G., and Conroy, M. J., Boston, MA:
10662 Springer US, vol. 3, pp. 237–254.
- 10663 Skalski, J. R., Millspaugh, J. J., and Spencer, R. D. (2005), "Population estimation and
10664 biases in paintball, mark-resight surveys of elk," *Journal of Wildlife Management*, 69,
10665 1043–1052.
- 10666 Sklyar, O., Murdoch, D., Smith, M., Eddelbuettel, D., and François, R. (2010), *inline:
10667 Inline C, C++, Fortran function calls from R*, r package version 0.3.8.
- 10668 Smith, M. H., Blessing, R., Chelton, J. G., Gentry, J. B., Golley, F. B., and McGinnis,
10669 J. T. (1971), "Determining density for small mammal populations using a grid and
10670 assessment lines," *Acta Theriologica*, 16, 105–125.
- 10671 Soisalo, M. K. and Cavalcanti, S. M. C. (2006), "Close-up Space in "Radio-telemetry","
10672 *Biological Conservation*, 129, 487–496.
- 10673 Sollmann, R., Furtado, M. M., Gardner, B., Hofer, H., Jacomo, A. T. A., Trres, N. M., and
10674 Silveira, L. (2011), "Improving density estimates for elusive carnivores: Accounting for
10675 sex-specific detection and movements using spatial capture-recapture models for jaguars
10676 in central Brazil," *Biological Conservation*, 144, 1017–1024.
- 10677 Sollmann, R., Gardner, B., Chandler, R. B., Shindle, D., Onorato, D. P., Royle, J. A., and
10678 O'Connell, A. F. (in revision), "Using multiple data sources provides density estimates
10679 for endangered Florida panther," *Journal of Applied Ecology*.
- 10680 Sollmann, R., Gardner, B., Parsons, A., Stocking, J., McClintock, B., Simons, T., Pollock,
10681 K., and O'Connell, A. (2012), "A spatial mark-resight model augmented with telemetry
10682 data," *Ecology*.
- 10683 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.
- 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.
- Stabler, B. (2006), *shapefiles: Read and Write ESRI Shapefiles*, r package version 0.6.
- Stanley, T. and Burnham, K. (1999), "A closure test for time-specific capture-recapture data," *Environmental and Ecological Statistics*, 6, 197–209.
- Stoyan, D. and Penttinen, A. (2000), "Recent Applications of Point Process Methods in Forestry Statistics," *Statistical Science*, 15, 61–78.
- Sturtz, S., Ligges, U., and Gelman, A. (2005), "R2WinBUGS: A Package for Running WinBUGS from R," *Journal of Statistical Software*, 12, 1–16.
- Su, Y.-S. and Yajima, M. (2011), *R2jags: A Package for Running jags from R*, r package version 0.02-17.
- Taberlet, P. and Bouvet, J. (1992), "Bear conservation genetics," *Nature*, 358, 197–197.
- Tanner, M. A. and Wong, W. H. (1987), "The calculation of posterior distributions by data augmentation," *J Am Stat Assoc*, 82, 528–540.
- Thomas, A., O'Hara, B., Ligges, U., and Sturtz, S. (2006), "Making BUGS Open," *R News*, 6, 12–17.
- Thomas, M. (1949), "A generalization of Poisson's binomial limit for use in ecology," *Biometrika*, 36, 18–25.
- Thompson, C., Royle, J. A., and Garner, J. (2012), "A framework for inference about carnivore density from unstructured spatial sampling of scat using detector dogs," *The Journal of Wildlife Management*, 76, 863–871.
- Tierney, L., Rossini, A. J., Li, N., and Sevcikova, H. (2011), *snow: Simple Network of Workstations*, r package version 0.3-7.
- Tilman, D. and Kareiva, P. (1997), *Spatial ecology: the role of space in population dynamics and interspecific interactions*, vol. 30, Princeton University Press.
- Tobler, M. W., Hibert, F., Debeir, L., and Hansen, C. (2012), "Density and sustainable harvest estimates for the lowland tapir in the Amazon of French Guiana using a spatial capture-recapture model," *none yet*, none, none.
- Trolle, M. and Kéry, M. (2003), "Estimation of ocelot density in the Pantanal using capture-recapture analysis of camera-trapping data," *Journal of Mammalogy*, 84, 607–614.
- (2005), "Camera-trap study of ocelot and other secretive mammals in the northern Pantanal," *Mammalia*, 69, 409–416.
- Tyre, A. J., Tenhumberg, B., Field, S. A., Niejalke, D., Parris, K., and Possingham, H. P. (2003), "Improving precision and reducing bias in biological surveys: estimating false-negative error rates," *Ecological Applications*, 13, 1790–1801.
- Valiere, N. and Taberlet, P. (2000), "Urine collected in the field as a source of DNA for species and individual identification," *Molecular Ecology*, 9, 2150–2152.
- van Etten, J. (2011), *Package gdistance*, r package version 1.1-2.
- van Etten, R. J. H. . J. (2012), *raster: Geographic analysis and modeling with raster data*, r package version 1.9-67.
- Venables, W. and Ripley, B. (2002), *Modern applied statistics with S*, Springer verlag.
- Venables, W., Smith, D., and Team, R. D. C. (2012), "An introduction to R," .

- 10730 Ver Hoef, J. and Boveng, P. (2007), "Quasi-poisson vs. negative binomial regression: How
10731 should we model overdispersed count data?" *Ecology*, 88, 2766–2772.
- 10732 Ver Hoef, J. M. (2012), "Who Invented the Delta Method?" *The American Statistician*,
10733 66, 124–127.
- 10734 Wallace, R. B., Gomez, H., Ayala, G., and Espinoza, F. (2003), "Camera trapping for
10735 jaguar (*Panthera onca*) in the Tuichi Valley, Bolivia," *Journal of Neotropical Mammal-*
10736 *ogy*, 10, 133–139.
- 10737 Wegan, M., Curtis, P., Rainbolt, R., and Gardner, B. (2012), "Temporal sampling frame
10738 selection in DNA-based capture mark-recapture investigations." *Ursus*, 23, 42 – 51.
- 10739 Wegan, M. T. (2008), "Aversive conditioning, population estimation, and habitat prefer-
10740 ence of black bears (*Ursus americanus*) on Fort Drum Military Installation in northern
10741 New York," Ph.D. thesis, Cornell University, Jan.
- 10742 White, G. (1996), "NOREMARK: population estimation from mark-resighting surveys,"
10743 *Wildlife Society Bulletin*, 24, 50–52.
- 10744 White, G. and Bennetts, R. (1996), "Analysis of frequency count data using the negative
10745 binomial distribution," *Ecology*, 77, 2549–2557.
- 10746 White, G. and Shenk, M. (2001), "Poplation estimation with radio-marked inividualls." in
10747 *Radio tracking adn animal populations*, eds. Millspaugh, J. and Marzluff, J., San Diego,
10748 USA: Academic Press, pp. 329–350.
- 10749 White, G. C., Anderson, D. R., Burnham, K. P., and Otis, D. (1982), *Capture-recapture*
10750 *and removal methods for sampling closed populations*, Los Alamos: Los Alamos National
10751 Laboratory.
- 10752 White, G. C. and Garrot, R. (1990), *Analysis of wildlife radiolocation data*, New York,
10753 USA: Academic Press.
- 10754 White, G. C. and Shenk, T. M. (2000), *Population estimation with radio-marked animals*,
10755 San Diego, California: Academic Press.
- 10756 Whitman, J., Ballard, W., and Gardner, C. (1986), "Home range and habitat use by
10757 wolverines in southcentral Alaska," *The Journal of wildlife management*, 460–463.
- 10758 Wickham and Hadley (2007), "Reshaping data with the reshape package," *Journal of*
10759 *Statistical Software*, 21.
- 10760 Williams, B. K., Nichols, J. D., and Conroy, M. J. (2002), *Analysis and management of*
10761 *animal populations: modeling, estimation, and decision making*, Academic Pr.
- 10762 Wilson, K. R. and Anderson, D. R. (1985a), "Evaluation of a Density Estimator Based
10763 on a Trapping Web and Distance Sampling Theory," *Ecology*, 66, 1185–1194.
- 10764 — (1985b), "Evaluation of Two Density Estimators of Small Mammal Population Size,"
10765 *Journal of Mammalogy*, 66, 13–21.
- 10766 Wolpert, R. L. and Ickstadt, K. (1998), "Poisson/gamma random field models for spatial
10767 statistics," *Biometrika*, 85, 251 –267.
- 10768 Woods, J. G., Paetkau, D., Lewis, D., McLellan, B. N., Proctor, M., and Strobeck, C.
10769 (1999), "Genetic tagging of free-ranging black and brown bears," *Wildlife Society Bul-*
10770 *letin*, 27, 616–627.
- 10771 Wright, J., Barker, R., Schofield, M., Frantz, A., Byrom, A., and Gleeson, D. (2009), "In-
10772 corporating Genotype Uncertainty into Mark–Recapture-Type Models For Estimating
10773 Abundance Using DNA Samples," *Biometrics*, 65, 833–840.
- 10774 Yang, H. C. and Chao, A. (2005), "Modeling Animals' Behavioral Response by Markov
10775 Chain Models for Capture–Recapture Experiments," *Biometrics*, 61, 1010–1017.

- 10776 Yoshizaki, J., Pollock, K. H., Brownie, C., and Webster, R. A. (2009), "Modeling misiden-
10777 tification errors in capture-recapture studies using photographic identification of evolv-
10778 ing marks," *Ecology*, 90, 3–9.
- 10779 Zuur, A. F., Ieno, E. N., Walker, N. J., Saveliev, A. A., and Smith, G. M. (2009), *Mixed*
10780 *effects models and extensions in ecology with R*, Springer Verlag.
- 10781 Zylstra, E., Steidl, R., and Swann, D. (2010), "Evaluating survey methods for monitoring
10782 a rare vertebrate, the Sonoran desert tortoise," *The Journal of Wildlife Management*,
10783 74, 1311–1318.