

1

2

Spatial Capture-Recapture

3

J. Andrew Royle
Richard B. Chandler
Rahel Sollmann
Beth Gardner

4

5

6

7

8

USGS Patuxent Wildlife Research Center
North Carolina State University

9

ACADEMIC PRESS

CONTENTS

10

11

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

41	2.2	Common Probability Distributions	28
42	2.2.1	The binomial distribution	28
43	2.2.2	The Bernoulli distribution	29
44	2.2.3	The multinomial and categorical distributions	30
45	2.2.4	The Poisson distribution	32
46	2.2.5	The uniform distribution	32
47	2.2.6	Other distributions	33
48	2.3	Statistical Inference and Parameter Estimation	35
49	2.4	Joint, Marginal, and Conditional Distributions	38
50	2.5	Hierarchical Models and Inference	41
51	2.6	Characterization of SCR Models	43
52	2.7	Summary and Outlook	47
53	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	79
78	3.8.1	Binomial regression	79
79	3.8.2	Example: waterfowl banding data	80
80	3.9	Bayesian Model Checking and Selection	82
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	143
126	5.7.1	Analysis using data augmentation in WinBUGS	145
127	5.7.2	Implied home range area	147
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	167
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 Variation In 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	216
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	218
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	220
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	280
219	10.1.1	Model-based not design-based	280
220	10.1.2	Sampling space or sampling individuals?	281
221	10.1.3	Scope of inference vs. state-space	281
222	10.2	Study design for (spatial) capture-recapture	282
223	10.3	Trap spacing and array size relative to animal movement	284
224	10.3.1	Example: Black bears from Pictured Rocks National Lakeshore: .	286
225	10.3.2	Final musings: SCR models, trap spacing and array size . . .	287
226	10.4	Spacing of traps with telemetered individuals	288
227	10.5	Sampling over large scales	288
228	10.6	Model-based Spatial Design	288
229	10.6.1	Formalization of the Design Problem for SCR Studies . . .	289
230	10.6.2	An Optimal Design Criterion for SCR	290
231	10.6.3	Optimization of the criterion	292
232	10.6.4	Illustration	293
233	10.7	Covariate models	294
234	10.8	Summary and Outlook	294
235	III	Advanced SCR Models	295
236	11	Modeling Spatial Variation in Density	297
237	11.1	Homogeneous point process revisited	298
238	11.2	Inhomogeneous point processes	301
239	11.3	Observed Point Processes	303

241	11.4 Fitting inhomogeneous point process SCR models	306
242	11.4.1 Continuous space	306
243	11.4.2 Discrete space	309
244	11.5 Ecological Distance and Density Covariates	310
245	11.6 The Jaguar Data	313
246	11.7 Summary and Outlook	316
247	12 Modeling Landscape Connectivity	319
248	12.1 Shortcomings of Euclidean Distance Models	320
249	12.2 Least-Cost Path Distance	321
250	12.2.1 Example of Computing Cost-weighted distance	322
251	12.3 Simulating SCR Data using Ecological Distance	325
252	12.4 Likelihood Analysis of Ecological Distance Models	328
253	12.4.1 Example of SCR with Least-Cost Path	329
254	12.5 Bayesian Analysis	329
255	12.6 Simulation Evaluation of the MLE	330
256	12.6.1 Simulation Results	330
257	12.7 Distance In an Irregular Patch	333
258	12.7.1 Basic Geographic Analysis in R	333
259	12.8 Summary and Outlook	337
260	13 Integrating Resource Selection with Spatial Capture-Recapture Models	339
261	13.1 A Simple Model of Space Usage	340
262	13.1.1 Poisson use model	344
263	13.1.2 Thinning	344
264	13.1.3 Capture-recapture Data	345
265	13.2 The Joint RSF/SCR Likelihood	346
266	13.3 Application: New York Black Bear Study	347
267	13.4 Simulation Study	349
268	13.5 Summary and Outlook	352
270	14 Stratified Populations: Multi-session and Multi-site Data	357
271	14.1 Data Structure	358
272	14.2 Multinomial Abundance Models	359
273	14.2.1 Observation Models	360
274	14.2.2 Simulating group structured capture-recapture data	362
275	14.2.3 Fitting in BUGS	362
276	14.2.4 Approach B modeling ψ	362
277	14.3 Spatial Capture-Recapture	362
278	14.4 Application	363
279	14.4.1 Results	365
280	14.5 Topics in Multi-Session models	368

281	14.5.1 Temporal models	368
282	14.5.2 Dependence – is it a problem?	368
283	14.6 Multi-session models in secr	368
284	14.6.1 Ovenbird data in WinBUGS?	370
285	14.6.2 Converse data in secr ?	370
286	14.7 Summary and Outlook	370
287	15 Models for Search-Encounter Data	373
288	15.1 Search-Encounter sampling designs	374
289	15.2 A Model for Search-Encounter Data	376
290	15.2.1 Ecological process model	378
291	15.2.2 Other stuff	379
292	15.3 Examples	379
293	15.3.1 Hard plot boundaries	380
294	15.3.2 Analysis of other protocols	381
295	15.4 Design 3: Ad hoc implementation of Design 1.	381
296	15.5 Capricailie crap	381
297	15.6 Design 4 – no location info	381
298	15.7 Summary and Outlook	382
299	16 Open Population Models	383
300	16.1 Introduction	383
301	16.1.1 Overview of Population Dynamics	384
302	16.1.2 Animal movement related to population demography	385
303	16.1.3 Basic assumptions of JS and CJS models	385
304	16.2 Traditional Jolly-Seber Models	386
305	16.2.1 Data Augmentation for the Jolly-Seber Model	388
306	16.2.2 Mist-netting example	390
307	16.2.3 Shortcomings of the traditional JS models	391
308	16.3 Spatial Jolly-Seber Models	392
309	16.3.1 Mist-netting example	392
310	16.4 Traditional CJS models	397
311	16.4.1 Migratory fish example	397
312	16.5 Multi-state CJS models	399
313	16.5.1 Migratory fish example	401
314	16.6 Spatial CJS models	403
315	16.6.1 Migratory fish example	404
316	16.7 Moving Activity Centers	406
317	16.7.1 Migratory Fish Example Notes	407
318	16.8 Summary and Outlook	408

319	IV Super-Advanced SCR Models	409
320	17 Developing Markov Chain Monte Carlo Samplers	411
321	17.0.1 Why build your own MCMC algorithm?	411
322	17.1 MCMC and posterior distributions	412
323	17.2 Types of MCMC sampling	414
324	17.2.1 Gibbs sampling	414
325	17.2.2 Metropolis-Hastings sampling	416
326	17.2.3 Metropolis-within-Gibbs	421
327	17.2.4 Rejection sampling and slice sampling	424
328	17.3 MCMC for closed capture-recapture Model Mh	424
329	17.4 MCMC algorithm for model SCR0	427
330	17.4.1 SCR model with binomial encounter process	430
331	17.4.2 Looking at model output	431
332	17.4.3 Posterior density plots	432
333	17.4.4 Serial autocorrelation and effective sample size	433
334	17.4.5 Summary results	435
335	17.4.6 Other useful commands	436
336	17.5 Manipulating the state-space	437
337	17.6 Increasing computational speed	440
338	17.6.1 Parallel computing	441
339	17.6.2 Using C++	443
340	17.7 Summary and Outlook	445
341	18 Spatial Capture-Recapture for Unmarked Populations	451
342	18.1 Existing Models for Inference About Density in Unmarked Populations	452
343	18.2 Spatial Correlation as Information	453
344	18.3 Data	454
345	18.4 Model	454
346	18.5 Northern Parula Example	456
347	18.6 Improving Precision with Prior Information	457
348	18.7 Design issues	460
349	18.7.1 How Much Correlation Is Enough?	460
350	18.7.2 Linear Designs	460
351	18.7.3 Quadrat counts	460
352	18.8 Alternative Observation Models	460
353	18.8.1 Spatial point process models	460
354	18.9 Conclusion	461
355	19 Spatial Mark-Resight Models for partially identifiable populations	463
356	19.1 Background	464
357	19.1.1 Types of partial ID data	464

359	19.1.2 A short history of mark-resight models	466
360	19.2 Known number of marked individuals	467
361	19.2.1 MCMC for a spatial mark-resight model	468
362	19.2.2 Binomial encounter model	470
363	19.3 Unknown number of marked individuals	471
364	19.4 Imperfect identification of marked individuals	474
365	19.5 How much information do marked and unmarked individuals con- tribute?	476
366	19.6 Incorporating telemetry data	479
367	19.7 Summary and Outlook	482
369	20 2012: A Spatial Capture-Recapture Odyssey	485
370	20.1 10 thesis or dissertation topics	486
371	20.2 Three dimesional space	486
372	20.3 Gregarious species	486
373	V Appendices	487
374	20.4 WinBUGS	489
375	20.4.1 WinBUGS through R	489
376	20.5 OpenBUGS	490
377	20.5.1 OpenBUGS through R	490
378	20.6 JAGS	490
379	20.6.1 JAGS through R	491
380	20.7 R	492
381	20.7.1 R packages	492
382	Bibliography	495

383 Preface

384

387

Part I

388

389

Background and Concepts

390
391

1

392

INTRODUCTION

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

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

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

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

1.1 THE STUDY OF POPULATIONS BY CAPTURE-RECAPTURE

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

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

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

459 spatial capture-recapture (SCR) models¹.

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

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

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

485 1.2.1 Camera trapping

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

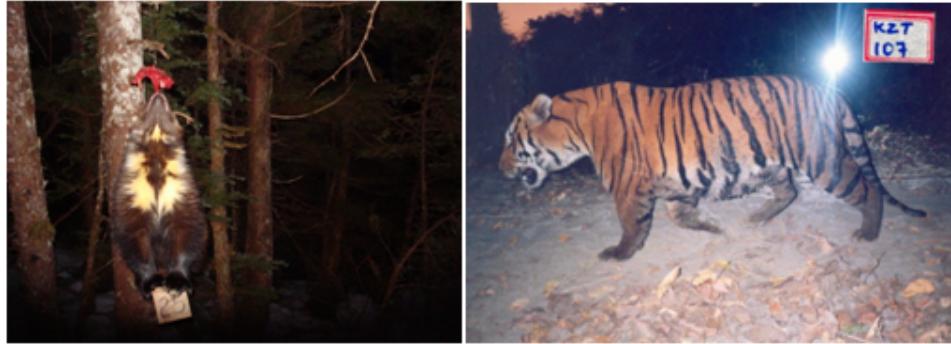


Figure 1.1. Left: Wolverine being encounter by a camera trap (Photo credit: Audrey Magoun). Right: Tiger encountered by camera trap (Photo credit: Ullas Karanth).

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

507 1.2.2 DNA sampling

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

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

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

1.3 CAPTURE-RECAPTURE FOR MODELING ENCOUNTER PROBABILITY

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

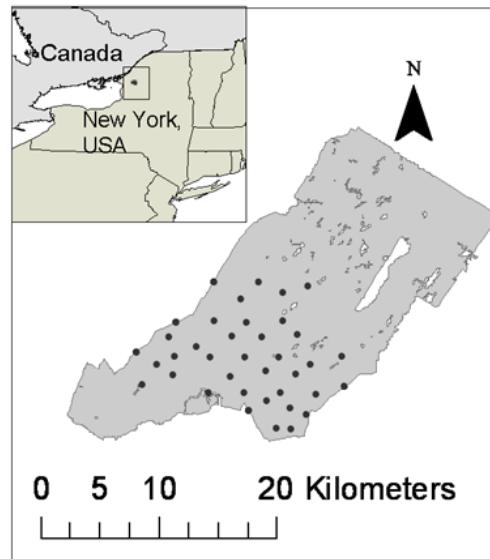


Figure 1.4. Locations of hair snares on Fort Drum, New York, operated during the summer of 2006 to sample black bears.

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

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

594 The mean female home range radius was estimated (Wegan, 2008) for this study
595 region to be 2.19 km, and the area of the convex hull buffered by 2.19 km is
596 277.01 km^2 . (**R** commands to compute the convex hull, buffer it, and compute the
597 area are given in the **R** package **scrbook** which accompanies the book). Hence,
598 the estimated density here is approximately 0.178 bears/km^2 using the estimated
599 population size obtained by model M_0 . We could assert that the problem has been
600 solved, go home, and have a beer. But then, on the other hand, maybe we should
601 question the use of the estimated home range radius – after all, this is only the
602 female home range radius and the home ranges change for many reasons. Instead,
603 we may decide to rely on a buffer width based on one-half mean maximum distance
604 moved (MMDM) estimated from the actual hair snare data as is more customary
605 (Dice, 1938). In that case the buffer width is 1.19 km, and the resulting estimated
606 density is increased to 0.225 bears/km^2 about 27 % larger. But wait – some studies
607 actually found the full MMDM (Parmenter et al., 2003) to be a more appropriate
608 measure of movement (e.g. Soisalo and Cavalcanti (2006)). So maybe we should use
609 the full MMDM which is 2.37 km, pretty close to the telemetry-based estimate and
610 therefore providing a similar estimate of density (0.171 bears/km^2). So in trying to
611 decide how to buffer our trap array we have already generated 3 density estimates.
612 The crux of the matter is obvious: Although it is intuitive that N should scale with
613 area – the number of bears should go up as area increases and go down as area
614 decreases – in this ad hoc approach of accounting for animal movement N remains

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

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

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

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

So what are we left with? Our density estimates span a range from 0.17 to 0.43 bears/km² depending on which estimator of N we use and what buffer strip we apply. Should we feel strongly about one or the other? Which buffer should we prefer? AIC favors model M_h , but did it adequately account for the differences in exposure of individuals to the trap array? Are we happy with a purely phenomenological model for heterogeneity? It assumes that all individuals are independent and identically distributed (*iid*) draws from some distribution, but does not account for the explicit mechanism of induced heterogeneity. And, further, we

have information about that (trap of capture) which model M_h ignores. And if we choose one type of buffer, how do we compare our density estimates to those from other studies that may opt for a different kind of buffer? The fact that N does not scale with A , as part of the model, renders this choice arbitrary.

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

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

1.3.2 Inadequacy of non-spatial capture-recapture

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

The capture-recapture models that we used apply to truly closed populations – a population of goldfish in a fish bowl. Yet here we are applying them to a population of bears that inhabit a rich two-dimensional landscape of varied habitats, exposed to trapping by an irregular and sparse array of traps. It seems questionable that the same model that is completely sensible for a population of goldfish in a bowl, should also be the right model for this population of bears distributed over a broad landscape. Ordinary capture-recapture methods are distinctly non-spatial. They don’t admit spatial indexing of either sampling (the observation process) or of individuals (the ecological process). This leads immediately to a number of practical deficiencies: (1) Ordinary CR models do not provide a coherent basis for estimating density, a problem we struggled with in the black bear study. (2) Ordinary CR model and sampling methods *induce* a form of heterogeneity that can only at best be approximated by classical models of latent heterogeneity. SCR models formally accommodate heterogeneity due to the juxtaposition of individuals with the encounter devices. (3) Ordinary CR models do not accommodate trap-

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

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

1.4 HISTORICAL CONTEXT: A BRIEF SYNOPSIS

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

1.4.1 Buffering

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

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

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

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

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

1.5 EXTENSION OF CLOSED POPULATION MODELS

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

799 1.5.1 Towards spatial explicitness: Efford’s formulation

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

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

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

822 **1.5.2 Abundance as the aggregation of a point process**

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

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

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

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

852 1.5.3 The activity center concept

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

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

875 1.5.4 The state-space

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

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

891 **1.5.5 Abundance and density**

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

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

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

1.6 CHARACTERIZATION OF SCR MODELS

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

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

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

1.7 SUMMARY AND OUTLOOK

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

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

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

STATISTICAL MODELS AND SCR

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

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

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

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

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

2.1 RANDOM VARIABLES AND PROBABILITY DISTRIBUTIONS

2.1.1 Stochasticity in ecology

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

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

1034 we need to first mention a few issues related to notation.

1035 Statisticians make things easier for themselves, and more complicated for ev-
 1036 eryone else, by using different notation for probability distributions. Sometimes
 1037 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
 1038 $\pi(X|K, p)$ or $\mathbb{P}(X|K, p)$ or $[X|K, p]$ or even just $[X]!$ Just remember that these
 1039 expressions all have the same meaning—they are all probability distributions that
 1040 tell us the probability of observing any possible realization of the random variable
 1041 X . In this book, we will almost always use bracket notation (the last two examples
 1042 above) to represent arbitrary probability distributions. Hence, from here on out,
 1043 when you see $[X|K, p]$, just remember that this is equivalent to the more traditional
 1044 expression $\Pr(X = x|K, p)$. In addition, from here on, to achieve a more concise
 1045 presentation, we will no longer use uppercase letters to denote random variables
 1046 and lowercase letters for realized values. Rather, we will define a random vari-
 1047 able by some symbol (x, N , etc...) and let the context determine whether we are
 1048 talking about the random variable itself, or realized values of it. In some limited
 1049 cases, we will want upper- and lower-case letters to represent different variables.
 1050 For example, we will often let N denote population size and n denote the number
 1051 of individuals actually detected.

1052 When we wish to be specific about a probability distribution, we will do so in
 1053 one of two ways, one mathematically precise and one symbolic. Before explaining
 1054 these two options, let's choose a specific distribution as a model for the data in our
 1055 example. In this case, the natural choice for $[x|K, p]$ is the binomial distribution,
 1056 the mathematically precise representation of which is

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

1057 The right-hand side of this equation is the binomial pmf (described in more detail
 1058 in Sec. 2.2), and plugging in values for the parameters K , and p will return the
 1059 probability of observing any realized value of the random variable x . This is precise,
 1060 but it is also cumbersome to write repetitively, and it may make the eyes glaze over
 1061 when seen too often. Thus, we will often simplify Eq. 2.1.1 using the symbolic
 1062 notation:

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

1063 The “ \sim ” symbol is meant to represent a stochastic relationship, and can be read
 1064 “is distributed as.” Another reason for using this notation is that it resembles the
 1065 syntax of the **BUGS** language, which we will frequently use to conduct Bayesian
 1066 inference.

1067 Note that once we choose a probability distribution, we have chosen a model. In
 1068 our example, we have specified our model as $x \sim \text{Binomial}(K, p)$, and because we
 1069 are assuming that the parameters are known, we can make probability statements
 1070 about future outcomes. Continuing with our fish example, we might want to know
 1071 the probability of catching $x = 7$ again after $K = 20$ casts on a future fishing

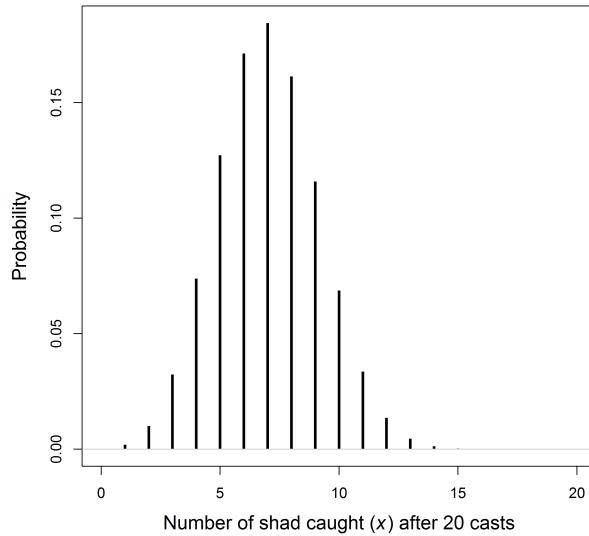


Figure 2.1. The binomial probability mass function with $N = 20$ and $p = 0.35$.

1072 outing, assuming that we know $p = 0.35$. Evaluating the binomial pmf returns a
 1073 probability of approximately 0.18, as show using this bit of **R** code:

```
1074 > dbinom(7, 20, 0.35)
1075 [1] 0.1844012
```

1076 By definition, the pmf allows us to evaluate the probability of observing any x given
 1077 $K = 20$ and $p = 0.35$, thus the distribution of the random variable can be visualized
 1078 by evaluating it for all values of x that have non-negligible probabilities, as can be
 1079 easily done in **R**:

```
1080 plot(0:20, dbinom(0:20, 20, 0.35), type="h", ylab="Probability",
1081       xlab="Number of shad caught (X)")
```

1082 the result of which is shown in Fig. 2.1 with some extra details.

1083 The purpose of this little example is to show that once we specify a model for the
 1084 random variable(s) being studied, we can begin drawing conclusions, i.e. making
 1085 inferences, about the processes of interest, even in the face of uncertainty. Prob-
 1086 ability distributions are essential to this process, and thus we need to understand
 1087 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	$\mathbf{x} \sim \text{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	$(2\pi)^{-k/2} \boldsymbol{\Sigma} ^{-1/2} \exp(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}))$	$x_k \in [-\infty, \infty]$	$\boldsymbol{\mu}$	$\boldsymbol{\Sigma}$

1088 **2.1.2 Properties of probability distributions**

1089 A pdf or a pmf is a function like any other function in the sense that it has one
 1090 or more arguments whose values determine the result of the function. However,
 1091 probability functions have a few properties that distinguish them from other func-
 1092 tions. The first is that the function must be non-negative for all possible values of
 1093 the random variable, i.e. $[x] \geq 0$. The second requirement is that the integral of
 1094 a pdf must be unity, $\int_{-\infty}^{\infty} [x] dx = 1$, and similarly for a pmf, the summation over
 1095 all possible values is unity, $\sum_x [x] = 1$. The following **R** code demonstrates this for
 1096 the normal and binomial distributions:

```
1097 > integrate(dnorm, -Inf, Inf, mean=0, sd=1)$value
1098 [1] 1
1099 > sum(dbinom(0:5, size=5, p=0.1))
1100 [1] 1
```

1101 This requirement is important to remember when one develops a non-standard
 1102 probability distribution. For example, in Chapt. 11 and 13, we work with resource
 1103 selection functions whose probability density function is not one that is pre-defined
 1104 in software packages such as **R** or **BUGS**.

1105 Another feature of probability distributions is that they can be used to compute
 1106 important summaries of random variables. The two most important summaries
 1107 are the expected value, $\mathbb{E}(x)$, and the variance $\text{Var}(x)$. The expected value, or
 1108 mean, can be thought of as the average of a very large sample from the specified
 1109 distribution. For example, one way of approximating the expected values of a
 1110 binomial distribution with $K = 20$ trials and $p = 0.35$ can be implemented in
 1111 **R** using:

```
1112 > mean(rbinom(10000, 20, 0.3))
1113 [1] 6.9865
```

1114 For most probability distributions used in this book, the expected values are known
 1115 exactly, as shown in Table 2.1, and thus we don't need to resort to such Monte Carlo
 1116 approximations. For instance, the expected value of the binomial distribution is
 1117 exactly $\mathbb{E}(x) = Kp = 20 \times 0.35 = 7$. In this case, it happens to take an integer
 1118 value, but this is not a necessary condition, even for discrete random variables.

1119 A more formal definition of an expected value is the average of all possible
 1120 values of the random variable, weighted by their probabilities. For continuous
 1121 random variables, this weighted average is found by integration:

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

1122 For example, if $[x]$ is normally distributed with mean 3 and unit variance, we could
 1123 find the expected value using the following code.

```
1124 > integrate(function(x) x*dnorm(x, 3, 1), -Inf, Inf)
1125 3 with absolute error < 0.00033
```

1126 Of course, the mean *is* the expected value of the normal distribution, so we didn't
 1127 need to compute the integral but, the point is, that Eq. 2.1.3 is generic. For
 1128 discrete random variables, the expected value is found by summation rather than
 1129 integration:

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

1130 where the summation is over all possible values of x . Earlier we approximated the
 1131 expected value of the binomial distribution with $K = 20$ trials and $p = 0.35$ by
 1132 taking a Monte Carlo average. Eq. 2.1.4 let's us find the exact answer, using this
 1133 bit of R code:

```
1134 > sum(dbinom(0:100, 20, 0.35)*0:100)
1135 [1] 7
```

1136 This is great. But of what use is it? One very important concept to understand is
 1137 that when we fit models, we are often modeling changes in the expected value of
 1138 some random variable. For example, in Poisson regression, we model the expected
 1139 value of the random variable, which may be a function of environmental variables.

1140 The ability to model the expected value of a random variable gets us very far,
 1141 but we also need a model for the variance of the random variable. The variance
 1142 describes the amount of variation around the expected value. Specifically, $\text{Var}(x) =$
 1143 $\mathbb{E}((x - \mathbb{E}(x))^2)$. Clearly, if the variance is zero, the variable is not random as
 1144 there is no uncertainty in its outcome. For some distributions, notably the normal
 1145 distribution, the variance is a parameter to be estimated. Thus, in ordinary linear
 1146 regression, we estimate both the expected value $\mu = \mathbb{E}(x)$, which may be a function
 1147 of covariates, and the variance σ^2 , or similarly the residual standard error σ . For
 1148 other distributions, the variance is not an explicit parameter to be estimated, and
 1149 instead, the mean to variance ratio is fixed. In the case of the Poisson distribution,
 1150 the mean is equal to the variance, $\mathbb{E}(x) = \text{Var}(x) = \lambda$. A similar situation is true
 1151 for the binomial distribution—the variance is determined by the two parameters K
 1152 and p , $\text{Var}(x) = Kp(1-p)$. In our earlier example with $K = 20$ and $p = 0.35$, the
 1153 variance is 4.55. Toying around with these ideas using random number generators
 1154 may be helpful. Here is some code to illustrate some of these basic concepts:

```
1155 > 20*0.35*(1-0.35)                      # Exact variance, Var(x)
1156 [1] 4.55
1157 > x <- rbinom(100000, 20, 0.35)
1158 > mean((x-mean(x))^2)                   # Monte Carlo approximation
1159 [1] 4.545525
```

2.2 COMMON PROBABILITY DISTRIBUTIONS

1160 We got a little ahead of ourselves in the previous sections by using the binomial
 1161 and Poisson distributions without describing them in detail. A solid understanding
 1162 of the binomial, Poisson, multinomial, uniform, and normal (or Gaussian) distri-
 1163 butions is absolutely essential throughout the remainder of the book. We will
 1164 occasionally make use of other distributions such as the beta, log-normal, gamma,
 1165 Dirichlet, etc... that can be helpful when modeling capture-recapture data, but
 1166 these distributions can be readily understood once you are comfortable with the
 1167 more commonly used distributions described in this section.

1168 **2.2.1 The binomial distribution**

1169 The binomial distribution plays a critical role in ecology. It is used for purposes
 1170 as diverse as modeling count data, survival probability, occurrence probability, and
 1171 capture probability, just to name a few. To describe the properties of the binomial
 1172 distribution, and related distributions, we will introduce a new example. Suppose
 1173 we are conducting a bird survey at a site in which $N = 10$ chestnut-sided warblers
 1174 (*Setophaga pensylvanica*) occur, and each of these individuals has a detection prob-
 1175 ability of $p = 0.5$. The binomial distribution is the natural choice for describing
 1176 the number of individuals that we would expect to detect (n) in this situation, and
 1177 using our notation, we can write the model as: $n \sim \text{Binomial}(10, 0.5)$. When $p < 1$,
 1178 we can expect that we will observe a different number of warblers on each of K
 1179 replicate survey occasions. To see this, we simulate data under this simple model
 1180 with $K = 3$.

```
1181 > n <- rbinom(3, size=10, prob=0.5) # Generate 3 binomial outcomes
1182 > n                                     # Display the 3 values
1183 [1] 6 4 8
```

1184 The vector of counts will typically differ each time you issue this command; however,
 1185 we know the probability of observing any value of n_k because it is defined by the
 1186 binomial pmf. As we demonstrated earlier, in R this probability can be found using
 1187 the `dbinom` function. For example, the probability of observing $n_k = 5$ is given by:

```
1188 > dbinom(5, 10, 0.5)
```

1189 This simply evaluates the function shown in Table 2.1. We could do the same more
 1190 transparently, but less efficiently, using any of the following:

```
1191 > n <- 5; N <- 10; p <- 0.5
1192 > factorial(N)/(factorial(n)*factorial(N-n))*p^n*(1-p)^(N-n)
1193 > exp(lgamma(N+1) - (lgamma(n+1) + lgamma(N-n+1)))*p^n*(1-p)^(N-n)
1194 > choose(N, n)*p^n*(1-p)^(N-n)
```

1195 Note that the last three lines of code differ only in how they compute the binomial
 1196 coefficient $\binom{N}{n}$, which is the number of different ways we could observe $n = 5$ of
 1197 the $N = 10$ chestnut-sided warblers at the site. The binomial coefficient, which is
 1198 read “N choose n” is defined as

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

1199 Now that we know how to simulate binomial data and compute the probabilities
 1200 of observing any particular outcome n , conditional on the parameters N and
 1201 p , we can contemplate the relevance of the binomial distribution in spatial capture-
 1202 recapture models. One important application of the binomial distribution is as a
 1203 model encounter frequencies. Indeed, one of the most important encounter models
 1204 in SCR will be referred to as the “binomial encounter model”, in which the number
 1205 of times individual i is captured at “trap” j after K survey occasions is modeled as
 1206 $y_{ij} \sim \text{Binomial}(K, p_{ij})$. Here, p_{ij} is the encounter probability determined, in part,
 1207 by the distance between an animal’s activity center and the trap location. This
 1208 binomial encounter model is described in detail in Sec. 7.1. Another important application
 1209 of the binomial distribution is as a prior for the population size parameter
 1210 in Bayesian analyses, as is discussed in Chapt. 4.

1211 2.2.2 The Bernoulli distribution

1212 Above, we showed 3 alternatives to `dbinom` for evaluating the binomial pmf. These
 1213 three commands differed only in how they computed the binomial coefficient, which
 1214 we needed because of the numerous ways in which we could observe $n = 5$ given
 1215 $N = 10$. To conceptualize this, let y_i be a binary variable indicating if individual i
 1216 was detected or not. Hence, given that 5 individuals were detected, the vector of
 1217 individual detections could be something like $\mathbf{y} = (0, 0, 1, 1, 1, 1, 0, 0, 0)$, indicating
 1218 that we detected individuals 3-7 but not 1-2 or 8-10. For $N = 10$ and $n = 5$,
 1219 the binomial coefficient tells us that there are 252 possible vectors \mathbf{y} with 5 ones.
 1220 However, when $N \equiv 1$, this term drops from the pmf and the result is the pmf for
 1221 the Bernoulli distribution. That is, the Bernoulli distribution is simply the binomial
 1222 distribution when $N \equiv 1$. Alternatively, we could say that the binomial distribution
 1223 is the outcome of N iid Bernoulli trials. We use the standard abbreviation “iid”
 1224 to mean *independent, identically distributed*.

1225 The utility of the Bernoulli distribution is evident when we imagine that not all
 1226 of the chestnut-sided warblers have the same detection probability. Thus, if some
 1227 individuals can be detected with probability 0.3 and others have a 0.7 detection
 1228 probability, then the model $n \sim \text{Binomial}(N, p)$ is no longer an accurate description
 1229 of system since p is no longer constant for all individuals.

To properly account for variation in p , we could redefine our model for the

counts of chestnut-sided warblers as

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

1230 This states that individual i is detected with probability p_i , and the observed count
 1231 is the sum of the N Bernoulli outcomes.

1232 An important point is that the individual-specific data y_{ik} can only be observed
 1233 if the individuals are uniquely distinguishable, such as when they are marked by
 1234 biologists with color bands. In such cases, the Bernoulli distribution allows us
 1235 to model variation in detection probability among individuals and thus would be
 1236 preferable to the binomial distribution, which assumes that each of the N indi-
 1237 viduals have the same p . For this reason, the Bernoulli distribution, as simple as
 1238 it is, is of paramount importance in capture-recapture models, including spatial
 1239 capture-recapture models in which there is virtually always substantial and impor-
 1240 tant variation in capture probability among individuals. Indeed, it could be said
 1241 that the Bernoulli model is the canonical model in capture-recapture studies, and
 1242 most of the different flavors of capture-recapture models differ primarily in how p_i
 1243 is specified.

1244 The Bernoulli pmf is given by $p^n(1-p)^{1-n}$ and hence we do not need canned
 1245 functions to facilitate its evaluation. Of course, if you wanted to, you could always
 1246 use `dbinom` with the `size` argument set to 1. For example, `dbinom(1, 1, 0.3)`
 1247 returns the Bernoulli probability of observing $n = 1$ given $p = 0.3$.

1248 2.2.3 The multinomial and categorical distributions

1249 The binomial distribution is used when we are accumulating a binary response—
 1250 that is, one in which there are two possible categories such as success/failure or
 1251 captured/not-captured. The multinomial distribution is a multivariate extension
 1252 of the binomial used when there are $G > 2$ categories. The multinomial distribution
 1253 can be thought of as a model for placing N items in the G categories, which are
 1254 also called bins or cells. Each bin has its own probability π_g and these probabilities
 1255 must sum to one. In ecology, N is often population size or the number of individuals
 1256 detected, but the definition of the G bins varies among applications. For example,
 1257 in distance sampling, when the distance data are aggregated into intervals, the
 1258 bins are the distance intervals, and the cell probabilities are functions of detection
 1259 probability in each interval (Royle et al., 2004).

1260 The multinomial distribution is widely used to model data from traditional,
 1261 non-spatial capture-recapture studies. Earlier we let y_{ik} denote a binary random
 1262 variable indicating if warbler i was detected on survey k . The vector of observations
 1263 for an individual, \mathbf{y}_i , is often referred to as the individual's "encounter history".

1264 The number of possible encounter histories depends on K , the number of survey
 1265 occasions. Specifically, there are 2^K possible encounter histories¹. If we tabulate the
 1266 number of individuals with each encounter history, the frequencies can be modeled
 1267 using the multinomial distribution.

1268 Going back to our chestnut-sided warbler example, suppose the 10 individuals
 1269 are marked and we make $K = 2$ visits to the site such that there are $2^K = 4$ pos-
 1270 sible encounter histories: (11, 10, 01, 00), where, for example, “10” is the encounter
 1271 history for an individual detected on the first visit but not the second. If $p = 1$,
 1272 then the encounter history for each of the 10 individuals must be “11”. That is, we
 1273 would detect each individual on both occasions. In this case, we the data would be:
 1274 $\mathbf{h} = (10, 0, 0, 0)$, which indicates that all 10 warblers had the first encounter history.
 1275 The corresponding cell probabilities would be $\boldsymbol{\pi} = (1, 0, 0, 0)$. What about the sit-
 1276 uation where $p < 1$, e.g. $p = 0.3$? In this case, the probability of observing the
 1277 capture history “11” (detected on both occasions) is $p \times p = 0.3 \times 0.3 = 0.09$. The
 1278 probability of observing “10” is $p \times (1 - p) = 0.21$. Following this logic, the vector
 1279 of cell probabilities is $\boldsymbol{\pi} = (0.09, 0.21, 0.21, 0.49)$. We can simulate data under this
 1280 model as follows:

```
1281 > caphist.probs <- c("11"=0.09, "10"=0.21, "01"=0.21, "00"=0.49)
1282 > drop(rmultinom(1, 10, caphist.probs))
1283 11 10 01 00
1284 0 3 2 5
```

1285 The result of our simulation is that zero individuals were observed with the capture
 1286 history “11” and 5 individuals were observed with the capture history “00”. The
 1287 other 5 individuals were observed one out of the two occasions. This is not such a
 1288 surprising outcome given $p = 0.3$.

1289 As in non-spatial capture-recapture studies, the multinomial distribution turns
 1290 out to be very important in spatial capture-recapture studies. However, N is not
 1291 defined as population size. Rather, we use the multinomial distribution when an
 1292 individual can only be captured in a single trap during an occasion. Thus $N = 1$
 1293 and the cell probabilities are the probabilities of being captured in each trap. A
 1294 thorough discussion of this point can be found in Chapt. 9. Another application
 1295 of the multinomial distribution in SCR models is discussed in Chapt. 11 where we
 1296 discuss how to model the probability that an individual’s activity center is located
 1297 in one of the cells of a raster defining the spatial region of interest.

1298 Just as the Bernoulli distribution is the elemental form of the binomial distri-
 1299 bution (being the case $N = 1$), the categorical distribution is essentially equivalent
 1300 to the multinomial distribution with size parameter $N \equiv 1$. The only difference is
 1301 that, rather than returning a vector with a single element equal to 1, it returns the
 1302 element *location* where the 1 occurs. For example, if $\mathbf{y} = (0, 0, 1, 0)$ is an outcome

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

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

```
1309 > which(rmultinom(1, 1, c(0.1, 0.7, 0.2)) == 1)
1310 [1] 2
```

1311 2.2.4 The Poisson distribution

1312 The Poisson distribution is the canonical model for count data in ecology. More
 1313 generally, the Poisson distribution is a model for random variables taking on non-
 1314 negative, integer values. Although it is a simple model having just one parameter,
 1315 $\lambda = \mathbb{E}(x) = \text{Var}(x)$, its applications are highly diverse, including as a model of
 1316 spatial variation in abundance or as a model for the frequency of behaviors over
 1317 time. Just as logistic regression is the standard generalized linear model (GLM)
 1318 used to model binary data, Poisson regression is the default GLM for modeling
 1319 count data and variation in λ .

1320 The Poisson distribution is related to both the binomial and multinomial distri-
 1321 butions, and the following three bits of trivia are occasionally worth knowing. First,
 1322 it is the limit of the binomial distribution as $N \rightarrow \infty$ and $p \rightarrow 0$, which means that
 1323 for high values of N and low values of p , $\text{Poisson}(N \times p)$ is approximately equal
 1324 to $\text{Binomial}(N, p)$. Second, if $\{n_1 \sim \text{Poisson}(\lambda_1), \dots, n_K \sim \text{Poisson}(\lambda_K)\}$ then the
 1325 vector of counts is multinomial, $\{n_1, \dots, n_K\} \sim \text{Multinomial}(\sum_k n_k, \{\frac{\lambda_1}{\sum_k \lambda_k}, \dots, \frac{\lambda_K}{\sum_k \lambda_k}\})$.
 1326 Third, the sum of two Poisson random variables $x_1 \sim \text{Poisson}(\lambda_1)$ and $x_2 \sim$
 1327 $\text{Poisson}(\lambda_2)$ is also Poisson: $x_1 + x_2 \sim \text{Poisson}(\lambda_1 + \lambda_2)$.

1328 The Poisson distribution has two important uses in spatial capture-recapture
 1329 models: (1) as a prior distribution for the population size parameter N , and (2) as a
 1330 model for the frequency of captures in a trap. In the first context, the Poisson prior
 1331 for N results in a Poisson point process for the location of the N activity centers
 1332 in the region of interest. This topic is discussed in Chapt. 5 and Chapt 11. The
 1333 second use of the Poisson distribution in spatial capture-recapture is to describe
 1334 data from sampling methods in which an individual can be detected multiple times
 1335 at a trap during a single occasion. For example, in camera trapping studies we
 1336 might obtain multiple pictures of the same individual at a trap during a single
 1337 sampling occasion. Thus, λ in this case would be defined as the expected number
 1338 of detections or captures per occasion.

1339 2.2.5 The uniform distribution

1340 The lowly uniform distribution is a continuous distribution whose only two pa-
 1341 rameters are the lower and upper bounds that restrict the possible values of the

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

1350 Another common usage of the uniform distribution is as a prior for the coor-
 1351 dinates of points in the real plane, i.e. in two-dimensional space. Such a use of
 1352 the uniform distribution implies that a point process is “homogeneous”, meaning
 1353 that the location of one point does not affect the location of another point and
 1354 that the expected density of points is constant throughout the region. Thus, to
 1355 simulate a realization from a homogeneous Poisson point process in the unit square
 1356 $[0, 1] \times [0, 1]$, we could use the following **R** code:

```
1357 D <- 100      # points per unit area
1358 A <- 1        # Area of unit square
1359 N <- rpois(1, D*A)
1360 plot(s <- cbind(runif(N), runif(N)))
```

1361 where \mathbf{s} is a matrix of coordinates with N rows and 2 columns. We will often
 1362 represent the uniform point process using the following notation:

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

1363 where \mathcal{S} is some specific unit of space called the state-space of the random variable
 1364 \mathbf{s} . It would be more correct to somehow distinguish this two-dimensional uniform
 1365 distribution for the univariate one. That is, it might be more clear to use notation
 1366 such as $\mathbf{s} \sim \text{Uniform}_2(\mathcal{S})$ instead, but this is somewhat cumbersome, so we will opt
 1367 for the former expression.

1368 2.2.6 Other distributions

1369 The other continuous distributions that are regularly encountered in SCR models
 1370 are primarily used as priors in Bayesian analyses, and thus we will avoid a lengthy
 1371 discussion of their properties. The normal distribution, also called the Gaussian
 1372 distribution, is perhaps the most widely recognized and applied probability model
 1373 in statistics, but it plays only a minor role in SCR models other than as a model for
 1374 signal strength in acoustic SCR models (Efford et al., 2009b; Dawson and Efford,
 1375 2009), and see Sec. 9.4. Nonetheless, it is the canonical prior for any continuous
 1376 random variable with infinite support, and thus it is often used as a prior when
 1377 applying Bayesian methods. One common usage is as a prior for the β coefficients
 1378 of a linear model defining some parameter as a function of covariates (usually on

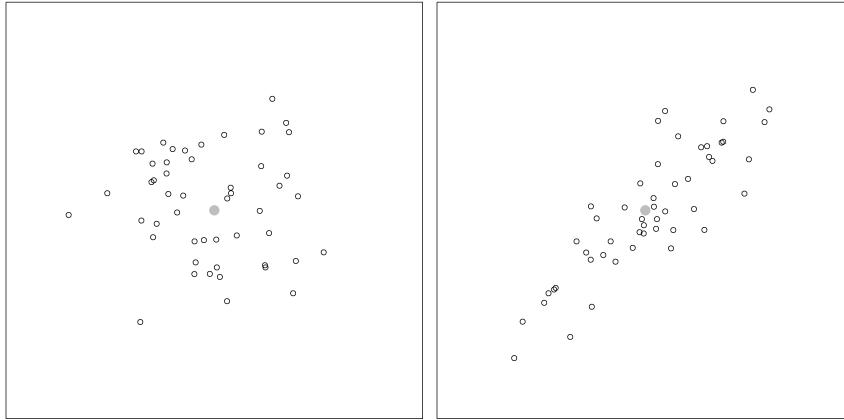


Figure 2.2. Two realized point patterns from the bivariate normal distribution.

1379 a transformed scale). An example, including a cautionary note, is provided in
 1380 Sec. 3.5.1. Be aware that although the normal distribution is typically parameter-
 1381 ized in terms of the variance parameter σ^2 , in the **BUGS** language, the inverse of
 1382 the variance, or precision, is used instead, $\tau = 1/\sigma^2$. In **R**, the **dnorm** function
 1383 requires the standard deviation σ , rather than the variance σ^2 .

1384 The bivariate normal distribution is a generalization of the normal distribution
 1385 and a special case of the multivariate normal distribution whose pdf is shown in
 1386 Table 2.1. The bivariate normal distribution is used to model two (possibly) depen-
 1387 dent continuous variables whose symmetric variance-covariance matrix is denoted
 1388 Σ . In SCR models, we most often use this model as a rudimentary description of
 1389 movement outcomes about a home range center. If there is no correlation, then the
 1390 model reduces to two independent normal draws along the coordinate axes. The
 1391 following code generates bivariate normal outcomes with no correlation ($\rho = 0$), as
 1392 well as outcomes in which the correlation is $\rho = 0.9$.

```
1393 library(mvtnorm)
1394 set.seed(3)
1395 mu <- c(0,0)
1396 Sigma <- matrix(c(1, .9, .9, 1), 2, 2)
1397 X1 <- cbind(rnorm(50, mu[1], Sigma[1,1]), # No correlation (rho=0)
1398             rnorm(50, mu[2], Sigma[2,2]))
1399 X2 <- rmvnorm(50, mu, Sigma)           # rho=0.9
```

1400 Fig. 2.2 shows the simulated points.

1401 Several of the parameters in capture-recapture models do not have infinite sup-
 1402 port, but instead are probabilities restricted to the range [0, 1], or are positive

1403 valued living between zero and ∞ . The beta distribution is the standard prior
 1404 used for probabilities because it can be used to express either a lack of knowledge
 1405 or very precise knowledge about a parameter. For example, a Beta(1, 1) distribu-
 1406 tion is equivalent to a Uniform(0, 1) distribution. However, unlike the uniform
 1407 distribution, the beta distribution can be used as an informative prior; for exam-
 1408 ple if published estimates of detection probability exist we can choose parameters
 1409 of the beta distribution to reflect that. To gain some familiarity with the beta
 1410 distribution, execute the following R commands:

```
1411 curve(dbeta(x, 1, 1), col="black", ylim=c(0,5))
1412 curve(dbeta(x, 10, 10), col="blue", add=TRUE)
1413 curve(dbeta(x, 10, 20), col="darkgreen", add=TRUE)
```

1414 Other parameters in SCR models are continuous but positive-valued and can be
 1415 modeled using the gamma distribution. As with the beta distribution, the gamma
 1416 distribution is typically favored over the uniform distribution when one is interested
 1417 in using an informative prior. It is also frequently used as a vague prior for the
 1418 inverse of variance parameters, but it is wise to compare this prior to a uniform to
 1419 assess its influence on the posterior.

2.3 STATISTICAL INFERENCE AND PARAMETER ESTIMATION

1420 If the parameters of a statistical model were known with absolute certainty, then it
 1421 would be possible to use pdfs and pmfs to make direct probability statements about
 1422 unknowns such as future outcomes. However, we almost never know the actual
 1423 values of parameters, and instead we have to estimate them from observations
 1424 (i.e., data). Our inferences must then acknowledge the uncertainty associated with
 1425 our imperfect knowledge of the parameters. Doing so is most often accomplished
 1426 using one of two approaches: classical (frequentist) inference or Bayesian inference.
 1427 These two modes of inference regard the uncertainty about parameters in entirely
 1428 different ways. In the next chapter, we will review some of the important concepts
 1429 in Bayesian inference, so here, we will focus on the frequentist perspective.

1430 Suppose we count oak trees at J sites, and the resulting data $\{y_1, \dots, y_J\}$ can
 1431 be assumed to be *iid* outcomes from some distribution, such as the Poisson with
 1432 unknown parameter λ . We want to estimate this parameter. In classical inference,
 1433 the only uncertainty about λ is that attributable to sampling. For instance, we can
 1434 imagine repeatedly sampling the population (sites in this example) and obtaining
 1435 sample-specific estimates of λ . Typically, we entertain the idea that there are an
 1436 infinite number of possible samples and so we could obtain an infinite number of
 1437 estimates: $\{\hat{\lambda}_1, \hat{\lambda}_2, \dots, \hat{\lambda}_\infty\}$. If these estimates are produced using the method
 1438 of maximum likelihood, and as n tends to infinity, the distribution of estimates,
 1439 called the sampling distribution, will be normally distributed with $E(\hat{\lambda}) = \lambda$. The
 1440 standard deviation of the sampling distribution is called the standard error, which
 1441 can also be estimated as part of the maximum likelihood procedure. Of course, we

1442 almost always have just a single sample of data, and hence a single $\hat{\lambda}$ and a single
 1443 estimate of the standard error. However, under the assumption of a normally
 1444 distributed sampling distribution, we can construct a confidence interval that will
 1445 include the true value of λ with coverage probability $1 - \alpha$, where α is a prescribed
 1446 value like 0.05. An important point is that there is no uncertainty associated with
 1447 the actual parameter—it is regarded as a fixed value, and hence probability is only
 1448 used to characterize the estimator via its sampling distribution.

1449 Maximum likelihood is heuristically a method of finding the most “likely” value
 1450 of λ , given the observed data, and of characterizing the variance of the sampling dis-
 1451 tribution. Of course, it also applies to cases where the observations are multivariate,
 1452 or the probability distribution is a function of multiple parameters. Endless num-
 1453 bers of textbooks and online resources are available for those interested in a detailed
 1454 explanation of maximum likelihood. For our purposes, we wish to keep it simple
 1455 and focus on *how* to do it. The first step is to define the likelihood function, which
 1456 is the joint distribution of the data regarded as a function of the parameter(s). If
 1457 the joint distribution of the observations is denoted by $[y_1, y_2, \dots, y_n | \lambda]$, we usually
 1458 denote the likelihood by flipping the arguments: $\mathcal{L}(\lambda | \mathbf{y}) = [\lambda | y_1, y_2, \dots, y_n]$.

1459 If the observations are *iid*, the likelihood simplifies to

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

1460 where $[y_i | \lambda]$ is a probability distribution, like those discussed in the previous sec-
 1461 tions. For example, if y_i is Poisson distributed, then $[y_i | \lambda] = \text{Poisson}(\lambda) = \frac{\lambda^{y_i} e^{-\lambda}}{y_i!}$.
 1462 Although likelihoods are typically shown on the natural scale, we almost always
 1463 maximize the logarithm of the likelihood to avoid computational problems that
 1464 arise when multiplying very small probabilities. Thus, we rewrite Eq. 2.3.1 as

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

1465 Here is some simple **R** code to simulate independent Poisson outcomes and esti-
 1466 mate λ (as though we did not know it) using the method of maximum likelihood.
 1467 Actually, we will minimize the negative log-likelihood because it is equivalent and
 1468 is the default for **R**’s optimizers like `optim` and `nls`.

```
1469 > lambda <- 3                      # Actual parameter value
1470 > y1 <- rpois(100, lambda)        # Realized values (data)
1471 > negLogLike1 <- function(par) -sum(dpois(y1, par, log=TRUE))
1472 > starting.value <- c('lambda'=1)
1473 > optim(starting.value, negLogLike1)$par # MLE
1474   lambda
1475 3.039844
```

1476 Explicitly maximizing the likelihood, numerically, isn't actually necessary here be-
 1477 cause the MLE of λ is given by the mean of the observations. A more interesting
 1478 example is when there are covariates of λ . For example, suppose λ is a function of
 1479 elevation and vegetation height according to: $\log(\lambda_i) = \beta_0 + \beta_1 ELEV_i + \beta_2 VEGHT_i$.
 1480 This is a standard Poisson regression problem, with likelihood:

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

1481 This likelihood is almost identical to the previous one except that λ is now a
 1482 function, and so we need to estimate the parameters of the function, i.e. the β 's.
 1483 Some code to fit this model to simulated data is shown here:

```
1484 > nsites <- 100
1485 > elevation <- rnorm(100)
1486 > veght <- rnorm(100)
1487 > beta0 <- 1
1488 > beta1 <- -1
1489 > beta2 <- 0
1490 > lambda <- exp(beta0 + beta1*elevation + beta2*vegght)
1491 > y2 <- rpois(nsites, lambda)
1492 > negLogLike2 <- function(pars) {
1493   +   beta0 <- pars[1]
1494   +   beta1 <- pars[2]
1495   +   beta2 <- pars[3]
1496   +   lambda <- exp(beta0 + beta1*elevation + beta2*vegght)
1497   +   -sum(dpois(y2, lambda, log=TRUE))
1498   +
1499 > starting.values <- c('beta0'=0, 'beta1'=0, 'beta2'=0)
1500 > optim(starting.values, negLogLike2)$par
1501       beta0      beta1      beta2
1502     0.98457756 -1.03025173 -0.01218292
```

1503 We see that the maximum likelihood estimates (MLEs) are very close to the true
 1504 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:

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

1505 This is an example of a mixed effects model or a hierarchical model. How do we
 1506 estimate the parameters of a model that includes random effects? Earlier the like-
 1507 lihood function was written as the product of probabilities determined by a single
 1508 pmf or pdf, $[y|\lambda]$, but now we have an additional random variable, and we are forced
 1509 to think about conditional relationships, because y depends upon β_i and β_i depends
 1510 upon other parameters, specifically μ and σ^2 . This type of conditional dependence
 1511 among parameters is the essence of hierarchical models, and statistical analysis
 1512 of hierarchical models requires that we discuss joint distributions, marginal distri-
 1513 butions and conditional distributions. These concepts will be used extensively in
 1514 Chapt. 6 where we demonstrate how to estimate parameters of hierarchical models
 1515 using maximum likelihood.

2.4 JOINT, MARGINAL, AND CONDITIONAL DISTRIBUTIONS

1516 So far we have restricted our attention to situations in which we wish to make
 1517 inference about a single random variable. However, in ecology, we often are inter-
 1518 ested in multiple random variables and how they are related. Let Y be a random
 1519 variable that may or may not be independent of X (here again we will distinguish
 1520 between random variables and realized values for conceptual clarity). Inference
 1521 about these two random variables can be made using the joint, marginal, or condi-
 1522 tional distributions—or, we may make use of all of them depending on the question
 1523 being asked. In the case of discrete random variables, the joint distribution is the
 1524 probability that X takes on the value x and that Y takes on the value y , which
 1525 is written $[X = x, Y = y]$. To clarify this concept, let's go back to our original
 1526 example where X was the number of fish caught after 20 casts, which we said
 1527 was an *iid* binomial random variable. Now, let's suppose that X depends on the
 1528 random variable Y , which is the number of other fisherman at the hole. Specifi-
 1529 cally, let's say that the probability of catching a fish p is related to Y according
 1530 to $\text{logit}(p) = -0.6 + -2y$. Furthermore, let's make the intuitive assumption that
 1531 the number of fishermen at the hole is a Poisson random variable with mean 0.6,
 1532 i.e. $Y \sim \text{Poisson}(0.6)$. Our model is now fully specified, and so we can answer the
 1533 question: “what is the probability of catching x fish and of there being y fishermen
 1534 at the hole”. This joint distribution is given by the product of the binomial pmf
 1535 (with p determined by y) and the Poisson pmf with $\lambda = 0.6$. The following R code
 1536 creates the joint distribution.

```
1537 > X <- 0:20 # All possible values of X
1538 > Y <- 0:10 # All possible values of Y
1539 > lambda <- 0.6
```

```

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

```

1574 This matrix tells us the probability of all possible combinations of x and y , and
 1575 we see that the most likely value is $(X = 1, Y = 1)$, i.e. we will catch 1 fish and
 1576 there will be 1 other fisherman. This matrix also demonstrates the law of total
 1577 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:

```
1583 > margX <- rowSums(joint)
1584 > round(margX, 2)
1585   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
1586 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
1587 X=15  X=16  X=17  X=18  X=19  X=20
1588 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 .

```
1591 > XgivenY <- joint/matrix(margY, nrow(joint), ncol(joint), byrow=TRUE)
1592 > round(XgivenY, 2)
1593   Y=0  Y=1  Y=2  Y=3  Y=4  Y=5  Y=6  Y=7  Y=8  Y=9  Y=10
1594   X=0  0.00 0.25 0.82 0.97  1  1  1  1  1  1  1
1595   X=1  0.00 0.36 0.16 0.03  0  0  0  0  0  0  0
1596   X=2  0.01 0.25 0.02 0.00  0  0  0  0  0  0  0
1597   X=3  0.03 0.11 0.00 0.00  0  0  0  0  0  0  0
1598   X=4  0.07 0.03 0.00 0.00  0  0  0  0  0  0  0
1599   X=5  0.13 0.01 0.00 0.00  0  0  0  0  0  0  0
1600   X=6  0.17 0.00 0.00 0.00  0  0  0  0  0  0  0
1601   X=7  0.18 0.00 0.00 0.00  0  0  0  0  0  0  0
```

1604	X=8	0.16	0.00	0.00	0.00	0	0	0	0	0	0	0
1605	X=9	0.12	0.00	0.00	0.00	0	0	0	0	0	0	0
1606	X=10	0.07	0.00	0.00	0.00	0	0	0	0	0	0	0
1607	X=11	0.03	0.00	0.00	0.00	0	0	0	0	0	0	0
1608	X=12	0.01	0.00	0.00	0.00	0	0	0	0	0	0	0
1609	X=13	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
1610	X=14	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
1611	X=15	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
1612	X=16	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
1613	X=17	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
1614	X=18	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
1615	X=19	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
1616	X=20	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0

1617 Note that we have 11 probability distributions for X , one for each possible value of
 1618 Y , and each pmf sums to unity as it should. Note also that if you show up at the
 1619 hole and there are > 2 fisherman, your chance of catching a fish is very low. Go
 1620 home. These concepts are explained in more detail in other texts such as Casella
 1621 and Berger (2002), Royle and Dorazio (2008), and Link and Barker (2010), but
 1622 hopefully, the code shown here complements the equations and makes it easier for
 1623 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:

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

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

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

1624 From here on out, when you see such notation, you should immediately grasp
 1625 the fact that Y is a random variable independent of X , but X depends upon
 1626 Y through p . Now you have the tools to make probability statements about the
 1627 random variables in this system. The one caveat faced in reality is that we typically
 1628 do not know the values of the parameters, and instead we have to estimate them.
 1629 Maximum likelihood methods for hierarchical models are covered in Chapt. 6.

2.5 HIERARCHICAL MODELS AND INFERENCE

1630 The term hierarchical modeling (or hierarchical model) has become something of
 1631 a buzzword over the last decade with hundreds of papers published in ecological
 1632 journals using that term. So then, what exactly is a hierarchical model, anyhow?
 1633 Obviously, this term stems from the root “hierarchy” which means:

1634 **Definition:** *hierarchy* (noun) – a series of ordered groupings of people or things
 1635 within a system;

1636 In the case of a hierarchical model (hierarchical being the adjective form of hi-
 1637 erarchy), the “things” are probability distributions, and they are ordered according
 1638 to their conditional probability structure. Thus, a hierarchical model is *an ordered*
 1639 *series of models, ordered by their conditional probability structure.*

1640 A canonical hierarchical model in ecology is this elemental model of species
 1641 occurrence or distribution (MacKenzie et al., 2002; Tyre et al., 2003; Kéry, 2011):

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

1642

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

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

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

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

1661 Most models considered in this book describe the encounter of individuals con-
 1662 ditional on the “activity center” of the individual, which is a latent variable (i.e.,
 1663 unobserved random effect). The definition of an activity center will be context-
 1664 dependent as discussed in Chapt. 5, but often it can be thought of as an individual’s
 1665 home range center. The collection of these latent variables represents the outcome
 1666 of an ecological process describing how individuals distribute themselves over the
 1667 landscape. Moreover, how individuals are encountered in traps is, in some cases,
 1668 the result of a model governing movement. As such, these models are examples of
 1669 hierarchical models that contain formal model components representing both eco-
 1670 logical process and also the observation of that process. That is, they are explicit
 1671 hierarchical models (Royle and Dorazio, 2008) as opposed to implicit hierarchical
 1672 models.

2.6 CHARACTERIZATION OF SCR MODELS

1673 For the purposes of this book, an SCR model is any “individual encounter model”
 1674 (not just “capture-recapture”!) where auxiliary spatial information is also obtained.
 1675 To be more precise we could as well use the term “spatial capture and/or recap-
 1676 ture” but that is slightly unwieldy and, besides, it also abbreviates to SCR. The
 1677 class of SCR models includes traditional capture-recapture models with auxiliary
 1678 spatial information and even some models that do not even require “recapture”
 1679 (e.g., distance sampling). There is even a class of models (Chapt. 18) which don’t
 1680 require capture or unique identification of individuals.

1681 Conceptually, SCR models involve a collection of random variables, \mathbf{s} , \mathbf{u} and
 1682 y where \mathbf{s} is the activity center, or home range center, \mathbf{u} is the location of the
 1683 individual at the time of sampling, which we may think of as a realization from some
 1684 movement model, and y is the “response variable”—what the observer records. For
 1685 example, $y = 1$ means “detected” and $y = 0$ means “not detected”, but many other
 1686 types of responses are possible (Chapt 9). A broad class of models for estimating
 1687 density are unified by a hierarchical model involving explicit models for animal
 1688 activity centers \mathbf{s} , movement outcomes \mathbf{u} , and encounter data y . In some cases, we
 1689 don’t observe y but rather summaries of y , say $n(y)$, yet it might be convenient
 1690 in such cases to retain an explicit focus on y in terms of model construction. We
 1691 thus introduce a sequence of models—a hierarchical model—to relate these random
 1692 variables, which can be written as

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

1693 Every model we talk about in this book has a subset of these components although
 1694 we never fit the full model because we have not encountered a situation requiring
 1695 that we do so. However, a detailed description of this model and its various com-
 1696 ponents is the subject of this book, and we will not pretend to condense hundreds
 1697 of pages of material into the next few paragraphs. However, we give a cursory
 1698 overview here to whet the appetite and provide some indication of where we are
 1699 going. Don’t worry if some of this material doesn’t sink in just yet—we will walk
 1700 through it slowly in the subsequent chapters.

1701 Let’s begin with the model $[\mathbf{s}]$ that describes the distribution of the activity
 1702 centers of each animal in the spatial region \mathcal{S} (the state-space as we called it previ-
 1703 ously). As will be explained in Chapt. 5 and Chapt. 11, $[\mathbf{s}]$ defines a spatial point
 1704 process, which may be inhomogeneous if there exists spatial variation in density, or
 1705 it may be homogeneous if density is constant throughout \mathcal{S} . In the later case, we can
 1706 write $[\mathbf{s}] = \text{Uniform}(\mathcal{S})$, which is to say that the N activity centers are uniformly
 1707 distributed in the polygon \mathcal{S} . A point process is also a model for the number of indi-
 1708 viduals in the population N . So we could write $[\mathbf{s}|\mu]$ where μ is an intensity param-
 1709 eter defined as the number of points per unit area. In other words, μ is population
 1710 density, and we often model population size as either $N \sim \text{Poisson}(\mu A(\mathcal{S}))$, where
 1711 $A(\mathcal{S})$ is the area of the state-space; or, $N \sim \text{Binomial}(M, \psi)$ where $\psi = \mu A(\mathcal{S})/M$

1712 and M is some large integer used simply as a convenience measure when conducting
 1713 Bayesian analysis. As it turns out, there is very little practical difference in the
 1714 Poisson prior versus a binomial models for N (Chapt. 11).

1715 The model $[\mathbf{u}|\mathbf{s}]$ describes the locations of animals conditional on their activity
 1716 center. In the original formulation of SCR models (Efford, 2004), this model com-
 1717 ponent was intentionally ignored. Indeed when movement is not of direct interest,
 1718 or when \mathbf{s} is defined in a way not related to a home range center, it may be prefer-
 1719 able to ignore this model component (Borchers, 2012). In other cases, we might use
 1720 an explicit model, such as the bivariate normal model (Royle and Young, 2008).

1721 The third component of the model, $[y|\mathbf{u}]$, describes how the observed data—the
 1722 so-called capture-histories—arise conditional on the locations of animals. However,
 1723 as mentioned previously, most SCR models do not contain a movement model, and
 1724 thus, we typically entertain the model $[y|\mathbf{s}]$ instead of $[y|\mathbf{u}]$. This encounter model
 1725 generally has at least two parameters, say p_0 and σ , describing the probability of
 1726 capturing or detecting an individual given the distance between \mathbf{s} and the trap.
 1727 The most basic model is often called the half-normal model, although we typically
 1728 refer to it as the Gaussian model since, in two-dimensional space, it is the kernel
 1729 of a bivariate normal distribution. The model is $p_{ij} = p_0 \exp(-\|\mathbf{x}_j - \mathbf{s}_i\|/(2\sigma^2))$
 1730 where p_0 is the capture probability when the activity center occurs at the trap
 1731 location \mathbf{x}_j , and σ is a spatial scale parameter determining how rapidly capture
 1732 probability declines with distance. One common design leads to the model $[y_{ij}|\mathbf{s}_i] =$
 1733 Bernoulli(p_{ij}). Chapt. 5 and Chapt. 9 describe many other possible encounter
 1734 models.

1735 When individuals are marked by biologists or have natural markings permit-
 1736 ting individual recognition, y_{ij} is the observed data. However, some or all of the
 1737 individuals cannot be uniquely identified, then we cannot record this individual-
 1738 specific encounter history data. Instead, the data might be simply the number of
 1739 detections at a trap or perhaps binary detection/non-detection data at each trap on
 1740 each survey occasion. We call this reduced information data $n(y)$, and Chapt. 18
 1741 and Chapt. 19 describe models for $[n(y)|y]$ that still allow for density estimation.
 1742 The basic strategy is to view y as “missing data” and to use the spatial correlation
 1743 in the counts, or other sources of information, to provide information about these
 1744 latent encounter histories.

1745 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)

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

1752 We realize that many the model description in Eq. 2.6.2 may not be self-evident
 1753 to some ecologists. However, it is absolutely essential that one can understand
 1754 such a model description—not just for being able to read this book, but also for
 1755 understanding any statistical model in ecology. One of the best ways of familiarizing
 1756 oneself with this notation is to translate it into **R** code that simulates outcomes
 1757 from the model. The following code is an example.

```
1758 set.seed(36372)
1759 Area <- 1 # area of state-space (unit square)
1760 x <- cbind(rep(seq(.1,.9,.2), each=5), # trap locations
1761             rep(seq(.1,.9,.2), times=5))
1762 p0 <- 0.3 # baseline capture probability
1763 sigma <- 0.05 # Gaussian scale parameter
1764 mu <- 50 # population density
1765 N <- rpois(1, mu*Area) # population size
1766 s <- cbind(runif(N, 0, 1), # activity centers in unit square
1767             runif(N, 0, 1))
1768 K <- 5
1769 y <- matrix(NA, N, nrow(x)) # capture data
1770 for(i in 1:N) {
1771   d.ij <- sqrt((x[,1] - s[i,1])^2 + # distance between x and s[i]
1772                 (x[,2] - s[i,2])^2)
1773   p.ij <- p0*exp(-d.ij^2 / (2*sigma^2)) # capture probability
1774   y[i,] <- rbinom(nrow(x), K, p.ij) # capture history for animal i
1775 }
```

1776 Fig. 2.3 shows the results of this simulation from a basic, yet very useful, SCR
 1777 model.

1778 Having briefly explained each of the model components in Eq. 2.6.1, and having
 1779 shown how a subset of these components results in a basic SCR model, we can
 1780 now discuss other relevant arrangements. Examples include: (1) Classical distance
 1781 sampling (Buckland et al., 2001; Borchers et al., 2002), (2) Spatial capture-recapture
 1782 models with fixed arrays of traps (Efford, 2004; Borchers and Efford, 2008; Royle
 1783 et al., 2009a,b; Gardner et al., 2010a; Royle et al., 2011b), and (3) Search-encounter
 1784 models (Royle and Young, 2008; Royle et al., 2011a). We will now elaborate on
 1785 some of these distinctions.

1786 1. **Distance sampling.** The last 2 stages of the hierarchy are confounded
 1787 (implicitly) and so analysis is based on the model $[y|\mathbf{u}][\mathbf{u}]$. The “process
 1788 model” is that of “uniformity”: $\mathbf{u} \sim \text{Uniform}(\mathcal{S})$.

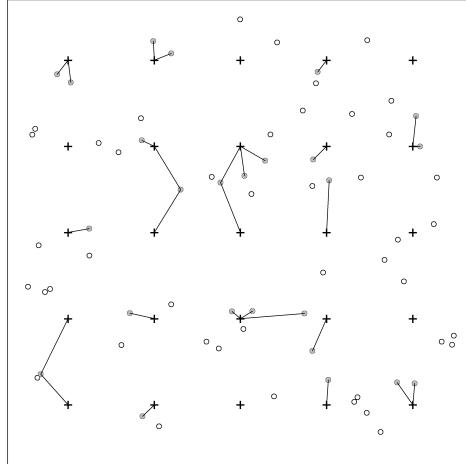


Figure 2.3. Population of $N = 69$ home-range centers (s , circles) and 25 trap locations (x , crosses). Lines connect activity centers to the traps where the individuals were detected. As in many SCR models, movement outcomes (u) are ignored.

1789 2. **Spatial capture-recapture model with a fixed array of traps.** SCR
 1790 models appear to have little in common with distance sampling because ob-
 1791 servations are made only at a pre-defined set of discrete locations—where
 1792 traps are placed. However, the models are closely related in terms of our
 1793 hierarchical representation above. In SCR models based on fixed arrays, we
 1794 cannot estimate both $\Pr(y = 1|u)$ and $\Pr(u|s)$ —the probability that an in-
 1795 dividual “moves to u ” cannot be separated from the probability that it is
 1796 detected given that it moves to u , because of the fact that the observation
 1797 locations are fixed by design. Formally, such SCR models confound $[y|u]$
 1798 with $[u|s]$ so that the observation model arises as:

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

1799 This confounding happens because SCR sampling is spatially biased—restricted
 1800 to a fixed pre-determined set of locations. Conversely, distance sampling
 1801 confounds $[u|s][s]$ because, essentially, there is only a single realization of the
 1802 encounter process. It is probably reasonable to assume that $\Pr(y = 1|u) = 1$
 1803 or at least it is locally constant for most devices (e.g., cameras, etc..), and
 1804 thus the detection model will have the interpretation in terms of movement
 1805 (see Chapt. 13 and 12).

1806 3. **Search-encounter models.** What we call “search-encounter” models (Royle

1807 and Young, 2008; Royle et al., 2011a) are kind of a hybrid model combining
1808 features of SCR models and features of distance sampling. Like distance
1809 sampling they allow for encounters in continuous space which provide di-
1810 rect observations from $[\mathbf{u}|\mathbf{s}]$. Thus, the hierarchical model is fully identified.
1811 These models are described in Chapt. chapt.search-encounter.

2.7 SUMMARY AND OUTLOOK

1812 Spatial capture-recapture models are hierarchical models, and hierarchical models
1813 are models of multiple random variables that are conditionally related. It is there-
1814 fore important that the basic rules of modeling random variables are understood,
1815 and we hope that this chapter has made some of the basic concepts accessible to
1816 ecologists with rudimentary background in statistics. If some of this material still
1817 seems difficult to grasp, we recommend working with the provided **R** code, which
1818 is perhaps the best way of making the equations more tangible.

1819 In some respects, it is possible to understand the jist of SCR without knowing
1820 anything about marginal and conditional relationships. One can always fit models
1821 using canned software and interpret the output without understanding the guts of
1822 the model or the details of the estimation process. For some applied ecologists,
1823 this may be perfectly fine, and this book is meant to be useful for both statistical
1824 novices and ecologists with more advanced quantitative skills. In most chapters, we
1825 begin with a basic conceptual discussion, then we explain the technical details that
1826 require an understanding of the concepts in this chapter, and finally we end with
1827 one or more worked examples. For those not interested in the technical details,
1828 we recommend focusing on the chapter introductions and the examples. However,
1829 taking the time to understand the concepts presented in this chapter can only
1830 increase one's ability to tackle the unique and complex problems that often present
1831 themselves when modeling spatial and temporal aspects of population dynamics.

1832
1833

1834

3

GLMS AND BAYESIAN ANALYSIS

1835 A major theme of this book is that spatial capture-recapture models are, for the
1836 most part, just generalized linear models (GLMs) wherein the covariate, distance
1837 between trap and home range center, is partially or fully unobserved – and therefore
1838 regarded as a random effect. Outside of capture-recapture, such models are usually
1839 referred to as generalized linear mixed models (GLMMs) and, therefore, SCR mod-
1840 els can be thought of as a specialized type of GLMM. Naturally then, we should
1841 consider analysis of these slightly simpler models in order to gain some experience
1842 and, hopefully, develop a better understanding of spatial capture-recapture models.

1843 In this chapter, we consider classes of GL(M)Ms – Poisson and binomial (i.e.,
1844 logistic regression) models – that will prove to be enormously useful in the analysis
1845 of capture-recapture models of all kinds. Many readers are likely familiar with these
1846 models already because they are among the most useful models in ecology and,
1847 as such, have received considerable attention in many introductory and advanced
1848 texts. We focus on them here in order to introduce the readers to the analysis of
1849 such models in **R** and **WinBUGS** or **JAGS**, which we will translate directly to
1850 the analysis of SCR models in subsequent chapters.

1851 Bayesian analysis is convenient for analyzing GL(M)Ms because it allows us to
1852 work directly with the conditional model – i.e., the model that is conditional on the
1853 random effects, using computational methods known as Markov chain Monte Carlo
1854 (MCMC). Learning how to do Bayesian analysis of GLMs and GLMMs using the
1855 **BUGS** language is, in part, the purpose of this chapter. We focus here on the use of
1856 **WinBUGS** because it is the most popular “**BUGS** engine”. However, later in the
1857 book we transition to another popular **BUGS** engine known as **JAGS** (Plummer,
1858 2009) which stands for *Just Another Gibbs Sampler*. For most of our purposes, the
1859 specification of models in either platform is the same, but **JAGS** is under active
1860 development at the present time while **WinBUGS** no longer is, having transitioned

1861 to **OpenBUGS** (Lunn et al., 2009) which is still in active development. While we
 1862 use **BUGS** of one sort or another to do the Bayesian computations, we organize and
 1863 summarize our data and execute **WinBUGS** or **JAGS** from within **R** using the
 1864 packages **R2WinBUGS** (Sturtz et al., 2005), **R2jags** (Su and Yajima, 2011) or **rjags**
 1865 (Plummer, 2009). Kéry (2010), and Kéry and Schaub (2012) provide excellent
 1866 and accessible introductions to the basics of Bayesian analysis and GL(M)Ms using
 1867 **WinBUGS**. We don't want to be too redundant with those books and so we avoid
 1868 a detailed treatment of Bayesian methodology and software usage - instead just
 1869 providing a cursory overview so that we can move on and attack the problems
 1870 we're most interested in related to spatial capture-recapture. In addition, there are
 1871 a number of texts that provide general introductions to Bayesian analysis, MCMC,
 1872 and their applications in ecology including McCarthy (2007), Kéry (2010), Link
 1873 and Barker (2010), and King et al. (2008).

1874 While this chapter is about Bayesian analysis of GL(M)Ms, such models are
 1875 routinely analyzed using likelihood methods too. Later in this book (Chapt. 6), we
 1876 will use likelihood methods to analyze SCR models but, for now, we concentrate on
 1877 providing a basic introduction to Bayesian analysis because that is the approach
 1878 we will use in a majority of cases in later chapters.

3.1 GLMS AND GLMMS

1879 We have asserted already that SCR models work out most of the time to be variations
 1880 of GL(M)Ms. You might therefore ask: What are these GLM and GLMM
 1881 models, anyhow? These models are covered extensively in many very good applied
 1882 statistics books and we refer the reader elsewhere for a detailed introduction. The
 1883 classical references for GLMs are Nelder and Wedderburn (1972) and McCullagh
 1884 and Nelder (1989). In addition, we think Kéry (2010), Kéry and Schaub (2012),
 1885 and Zuur et al. (2009) are all accessible treatments. Here, we'll give the 1 minute
 1886 treatment of GL(M)Ms, not trying to be complete but rather only to preserve a
 1887 coherent organization to the book.

1888 The GLM is an extension of standard linear models allowing the response variable
 1889 to have some distribution from the exponential family of distributions. This
 1890 includes the normal distribution but also others such as the Poisson, binomial,
 1891 gamma, exponential, and many more. In addition, GLMs allow the response variable
 1892 to be related to the predictor variables (i.e., covariates) using a link function,
 1893 which is usually nonlinear. The GLM consists of three components:

- 1894 1. A probability distribution for the dependent (or response) variable y , from the
 1895 exponential family of probability distributions.
- 1896 2. A “linear predictor” $\eta = \beta_0 + x\beta_1$, where x is a predictor variable (i.e., a covariate).
- 1897 3. A link function g that relates the expected value of y , $\mathbb{E}(y)$, to the linear predictor,
 1898 $\mathbb{E}(y) = \mu = g^{-1}(\eta)$. Therefore $g(\mathbb{E}(y)) = \eta = \beta_0 + x\beta_1$.

1900 A key aspect of GLMs is that $g(\mathbb{E}(y))$ is assumed to be a linear function of the
 1901 predictor variable(s), here x , with unknown parameters, here β_0 and β_1 , to be
 1902 estimated. In standard GLMs, the variance of y is a function V of the mean of y :
 1903 $\text{Var}(y) = V(\mu)$ (see below for examples). As an example, a Poisson GLM posits
 1904 that $y \sim \text{Poisson}(\lambda)$ with $\mathbb{E}(y) = \lambda$ and usually the model for the mean is specified
 1905 using the *log link function* by

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

1906 The variance function is $V(y_i) = \lambda_i$. To see how a Poisson GLM works, use the **R**
 1907 code below to simulate some data and then estimate the parameters:

```
1908 > set.seed(13)
1909 > n <- 100          # set sample size
1910 > beta0 <- -2        # set intercept term
1911 > beta1 <- 1.5       # set coefficient
1912 > x <- rnorm(n, 0,1) # generate a predictor variable, x
1913
1914 > linpred <- beta0 + beta1*x # calculate linear predictor of E(y)
1915 > y <- rpois(n, exp(linpred)) # generate observations from model
```

1916 The **R** function `glm()` fits a GLM to the data we just generated and returns estimates of
 1917 β_0 and β_1 , which we see are fairly close to the data generating values above:

```
1918 > glm(y ~ 1 + x, family='poisson')      # the fit model
```

1919 This produces the output:

```
1920 Call: glm(formula = y ~ 1 + x, family = "poisson")
1921
1922 Coefficients:
1923 (Intercept)           x
1924     -2.007            1.446
1925
1926 [... some output deleted ...]
```

1927 In this summary output, the maximum likelihood estimates (MLEs) of the regression
 1928 parameters β_0 and β_1 are labeled “Coefficients.” We see that these are not too different
 1929 from the data-generating values (-2 and 1.5, respectively).

1930 The binomial GLM posits that $y_i \sim \text{Binomial}(K, p)$ where K is the fixed sample size
 1931 parameter and $\mathbb{E}(y_i) = K \times p_i$. Usually the model for the mean is specified using the *logit*
 1932 *link function* according to

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

1933 Where $\text{logit}(p) = \log(p/(1-p))$. The inverse-logit function, consequently, is $\text{logit}^{-1}(p) =$
 1934 $\exp(p)/(1 + \exp(p))$.

1935 A GLMM is the extension of GLMs to accommodate “random effects”. Often this
 1936 involves adding a normal random effect to the linear predictor. One simple example is
 1937 using a random intercept, α :

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

1938 where

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

1939 Many other probability distributions and formulations of the linear predictor might be
 1940 considered. GLMMs are enormously useful in ecological modeling applications for mod-
 1941 eling variation due to subjects, observers, spatial or temporal stratification, clustering,
 1942 and dependence that arises from any kind of group structure and, of course, because SCR
 1943 models prove to be a type of GLM with a random effect, but one that does not enter the
 1944 mean linearly.

3.2 BAYESIAN ANALYSIS

1945 Bayesian analysis is less familiar to many ecological researchers because they are often
 1946 educated only in the classical statistical paradigm of frequentist inference. But advances
 1947 in technology and increasing exposure to the benefits of Bayesian analysis are fast mak-
 1948 ing Bayesians out of people or at least making Bayesian analysis an acceptable, general
 1949 alternative to classical, frequentist inference.

1950 Conceptually, the main thing about Bayesian inference is that it uses probability
 1951 directly to characterize uncertainty about things we don't know. "Things", in this case,
 1952 are parameters of models and, just as it is natural to characterize uncertain outcomes of
 1953 stochastic processes using probability, it seems natural also to characterize information
 1954 about unknown parameters using probability. At least this seems natural to us and, we
 1955 think, most ecologists either explicitly adopt that view or tend to fall into that point
 1956 of view naturally. Conversely, frequentists use probability in many different ways, but
 1957 never to characterize uncertainty about parameters¹. Instead, frequentists use probability
 1958 to characterize the behavior of *procedures* such as estimators or confidence intervals (see
 1959 below). It is surprising that people readily adopt a philosophy of statistical inference in
 1960 which the things you don't know (i.e., parameters) should *not* be regarded as random
 1961 variables, so that, as a consequence, one cannot use probability to characterize one's state
 1962 of knowledge about them.

1963 3.2.1 Bayes' rule

1964 As its name suggests, Bayesian analysis makes use of Bayes' rule in order to make direct
 1965 probability statements about model parameters. Given two random variables z and y ,
 1966 Bayes' rule relates the two conditional probability distributions $[z|y]$ and $[y|z]$ by the
 1967 relationship:

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

1968 Bayes' rule itself is a mathematical fact and there is no debate in the statistical community
 1969 as to its validity and relevance to many problems. Generally speaking, these distributions
 1970 are characterized as follows: $[y|z]$ is the conditional probability distribution of y given z ,
 1971 $[z]$ is the marginal distribution of z and $[y]$ is the marginal distribution of y . In the context
 1972 of Bayesian inference we usually associate specific meanings in which $[y|z]$ is thought of
 1973 as "the likelihood", $[z]$ as the "prior" and so on. We leave this for later because here the
 1974 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

2010 all unknown quantities of a model as realizations of random variables – this includes data,
 2011 latent variables, and also parameters. Classical (non-Bayesian) analysts sometimes object
 2012 to regarding parameters as outcomes of random variables. Classically, parameters are
 2013 thought of as “fixed but unknown” (using the terminology of classical statistics). Indeed,
 2014 a common misunderstanding on the distinction between Bayesian and frequentist infer-
 2015 ence goes something like this “in frequentist inference parameters are fixed but unknown
 2016 but in a Bayesian analysis parameters are random.” At best this is a sad caricature of the
 2017 distinction and at worst it is downright wrong. In Bayesian analysis the parameters are
 2018 also unknown and, in fact, there is a single data-generating value of each parameter, and
 2019 so they are also fixed. The difference is that the fixed but unknown values are regarded
 2020 as having been generated from some probability distribution. Specification of that prob-
 2021 ability distribution is necessary to carry out Bayesian analysis, but it is not required in
 2022 classical frequentist inference.

2023 To see the general relevance of Bayes’ rule in the context of statistical inference, let y
 2024 denote observations - i.e., data - and let $[y|\theta]$ be the observation model (often colloquially
 2025 referred to as the “likelihood”). Suppose θ is a parameter of interest having (prior)
 2026 probability distribution $[\theta]$ (also simply referred to as the prior). These are combined to
 2027 obtain the posterior distribution using Bayes’ rule, which is:

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

2028 Asserting the general relevance of Bayes’ rule to all statistical problems, we can conclude
 2029 that the two main features of Bayesian inference are that: (1) parameters, θ , are regarded
 2030 as realizations of a random variable and, as a result, (2) inference is based on the prob-
 2031 ability distribution of the parameters given the data, $[\theta|y]$, which is called the posterior
 2032 distribution. This is the result of using Bayes’ rule to combine the “likelihood” and the
 2033 prior distribution. The key concept is regarding parameters as realizations of a random
 2034 variable because, once you admit this conceptual view, this leads directly to the posterior
 2035 distribution, a very natural quantity upon which to base inference about things we don’t
 2036 know - including parameters of statistical models. In particular, $[\theta|y]$ is a probability
 2037 distribution for θ and therefore we can make direct probability statements to characterize
 2038 uncertainty about θ .

2039 The denominator of our invocation of Bayes’ rule, $[y]$, is the marginal distribution of
 2040 the data y . We note without further remark right now that, in many practical problems,
 2041 this can be an enormous pain to compute. The main reason that the Bayesian paradigm
 2042 has become so popular in the last 20 years or so is because methods have been developed
 2043 for characterizing the posterior distribution that do not require that we possess a math-
 2044 ematical understanding of $[y]$. This means we never have to compute it or know what it
 2045 looks like, or know anything specific about it.

2046 While we can understand the conceptual basis of Bayesian inference merely by under-
 2047 standing Bayes’ rule – that’s really all there is to it – it is not so easy to understand the
 2048 basis of classical frequentist inference. What is mostly coherent in frequentist inference is
 2049 the manner in which procedures are evaluated – the performance of a given procedure is
 2050 evaluated by “averaging over” hypothetical realizations of y , regarding the *estimator* as a
 2051 random variable. For example, if $\hat{\theta}$ is an estimator of θ then the frequentist is interested
 2052 in $E_y(\hat{\theta}|y)$ which is used to characterize bias. If the expected value of $\hat{\theta}$, when averaged
 2053 over realizations of y , is equal to θ , then $\hat{\theta}$ is unbiased.

2054 The view of parameters as being random variables allows Bayesians to use probability
 2055 to make direct probability statements about parameters. Frequentist inference procedures
 2056 do not permit direct probability statements to be made about parameter values. Instead,
 2057 the view of parameters as fixed constants and estimators as random variables leads to
 2058 interpretations that are not so straightforward. For example confidence intervals having
 2059 the interpretation “95% probability that the interval contains the true value” and p-values
 2060 being “the probability of observing an outcome of the test statistic as extreme or more
 2061 than the one observed.” These are far from intuitive interpretations to most people.
 2062 Moreover, this is conceptually problematic to some because we will never get to observe
 2063 the hypothetical realizations that characterize the performance of our procedure.

2064 While we do tend to favor Bayesian inference for the conceptual simplicity (parameters
 2065 are random, posterior inference), we mostly advocate for a pragmatic non-partisan
 2066 approach to inference because, frankly, some of the frequentist methods are actually very
 2067 convenient in certain situations, and will generally yield very similar inferences about
 2068 parameters, as we will see in later chapters.

2069 3.2.3 Prior distributions

2070 The prior distribution $[\theta]$ is an important feature of Bayesian inference. As a conceptual
 2071 matter, the prior distribution characterizes “prior beliefs” or “prior information” about
 2072 a parameter. Indeed, an oft-touted benefit of Bayesian analysis is the ease with which
 2073 prior information can be included in an analysis. However, more commonly, the prior
 2074 is chosen to express a lack of prior information, even if previous studies have been done
 2075 and even if the investigator does in fact know quite a bit about a parameter. This is
 2076 because the manner in which prior information is embodied in a prior (and the amount
 2077 of information) is usually very subjective and thus the result can wind up being very
 2078 contentious; e.g., different investigators might report different results based on subjective
 2079 assessments of prior information. Thus it is usually better to “let the data speak” and
 2080 use priors that reflect absence of information beyond the data set being analyzed. An
 2081 example for an uninformative prior is a Uniform(0, 1) for a probability, or a Uniform($-\infty$,
 2082 ∞) (also called a “flat” or “improper” prior) for an unbounded continuous parameter.
 2083 Alternatively, people use “diffuse priors”; these contain some information, but (ideally)
 2084 not enough to exert meaningful influence on the posterior. An example for a diffuse prior
 2085 could be a normal distribution with a large standard deviation.

2086 But still the need occasionally arises to embody prior information or beliefs about a
 2087 parameter formally into the estimation scheme. In SCR models we often have a parameter
 2088 that is closely linked to “home range size” and thus auxiliary information on the home
 2089 range size of a species can be used as prior information, which may improve parameter
 2090 estimation (e.g., see Chandler and Royle (In press); also Chapt. 18).

2091 At times the situation arises where a prior can inadvertently impose substantial effect
 2092 on the posterior of a parameter, and that is not desirable. For example, we use data
 2093 augmentation to deal with the fact that the population size N is an unknown parameter
 2094 (Royle et al., 2007) which is equivalent to imposing a Binomial(M, ψ) prior on N for some
 2095 integer M (see Sec. 4.2). One has to take care to make sure that M is sufficiently large so
 2096 as to not affect the posterior distribution on N (see Fig. 17.6, and also Kéry and Schaub
 2097 (2012, Ch. 5)). Another situation that we have to be careful of is that prior distributions

2098 are *not* invariant to transformation of the parameter, and therefore neither are posterior
 2099 distributions (Link and Barker, 2010, Sec. 6.2.1). Thus, a prior that is ostensibly non-
 2100 informative on one scale, may be very informative on another scale. For example, if we
 2101 have a flat prior on $\text{logit}(p)$ for some probability parameter p , this is very different from
 2102 having a Uniform(0,1) prior on p . We show an example where this makes a difference in
 2103 Chapt. 5. Nonetheless, it is always possible to assess the influence of prior choice, and
 2104 it is often the case (with sufficient data and a structurally identifiable model) that the
 2105 influence of priors is negligible.

2106 **3.2.4 Posterior inference**

2107 In Bayesian inference, we are not focusing on estimating a single point or interval but
 2108 rather on characterizing a whole distribution – the posterior distribution – from which
 2109 one can report any summary of interest. A point estimate might be the posterior mean,
 2110 median, mode, etc.. In many applications in this book, we will compute 95% Bayesian
 2111 confidence intervals using the 2.5% and 97.5% quantiles of the posterior distribution. For
 2112 such intervals, it is correct to say $\Pr(L < \theta < U) = 0.95$. That is, “the probability that θ
 2113 lies between L and U is 0.95”.

2114 As an example, suppose we conducted a Bayesian analysis to estimate detection prob-
 2115 ability (p) of some species at a study site, and we obtained a posterior distribution of
 2116 beta(20,10) for the parameter p . The following R commands demonstrate how we make
 2117 inferences based upon summaries of the posterior distribution:

```
2118 > post.median <- qbeta(0.5, 20, 10)
2119 [1] 0.6704151
2120
2121 > post.95ci <- qbeta(c(0.025, 0.975), 20, 10)
2122 [1] 0.4916766 0.8206164
```

2123 Thus, we can state that there is a 95% probability that θ lies between 0.49 and 0.82. Fig.
 2124 3.1 shows the posterior along with the summary statistics. It is not a subtle thing that
 2125 such statements cannot be made using frequentist methods, although people tend to say
 2126 it anyway and not really understand why it is wrong or even that it is wrong.

2127 **3.2.5 Small sample inference**

2128 The posterior distribution is an exhaustive summary of the state-of-knowledge about an
 2129 unknown quantity. It is *the* posterior distribution - not an estimate of that thing. It is
 2130 also not, usually, an approximation except to within Monte Carlo error (in cases where
 2131 we use simulation to calculate it, see Sec. 3.5.2). One of the great virtues of Bayesian
 2132 analysis which is not widely appreciated is that posterior inference is not “asymptotic”,
 2133 which is to say, valid in a limiting sense as the sample size tends to infinity. Rather,
 2134 posterior inference is valid for *any* sample size and, in particular, *the* sample size on-hand.
 2135 Conversely, almost all frequentist procedures are based on asymptotic approximations to
 2136 the procedure which is being employed.

2137 There seems to be a prevailing view in statistical ecology that classical likelihood-based
 2138 procedures are virtuous because of the availability of simple formulas and procedures for

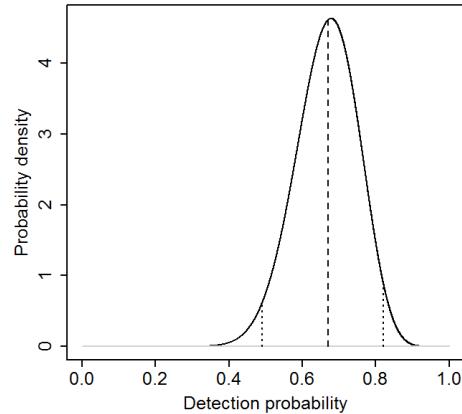


Figure 3.1. Probability density plot of a hypothetical posterior distribution of $\text{beta}(20,10)$; dashed lines indicate mean and upper and lower 95% interval

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

2161 may be all that is available for species that are extraordinarily important for conservation
 2162 and management. The Bayesian framework for inference provides a valid, rigorous, and
 2163 flexible framework that is theoretically justifiable in arbitrary sample sizes. This is not to
 2164 say that one will obtain precise estimates of density or other parameters, just that your
 2165 inference is coherent and justifiable from a conceptual and technical statistical point of
 2166 view. That is, for example when we estimate the density D of some animal population,
 2167 we report the posterior probability $\Pr(D|data)$ which is easily interpretable and just what
 2168 it is advertised to be and we don't need to do a simulation study to evaluate how well
 2169 the reported $\Pr(D|data)$ deviates from the "true" $\Pr(D|data)$ because they are the same
 2170 quantity.

3.3 CHARACTERIZING POSTERIOR DISTRIBUTIONS BY MCMC SIMULATION

2171 In practice, it is not really feasible to ever compute the marginal probability distribution
 2172 [y], the denominator resulting from application of Bayes' rule (Eq. 3.2.1). For decades
 2173 (even centuries!) this impeded the adoption of Bayesian methods by practitioners. Or,
 2174 the few Bayesian analyses done were based on asymptotic normal approximations to the
 2175 posterior distribution. While this was useful from a theoretical and technical standpoint
 2176 and, practically, it allowed people to make the probability statements that they naturally
 2177 would like to make, it was kind of a bad joke around the Bayesian water-cooler to, on
 2178 one hand, criticize classical statistics for being, essentially, completely ad hoc in their
 2179 approach to things but then, on the other hand, have to devise various approximations to
 2180 what they were trying to characterize. The advent of Markov chain Monte Carlo (MCMC)
 2181 methods has made it easier to calculate posterior distributions for just about any problem
 2182 to sufficient levels of precision.

2183 Broadly speaking, MCMC is a class of methods for drawing random samples (i.e.,
 2184 simulating from or just "sampling") from the target posterior distribution. Thus, even
 2185 though we might not recognize the posterior as a named distribution or be able to analyze
 2186 its features analytically, e.g., devise mathematical expressions for the mean and variance,
 2187 we can use these MCMC methods to obtain a large sample from the posterior and then
 2188 use that sample to characterize features of the posterior. What we do with the sample
 2189 depends on our intentions – typically we obtain the mean or median for use as a point
 2190 estimate, and take a confidence interval based on Monte Carlo estimates of the quantiles.

2191 3.3.1 What goes on under the MCMC hood

2192 We will develop and apply MCMC methods in some detail for spatial capture-recapture
 2193 models in Chapt. 17. Here we provide a simple illustration of some basic ideas related to
 2194 the practice of MCMC.

2195 A type of MCMC method relevant to most problems is Gibbs sampling (Geman and
 2196 Geman, 1984) which we address in more detail in Chapt. 17. Gibbs sampling involves iter-
 2197 ative simulation from the "full conditional" distributions (also called conditional posterior
 2198 distributions). The full conditional distribution for an unknown quantity is the conditional
 2199 distribution of that quantity given every other random variable in the model - the data
 2200 and all other parameters (see Sec. 3.3.2 for rules of how to construct full conditionals).

2201 For example, for a normal regression model ³ with $y \sim \text{Normal}(\beta_0 + \beta_1(x - \bar{x}), \sigma^2)$ where
 2202 lets say σ^2 is known, the full conditionals are, using “bracket notation”,

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

2203 and

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

2204 We might use our knowledge of probability to identify these mathematically. In particular,
 2205 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
 2206 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
 2207 some algebra reveals that

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

2208 where $\tau = 1/\sigma^2$ and $\tau_{\beta_0} = 1/\sigma_{\beta_0}^2$ (the inverse of the variance is sometimes called *precision*),
 2209 and $w = \tau n / (\tau n + \tau_{\beta_0})$. We see in this case that the posterior mean is a *precision-weighted*
 2210 sum of the sample mean \bar{y} and the prior mean μ_{β_0} , and the posterior *precision* is the
 2211 sum of the precision of the likelihood and that of the prior. These results are typical of
 2212 many classes of problems. In particular, note that as the prior precision tends to 0, i.e.,
 2213 $\tau_{\beta_0} \rightarrow 0$, then the posterior of β_0 tends to $\text{Normal}(\bar{y}, \sigma^2/n)$. We recognize the variance of
 2214 this distribution as that of the variance of the sampling distribution of \bar{y} and its mean is
 2215 in fact the MLE of β_0 for this model. The conditional posterior of β_1 has a very similar
 2216 form:

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

2217 which might look slightly unfamiliar, but note that if $\tau_{\beta_1} = 0$, then the mean of this
 2218 distribution is the familiar $\hat{\beta}_1$, and the variance is, in fact, the sampling variance of $\hat{\beta}_1$.
 2219 The MCMC algorithm for this model has us simulate in succession, repeatedly, from
 2220 those two distributions. See Gelman et al. (2004) for more examples of Gibbs sampling
 2221 for the normal model, and we also provide another example in Chapt. 17. A conceptual
 2222 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
}

```

2224 As we just saw for this simple “normal-normal” model, it is sometimes possible to
 2225 specify the full conditional distributions analytically. In general, when certain so-called
 2226 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

2227 sense, “matches” the likelihood, then the form of the full conditional distributions is also
 2228 similar to that of the observation model. In this normal-normal case, the normal distribu-
 2229 tion for the mean parameters is the conjugate prior for the normal observation model, and
 2230 thus the full-conditional distributions are also normal. This is convenient because, in such
 2231 cases, we can simulate directly from them using standard methods (or **R** functions). But,
 2232 in practice, we don’t really ever need to know such things because most of the time we
 2233 can get by using a simple algorithm, called the Metropolis-Hastings (henceforth “MH”)
 2234 algorithm, to obtain samples from these full conditional distributions without having to
 2235 recognize them as specific, named, distributions. This gives us enormous freedom in devel-
 2236 oping models and analyzing them without having to resolve them mathematically because
 2237 to implement the MH algorithm we need only identify the full conditional distribution up
 2238 to a constant of proportionality, that being the marginal distribution in the denominator
 2239 (e.g., $[y|\beta_1]$ above).

2240 We will talk about the Metropolis-Hastings algorithm shortly, and we will use it ex-
 2241 tensively in the analysis of SCR models (e.g., Chapt. 17).

2242 **3.3.2 Rules for constructing full conditional distributions**

2243 The basic strategy for constructing full-conditional distributions for devising MCMC al-
 2244 gorithms can be reduced conceptually to a couple of basic steps summarized as follows:

- 2245 **(step 1)** Identify all stochastic components of the model and collect their probability
 distributions;
- 2246 **(step 2)** Express the full conditional in question as proportional to the product of all
 probability distributions identified in step 1;
- 2247 **(step 3)** Remove the ones that don’t have the focal parameter in them.
- 2248 **(step 4)** Do some algebra on the result in order to identify the resulting probability
 distribution function (pdf) or mass function (pmf).

2252 Of the 4 steps, the last of those is the main step that requires quite a bit of statistical
 2253 experience and intuition because various algebraic tricks can be used to reshape the mess
 2254 into something recognizable – i.e., a standard, named distribution. But step 4 is not
 2255 necessary if we decide instead to use the Metropolis-Hastings algorithm as described below.

2256 In the context of our simple linear regression model that we’ve been working with,
 2257 to characterize $[\beta_0|y, \beta_1]$ we first apply step 1 and identify the model components as:
 2258 $[y|\beta_0, \beta_1]$, with prior distributions $[\beta_0]$ and $[\beta_1]$. Step 2 has us write $[\beta_0|y, \beta_1] \propto [y|\beta_0, \beta_1][\beta_0][\beta_1]$.
 2259 Step 3: We note that $[\beta_1]$ is not a function of β_0 and therefore we remove it to obtain
 2260 $[\beta_0|y, \beta_1] \propto [y|\beta_0, \beta_1][\beta_0]$. Similarly, applying step 2 and 3 for β_1 we obtain $[\beta_1|y, \beta_0] \propto$
 2261 $[y|\beta_0, \beta_1][\beta_1]$. We apply step 4 and manipulate these algebraically to arrive at the re-
 2262 sult (which we provided in Eqs. 3.3.1 and 3.3.2) or, alternatively, we can sample them
 2263 indirectly using the Metropolis-Hastings algorithm, which we discuss now.

2264 **3.3.3 Metropolis-Hastings algorithm**

2265 The Metropolis-Hastings (MH) algorithm is a completely generic method for sampling
 2266 from any distribution, say $[\theta]$. In our applications, $[\theta]$ will typically be the full conditional
 2267 distribution of θ . While we sometimes use Gibbs sampling, we seldom use “pure” Gibbs

2268 sampling because full conditionals do not always take the form of known distributions we
 2269 can sample from directly. In such cases, we use MH to sample from the full conditional
 2270 distributions. When the MH algorithm is used to sample from full conditional distributions
 2271 of a Gibbs sampler the resulting hybrid algorithm is called *Metropolis-within-Gibbs*. In
 2272 Sec. 3.6.3 we will construct such an algorithm for a simple class of models. We discuss
 2273 both the Gibbs and the MH algorithm, as well as their hybrid in more depth in Chapt.
 2274 17.

2275 The MH algorithm generates candidate values for the parameter(s) we want to estimate
 2276 from some proposal or candidate-generating distribution that may be conditional on the
 2277 current value of the parameter, denoted by $h(\theta^*|\theta^{t-1})$. Here, θ^* is the *candidate* or
 2278 proposed value and θ^{t-1} is the value of θ at the previous time step, i.e., at iteration $t - 1$
 2279 of the MCMC algorithm. The proposed value is accepted with probability

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

2280 which is called the MH acceptance probability. This ratio can sometimes be > 1 in which
 2281 case we set it equal to 1. It is useful to note that $h()$ can be any probability distribution.

2282 In the context of using the MH algorithm to do MCMC (in which case the target
 2283 distribution is a full-conditional or posterior distribution), an important fact is, no matter
 2284 the choice of $h()$, we can compute the MH acceptance probability directly because the
 2285 marginal distribution of y cancels from both the numerator and denominator of r . This
 2286 is the magic of the MH algorithm.

3.4 BAYESIAN ANALYSIS USING THE BUGS LANGUAGE

2287 We won't be too concerned with devising our own MCMC algorithms for every analysis,
 2288 although we will do that a few times for fun. More often, we will rely on the freely available
 2289 software package **WinBUGS** or **JAGS** for doing this. We will always execute these
 2290 **BUGS** engines from within **R** using the **R2WinBUGS** (Sturtz et al., 2005) or, for **JAGS**,
 2291 the **R2jags** (Su and Yajima, 2011) or **rjags** (Plummer, 2009) packages. **WinBUGS** and
 2292 **JAGS** are MCMC black boxes that take a pseudo-code description (i.e., written in the
 2293 **BUGS** language) of all of the relevant stochastic and deterministic elements of a model
 2294 and generate an MCMC algorithm for that model. But you never get to see the algorithm.
 2295 Instead, **WinBUGS/JAGS** will run the algorithm and return the Markov chain output
 2296 - the posterior samples of model parameters.

2297 The great thing about using the **BUGS** language is that it forces you to become
 2298 intimate with your statistical model - you have to write each element of the model down,
 2299 admit (explicitly) all of the various assumptions, understand what the actual probability
 2300 assumptions are and how data relate to latent variables and data and latent variables
 2301 relate to parameters, and how parameters relate to one another.

2302 While we normally use **WinBUGS**, we note that **OpenBUGS** is the current active
 2303 development tree of the **BUGS** project. See Kéry (2010) and Kéry and Schaub (2012,
 2304 especially Appendix 1) for more on practical analysis in **WinBUGS**. Those books should
 2305 be consulted for a more comprehensive introduction to using **WinBUGS**. Recently we
 2306 have migrated many of our analyses to **JAGS** (Plummer, 2009), which we adopt later in

2307 the book. You can refer to Hobbs (2011) for an ecological introduction to **JAGS**. Next,
 2308 we provide an example of a Bayesian analysis using **WinBUGS**.

2309 **3.4.1 Linear regression in WinBUGS**

2310 We provide a brief introductory example of a normal regression model using a small
 2311 simulated data set. The following commands are executed from within your **R** workspace.
 2312 First, simulate a covariate x and observations y having prescribed intercept, slope and
 2313 variance:

```
2314 > x <- rnorm(10)
2315 > mu <- -3.2 + 1.5*x
2316 > y <- rnorm(10, mu, sd=4)
```

2317 The **BUGS** model specification for a normal regression model is written within **R** as
 2318 a character string input to the command `cat()` and then dumped to a text file named
 2319 `normal.txt`:

```
2320 > cat("
2321   model{
2322     for (i in 1:10){
2323       y[i] ~ dnorm(mu[i],tau)      # the likelihood
2324       mu[i] <- beta0 + beta1*x[i]  # the linear predictor
2325     }
2326     beta0 ~ dnorm(0,.01)         # prior distributions
2327     beta1 ~ dnorm(0,.01)
2328     sigma ~ dunif(0,100)
2329     tau <- 1/(sigma*sigma)      # tau is the precision
2330   }                                # and a derived parameter
2331 ",file="normal.txt")
```

2332 Alternatively, you can write the model specifications directly within a text file and save it
 2333 in your current working directory, but we do not usually take that approach in this book.

2334 The **BUGS** dialects⁴ parameterize the normal distribution in terms of the mean and
 2335 inverse-variance, called the precision. Thus, `dnorm(0,.01)` implies a variance of 100.
 2336 We typically use diffuse normal priors for mean parameters, β_0 and β_1 in this case, but
 2337 sometimes we might use uniform priors with suitable bounds $-B$ and $+B$. Also, we
 2338 typically use a Uniform($0, B$) prior on standard deviation parameters (Gelman, 2006).
 2339 But sometimes we might use a gamma prior on the precision parameter τ . In a **BUGS**
 2340 model file, every variable referenced in the model description has to be either data, which
 2341 will be input (see below), a random variable which must have a probability distribution
 2342 associated with it using the tilde character “~” (a.k.a. “twiddle”) or it has to be a derived
 2343 parameter connected to variables and data using an assignment arrow: “<-”.

2344 To fit the model, we need to describe various data objects to **WinBUGS**. In particular,
 2345 we create an **R** list object called `data` which are the data objects identified in the **BUGS**
 2346 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**

2347 **R** workspace and also in the **WinBUGS** model definition. We also create an **R** function
 2348 that produces a list of starting values, **inits**, that get sent to **WinBUGS**. In general,
 2349 starting values are optional. We recommend to always provide reasonable starting values
 2350 where possible, both for structural parameters and also random effects⁵. Finally, we
 2351 identify the names of the parameters (labeled correspondingly in the **WinBUGS** model
 2352 specification) that we want **WinBUGS** to save the MCMC output for. In this example,
 2353 we will “monitor” the parameters β_0 , β_1 , σ and τ . **WinBUGS** is executed using the
 2354 **R** command **bugs()**. We set the option **debug=TRUE** if we want the **WinBUGS** GUI to
 2355 stay open (useful for analyzing MCMC output and looking at the **WinBUGS** error log).
 2356 Also, we set **working.dir=getwd()** so that **WinBUGS** output files and the log file are
 2357 saved in the current **R** working directory (note that sometimes you will need to specify the
 2358 place where you installed **WinBUGS** within the **bugs()** call, using the **bugs.directory**
 2359 argument). All of these activities together look like this:

```
2360 > library(R2WinBUGS)      # "load" the R2WinBUGS package
2361 > data <- list( y=y, x=x)
2362 > inits <- function()
2363 > list ( beta1=rnorm(1),beta0=rnorm(1),sigma=runif(1,0,2) )
2364 > parameters <- c("beta0","beta1","sigma","tau")
2365 > out <- bugs(data, inits, parameters, "normal.txt", n.thin=1, n.chains=2,
2366   n.burnin=2000, n.iter=6000, debug=TRUE,working.dir=getwd())
```

2367 Note that the previously created objects defining data, initial values and parameters to
 2368 monitor are passed to the function **bugs()**. In addition, various other things are declared:
 2369 The number of parallel Markov chains (**n.chains**), the thinning rate (**n.thin**), the number
 2370 of burn-in iterations (**n.burnin**) and the total number of iterations (**n.iter**). To develop
 2371 a detailed understanding of the various parameters and settings used for MCMC, consult
 2372 a basic reference such as Kéry (2010). We also come back to these issues in the following
 2373 section (3.5) and in Chapt. 17. A common question is “how should my data be formatted?”
 2374 That depends on how you describe the model in the **BUGS** language, and how your data
 2375 are input into **R**. There is no unique way to describe any particular model and so you have
 2376 some flexibility. We talk about data format further in the context of capture-recapture
 2377 models and SCR models in Chapt. 5 and elsewhere.

2378 You should execute all of the commands given above and then close the **WinBUGS**
 2379 GUI, and the data will be read back into **R** (or specify **debug=FALSE** in the **bugs()** call).
 2380 We don’t want to give instructions on how to navigate and use the GUI – but you can
 2381 fire up **WinBUGS** and read the help files, or see Chapt. 4 from Kéry (2010) for a brief
 2382 introduction. The **print** command applied to the object **out** prints some basic summary
 2383 output (this is slightly edited):

```
2384 > print(out,digits=2)
2385 Inference for Bugs model at "normal.txt", fit using WinBUGS,
2386 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**

```

2387 n.sims = 8000 iterations saved
2388      mean   sd 2.5% 25% 50% 75% 97.5% Rhat n.eff
2389 beta0    -6.62 1.64 -9.77 -7.63 -6.64 -5.63 -3.29     1  4200
2390 beta1     0.81 1.20 -1.63  0.09  0.80  1.54  3.24     1  5100
2391 sigma     4.99 1.56  2.93  3.92  4.66  5.70  8.85     1  8000
2392 tau       0.05 0.03  0.01  0.03  0.05  0.07  0.12     1  8000
2393 deviance 58.72 3.21 55.06 56.35 57.85 60.26 67.15     1  6200
2394
2395 For each parameter, n.eff is a crude measure of effective sample size,
2396 and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
2397
2398 DIC info (using the rule, pD = Dbar-Dhat)
2399 pD = 2.5 and DIC = 61.3

```

2400 In the **WinBUGS** output you see a column called “Rhat”, as well as one called
2401 “n.eff”. These are convergence diagnostics (the \hat{R} or Brooks-Gelman-Rubin statistic
2402 and the effective sample size) and we will discuss those in the following section, 3.5.2.
2403 DIC is the deviance information criterion (Spiegelhalter et al. (2002), see section 3.9)
2404 which some people use in a manner similar to AIC although it is recognized to have some
2405 problems in hierarchical models (Millar, 2009). We consider use of DIC in the context of
2406 SCR models in Chapt. 8.

3.5 PRACTICAL BAYESIAN ANALYSIS AND MCMC

2407 The mere execution of a Bayesian analysis using the **BUGS** language, as demonstrated
2408 with the linear regression example, is fairly straight forward. There are, however, a number
2409 of really important practical issues to be considered in any Bayesian analysis and we cover
2410 some of these briefly here before we move on to implementing slightly more complex
2411 GL(M)Ms in a Bayesian framework.

2412 3.5.1 Choice of prior distributions

2413 Bayesian analysis requires that we choose prior distributions for all of the structural pa-
2414 rameters of the model (we use the term structural parameter to mean all parameters that
2415 aren’t customary thought of as latent variables). We will strive to use priors that are
2416 meant to express little or no prior information - default or customary “non-informative”
2417 or diffuse priors. This will be $\text{Uniform}(a, b)$ priors for parameters that have a natural
2418 bounded support and, for parameters that live on the real line we use either (1) diffuse
2419 normal priors, as we did in the linear regression example above; (2) improper uniform
2420 priors which have unbounded support, e.g., $[\theta] \propto 1$, or (3) sometimes even a bounded
2421 $\text{Uniform}(a, b)$ prior, if that greatly improves the performance of **WinBUGS** or other
2422 software doing the MCMC for us. In **WinBUGS** a prior with low precision, τ , where
2423 $\tau = 1/\sigma^2$, such as $\text{Normal}(0, .01)$ will typically be used. Of course $\tau = 0.01$ ($\sigma^2 = 100$)
2424 might be very informative for a regression parameter depending on its magnitude and
2425 scaling of x . Therefore, we recommend that predictor variables (covariates) *always* be
2426 standardized to have mean 0 and variance 1.

2427 **Lack of invariance of priors to transformation.** Clearly there are a lot of choices
 2428 for ostensibly non-informative priors, and the degree of non-informativeness depends on
 2429 the parameterization. For example, a natural non-informative prior for the intercept of a
 2430 logistic regression

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

2431 would be a very diffuse normal prior, $[\beta_0] = \text{Normal}(0, \text{Large})$ or even $\beta_0 \sim \text{Uniform}(-\text{Large}, \text{Large})$.
 2432 However, we might also use a prior on the parameter $p_0 = \text{logit}^{-1}(\beta_0)$, which is $\Pr(y=1)$
 2433 for the value $x=0$. Since p_0 is a probability a natural choice is $p_0 \sim \text{Uniform}(0, 1)$. These
 2434 priors are very different in their implications. For example, if we choose the normal prior
 2435 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

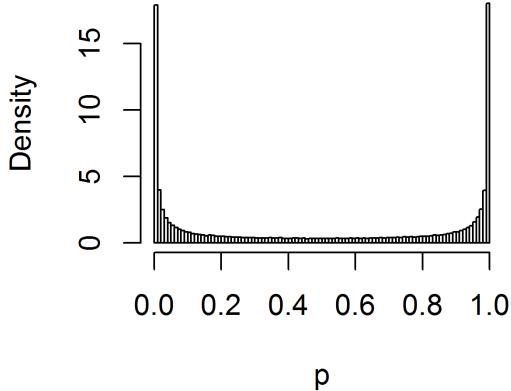


Figure 3.2. Implied prior for $p_0 = \exp(\beta_0)/(1 + \exp(\beta_0))$ if $\beta_0 \sim \text{Normal}(0, 5^2)$.

2436 affect results (see Sec. 4.4.2 for an illustration of this for a real data set), yet they are
 2437 both sensible non-informative priors. Despite this, it is often the case that priors will have
 2438 little or no impact on the results. Choice of priors and parameterization is very much
 2439 problem-specific and often largely subjective. Moreover, it also affects the behavior of
 2440 MCMC algorithms and therefore the analyst needs to pay some attention to this issue
 2441 and possibly try different things out. Most standard Bayesian analysis books address
 2442 issues related to specification and effect of prior distribution choice in some depth. Some
 2443 good references include Kass and Wasserman (1996), Gelman (2006) and Link and Barker
 2444 (2010).

2446 **3.5.2 Convergence and so-forth**

2447 Once we have carried out an analysis by MCMC, there are many other practical issues
 2448 that we have to confront. One characteristic of MCMC sampling is that Markov chains
 2449 take some time to converge to their stationary distribution - in our case the posterior
 2450 distribution for some parameter given data, $[\theta|y]$. Only when the Markov chain has
 2451 reached its stationary distribution, the generated samples can be used to characterize the
 2452 posterior distribution. Thus, one of the most important issues we need to address is “have
 2453 the chains converged?” Since we do not know what the stationary posterior distribution
 2454 of our Markov chain should look like (this is the whole point of doing an MCMC analysis),
 2455 we effectively have no means to assess whether or not it has truly converged to this desired
 2456 distribution. Most MCMC algorithms only guarantee that, eventually, the samples being
 2457 generated will be from the target posterior distribution, but no-one can tell us how long
 2458 this will take. Also, you only know the part of your posterior distribution that the Markov
 2459 chain has explored so far – for all you know the chain could be stuck in a local maximum,
 2460 while other maxima remain completely undiscovered. Acknowledging that there is truly
 2461 nothing we can do to ever prove convergence of our MCMC chains, there are several things
 2462 we can do to increase the degree of confidence we have about the convergence of our chains.
 2463 Some problems are easily detected using simple plots, such as a time-series plot, where
 2464 parameter values of each MCMC iteration are plotted against the number of iterations.
 2465 Fig. 3.3 shows the time series plots for the three parameters – β_0 , β_1 and σ – from our
 2466 linear regression example, taken from the **WinBUGS** GUI before closing it to return to
 2467 **R**.

2468 Typically a period of transience is observed in the early part of the MCMC algorithm,
 2469 and this is usually discarded as the “burn-in” period. In our linear regression example,
 2470 within the `bugs()` call we set the burn-in period as 2000 iterations so these are auto-
 2471 matically removed by **WinBUGS** and are not part of the output (but Fig. 3.6 shows a
 2472 time-series plot that starts at iteration 0 with a clearly visible burn-in period). The quick
 2473 diagnostic to whether convergence has been achieved is that your Markov chains look
 2474 “grassy” – this seems a reasonable statement for the plots in Fig. 3.3. Another way to
 2475 check convergence is to update the parameters some more and see if the posterior changes.
 2476 If the chains have converged to the posterior, the posterior mean, confidence intervals, and
 2477 other summaries should be relatively static as we continue to run the algorithm. Yet an-
 2478 other option, and one generally implemented in **WinBUGS**, is to run several Markov
 2479 chains and to start them off at different initial values that are over-dispersed relative to
 2480 the posterior distribution. Such initial values help to explore different areas of the param-
 2481 eter space simultaneously; if, after a while, all chains oscillate around the same average
 2482 value, chances are good that they indeed converged to the posterior distribution. Gelman
 2483 and Rubin came up with the so-called “R-hat” statistic (\hat{R}) or Brooks-Gelman-Rubin
 2484 statistic that essentially compares within-chain and between-chain variance to check for
 2485 convergence of multiple chains (Gelman et al., 1996). The R-hat statistic should be close
 2486 to 1 if the Markov chains have converged and sufficient posterior samples have been ob-
 2487 tained. For the linear regression example, we ran two parallel chains (also specified in the
 2488 `bugs()` call) and **WinBUGS** returns the \hat{R} statistic for us as part of the summary model
 2489 output. If you look back to Sec. 3.4.1 you see that $\hat{R} = 1$ for all parameters of the linear
 2490 model. In practice, $\hat{R} \leq 1.2$ may be good enough for some problems. For some models you
 2491 can’t actually realize a low \hat{R} . E.g., if the posterior is a discrete mixture of distributions

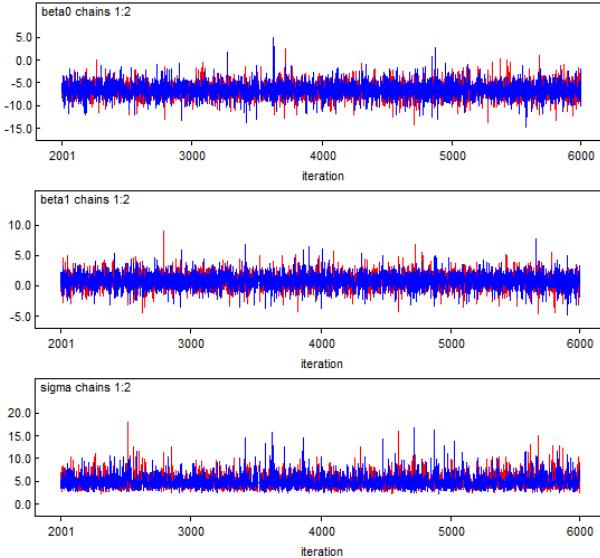


Figure 3.3. Time-series plots for parameters from a linear regression run in **WinBUGS** using two parallel Markov chains.

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

2512 (MacEachern and Berliner, 1994; Link and Eaton, 2011). Practical considerations might
 2513 necessitate thinning, even though it is statistically inefficient. For example, in models
 2514 with many parameters or other unknowns being tabulated, the output files might be
 2515 enormous and unwieldy to work with. In such cases, thinning is perfectly reasonable. In
 2516 many cases, how well the Markov chains mix is strongly influenced by parameterization,
 2517 standardization of covariates, and the prior distributions being used. Some things work
 2518 better than others, and the investigator should experiment with different settings and
 2519 remain calm when things don't work out perfectly.

2520 **Is the posterior sample large enough?** The subsequent samples generated from
 2521 a Markov chain are not *independent* samples from the posterior distribution, due to the
 2522 correlation among samples introduced by the Markov process⁶ and the sample size has
 2523 to be adjusted to account for the autocorrelation in subsequent samples (see Chapt. 8 in
 2524 Robert and Casella (2010) for more details). This adjusted sample size is referred to as the
 2525 effective sample size. Checking the degree of autocorrelation in your Markov chains and
 2526 estimating the effective sample size your chain has generated should be part of evaluating
 2527 your model output. **WinBUGS** will automatically return the effective sample size for
 2528 all monitored parameters, as we saw in our linear regression example (the “n.eff” column
 2529 of the summary output). If you find that your supposedly long Markov chain has only
 2530 generated a very short effective sample, you should consider a longer run. What exactly
 2531 constitutes a reasonable effective sample size is hard to say. A more palpable measure
 2532 of whether you've run your chain for enough iterations is the time-series or Monte Carlo
 2533 error - the “noise” introduced into your samples by the stochastic MCMC process. The
 2534 MC error is printed by default in summaries produced in the **WinBUGS** GUI, which
 2535 can be reproduced in **R** using `bugs.log('log.txt')$stats` (note that “log.txt” refers
 2536 to a model log file that **WinBUGS** automatically creates in the working directory; it is
 2537 overwritten with every new model you run unless you save it under a different name).

```
2538 > bugs.log('log.txt')$stats
2539 $stats
2540      mean      sd   MCerror    2.5%   median   97.5% start sample
2541 beta0    -6.64700 1.60300 0.0179400 -9.7140 -6.70800 -3.2730  2001  8000
2542 beta1     0.82100 1.19000 0.0116800 -1.4900  0.82560  3.1800  2001  8000
2543 deviance 58.66000 3.08800 0.0506800 55.0700 57.93000 66.8400  2001  8000
2544 sigma     4.96800 1.52300 0.0248300  2.9350  4.68100  8.7410  2001  8000
2545 tau       0.05074 0.02677 0.0003651  0.0131  0.04564  0.1162  2001  8000
```

2546 When using **JAGS** the `summary` command will automatically produce the MC error
 2547 (which is called “Time-series SE” in **JAGS**). You want the MC error to be smallish relative
 2548 to the magnitude of the parameter and what smallish means will depend on the purpose
 2549 of the analysis. For a preliminary analysis you might settle for a few percent whereas
 2550 for a final analysis then certainly less than 1% is called for. You can run your MCMC
 2551 algorithm as long as it takes to achieve that. A consequence of the MC error is that even
 2552 for the exact same model, results will usually be slightly different. Thus, as a good rule of
 2553 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, $\theta^{(t-1)}$.

2554 **3.5.3 Bayesian confidence intervals**

2555 The 95% Bayesian confidence interval based on percentiles of the posterior is not a unique
 2556 interval - there are many of them. The so-called “highest posterior density” (HPD) inter-
 2557 val is an alternative, defined as the narrowest interval that contains *at least* 95% of the
 2558 posterior mass. As a result (of the *at least* clause), for discrete parameters, the 95% HPD
 2559 is not often exactly 95% but usually slightly more conservative than nominal.

2560 **3.5.4 Estimating functions of parameters**

2561 A benefit of analysis by MCMC is that we can seamlessly estimate functions of parameters
 2562 by simply tabulating the desired function of the simulated posterior draws. For example,
 2563 if θ is the parameter of interest and let $\theta^{(i)}$ for $i = 1, 2, \dots, M$ be the posterior samples
 2564 of θ . Let $\eta = \exp(\theta)$, then a posterior sample of η can be obtained simply by computing
 2565 $\exp(\theta^{(i)})$ for $i = 1, 2, \dots, M$. Almost all SCR models in this book involve at least 1 derived
 2566 parameter. For example, density D is a derived parameter, being a function of population
 2567 size N and the area A of the underlying state-space of the point process (see Chapt. 5).

2568 **Example: Finding the optimum value of a covariate.** As another example of
 2569 estimating functions of model parameters, suppose that the normal regression model from
 2570 Sec. 3.4.1 had a quadratic response function of the form

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

2571 Then the optimum value of x , i.e., that corresponding to the optimal expected response,
 2572 can be found by setting the derivative of this function to 0 and solving for x . We find that

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

2573 yields that $x_{opt} = -\beta_1/(2 * \beta_2)$. We can just take our posterior draws for β_1 and β_2
 2574 and obtain a posterior sample of x_{opt} by this simple calculation applied to the posterior
 2575 output. As an exercise, take the normal model above and simulate a quadratic response
 2576 and then describe the posterior distribution of x_{opt} .

3.6 POISSON GLMS

2577 The Poisson GLM (also known as “Poisson regression”) is probably the most relevant
 2578 and important class of models in all of ecology. The basic model assumes observations
 2579 $y_i; i = 1, 2, \dots, n$ follow a Poisson distribution with mean λ which we write

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

2580 Commonly y_i is a count of animals or plants at some point in space (“site”) i , and λ
 2581 might vary over sites as well. For example, i might index point count locations in a
 2582 forest, survey route centers, or sample quadrats, or similar, and we are interested in how
 2583 λ depends on site characteristics such as habitat. If covariates are available it is typical to
 2584 model them as linear effects on the log mean. If x_i is some measured covariate associated
 2585 with observation i , then,

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

2586 While we only specify the mean of the Poisson model directly, the Poisson model (and
 2587 all GLMs) has a “built-in” variance which is directly related to the mean. In this case,
 2588 $\text{Var}(y) = \mathbb{E}(y) = \lambda$. Thus the model accommodates a linear increase in variance with the
 2589 mean.

2590 **3.6.1 Example: Breeding Bird Survey data**

2591 As an example we consider a classical situation in ecology where counts of an organism
 2592 are made at a collection of spatial locations. In this particular example, we have
 2593 mourning dove (*Zenaida macroura*) counts made along North American Breeding Bird
 2594 Survey (BBS) routes in Pennsylvania, USA. A route consists of 50 stops separated by
 2595 0.5 miles. For the purposes here we are defining y_i = route total count and the sample
 2596 location will be marked by the center point of the BBS route. The survey is run annually
 2597 and the data set we analyze is 1966-1998. BBS data can be obtained online at
 2598 <http://www.pwrc.usgs.gov/bbs/>, but the particular chunk of data we will be using here
 2599 is also included in the **scrbook** package (**data(bbsdata)**). We will make use of the whole
 2600 data set shortly but for now we’re going to focus on a specific year of counts (1990) for
 2601 the sake of building a simple model. In 1990 there were 77 active routes; this data set
 2602 contains rows which index the unique route, column 1 is the route ID, columns 2-3 are
 2603 the route coordinates (longitude/latitude), column 4 is a habitat covariate “forest cover”
 2604 (standardized, see below) and the remaining columns are the yearly counts. Years for
 2605 which a survey was not conducted on a route are coded as “NA” in the data matrix. We
 2606 imagine that this will be a typical format for many ecological studies, perhaps with more
 2607 columns representing covariates. To read in the data and display the first few elements of
 2608 the data frame containing the counts, do this:

```
2609 > data(bbsdata)           #  loads data frame 'bbs'  

2610 > bbsdata$counts[1:2,1:6]  

2611  

2612      X     lon     lat   habitat X66 X67  

2613 1 72002 -80.445 41.501 -0.3871372 NA 24  

2614 2 72003 -80.347 41.214 -1.0171629 NA NA
```

2615 It is useful to display the spatial pattern in the observed counts. For that we use a
 2616 spatial dot plot – where we plot the coordinates of the observations and mark the color
 2617 of the plotting symbol based on the magnitude of the count. We have a special plotting
 2618 function for that which is called **spatial.plot()** and it is available with the supplemental
 2619 **R** package **scrbook**. Actually, what we want to do here is plot the log-counts (+1 of
 2620 course) which (Fig. 3.4) display a notable pattern that could be related to something.
 2621 The **R** commands for obtaining this figure are:

```
2622 > library(scrbook)  

2623 > data(bbsdata)  

2624 > library(maps)  

2625  

2626 > y <- bbsdata$counts[, "X90"] # Pick year 1990  

2627 > notna <- !is.na(y)
```

```

2628 > y <- y[notna]
2629 > locs <- bbsdata$counts[notna,c("lon","lat")]
2630 > sz <- y/max(y)
2631
2632 > par(mar=c(3,3,3,6))
2633 > plot(locs,pch=" ",axes=FALSE,xlim=range(locs[,1])+c(-.3,+.3),
2634   ylim=c(range(locs[,2]) + c(-.6,.6)), xlab=" ",ylab=" ")
2635 > map('state', regions='pennsylvania', add=TRUE, lwd=2)
2636 > spatial.plot(bbsdata$counts[notna,2:3], y, cx=1+sz*6, add=TRUE)

```

2637 We can ponder the potential effects that might lead to dove counts being high - corn
 2638 fields, telephone wires, barn roofs along with misidentification of pigeons, these could all
 2639 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-

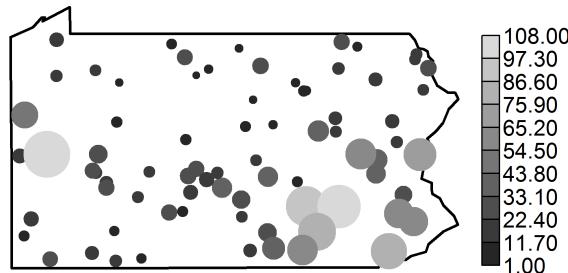


Figure 3.4. Mourning dove counts along North American Breeding Bird Survey routes in Pennsylvania (year = 1990). Plot symbol shading and circle size is proportional to raw count.

```

2640 vided in the data frame bbsdata$habitat) which can be plotted using the spatial.plot
2641 function with the following R commands
2642

```

```

2643 > habdata <- bbsdata$habitat
2644 > map('state',regions="penn",lwd=2)
2645 > I <- matrix(NA, nrow=30, ncol=40)
2646 > I <- matrix(habdata[,"dfor"], ncol=40, byrow=FALSE)
2647 > ux <- unique(habdata[,2])

```

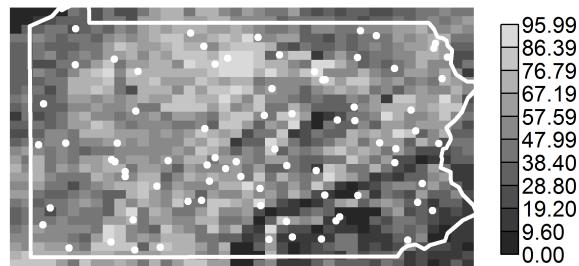


Figure 3.5. Forest cover (percent deciduous) in Pennsylvania. BBS route locations are shown by white dots.

```

2648 > uy <- sort(unique(habdata[,3]))
2649
2650 > par(mar=c(3,3,3,6))
2651 > plot(locs,pch=" ", axes=FALSE, xlim=range(locs[,1])+c(-.3,+.3),
2652   ylim=c(range(locs[,2]) + c(-.6,.6)), xlab=" ",ylab=" ")
2653 > image(ux,uy,rot(I), add=TRUE, col=gray(seq(3,17,,10)/20) )
2654 > map('state', regions='pennsylvania', add=TRUE, lwd=3, col="white")
2655 > image.scale(I, col=gray(seq(3,17,,10)/20) )
2656 > points(locs,pch=20, col="white")

```

2657 The result appears in Fig. 3.5. We see a prominent pattern that indicates high forest
2658 coverage in the central part of the state and low forest cover in the SE. Inspecting the
2659 previous figure of the raw counts suggests a relationship between counts and forest cover
2660 which is perhaps not surprising.

2661 3.6.2 Doing it in WinBUGS

2662 Here we demonstrate how to fit a Poisson GLM in **WinBUGS** using the covariate $x_i =$
2663 forest cover along BBS route i . It is advisable that x_i be standardized in most cases as
2664 this will improve mixing of the Markov chains. We have pre-standardized the forest cover
2665 covariate for the BBS route locations, and so we don't have to worry about that here. To
2666 read the BBS data into **R** and get things set up for **WinBUGS** we issue the following
2667 commands:

```

2668 > library(scrbook)
2669 > data(bbsdata)
2670
2671 > y <- bbsdata$counts[, "X90"] # Pick year 1990
2672 > notna <- !is.na(y)
2673 > y <- y[notna]
2674
2675 ## Forest cover already standardized here:
2676 > habitat <- bbsdata$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
 2710 more detail in Chapt. 17 where you can also find step-by-step instructions for Metropolis-
 2711 within-Gibbs samplers, should the following section move through all this material too
 2712 quickly.

2713 We will assume that the two parameters, β_0 and β_1 , have diffuse normal priors, say
 2714 $[\beta_0] = \text{Normal}(0, 100)$ and $[\beta_1] = \text{Normal}(0, 100)$ where each has *standard deviation* 100
 2715 (recall that **WinBUGS** parameterizes the normal in terms of $1/\sigma^2$). We need to assem-
 2716 ble the relevant elements of the model which are these two prior distributions and the
 2717 likelihood $[\mathbf{y}|\beta_0, \beta_1] = \prod_i [y_i|\beta_0, \beta_1]$ which is, mathematically, the product of the Poisson
 2718 pmf evaluated at each y_i , given particular values of β_0 and β_1 . Next, we need to identify
 2719 the full conditionals $[\beta_0|\beta_1, \mathbf{y}]$ and $[\beta_1|\beta_0, \mathbf{y}]$. We use the all-purpose rule for constructing
 2720 full conditionals (section 3.3.2) to discover that:

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

2721 Mathematically, the full conditional is of the form

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

2722 which you can program as an **R** function with arguments β_0 , β_1 and \mathbf{y} without difficulty.

2723 The full-conditional for β_1 is:

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

2724 which has a similar mathematical representation except the prior is expressed in terms
 2725 of β_1 instead of β_0 . Remember, we could replace the “ \propto ” with “=” if we put $[y|\beta_1]$ or
 2726 $[y|\beta_0]$ in the denominator. But, in general, $[y|\beta_0]$ or $[y|\beta_1]$ will be quite a pain to compute
 2727 and, more importantly, it is a constant as far as the operative parameters (β_0 or β_1 ,
 2728 respectively) are concerned. Therefore, the MH acceptance probability will be the ratio
 2729 of the full-conditional evaluated at a candidate draw to that evaluated at the current
 2730 value, and so the denominator required to change \propto to $=$ winds up canceling from the
 2731 MH acceptance probability.

2732 Here we will use the so-called random walk candidate generator, which is a Normal
 2733 proposal distribution, so that, for example, $\beta_0^* \sim \text{Normal}(\beta_0^t, \delta)$ where δ is the standard-
 2734 deviation of the proposal distribution, which is just a tuning parameter that is set by
 2735 the user and adjusted to achieve efficient mixing of chains (see Sec. 17.2.2). We remark
 2736 also that calculations are often done on the log-scale to preserve numerical integrity of
 2737 things when quantities evaluate to small or large numbers, so keep in mind, for example,
 2738 $a * b = \exp(\log(a) + \log(b))$ for two positive numbers a and b . The “Metropolis within
 2739 Gibbs” algorithm for a Poisson regression turns out to be remarkably simple and is given
 2740 in Panel 3.1. It is also part of the **scrbook** package and you can run 1000 iterations of it
 2741 by calling `PoisGLMBBS(y=y, habitat=habitat, niter=1000)` (note that y = point count
 2742 data and `habitat` = forest cover have to be defined in your **R** workspace as shown in the
 2743 previous analysis of these data).

```

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

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

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

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

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

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

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

```

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

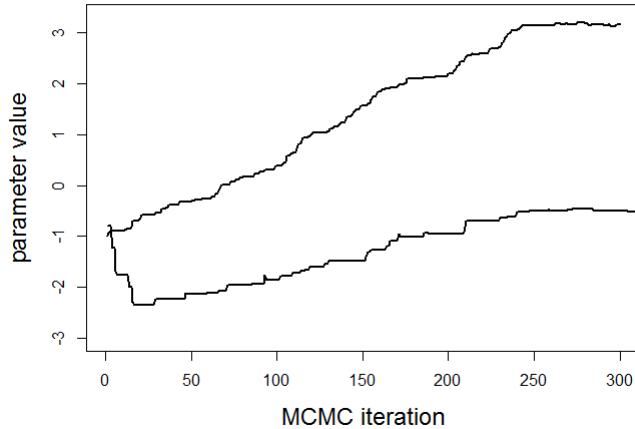


Figure 3.6. First 300 MCMC iterations for the Poisson GLM model parameters β_0 (top) and β_1 (bottom) using a Metropolis-Hastings tuning parameter of $\delta = 0.05$.

2744 The first 300 iterations of the MCMC history of each parameter are shown in Fig. 3.6.
 2745 These chains are not very appealing but a couple of things are evident: We see that the
 2746 burn-in takes about 250 iterations and that after that chains seem to mix reasonably well,
 2747 although this is not so clear given the scale of the y-axis, which we have chosen to get
 2748 both variables on the same graph. We generated 10,000 posterior samples, discarding the
 2749 first 500 as burn-in, and the result is shown in Fig. 3.7, this time on separate panels for
 2750 each parameter. The “grassy” look of the MCMC history is diagnostic of Markov chains
 2751 that are well-mixing and we would generally be very satisfied with results that look like
 2752 this.

2753 Note that we used a specific set of starting values for these simulations. It should be
 2754 clear that starting values closer to the mass of the posterior distribution might cause burn-
 2755 in to occur faster. Note also that we have used a different prior than in our **WinBUGS**
 2756 model specification given previously. We encourage you to evaluate whether this seems to
 2757 affect the result.

3.7 POISSON GLM WITH RANDOM EFFECTS

2758 In most of this book, we will be dealing with random effects in GLM-like models – similar
 2759 to what are usually referred to as generalized linear mixed models (GLMMs). We provide
 2760 a brief introduction of such a model by way of example, extending our Poisson regression
 2761 model to include a random effect.

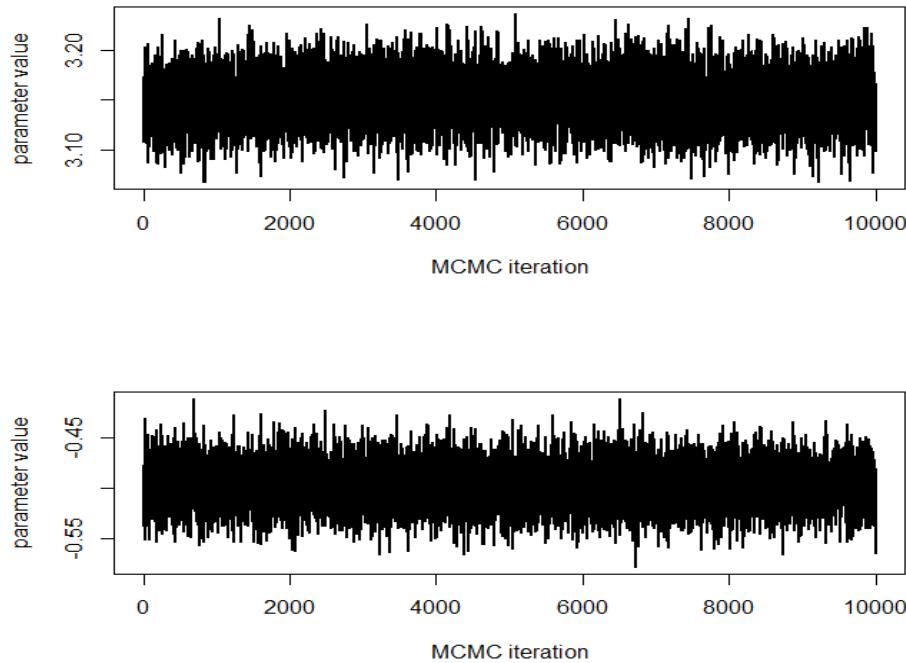


Figure 3.7. Nice grassy plots of 10,000 MCMC iterations for the Poisson GLM model parameters β_0 (top) and β_1 (bottom) using a Metropolis-Hastings tuning parameter of $\delta = 0.05$.

2762 **The Log-Normal mixture:** The classical situation involves a GLM with a normally
 2763 distributed random effect that is additive on the linear predictor. For the Poisson case,
 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
```

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

```

2774 > set.seed(2013)
2775 ### Dump the BUGS model into a file
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
2796 > library(R2WinBUGS)           # Load and run R2WinBUGS
2797 > out <- bugs (data, inits, parameters, "model.txt", n.thin=2,n.chains=2,
2798   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

Another extremely important class of models in ecology are binomial models. We use 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

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”

2841 it seems natural to view occupancy as being derived from local abundance N (Royle
 2842 and Nichols, 2003; Royle and Dorazio, 2006; Dorazio, 2007). Therefore, models of local
 2843 abundance in which $N_i \sim \text{Poisson}(A_i \lambda_i)$ for a habitat patch of area A_i implies a model
 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 distri-
 2848 butions, where $K = 1$ and the outcome is occurrence ($y = 1$) or not ($y = 0$) of some
 2849 species. Such examples abound in books (e.g., Royle and Dorazio (2008, ch. 3); Kéry
 2850 (2010, ch. 21); Kéry and Schaub (2012, ch. 13)) and in the literature. Therefore, instead,
 2851 we will consider an example involving band returns of waterfowl in the upper great plains
 2852 including some Canadian provinces, which were analyzed by Royle and Dubovsky (2001).

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
 2878 estimating the difference in p relative to category 1, we estimate a separate intercept term
 2879 for each category, so that we have 5 different β_0 parameters indexed by t . This reduces
 2880 the linear predictor to:

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

2881 The model can be implemented in the **BUGS** language for the mallard banding data
 2882 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 <- c('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)
```

2910 Look at the posterior summaries of model parameters in Table 3.2. The basic result
 2911 suggests a negative east-west gradient and a positive south to north gradient of band
 2912 recovery probabilities, but no interaction. A map of the response surface is shown in Fig.
 2913 3.8.

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

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

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

3.9 BAYESIAN MODEL CHECKING AND SELECTION

2914 In general terms, model checking – or assessing the adequacy of the model – and model
 2915 selection are quite thorny issues and, despite contrary and, sometimes, strongly held belief
 2916 among practitioners, there are not really definitive, general solutions to either problem.
 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 re-
 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

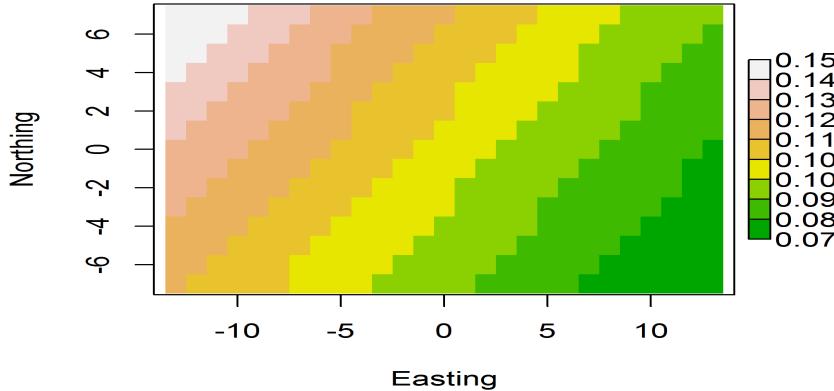


Figure 3.8. Predicted recovery rates of mallard bands in the upper great plains of North America. Note the negative gradient from the NW to the SE.

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

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

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

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

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

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

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:

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

2959 and the Bayesian p-value is simply the posterior probability $\text{Pr}(T(\mathbf{y}^{new}) > T(\mathbf{y}))$ which
 2960 should be close to 0.50 for a good model – one that “fits” in the sense that the observed
 2961 data set is consistent with realizations simulated under the model being fitted to the
 2962 observed data. In practice we judge “close to 0.50” as being “not too close to 0 or 1” and,
 2963 as always, closeness is somewhat subjective. We’re happy with anything $> .1$ and $< .9$
 2964 but might settle for $> .05$ and < 0.95 . Another useful fit statistic is the Freeman-Tukey
 2965 statistic, in which

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

2966 (Brooks et al., 2000), where y_i is the observed value of observation i and $\mathbb{E}(y_i)$ its expected
 2967 value. In contrast to a Chi-square discrepancy, the Freeman-Tukey statistic removes the
 2968 need to pool cells with small expected values. In summary, you can see that the Bayesian
 2969 p-value is easy to compute, and it is widely used as a result.

2970 3.9.2 Model selection

2971 In ecology, scientific hypotheses are often manifest as different models or parameters of
 2972 a model, and so evaluating the importance of different models is fundamental to many
 2973 ecological studies. For Bayesian model selection we typically use three different methods:
 2974 First is, let’s say, common sense. If a variable should plausibly be relevant to explaining
 2975 the data-generating processes, and it has posterior mass concentrated away from 0, then it
 2976 seems like it should be regarded as important - that is, it is “significant.” This approach
 2977 seems to have fallen out of favor in ecology over the last 10 or 15 years but in many
 2978 situations it is a reasonable thing to do.

2979 For regression problems we sometimes use the indicator variable method of Kuo and
 2980 Mallick (1998), in which we introduce a set of binary variables w_k for variable k , and
 2981 express the model as, e.g., for a single covariate model:

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

2982 where w_1 is given a Bernoulli prior distribution with some prescribed probability. E.g.,
 2983 $w_1 \sim \text{Bernoulli}(0.50)$ to provide a prior probability of 0.50 that variable x should be an
 2984 element of the linear predictor. The posterior probability of the event $w_1 = 1$ is a gage of
 2985 the importance of the variable x . i.e., high values of $\text{Pr}(w_1 = 1)$ indicate stronger evidence
 2986 to support that “ x is in the model” whereas values of $\text{Pr}(w_1 = 1)$ close to 0 suggest that
 2987 x is less important. Expansion of the model to include the binary variable w_1 defines a
 2988 set of 2 distinct models for which we can directly compute the posterior probabilities for,
 2989 merely by tallying up the posterior frequency of w_1 . See Royle and Dorazio (2008, Chapt.
 2990 3) for an example in the context of logistic regression.

2991 This approach seems to even work sometimes with fairly complex hierarchical models
 2992 of a certain form. E.g., Royle (2008) applied it to a random effects model to evaluate the
 2993 importance of the random effect component of the model. The main problem, which is

2994 really a general problem in Bayesian model selection, is that its effectiveness and results
 2995 will typically be highly sensitive to the prior distribution on the structural parameters
 2996 (e.g., see Royle and Dorazio (2008, table 3.6)). The reason for this is obvious: If $w_1 = 0$
 2997 for the current iteration of the MCMC algorithm, so that β is sampled from the prior
 2998 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 $w_1 = 1$, then the Markov chain may only jump from $w_1 = 0$ to $w_1 = 1$ 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, 2009b), 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						N	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	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 $M = 175$.

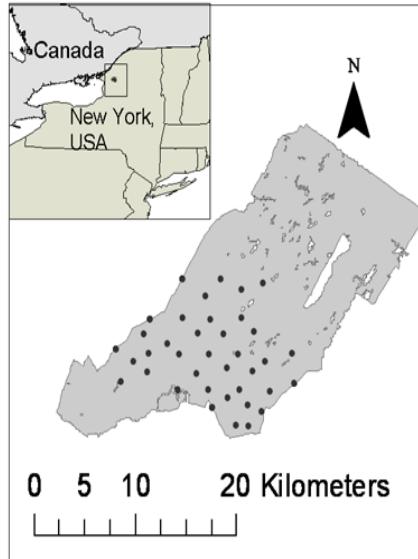


Figure 4.1. Fort Drum Black bear study area and the 38 baited hair snare locations operated for 8 weeks during June and July, 2006.

```

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.

3431 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....

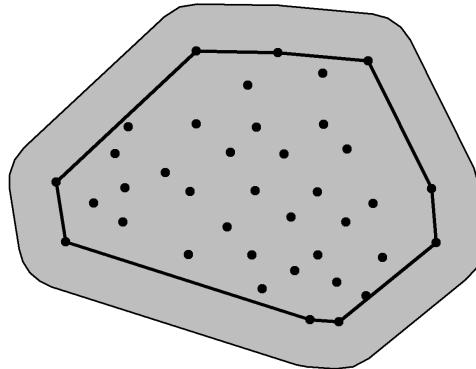


Figure 4.2. Convex hull of the bear hair snare array at Fort Drum, NY, buffered by mean female home range radius (2.19 km).

```

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.

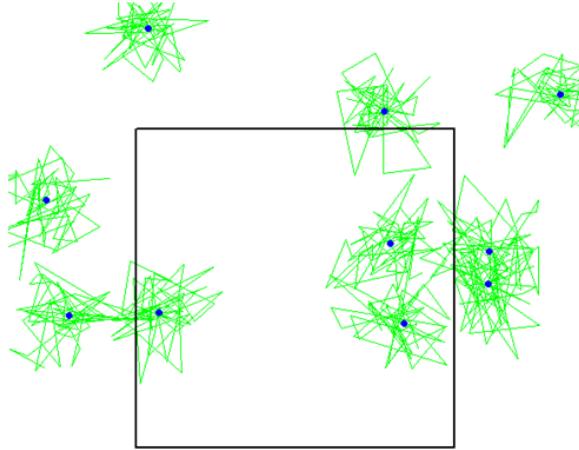


Figure 4.3. A quadrat searched for lizards over some period of time (simulated data). The locations of encounter for each of 10 lizards are connected by lines—the dots are activity centers.

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 > wbout <- 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(wbout$sims.list$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(wbout$sims.list$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

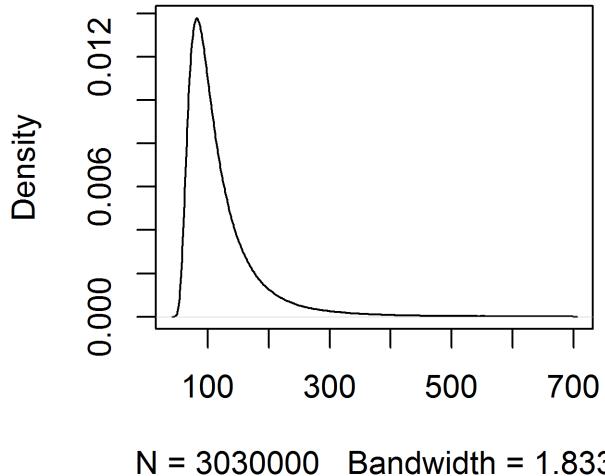


Figure 4.4. Posterior of N for Fort Drum bear study data under the logit-normal version of model M_h .

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:

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

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, 2009b). 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]
```

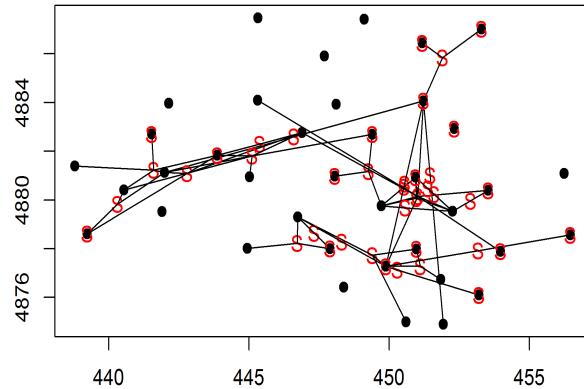


Figure 4.5. Spider plot of the Fort Drum study data. The black dots represent the 47 trap locations with the "S" symbols being the average capture location of each bear. i.e., its estimated home range center. All traps in which a bear was captured are connected to its estimated home range center with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

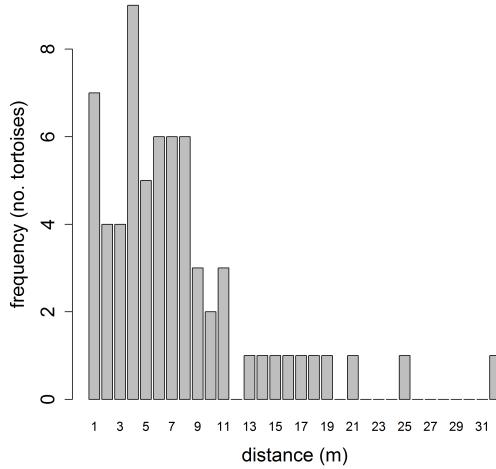


Figure 4.7. Distance histogram of $n = 65$ Sonoran desert tortoise detections from a total of 120 km of survey transect.

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

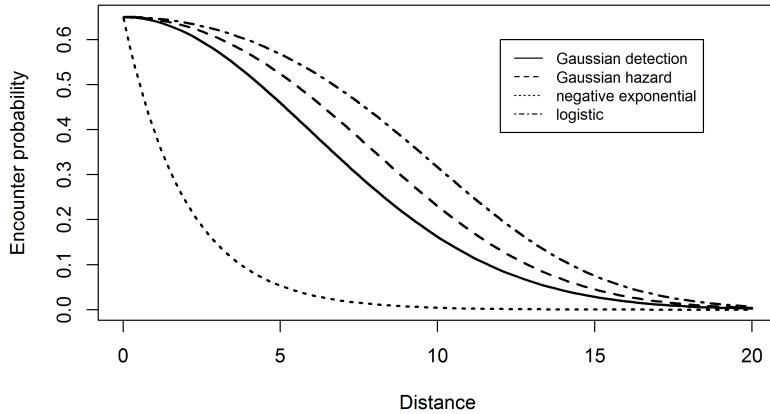


Figure 5.1. Some common encounter probability models showing the characteristic monotone decrease of encounter probability with distance between activity center and trap location.

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 $N = 50$?

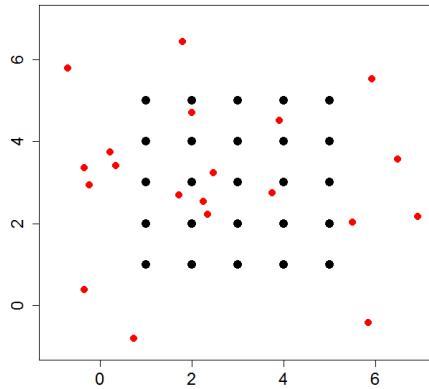


Figure 5.2. Realization (small dots) of a binomial point process with $N = 20$. The large dots represent trap locations.

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

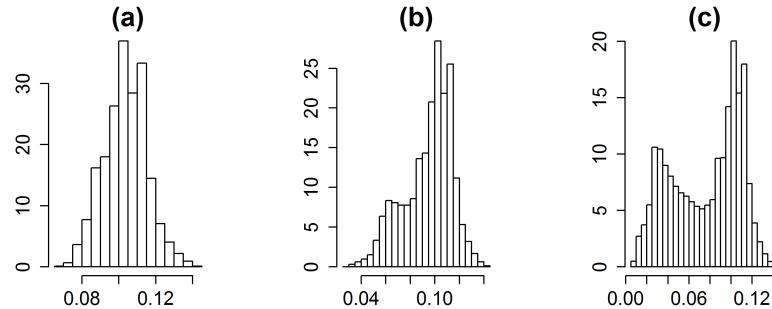


Figure 5.3. Implied distribution of p_i for a population of individuals as a function of the size of the state-space buffer around the trap array. The state-space buffer is 0.2, 0.5 and 1.0 for panels (a), (b), (c), respectively. In each case, the trap array is fixed and centered within a square state-space.

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} .

5.4.1 Bivariate normal case

4510 One encounter model that allows direct analytic computation of home range area is the
 4511 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.

5.4.2 Empirical analysis

4535 For any encounter model we can compute space usage quantiles empirically by taking a fine
 4536 grid of points and either simulating movement outcomes with probabilities proportional to
 4537 $p(\mathbf{x}, \mathbf{s})$ and accumulating area around \mathbf{s} , or else we can do this precisely by varying $B(\alpha)$
 4538 to find that value within which 95% of all movements are concentrated, i.e., the set of all
 4539 \mathbf{x} such that $\|\mathbf{x} - \mathbf{s}\| \leq B(q)$. Under any detection model, movement outcomes will occur
 4540 in proportion to $p(\mathbf{x}, \mathbf{s})$, as long as the probability of encounter is constant, *conditional on*
 4541 use, and so we can define our space usage distribution according to:

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

4542 Given the probabilities $\pi(\mathbf{x}, \mathbf{s})$ for all \mathbf{x} we can find the value of $B(q)$, for any q , such that

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

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 $\pi * .978^2 = 3.007$ spatial units. You can change the intercept of the model and find
 4586 that it has no effect. The true (analytic) value of σ that produces a home range area of 3.0
 4587 is 0.3993 which is the value we initially plugged in to the `hra` function. We can improve
 4588 on the numerical approximation to home range area (get it closer to 3.0) by increasing the
 4589 resolution of our spatial grid (increase the `ng` argument) along with the `tol` argument.

4590 We can also reverse this process, and find, for any detection model, the parameter
 4591 values that produce a certain $(1 - q)\%$ home range area, which we imagine would be
 4592 useful for doing simulation studies. The function `hra` will compute the value of the scale
 4593 parameter that achieves a certain target $(1 - q)\%$ home range area, by simply providing a
 4594 non-null value of the variable `target.area`. Here we use `target.area = 3.00735` (from
 4595 above) to obtain a close approximation to the value σ we started with (the parameter
 4596 argument is meaningless here):

```
4597 > hra(pGauss1,parms=c(-2,.3993),plot=FALSE,xlim,ylim,ng=500,  

4598     target.area=3.00735,tol=.0005)  

4599  

4600 Value of parm[2] to achieve 95% home range area of 3.00735: 0.3993674
```

4601 5.4.3 Relevance of understanding space usage

4602 One important reason that we need to be able to deduce “home range area” from a
 4603 detection model is so that we can compare different models with respect to a common
 4604 biological currency. Many encounter probability models have some “scale parameter”,
 4605 which we might call σ no matter the model, but this relates to 95% area in a different
 4606 manner under each model. Therefore, we want to be able to convert different models
 4607 to the same currency. Another reason to understand the relationship between models of
 4608 encounter probability and space usage is that it opens the door to combining traditional
 4609 resource selection data from telemetry with spatial capture-recapture data. In Chapt. 13
 4610 we consider this problem, for the case in which a sample of individuals produces encounter
 4611 history data suitable for SCR models and, in addition, we have telemetry relocations on a
 4612 sample of individuals. This is achieved by regarding the two sources of data as resulting
 4613 from the same underlying process of space usage but telemetry data produce “perfect”
 4614 observations, like always-on camera traps blanketing a landscape. We use this idea to
 4615 model the effect of a measured covariate at each pixel, say $C(\mathbf{x})$, on home range size and
 4616 geometry and, hence, the probability of encounter in traps.

4617 5.4.4 Contamination due to behavioral response

4618 Interpretation of encounter probability models as models of animal home range and space
 4619 usage can be complicated by a number of factors, including whether traps are baited or
 4620 not. In the case of baited traps, this might lead to a behavioral response (Sec. 7.2.3)
 4621 which could affect animal space usage. For example, if traps attract animals from a long
 4622 distance, it could make typical home ranges appear larger than normal. More likely, in our
 4623 view, it wouldn’t change the typical size of a range but would change how individuals use
 4624 their range e.g., by moving from baited trap to baited trap, so that observed movement
 4625 distances of individuals are typically larger than normal.

4626 In other cases, the reliance on Euclidean distance in models for encounter probability
 4627 might be unrealistic, and can lead to biased estimates of density (Royle et al., 2012a).
 4628 For example, animals might concentrate their movements along trails, roads, or other
 4629 landscape features. In this case, models that accommodate other distance metrics can be
 4630 considered. We present models based on least-cost path in Chapt. 12.

5.5 SIMULATING SCR DATA

4631 It is always useful to simulate data because it allows you to understand the system that
 4632 you're modeling and also calibrate your understanding with specific values of the model
 4633 parameters. That is, you can simulate data using different parameter values until you
 4634 obtain data that "look right" based on your knowledge of the specific situation that
 4635 you're interested in. Here we provide a simple script to illustrate how to simulate spatial
 4636 encounter history data. In this exercise we simulate data for 100 individuals and a 25 trap
 4637 array laid out in a 5×5 grid of unit spacing. The specific encounter model is the Gaussian
 4638 model given above and we used this code to simulate data used in subsequent analyses.
 4639 The 100 activity centers were simulated on a state-space defined by a 8×8 square within
 4640 which the trap array was centered (thus the trap array is buffered by 2 units). Therefore,
 4641 the density of individuals in this system is fixed at $100/64$.

```

4642 > set.seed(2013)
4643 # Create 5 x 5 grid of trap locations with unit spacing
4644 > traplocs <- cbind(sort(rep(1:5,5)),rep(1:5,5))
4645 > ntraps <- nrow(traplocs)
4646 # Compute distance matrix:
4647 > Dmat <- e2dist(traplocs,traplocs)

4648

4649

4650 # Define state-space of point process. (i.e., where animals live).
4651 # "buffer" just adds a fixed buffer to the outer extent of the traps.
4652 #
4653 > buffer <- 2
4654 > xlim <- c(min(traplocs[,1] - buffer),max(traplocs[,1] + buffer))
4655 > ylim <- c(min(traplocs[,2] - buffer),max(traplocs[,2] + buffer))

4656

4657 > N <- 100    # population size
4658 > K <- 20     # number nights of effort

4659

4660 > sx <- runif(N,xlim[1],xlim[2])    # simulate activity centers
4661 > sy <- runif(N,ylim[1],ylim[2])
4662 > S <- cbind(sx,sy)
4663 # Compute distance matrix:
4664 > D <- e2dist(S,traplocs)  # distance of each individual from each trap

4665

4666 > alpha0 <- -2.5      # define parameters of encounter probability
4667 > sigma <- 0.5        # scale parameter of half-normal
4668 > alpha1 <- 1/(2*sigma*sigma) # convert to coefficient on distance

4669

4670 # Compute Probability of encounter:
4671 #
4672 > probcap <- plogis(-2.5)*exp( - alpha1*D*D)

4673 # Generate the encounters of every individual in every trap

```

```

4675 > Y <- matrix(NA,nrow=N,ncol=ntraps)
4676 > for(i in 1:nrow(Y)){
4677   Y[i,] <- rbinom(ntraps,K,probcap[i,])
4678 }

```

4679 We remind the reader that, in presenting **R** or other code snippets throughout the
 4680 book, we will deviate from our standard variable expressions for some quantities. In
 4681 particular, we sometimes substitute words for integer variable designations: **nind** (for n),
 4682 **ntraps** (for J), and **nocc** (for K). In our opinion this leaves less to be inferred by the
 4683 reader in trying to understand code snippets.

4684 Subsequently we will generate data using this code packaged in an **R** function called
 4685 **simSCRO** in the package **scrbook** which takes a number of arguments including **discard0**
 4686 which, if TRUE, will return only the encounter histories for captured individuals. A second
 4687 argument is **array3d** which, if TRUE, returns the 3-dimensional encounter history array
 4688 instead of the aggregated **nind** \times **ntraps** encounter frequencies (see below). Finally we
 4689 provide a random number seed, **rnd** = 2013 to ensure repeatability of the analysis here.
 4690 We obtain a data set as above using the following command:

```

4691 > data <- simSCRO(discard0=TRUE, array3d=FALSE, rnd=2013)

```

4692 The **R** object **data** is a list, so let's take a look at what's in the list and then harvest some
 4693 of its elements for further analysis below.

```

4694 > names(data)
4695 [1] "Y"      "traplocs" "xlim"      "ylim"      "N"       "alpha0"    "beta"
4696 [8] "sigma"   "K"
4697
4698 ## Grab encounter histories from simulated data list
4699 > Y <- data$Y
4700 ## Grab the trap locations
4701 > traplocs <- data$traplocs

```

4702 5.5.1 Formatting and manipulating real data sets

4703 Conventional capture-recapture data are easily stored and manipulated as a 2-dimensional
 4704 array, an **nind** \times **K** (individuals by sample occasions) matrix, which is maximally informative
 4705 for any conventional capture-recapture model, but not for spatial capture-recapture
 4706 models. For SCR models we must preserve the spatial information in the encounter history
 4707 information. We will routinely analyze data from 3 standard formats:

- 4708 (1) The basic 2-dimensional data format, which is an **nind** \times **ntraps** encounter frequency
 4709 matrix such as that simulated previously. These are the total number of encounters in
 4710 each trap, summed over the K sample occasions.
- 4711 (2) The maximally informative 3-dimensional array, for which we establish here the con-
 4712 convention that it has dimensions **nind** \times **ntraps** \times **K**.
- 4713 (3) We use a compact format – the “encounter data file” – which we describe below in
 4714 Sec. 5.9.

4715 To simulate data in the most informative format - the “3-d array” - we can use the **R**
 4716 commands given previously but replace the last 4 lines with the following:

```
4717 > Y <- array(NA,dim=c(N,ntraps,K))
4718
4719 > for(i in 1:nrow(Y)){
4720   for(j in 1:ntraps){
4721     Y[i,j,1:K] <- rbinom(K,1,probcap[i,j])
4722   }
4723 }
```

4724 We see that a collection of K binary encounter events are generated for *each* individual
 4725 and for *each* trap. The probabilities of those Bernoulli trials are computed based on the
 4726 distance from each individual’s home range center and the trap (see calculation above),
 4727 and those are housed in the matrix `probcap`. Our data simulator function `simSRC0` will
 4728 return the full 3-d array if `array3d=TRUE` is specified in the function call. To recover the
 4729 2-d matrix from the 3-d array, and subset the 3-d array to individuals that were captured,
 4730 we do this:

```
4731 # Sum over the ‘‘sample occasions’’ dimension (3rd margin of the array)
4732 > Y2d <- apply(Y,c(1,2),sum)
4733
4734 # Compute how many times each individual was captured
4735 > ncaps <- apply(Y2d,1,sum)
4736
4737 # Keep those individuals that were captured
4738 > Y <- Y[ncaps>0,,]
```

5.6 FITTING MODEL SCR0 IN BUGS

4739 Clearly if we somehow knew the value of N then we could fit this model directly because,
 4740 in that case, it is a special kind of logistic regression model, one with a random effect (`s`)
 4741 that enters into the model in a peculiar fashion, and also with a distribution (uniform)
 4742 which we don’t usually think of as standard for random effects models. So our aim here is
 4743 to analyze the known- N problem, using our simulated data, as an incremental step in our
 4744 progress toward fitting more generally useful models. To begin, we use our simulator to
 4745 grab a data set and then harvest the elements of the resulting object for further analysis.

```
4746 > data <- simSRC0(discard0=FALSE,rnd=2013)
4747 > y <- data$Y
4748 > traplocs <- data$traplocs
4749
4750 # In this case nind=N because we’re doing the known-N problem
4751 #
4752 > nind <- nrow(y)
4753 > X <- data$traplocs
4754 > J <- nrow(X)    # number of traps
4755 > K <- data$K
```

```
4756 > xlim <- data$xlim
4757 > ylim <- data$ylim
```

4758 Note that we specify `discard0 = FALSE` so that we have a “complete” data set, i.e.,
 4759 one with the all-zero encounter histories corresponding to uncaptured individuals. Now,
 4760 within an **R** session, we can create the **BUGS** model file and fit the model using the
 4761 following commands.

```
4762 cat("
4763   model{
4764     alpha0 ~ dnorm(0,.1)
4765     logit(p0) <- alpha0
4766     alpha1 ~ dnorm(0,.1)
4767     sigma <- sqrt(1/(2*alpha1))
4768     for(i in 1:N){ # note N here -- N is KNOWN in this example
4769       s[i,1] ~ dunif(xlim[1],xlim[2])
4770       s[i,2] ~ dunif(ylim[1],ylim[2])
4771       for(j in 1:J){
4772         d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
4773         y[i,j] ~ dbin(p[i,j],K)
4774         p[i,j] <- p0*exp(- alpha1*d[i,j]*d[i,j])
4775       }
4776     }
4777   }
4778 ",file = "SCR0a.txt")
```

4779 This model describes the Gaussian encounter probability model, but it would be trivial
 4780 to modify that to various others including the logistic described above. One consequence
 4781 of using the half-normal is that we have to constrain the encounter probability to be in
 4782 $[0, 1]$ which we do here by defining `alpha0` to be the logit of the intercept parameter `p0`.
 4783 Note that the distance covariate is computed within the **BUGS** model specification given
 4784 the matrix of trap locations, `X`, which is provided to **WinBUGS** as data.

4785 Next we do a number of organizational activities including bundling the data for **Win-**
4786 BUGS, defining some initial values, the parameters to monitor and some basic MCMC
 4787 settings. We choose initial values for the activity centers `s` by generating uniform random
 4788 numbers in the state-space but, for the observed individuals, we replace those values by
 4789 each individual’s mean trap coordinate for all encounters

```
4790 ### Starting values for activity centers, s
4791 > sst <- cbind(runif(nind,xlim[1],xlim[2]),runif(nind,ylim[1],ylim[2]))
4792 > for(i in 1:nind){
4793   if(sum(y[i,])==0) next
4794   sst[i,1] <- mean( X[y[i,]>0,1] )
4795   sst[i,2] <- mean( X[y[i,]>0,2] )
4796 }
4797 > data <- list (y=y, X=X, K=K, N=nind, J=J, xlim=xlim, ylim=ylim)
4798 > inits <- function(){
```

```

4800     list (alpha0=rnorm(1,-4,.4), alpha1=runif(1,1,2), s=sst)
4801   }
4802
4803 > library(R2WinBUGS)
4804 > parameters <- c("alpha0","alpha1","sigma")
4805 > out <- bugs (data, inits, parameters, "SCR0a.txt", n.thin=1, n.chains=3,
4806           n.burnin=1000,n.iter=2000,debug=TRUE,working.dir=getwd())

```

4807 There is little to say about the preceding operations other than to suggest that you might
 4808 explore the output and investigate additional analyses by running the `simSCR0` script
 4809 provided in the **R** package `scrbook`.

4810 For purposes here, we ran 1000 burn-in and 1000 post-burn-in iterations, and 3 chains,
 4811 to obtain 3000 posterior samples. Because we know N for this particular data set we only
 4812 have 2 parameters of the detection model to summarize (`alpha0` and `alpha1`), along with
 4813 the derived parameter σ , the scale parameter of the Gaussian kernel, i.e., $\sigma = \sqrt{1/(2\alpha_1)}$.
 4814 When the object `out` is produced we print a summary of the results as follows:

```

4815 > print(out,digits=2)
4816 Inference for Bugs model at "SCR0a.txt", fit using WinBUGS,
4817   3 chains, each with 2000 iterations (first 1000 discarded)
4818   n.sims = 3000 iterations saved
4819     mean    sd   2.5%   25%   50%   75% 97.5% Rhat n.eff
4820 alpha0   -2.50  0.22 -2.95 -2.65 -2.48 -2.34 -2.09 1.01  190
4821 alpha1    2.44  0.42  1.64  2.15  2.44  2.72  3.30 1.00  530
4822 sigma     0.46  0.04  0.39  0.43  0.45  0.48  0.55 1.00  530
4823 deviance 292.80 21.16 255.60 277.50 291.90 306.00 339.30 1.01  380
4824
4825
4826 [...some output deleted...]
4827

```

4828 We know the data were generated with `alpha0 = -2.5` and `alpha1 = 2`. The estimates
 4829 look reasonably close to those data-generating values and we probably feel pretty good
 4830 about the performance of the Bayesian analysis and MCMC algorithm that **WinBUGS**
 4831 cooked-up based on our sample size of 1 data set. It is worth noting that the `Rhat`
 4832 statistics indicate reasonable convergence but, as a practical matter, we might choose to
 4833 run the MCMC algorithm for additional time to bring these closer to 1.0 and to increase
 4834 the effective posterior sample size (`n.eff`). Other summary output includes “deviance”
 4835 and related things including the deviance information criterion (DIC). We discuss general
 4836 issues of convergence and other MCMC considerations in Chapt. 17, and DIC and model
 4837 selection in Chapt. 8.

5.7 UNKNOWN N

4838 In all real applications N is unknown. We handled this important issue in Chapt. 4
 4839 using the method of data augmentation (DA) which we apply here to achieve a realistic
 4840 analysis of model SCR0. As with the basic closed population models considered previously,

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

```
4849 trap1 trap2 trap3 trap4
4850 [1,] 1 0 0 0
4851 [2,] 0 2 0 0
4852 [3,] 0 0 0 1
4853 [4,] 0 1 0 0
4854 [5,] 0 0 1 1
4855 [6,] 1 0 1 0
4856 [7,] 0 0 0 0
4857 [8,] 0 0 0 0
4858 [9,] 0 0 0 0
4859 [10,] 0 0 0 0
```

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

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

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

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

How big does the augmented data set have to be? We discussed this issue in Chapt. 4 where we noted that the size of the data set is equivalent to the upper limit of a uniform prior distribution on N . Practically speaking, it should be sufficiently large so that the posterior distribution for N is not truncated. On the other hand, if it is too large then unnecessary calculations are being done. An approach to choosing M by trial-and-error is indicated. Do a short MCMC run and then consider whether you need to increase M . See Chapt. 17 for an example of this. Kéry and Schaub (2012, Chapt. 6) provide an assessment of choosing M in closed population models. The useful thing about DA is that it removes N as an explicit parameter of the model. Instead, N is a derived parameter, computed by $N = \sum_{i=1}^M z_i$. Similarly, *density*, D , is also a derived parameter computed as $D = N/\text{area}(\mathcal{S})$.

4882 5.7.1 Analysis using data augmentation in WinBUGS

4883 We provide a complete **R** script for simulating and organizing a data set, and analyzing
 4884 the data in **WinBUGS**. As before we begin by obtaining a data set using our **simSCR0**
 4885 function and then harvesting the required data objects from the resulting data list. Note
 4886 that we use the **discard0=TRUE** option this time so that we get a “real looking” data set
 4887 with no all-zero encounter histories:

```
4888 ##  
4889 ## Simulate the data and extract the required objects  
4890 ##  
4891 > data <- simSCR0(discard0=TRUE,rnd=2013)  
4892 > y <- data$Y  
4893 > nind <- nrow(y)  
4894 > X <- data$traplocs  
4895 > K <- data$K  
4896 > J <- nrow(X)  
4897 > xlim <- data$xlim  
4898 > ylim <- data$ylim
```

4899 After harvesting the data we augment the data matrix **y** with $M - n$ all-zero encounter
 4900 histories, and create starting values for the variables z_i and also the activity centers \mathbf{s}_i
 4901 of which, for each, we require M values. One thing to take care of in using the **BUGS**
 4902 engines is the starting values for the activity centers. It is usually helpful to start the \mathbf{s}_i
 4903 for each observed individual at or near the trap(s) it was captured. All of this happens as
 4904 follows:

```
4905 ## Data augmentation  
4906 > M <- 200  
4907 > y <- rbind(y,matrix(0,nrow=M-nind,ncol=ncol(y)))  
4908 > z <- c(rep(1,nind),rep(0,M-nind))  
4909  
4910 ## Starting values for s  
4911 > sst <- cbind(runif(M,xlim[1],xlim[2]),runif(M,ylim[1],ylim[2]))  
4912 > for(i in 1:nind){  
4913   sst[i,1] <- mean( X[y[i,]>0,1] )  
4914   sst[i,2] <- mean( X[y[i,]>0,2] )  
4915 }
```

4916 Next, we write out the **BUGS** model specification and save it to an external file
 4917 called **SCR0b.txt**. The model specification now includes M encounter histories including
 4918 the augmented potential individuals, the data augmentation parameters z_i , and the data
 4919 augmentation parameter ψ :

```
4920 > cat("model{  
4921   alpha0 ~ dnorm(0,.1)  
4922   logit(p0) <- alpha0
```

```

4924 alpha1 ~ dnorm(0,.1)
4925 sigma <- sqrt(1/(2*alpha1))
4926 psi ~ dunif(0,1)

4927
4928 for(i in 1:M){
4929   z[i] ~ dbern(psi)
4930   s[i,1] ~ dunif(xlim[1],xlim[2])
4931   s[i,2] ~ dunif(ylim[1],ylim[2])
4932   for(j in 1:J){
4933     d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
4934     y[i,j] ~ dbin(p[i,j],K)
4935     p[i,j] <- z[i]*p0*exp(- alpha1*d[i,j]*d[i,j])
4936   }
4937 }
4938 N <- sum(z[])
4939 D <- N/64
4940 }
4941 ",file = "SCR0b.txt")

```

4942 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:

```

4945 > data <- list (y=y, X=X, K=K, M=M, J=J, xlim=xlim, ylim=ylim)
4946 > inits <- function(){
4947   list (alpha0=rnorm(1,-4,.4), alpha1=runif(1,1,2), s=sst, z=z)
4948 }
4949
4950 > library(R2WinBUGS)
4951 > parameters <- c("alpha0","alpha1","sigma","N","D")
4952 > out <- bugs (data, inits, parameters, "SCR0b.txt", n.thin=1,n.chains=3,
4953   n.burnin=1000,n.iter=2000,debug=TRUE,working.dir=getwd())

```

4954 Note the differences in this new **WinBUGS** model with that appearing in the known-
4955 N version – there are not many! The loop over individuals goes up to M now, and there is a
4956 model component for the DA variables z . We are also computing some derived parameters:
4957 population size $N(\mathcal{S})$ is computed by summing up all of the data augmentation variables
4958 z_i (as we've done previously in Chapt. 4) and density, D , is also a derived parameter,
4959 being a function of N . The input data has changed slightly too, as the augmented data
4960 set has more rows to include excess all-zero encounter histories. Previously we knew that
4961 $N = 100$ but in this analysis we pretend not to know N , but think that $N = 200$ is a
4962 good upper bound. This analysis can be run directly using the **SCR0bayes** function once
4963 the **scrbook** package is loaded, by issuing the following commands:

```

4964 > library(scrbook)
4965 > data <- simSCR0(discard0=TRUE,rnd=2013)
4966 > out1 <- SCR0bayes(data,M=200,engine="winbugs",ni=2000,nb=1000)

```

4967 Summarizing the output from **WinBUGS** produces:

```

4968 > print(out1,digits=2)
4969 Inference for Bugs model at "SCR0b.txt", fit using WinBUGS,
4970 3 chains, each with 2000 iterations (first 1000 discarded)
4971 n.sims = 3000 iterations saved
4972      mean     sd   2.5%   25%   50%   75% 97.5% Rhat n.eff
4973 alpha0    -2.57  0.23  -3.04  -2.72  -2.56  -2.41  -2.15 1.01   320
4974 alpha1     2.46  0.42   1.63   2.16   2.46   2.73   3.33 1.02   120
4975 sigma      0.46  0.04   0.39   0.43   0.45   0.48   0.55 1.02   120
4976 N        113.62 15.73  86.00 102.00 113.00 124.00 147.00 1.01   260
4977 D         1.78  0.25   1.34   1.59   1.77   1.94   2.30 1.01   260
4978 deviance 302.60 23.67 261.19 285.47 301.50 317.90 354.91 1.00 1400
4979
4980 [...some output deleted...]
4981

```

4982 The **Rhat** statistic (discussed in Secs. 3.5.2 and 17.4.5) for this analysis indicates
4983 satisfactory convergence. We see that the estimated parameters (α_0 and α_1) are comparable
4984 to the previous results obtained for the known- N case, and also not too different
4985 from the data-generating values. The posterior of N overlaps the data-generating value
4986 substantially.

4987 **Use of other BUGS engines: JAGS**

4988 There are two other popular **BUGS** engines in widespread use: **OpenBUGS** (Thomas
4989 et al., 2006) and **JAGS** (Plummer, 2003). Both of these are easily called from **R**. **Open-**
4990 **BUGS** can be used instead of **WinBUGS** by changing the package option in the **bugs**
4991 call to `package='OpenBUGS'`. **JAGS** can be called using the function **jags()** in package
4992 **R2jags** which has nearly the same arguments as **bugs()**. Or, it can be executed from the
4993 **R** package **rjags** (Plummer, 2011) which has a slightly different implementation that we
4994 demonstrate here as we reanalyze the simulated data set in the previous section (note:
4995 the same **R** commands are used to generate the data and package the data, inits and
4996 parameters to monitor). The function **jags.model** is used to initialize the model and run
4997 the MCMC algorithm for an adaptive period during which tuning of the MCMC algorithm
4998 might take place. These samples cannot be used for inference. Then the Markov chains
4999 are updated using **coda.samples()** to obtain posterior samples for analysis, as follows:

```

5000 > jinit <- jags.model("SCR0b.txt", data=data, inits=inits,
5001           n.chains=3, n.adapt=1000)
5002 > jout <- coda.samples(jinit, parameters, n.iter=1000, thin=1)

```

5003 These commands can be executed using the function **SCR0bayes** provided with the **R**
5004 package **scrbook**. Hobbs (2011) provides a good introduction to ecological modeling with
5005 **JAGS** which we recommend.

5006 **5.7.2 Implied home range area**

5007 Here we apply the method described in Sec. 5.4 to compute the effective home range
5008 area under different encounter probability models fit to simulated data. We simulated a
5009 data set from the Gaussian kernel model as in Sec. 5.7 and then we fitted 4 models to it:

Table 5.2. Posterior mean of model parameters for 4 different models fitted to a single simulated data set, and the effective home range area under each detection model.

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

(1) the true data-generating Gaussian encounter probability model; (2) the “hazard” or complementary log-log link model (Eq. 5.4.2); (3) the negative exponential model and (4) the logit model (Eq. 5.2.2). We modified the function `SCR0bayes` for this purpose which you should be able to do with little difficulty. We fit each model to the same simulated data set using **WinBUGS**, based only on 1000 post-burn-in samples and 3 chains, which produced the posterior summaries given in Table 5.2. The main thing we see is that, while the implied home range area can vary substantially, there are smaller differences in the estimated N and hence D .

5.7.3 Realized and expected density

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

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

There is a third type of inference objective that is relevant in practice and that is prediction of N for a population that was not sampled – i.e., a “new” population. To elaborate on this, consider a situation in which we are concerned about the tiger population in 2 distinct reserves in India. We do a camera trapping study on one of the reserves to estimate N_1 and we think the reserves are similar and homogeneous so we’re willing to apply a density estimate based on N_1 to the 2nd reserve. For the 2nd reserve, do we want a prediction of the realized population size, N_2 , or do we want an estimate of its expected

5044 value? We believe the former is the proper quantity for inference about the population
 5045 size in the 2nd reserve. An estimate of N_2 should include the uncertainty with which
 5046 the mean is estimated (from reserve 1) and it should also include “process variation” for
 5047 making the prediction of the latent variable N_2 .

5048 As a practical matter, to do a Bayesian analysis of this you could just define the state-
 5049 space to be the union of the two state-spaces, increase M so that the posterior of the
 5050 total population size is not truncated, and then have MCMC generate a posterior sample
 5051 of individuals on the joint state-space. You can tally-up the ones that are on \mathcal{S}_2 as an
 5052 estimate of N_2 . Alternatively, we can define $\mu = \psi M/A_1$ and then simulate posterior
 5053 samples of $N_s \sim \text{Binomial}(M, \mu A_2/M)$ for the new state-space area, A_2 .

5054 To carry out a classical likelihood analysis of this 2nd type of problem, what should we
 5055 do? The argument for making a prediction of a new value of N would go something like
 5056 this: If you obtain an MLE of N , say \hat{N} , then the inference procedure tells us the variance
 5057 of this *conditional* on N . i.e., $\text{Var}(\hat{N}|N)$. This is fine, if we care about the specific value
 5058 of N that generated our data set. However, if we don’t care about the specific one in
 5059 question then we want to “uncondition” on N to introduce a new variance component.
 5060 Law of total variance says:

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

5061 If \hat{N} is unbiased then we say the unconditional variance is

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

5062 The first part is estimation error and the 2nd component is the “process variance.” If
 5063 you do Bayesian analysis, then you don’t have to worry too much about how to compute
 5064 variances properly. You decide if you care about N , or its expected value, or predictions
 5065 of some “new” N , and you tabulate the correct posterior distribution from your MCMC
 5066 output.

5067 The considerations for estimating density are the same. Density can be N/A where
 5068 N is the realized population, which we understand it to be unless we put an expectation
 5069 operator around the N like $\mathbb{E}(N)/A$. Classically, density is thought of as being defined as
 5070 the expected value of N but this might not always be meaningful because the context of
 5071 whether we mean realized density, of an actual population, or expected density for some
 5072 hypothetical unspecified population, should matter. The formula for obtaining “expected
 5073 density” is slightly different depending on whether we assume N has a Poisson distribution
 5074 or whether we assume a binomial distribution (under data augmentation). In the latter
 5075 case ψ is related to the point process intensity (see Chapt. 11) in the sense that, under
 5076 the binomial prior:

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

5077 so, what we think of as “density”, D , is $D = M\psi/A$. Under the Poisson point process
 5078 model we have:

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

5079 In summary, there are 3 basic inference problems that relate to estimating population
 5080 size (or density):

- 5081 (1) What is the value of N for some population that was sampled. This is what Efford
 5082 and Fewster call “realized N” In general, we want the uncertainty to reflect having to
 5083 estimate n_0 , the part of the population not seen.

- 5084 (2) We need to estimate N for some population that we didn't sample but it is "similar"
5085 to the population that we have information on. In this case, we have to account for
5086 both variation in having to estimate parameters of the distribution of N and we have
5087 to account for process variation in N (i.e., due to the stochastic model of N).
5088 (3) In some extremely limited cases we might care about estimating the expected value of
5089 N , $\mathbb{E}(N)$. This is only useful as a hypothetical statement that we might use, e.g., if we
5090 were to establish a new million ha refuge somewhere, then we might say its expected
5091 population size is 200 tigers.

5.8 THE CORE SCR ASSUMPTIONS

5092 It's always a good idea to sit down and reflect on the meaning of any particular model,
5093 its various assumptions, and what they mean in a specific context. From the statistician's
5094 point of view, the basic assumption, the omnibus assumption, as in all of statistics, and
5095 for every statistical model, is that "the model is correctly specified". So, naturally, that
5096 precludes everything that isn't explicitly addressed by the model. To point this out to
5097 someone seems to cause a lot of anxiety, so we enumerate here what we think are the most
5098 important statistical assumptions of the basic SCR0 model:

- 5099 • **Demographic closure.** The model does not allow for demographic processes. There
5100 is no recruitment or entry into the sampled population. There is no mortality or exit
5101 from the sampled population.
- 5102 • **Geographic closure.** We assume no permanent emigration or immigration from the
5103 state-space. However, we allow for "temporary" movements around the state-space
5104 and variable exposure to encounter as a result. The whole point of SCR models is to
5105 accommodate this dynamic. In ordinary capture-recapture models we have to assume
5106 geographic closure to interpret N in a meaningful way.
- 5107 • **Activity centers are randomly distributed.** That is, uniformity and independence
5108 of the underlying point process s_1, \dots, s_N (see next section).
- 5109 • **Detection is a function of distance.** A detection model that describes how encounter
5110 probability declines as a function of distance from an individual's home range center.
- 5111 • **Independence of encounters** among individuals. Encounter of any individual is
5112 independent of encounter of each other individual.
- 5113 • **Independence of encounters** of the same individual. Encounter of an individual
5114 in any trap is independent of its encounter in any other trap, and subsequent sample
5115 occasion.

5116 It's easy to get worried and question the whole SCR enterprise just on the grounds that
5117 these assumptions combine to form such a simplistic model, one that surely can't describe
5118 the complexity of real populations. On this sentiment, a few points are worth making.
5119 First, you don't have inherently fewer assumptions by using an ordinary capture-recapture
5120 model but, rather, the SCR model relaxes a number of important assumptions compared
5121 to the non-spatial counterpart. For one, here, we're not assuming that p is constant for all
5122 individuals but rather that p varies substantially as a matter of the spatial juxtaposition of
5123 individuals with traps. So maybe the manner in which p varies isn't quite right, but that's
5124 not an argument that supports doing less modeling. Fundamentally a distance-based
5125 model for p has some basic biological justification in virtually every capture-recapture

study. Secondly, for some of these core assumptions such as uniformity, and independence of individuals and of encounters, we expect a fair amount of robustness to departures. They function primarily to allow us to build a model and an estimation scheme and we don't usually think they represent real populations (of course, no model does!). Third, we can extend these assumptions in many different ways and we do that to varying extents in this book, and more work remains to be done in this regard. Forth, we can also evaluate the reasonableness of the assumptions formally in some cases using standard methods of assessing model fit (Chapt. 8).

Finally, we return back to our sentiment about the omnibus assumptions which is that the model is properly specified. This precludes *everything* that isn't in the model. Sometimes you see in capture-recapture literature statements like "we assume no marks are lost", "marks are correctly identified" and similar things. We might as well also assume that, a shopping mall is not built, or a meteor does not crash down into our study area, the sun does not go super-nova, and so forth. Our point is that we should separate statistical assumptions about model parameters or aspects of the probability model from what are essentially logistical or operational assumptions about how we interpret our data, or based on our ability to conduct the study. It is pointless to enumerate all of the possible explanations for apparent *departures*, because there are an infinity of such cases.

5.9 WOLVERINE CAMERA TRAPPING STUDY

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

5.9.1 Practical data organization

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

- The encounter data file (EDF) containing a record of which traps and when each individual encounter occurred.
- The trap deployment file (TDF) which contains the coordinates of each trap, along with information indicating which sample occasions each trap was operating.

Encounter Data File (EDF) – We store the encounter data in the an efficient file format which is easily manipulated in **R** and easy to create in Excel and other spreadsheets which are widely used for data management. The file structure is a simple matrix with 4 columns, those being: (1) **session ID**: the trap *session* which usually corresponds to a year or a primary period in the context of a Robust Design situation, but it could also correspond to a distinct spatial unit (see Sec. 6.5.4 and Chapt. 14). For a single-year study (as considered here) this should be an integer that is the same for all records;

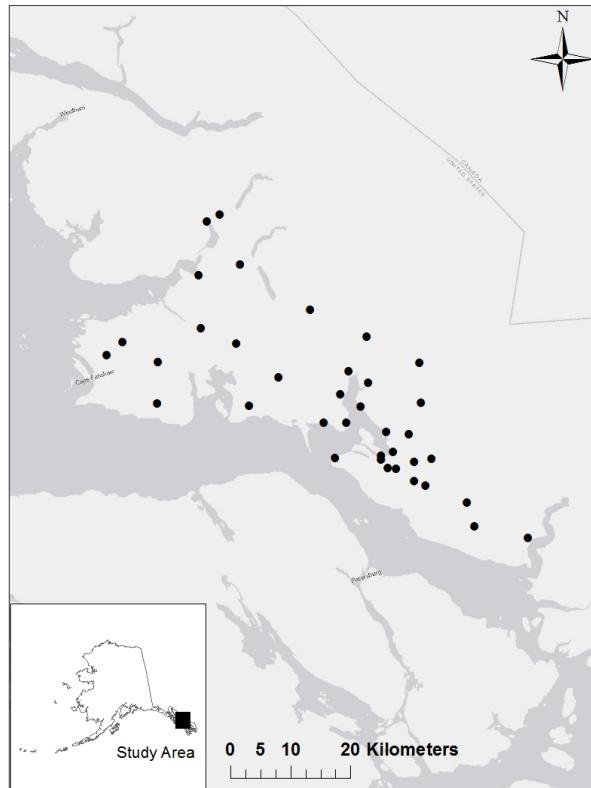


Figure 5.4. Wolverine camera trap locations (black dots) from a study that took place in SE Alaska. See Magoun et al. (2011) for details.

5166 (2) **individual ID:** the individual identity, being an integer from 1 to n (repeated for
 5167 multiple captures of the same individual) indicating which individual the record (row) of
 5168 the matrix belongs to; (3) **occasion ID:** The integer sample occasion which generated
 5169 the record, and (4) **trap ID:** the trap identity, an integer from 1 to J , the number of
 5170 traps. The structure of the EDF is the same as used in the **secr** package (Efford, 2011)
 5171 and similar to that used in the **SPACECAP** (Gopalaswamy et al., 2012a), and **SCRbayes**
 5172 (Russell et al., 2012) packages, both of which have a 3-column format (**trapID**, **indID**,
 5173 **sampID**). We note that the naming of the columns is irrelevant as far as anything we do in
 5174 this book, although **secr** and other software may have requirements on variable naming.
 5175 To illustrate this format, the wolverine data are available in the package **scrbook** by
 5176 typing:

```
5177 > data(wolverine)
```

5178 which contains a list having elements `wcaps` (the EDF) and `wtraps` (the TDF). We see
 5179 that `wcaps` has 115 rows, each representing a unique encounter event including the trap
 5180 identity, the individual identity and the sample occasion index (`sample`). The first 5 rows
 5181 of `wcaps` are:

```
5182 > wolverine$wcaps[1:5,]
5183   year individual day trap
5184 [1,]    1        2 127   1
5185 [2,]    1        2 128   1
5186 [3,]    1        2 129   1
5187 [4,]    1       18 130   1
5188 [5,]    1        3 106   2
```

5189 The 1st column here, labeled `year`, is an integer indicating the year or session of the
 5190 encounter. All these data come from a single year (2008) and so `year` is set to 1. Variable
 5191 `individual` is an integer identity of each individual captured, `day` is the sample occasion of
 5192 capture (in this case, the sample occasions correspond to days), and `trap` is the integer trap
 5193 identity. The variable `trapid` will have to correspond to the row of a matrix containing
 5194 the trap coordinates - in this case the TDF file `wtraps` which we describe further below.

5195 Note that the information provided in this encounter data file `wcaps` does not represent
 5196 a completely informative summary of the data. For example, if no individuals were
 5197 captured in a certain trap or during a certain period, then this compact data format will
 5198 have no record. Thus we will need to know J , the number of traps, and K , the number of
 5199 sample occasions when reformatting this SCR data format into a 2-d encounter frequency
 5200 matrix or 3-d array. In addition, the encounter data file does not provide information
 5201 about which periods each trap was operated. This additional information is also necessary
 5202 as the trap-specific sample sizes must be passed to **BUGS** as data. We provide this
 5203 information along with trap coordinates, in the “trap deployment file” (TDF) which is
 5204 described below.

5205 For our purposes, we need to convert the `wcaps` file into the $n \times J$ array of binomial
 5206 encounter frequencies, although more general models might require an encounter-history
 5207 formulation of the model which requires a full 3-d array. To obtain our encounter frequency
 5208 matrix, we do this the hard way by first converting the encounter data file into a 3-d array
 5209 and then summarize to trap totals. We have a handy function `SCR23darray` which takes
 5210 the compact encounter data file, and converts it to a 3-d array, and then we use the **R**
 5211 function `apply` to summarize over the sample occasion dimension (by convention here,
 5212 this is the 2nd dimension). To apply this to the wolverine data in order to compute the
 5213 3-d array we do this:

```
5214 > y3d <- SCR23darray(wolverine$wcaps,wolverine$wtraps)
5215 > y <- apply(y3d,c(1,2),sum)
```

5216 See the help file for more information on `SCR23darray`. The 3-d array is necessary to
 5217 fit certain types of models (e.g., behavioral response) and this is why we sometimes will
 5218 require this maximally informative 3-d data format but, here, we analyze the summarized
 5219 data.

5220 **Trap Deployment File (TDF)** – The other important information needed to fit SCR
 5221 models is the “trap deployment file” (TDF) which provides additional information not

5222 contained in the encounter data file. The traps file has $K + 3$ columns. The first column is
 5223 assumed to be a trap identifier, columns 2 and 3 are the easting and northing coordinates
 5224 (assumed to be in a Euclidean coordinate system), and columns 4 to $K + 3$ are binary
 5225 indicators of whether each trap was operational during each sample occasion. The first 10
 5226 rows (out of 37) and 10 columns (out of 167) of the trap deployment file for the wolverine
 5227 data are shown as follows:

```
5228 > wolverine$wtraps[1:10,1:10]
5229
5230      Easting Northing 1 2 3 4 5 6 7 8
5231 1   632538  6316012 0 0 0 0 0 0 0 0
5232 2   634822  6316568 1 1 1 1 1 1 1 1
5233 3   638455  6309781 0 0 0 0 0 0 0 0
5234 4   634649  6320016 0 0 0 0 0 0 0 0
5235 5   637738  6313994 0 0 0 0 0 0 0 0
5236 6   625278  6318386 0 0 0 0 0 0 0 0
5237 7   631690  6325157 0 0 0 0 0 0 0 0
5238 8   632631  6316609 0 0 0 0 0 0 0 0
5239 9   631374  6331273 0 0 0 0 0 0 0 0
5240 10  634068  6328575 0 0 0 0 0 0 0 0
```

5241 This tells us that trap 2 was operated during occasions (days) 1-7 but the other traps
 5242 were not operational during those periods. It is extremely important to recognize that
 5243 each trap was operated for a variable period of time and thus the binomial “sample size”
 5244 is different for each, and this needs to be accounted for in the **BUGS** model specification.
 5245 To compute the vector of sample sizes K , and extract the trap locations, we do this:

```
5246 > traps <- wolverine$wtraps
5247 > traplocs <- traps[,1:2]
5248 > K <- apply(traps[,3:ncol(traps)],1,sum)
```

5249 This results in a matrix `traplocs` which contains the coordinates of each trap and a vector
 5250 `K` containing the number of days that each trap was operational. We now have all the
 5251 information required to fit a basic SCR model in **BUGS**.

5252 Summarizing the data for the wolverine study, we see that 21 unique individuals were
 5253 captured a total of 115 times. Most individuals were captured 1-6 times, with 4, 1, 4, 3, 1,
 5254 and 2 individuals captured 1-6 times, respectively. In addition, 1 individual was captured
 5255 each 8 and 14 times and 2 individuals each were captured 10 and 13 times. The number
 5256 of unique traps that captured a particular individual ranged from 1-6, with 5, 10, 3, 1, 1,
 5257 and 1 individual captured in each of 1 to 6 different traps, respectively, for a total of 50
 5258 unique wolverine-trap encounters. These numbers might be hard to get your mind around
 5259 whereas some tabular summary is often more convenient. For that it seems natural to
 5260 tabulate individuals by trap and total encounter frequencies. The spatial information in
 5261 SCR data is based on multi-trap captures, and so, it is informative to understand how
 5262 many unique traps each individual is captured in, and the total number of encounters.
 5263 For the wolverine data, we reproduce Table 1 from Royle et al. (2011b) as Table 5.3.

Table 5.3. Individual frequencies of capture for wolverines captured in camera traps in Southeast Alaska in 2008. Rows index unique traps of capture for each individual and columns represent total number of captures (e.g., we captured 4 individuals 1 time, necessarily in only 1 trap; we captured 3 individuals 3 times but in 2 different traps).

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

5.9.2 Fitting the model in WinBUGS

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

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

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

It remains to define the state-space \mathcal{S} . For this, we nested the trap array (Fig. 5.4) in a rectangular state-space extending 20 km beyond the traps in each cardinal direction. We scaled the coordinate system so that a unit distance was equal to 10 km, producing a rectangular state-space of dimension 9.88×10.5 units ($\text{area} = 10374 \text{ km}^2$) within which the trap array was nested. As a general rule, we recommend scaling the state-space so that it is defined near the origin $(x, y) = (0, 0)$. While the scaling of the coordinate system is theoretically irrelevant, a poorly scaled coordinate system can produce Markov chains that mix poorly. The buffer of the state space should be large enough so that individuals beyond the state-space boundary are not likely to be encountered (Sec. 5.3.1). To evaluate this, we fit models for various choices of a rectangular state-space based on buffers from 1.0 to 5.0 units (10 km to 50 km). In the **R** package **scrbook** we provide a function **wolvSCR0** which will fit model SCR0. For example, to fit the model in **WinBUGS** using data augmentation with $M = 300$ potential individuals, using 3 Markov chains each of 12000 total iterations, discarding the first 2000 as burn-in, we execute the following **R** commands:

```

5289 > library(scrbook)
5290 > data(wolverine)
5291 > traps <- wolverine$wtraps
5292 > y3d <- SCR23darray(wolverine$wcaps,wolverine$wtraps)
5293 > wolv <- wolvSCR0(y3d,traps,nb=2000,ni=12000,buffer=1,M=300)

```

Table 5.4. Posterior summaries of SCR model parameters for the wolverine camera trapping data from SE Alaska, using state-space buffers from 10 up to 50 km. Each analysis was based on 3 chains, 12000 iterations, 2000 burn-in, for a total of 30000 posterior samples.

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

Table 5.5. Posterior summaries of SCR model parameters for the wolverine camera trapping data from SE Alaska. The model was run with the trap array centered in a state-space with a 20 km rectangular buffer.

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

5294 The argument `buffer` determines the buffer size of the state-space in the scaled units
 5295 (i.e., 10 km). Note that this analysis takes between 1-2 hours on many machines (in 2013)
 5296 so we recommend testing it with lower values of M and fewer iterations. The posterior
 5297 summaries are shown in Table 5.9.2.

5298 5.9.3 Summary of the wolverine analysis

5299 We see that the estimated density is roughly consistent as we increase the state-space
 5300 buffer from 15 to 55 km. We do note that the data augmentation parameter ψ (and,
 5301 correspondingly, N) increase with the size of the state space in accordance with the deter-
 5302 ministic relationship $N = D * A$. However, density is more or less constant as we increase
 5303 the size of the state-space beyond a certain point. For the 10 km state-space buffer, we see
 5304 a slight effect on the posterior distribution of D because the state-space is not sufficiently
 5305 large. The full results from the analysis based on 20 km state-space buffer are given in
 5306 Table 5.5.

5307 Our point estimate of wolverine density from this study, using the posterior mean from
 5308 the state-space based on the 20 km buffer, is approximately 5.77 individuals/1000 km²

5309 with a 95% posterior interval of [3.86, 8.29]. Density is estimated imprecisely which might
 5310 not be surprising given the low sample size ($n = 21$ individuals!). This seems to be a
 5311 basic feature of carnivore studies although it should not (in our view) preclude the study
 5312 of their populations by capture-recapture nor attempts to estimate density or vital rates.

5313 It is worth thinking about this model, and these estimates, computed under a rect-
 5314 angular state space roughly centered over the trapping array (Fig. 5.4). Does it make
 5315 sense to define the state-space to include, for example, ocean? What are the possible
 5316 consequences of this? What can we do about it? There's no reason at all that the state
 5317 space has to be a regular polygon – we defined it as such here strictly for convenience and
 5318 for ease of implementation in **WinBUGS** where it enables us to specify the prior for the
 5319 activity centers as uniform priors for each coordinate. While it would be possible to define
 5320 a more realistic state-space using some general polygon GIS coverage, it might take some
 5321 effort to implement that in the **BUGS** language but it is not difficult to devise custom
 5322 MCMC algorithms to do that (see Chapt. 17). Alternatively, we recommend using a
 5323 discrete representation of the state-space – i.e., approximate \mathcal{S} by a grid of G points. We
 5324 discuss this in Sec. 5.10.

5325 5.9.4 Wolverine space usage

5326 The parameter α_1 is related to the home range radius (Sec. 5.4). For the Gaussian model
 5327 we interpret the scale parameter σ , related to α_1 by $\alpha_1 = 1/(2\sigma^2)$, as the radius of a
 5328 bivariate normal model of space usage. In this case $\sigma = 0.64$ standardized units (10 km),
 5329 which corresponds to $0.64 \times 10 = 6.4$ km. It can be argued then that 95% of space used
 5330 by an individual is within $6.4 \times \sqrt{5.99} = 15.66$ km of the home range center. The effective
 5331 “home range area” is then the area of this circle, which is $\pi \times 15.66^2 = 770.4$ km². Using
 5332 our handy function **hra** we do this:

```
5333 hra(pGauss1,parms=c(-2,1/(2*.64*.64)),xlim=c(-1,7),ylim=c(-1,7))
5334
5335 [1] 7.731408
```

5336 which is in units of 100 km², so 773.1. The difference in this case is due to numerical
 5337 approximation of our all-purpose tool **hra**. This home range size is relatively huge for
 5338 measured home ranges, which range between 100 and 535 km² (Whitman et al., 1986).

5339 Royle et al. (2011b) reported estimates for σ in the range 6.3 – 9.8 km depending on
 5340 the model, which isn't too different than here¹. However, these estimates are larger than
 5341 the typical home range sizes suggested in the literature. One possible explanation is that
 5342 if a wolverine is using traps as a way to get yummy chicken, so it's moving from trap to
 5343 trap instead of adhering to “normal” space usage patterns, then the implied home range
 5344 size might not be worth much biologically. Thus, interpretation of detection models in
 5345 terms of home range area depends on some additional context or assumptions, such as
 5346 that traps don't effect individual space usage patterns. As such, we caution against direct

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

5347 biological interpretations of home range area based on σ , although SCR models can be
 5348 extended to handle more general, non-Euclidean, patterns of space usage. See Chaps. 12
 5349 and 13.

5350 We can calibrate the desired size of the state-space by looking at the estimated home
 5351 range radius of the species. We should target a buffer of width 2 to $3 \times \sigma$ in order that
 5352 the probability of encountering an individual is very close to 0 beyond the prescribed
 5353 state-space. Essentially, by specifying a state-space, we're setting $p = 0$ for individuals
 5354 beyond the prescribed state-space. For the wolverine data, with σ in the range of 6-9 km,
 5355 a state-space buffer of 20 km is sufficiently large.

5.10 USING A DISCRETE HABITAT MASK

5356 The SCR model developed previously in this chapter assumes that individual activity
 5357 centers are distributed uniformly over the prescribed state-space. Clearly this will not
 5358 always be a reasonable assumption. In Chapt. 11, we develop models that allow explicitly
 5359 for non-uniformity of the activity centers by modeling covariate effects on density. A
 5360 simplistic method of affecting the distribution of activity centers, which we address here,
 5361 is to modify the shape and organization of the state-space explicitly. For example, we
 5362 might be able to classify the state-space into distinct blocks of habitat and non-habitat.
 5363 In that case we can remove the non-habitat from the state-space and assume uniformity of
 5364 the activity centers over the remaining portions judged to be suitable habitat. There are
 5365 several ways to approach this: We can use a grid of points to represent the state-space, i.e.,
 5366 by the set of coordinates s_1, \dots, s_G , and assign equal probabilities to each possible value.
 5367 Alternatively, we can retain the continuous formulation of the state-space but attempt
 5368 to describe constraints analytically, or we can use polygon clipping methods to enforce
 5369 constraints on the state-space in the MCMC analysis. We focus here on the formulation of
 5370 the basic SCR model in terms of a discrete state-space but in Chapt. 17 we demonstrate
 5371 the latter approach based on using polygon operations to define an irregular state-space.
 5372 Use of a discrete state-space can be computationally expensive in **WinBUGS**. That said,
 5373 it isn't too difficult to perform the MCMC calculations in **R** (discussed in Chapt. 17).
 5374 The **R** package **SPACECAP** (Gopalaswamy et al., 2012a) arose from the **R** implementation
 5375 of the SCR model in Royle et al. (2009a).

5376 While clipping out non-habitat seems like a good idea, we think investigators should
 5377 go about this very cautiously. We might prefer to do it when non-habitat represents a
 5378 clear-cut restriction on the state-space such as a reserve boundary or a lake, ocean or
 5379 river. But, having the capability to do this also causes people to start defining "habitat"
 5380 vs. "non-habitat" based on their understanding of the system whereas it can't be known
 5381 whether the animal being studied has the same understanding. Moreover, differentiating
 5382 the landscape by habitat or habitat quality must affect the geometry and morphology of
 5383 home ranges (see Chapt. 13) much more so than the plausible locations of activity centers.
 5384 That is, a home range centroid could, in actual fact, occur in a shopping mall parking lot
 5385 if there is pretty good habitat around the shopping mall, so there is probably no sense
 5386 preclude it as the location for an activity center. It would generally be better to include
 5387 some definition of habitat quality in the model for the detection probability (Royle et al.,
 5388 2012a) which we address in Chaps. 12 and 13.

5.10.1 Evaluation of coarseness of habitat mask

5389 The coarseness of the state-space should not really have much of an effect on estimates
 5390 if the grain is sufficiently fine relative to typical animal home range sizes. Why is this?
 5391 We have two analogies that can help us understand. First is the relationship to model
 5392 M_h . As noted in Sec. 5.3.2 above, we can think about SCR models as a type of finite
 5393 mixture (Norris and Pollock, 1996; Pledger, 2004) where we are fortunate to be able to
 5394 obtain direct information about which group individuals belong to (group being location
 5395 of activity center). In the standard finite mixture models we typically find that a small
 5396 number of groups (e.g., 2 or 3 at the most) can explain high levels of heterogeneity and
 5397 are adequate for most data sets of small to moderate sample sizes. We therefore expect a
 5398 similar effect in SCR models when we discretize the state-space. We can also think about
 5399 discretizing the state-space as being related to numerical integration where we find (see
 5400 Chapt. 6) that we don't need a very fine grid of support points to evaluate the integral to
 5401 a reasonable level of accuracy. We demonstrate this here by reanalyzing simulated data
 5402 using a state-space defined by a different number of support points. We provide an **R**
 5403 script called **SCR0bayesDss** in the **R** package **scrbook**. We note that for this comparison
 5404 we generated the actual activity centers as a continuous random variable and thus the
 5405 discrete state-space is, strictly speaking, an approximation to truth. That said, we regard
 5406 all state-space specifications as approximations to truth in the sense that they represent
 5407 a component of the SCR model.

5408 As with our **R** function **SCR0bayes**, the modification **SCR0bayesDss** will use either
 5409 **WinBUGS** or **JAGS**. In addition, it requires a grid resolution argument (**ng**) which
 5410 is the dimension of 1 side of a square state-space. To execute this function we do, for
 5411 example:

```
5413 > library(scrbook)
5414 > data <- simSCR0(discard0=TRUE,rnd=2013)    # Generate data set
5415
5416 # Run with JAGS
5417 > out1 <- SCR0bayesDss(data,ng=8,M=200,engine="jags",ni=2000,nb=1000)
5418
5419 # Run with WinBUGS
5420 > out2 <- SCR0bayesDss(data,ng=8,M=200,engine="winbugs",ni=2000,nb=1000)
```

5421 We fit this model to the same simulated data set for 6×6 , 9×9 , 12×12 , 15×15
 5422 state-space grids. For **WinBUGS**, we used 3 chains of 5000 total length with 1000 burn-in,
 5423 which yields 12000 total posterior samples. Summary results are shown in Table 5.6.
 5424 The results are broadly consistent except for the 6×6 case. We see that the run time
 5425 increases with the size of the state-space grid (not unexpected), such that we imagine it
 5426 would be impractical to run models with more than a few hundred state-space grid points.
 5427 We found (not shown here) that the runtime of **JAGS** is much faster and, furthermore,
 5428 relatively *constant* as we increase the grid size. We suspect that **WinBUGS** is evaluating
 5429 the full-conditional for each activity center at all G possible values whereas it may be
 5430 that **JAGS** is evaluating the full-conditional only at a subset of values or perhaps using
 5431 previous calculations more effectively. While this might suggest that one should always
 5432 use **JAGS** for this analysis, we found in our analysis of the wolverine (next section) that
 5433 **JAGS** could be extremely sensitive to starting values, producing MCMC algorithms that

Table 5.6. Comparison of the effect of state-space grid coarseness on estimates of N for a simulated data set. Posterior summaries and run time are given. Results obtained using **WinBUGS** run from **R2WinBUGS**.

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

5434 often simply do not work for some problems, so be careful when using **JAGS**. To improve
 5435 its performance, always start the latent activity centers at values near where individuals
 5436 were captured. The performance of either should improve if we compute the full distance
 5437 matrix outside of **BUGS** and pass it as data, although we haven't fully evaluated this
 5438 approach.

5439 5.10.2 Analysis of the wolverine camera trapping data

5440 We reanalyzed the wolverine data using discrete state-space grids with points spaced by
 5441 2, 4 and 8 km (see Fig. 5.5). These were constructed from a 40 km buffered state-space,
 5442 and deleting the points over water (see Royle et al., 2011b). Our interest in doing this
 5443 was to evaluate the relative influence of grid resolution on estimated density because the
 5444 coarser grids will be more efficient from a computational stand-point and so we would
 5445 prefer to use them, but only if there is no strong influence on estimated density. The
 5446 posterior summaries for the 3 habitat grids are given in Table 5.7. We see that the
 5447 density estimates are quite a bit larger than obtained in our analysis (Table 5.9.2) based
 5448 on a rectangular, continuous state-space. We also see that there are slight differences
 5449 depending on the resolution of the state-space grid. Interestingly, the effectiveness of the
 5450 MCMC algorithms, as measured by effective sample size (**n.eff**) is pretty remarkably
 5451 different. Furthermore, the finest grid resolution (2 km spacing) took about 6 days to run
 5452 and thus it would not be practical for large problems or with many models.

5.11 SUMMARIZING DENSITY AND ACTIVITY CENTER LOCATIONS

5453 One of the most useful aspects of SCR models is that they are parameterized in terms of
 5454 individual locations – i.e., *where* each individual lives – and, thus, we can compute many
 5455 useful and interesting summaries of the activity centers using output from an MCMC sim-
 5456 ulation, including maps of density (the number of activity centers per unit area), estimates
 5457 of N for any well-defined polygon, or estimates of where the activity centers for specific
 5458 individuals reside. In Bayesian analysis by MCMC, obtaining such summaries entails no
 5459 added calculations, because we need only post-process the output for the individual ac-
 5460 tivity centers to obtain the desired summaries. We demonstrate that in this section. Note
 5461 that you have to be sure to retain the MCMC history for the **s** variables and also the data
 5462 augmentation variables z in order to do the following analyses.

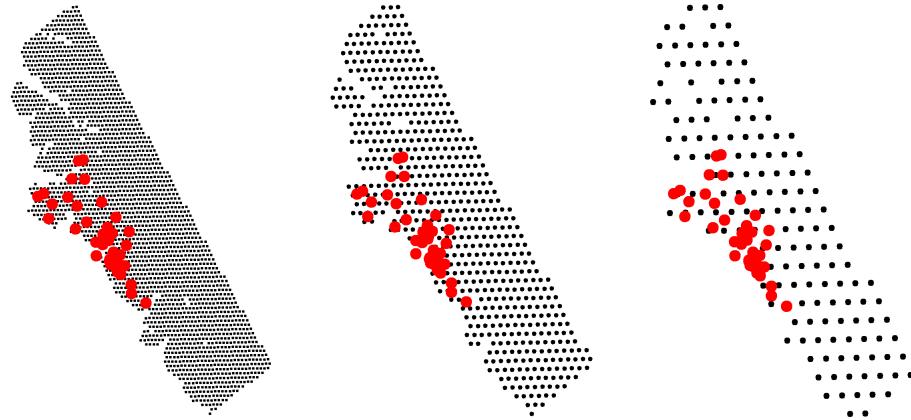


Figure 5.5. Three habitat mask grids used in the comparison of the effect of pixel size on the estimated density surface of wolverines. The 3 cases are 2 (left), 4 (center) and 8 (right) km spacing of state-space points, extending 40 km from the vicinity of the trap array.

5463 5.11.1 Constructing density maps

5464 Because SCR models are spatially-explicit, it is natural to want to summarize the results
 5465 of fitting a model by producing a map of density. Using Bayesian analysis by MCMC, it is
 5466 most easy to make a map of *realized* density. We can do this by tallying up the number of
 5467 activity centers \mathbf{s}_i in pixels of arbitrary size and then producing a nice multi-color spatial
 5468 plot of the result. Specifically, let $B(\mathbf{x})$ indicate a pixel centered at \mathbf{x} then

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

5469 (here, $I(arg)$ is the indicator function which evaluates to 1 if arg is true, and 0 otherwise)
 5470 is the population size of pixel $B(\mathbf{x})$, and $D(\mathbf{x}) = N(\mathbf{x})/\|B(\mathbf{x})\|$ is the local density. Note
 5471 that these $N(\mathbf{x})$ parameter are just “derived parameters” as we normally obtain from
 5472 posterior output using the appropriate Monte Carlo average (see Chapt. 3).

5473 One thing to be careful about, in the context of models in which N is unknown, is that,
 5474 for each MCMC iteration m , we only tabulate those activity centers which correspond to
 5475 individuals in the sampled population, i.e., for which the data augmentation variable
 5476 $z_i = 1$. In this case, we take all of the output for MCMC iterations $m = 1, 2, \dots, \text{niter}$
 5477 and compute this summary:

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

Table 5.7. Posterior summaries for the wolverine camera trapping data, using model SCR0, with a Gaussian hazard encounter probability model, and a discrete habitat mask of 3 different resolutions: 2, 4 and 8 km. Parameters are λ_0 = baseline encounter rate, $p_0 = 1 - \exp(-\lambda_0)$, σ is the scale parameter of the Gaussian kernel, ψ is the data augmentation parameter, N and D are population size and density, respectively. Models fitted using **WinBUGS**, 3 chains, each with 11000 iterations (first 1000 discarded) producing 30000 posterior samples.

2 km spacing										
	Mean	SD	2.5%	25%	50%	75%	97.5%	Rhat	n.eff	
N	86.56	16.94	57.00	75.00	85.00	97.00	124.00	1.00	510	
D	8.78	1.72	5.78	7.60	8.62	9.83	12.57	1.00	510	
λ_0	0.05	0.01	0.04	0.04	0.05	0.06	0.07	1.01	320	
p_0	0.05	0.01	0.03	0.04	0.05	0.05	0.06	1.01	320	
σ	0.62	0.05	0.54	0.59	0.62	0.65	0.73	1.01	160	
ψ	0.43	0.09	0.27	0.37	0.43	0.49	0.63	1.00	560	
4 km spacing										
	Mean	SD	2.5%	25%	50%	75%	97.5%	Rhat	n.eff	
N	89.25	17.44	59.00	77.00	88.00	100.00	127.00	1	1100	
D	9.01	1.76	5.96	7.77	8.88	10.10	12.82	1	1100	
λ_0	0.05	0.01	0.04	0.05	0.05	0.06	0.07	1	2500	
p_0	0.05	0.01	0.03	0.04	0.05	0.05	0.07	1	2500	
σ	0.61	0.04	0.53	0.58	0.61	0.64	0.71	1	1600	
ψ	0.45	0.09	0.28	0.38	0.44	0.50	0.64	1	1300	
8 km spacing										
	Mean	SD	2.5%	25%	50%	75%	97.5%	Rhat	n.eff	
N	83.18	16.14	56.00	72.00	82.00	93.00	119.00	1.00	700	
D	8.28	1.61	5.57	7.17	8.16	9.26	11.84	1.00	700	
λ_0	0.05	0.01	0.03	0.04	0.05	0.05	0.06	1.00	560	
p_0	0.05	0.01	0.03	0.04	0.04	0.05	0.06	1.00	560	
σ	0.68	0.05	0.59	0.64	0.67	0.71	0.77	1.01	220	
ψ	0.42	0.09	0.26	0.36	0.41	0.47	0.61	1.00	940	

5478 Thus, $N(\mathbf{x}, 1), N(\mathbf{x}, 2), \dots$, is the Markov chain for parameter $N(\mathbf{x})$. In what follows we
 5479 will provide a set of **R** commands for doing this calculation and making a basic image
 5480 plot from the MCMC output.

5481 **Step 1:** Define the center points of each pixel $B(\mathbf{x})$, or point at which local density will
 5482 be estimated:

```
5483 > xg <- seq(xlim[1], xlim[2], , 50)
  5484 > yg <- seq(ylim[1], ylim[2], , 50)
```

5485 **Step 2:** Extract the MCMC histories for the activity centers and the data augmentation
 5486 variables. Note that these are each $N \times \text{niter}$ matrices. Here we do this assuming that
 5487 **WinBUGS** was run producing the **R** object named **out**:

```
5488 > Sxout <- out$sims.list$s[, , 1]
  5489 > Syout <- out$sims.list$s[, , 2]
  5490 > z <- out$sims.list$z
```

5491 **Step 3:** We associate each coordinate with the proper pixel using the **R** command `cut()`.
 5492 Note that we keep only the activity centers for which $z = 1$ (i.e., individuals that belong
 5493 to the population of size N):

```
5494 > Sxout <- cut(Sxout[z==1], breaks=xg, include.lowest=TRUE)
5495 > Syout <- cut(Syout[z==1], breaks=yg, include.lowest=TRUE)
```

5496 **Step 4:** Use the `table()` command to tally up how many activity centers are in each
 5497 $B(\mathbf{x})$:

```
5498 > Dn <- table(Sxout, Syout)
```

5499 **Step 5:** Use the `image()` command to display the resulting matrix.

```
5500 > image(xg, yg, Dn/nrow(z), col=topo.colors(10))
```

5501 It is worth emphasizing here that density maps will not usually appear uniform despite
 5502 that we have assumed that activity centers are uniformly distributed. This is because
 5503 the observed encounters of individuals provide direct information about the location of
 5504 the $i = 1, 2, \dots, n$ activity centers and thus their “estimated” locations will be affected
 5505 by the observations. In a limiting sense, were we to sample space intensely enough,
 5506 every individual would be captured a number of times and we would have considerable
 5507 information about all N point locations. Consequently, the uniform prior would have
 5508 almost no influence at all on the estimated density surface in this limiting situation.
 5509 Thus, in practice, the influence of the uniformity assumption decreases as the fraction of
 5510 the population encountered, and the total number of encounters per individual, increases.

5511 **On the non-intuitiveness of `image()`** – the **R** function `image()`, invoked for a
 5512 matrix M by `image(M)`, might not be very intuitive to some – it plots $M[1, 1]$ in the lower
 5513 left corner. If you want $M[]$ to be plotted “as you look at it” then $M[1, 1]$ should be in the
 5514 upper left corner. We have a function `rot()` which does that. If you do `image(rot(M))`
 5515 then it puts it on the monitor as if it was a map you were looking at. You can always
 5516 specify the x - and y -labels explicitly as we did above.

5517 **Spatial dot plots** – A cruder version of the density map can be made using our
 5518 “spatial dot map” function `spatial.plot` (in `scrbook`). This function requires, as input,
 5519 point locations and the value to be displayed. A simplified version of this function is as
 5520 follows:

```
5521 > spatial.plot <- function(x,y){
  5522   nc <- as.numeric(cut(y,20))
  5523   plot(x,pch=" ")
  5524   points(x,pch=20,col=topo.colors(20)[nc],cex=2)
  5525   image.scale(y,col=topo.colors(20))
  5526 }
  5527 #
  5528 # To execute the function do this:
  5529 #
  5530 > spatial.plot(cbind(xg,yg), Dn/nrow(z))
```

5.11.2 Example: Wolverine density map

5531 We return to the wolverine study which took place in 2008 in SE Alaska (Fig. 5.4) and
 5532 we produce a density map of wolverines from that analysis. We include the function
 5533 **SCRdensity** which requires a specific data structure as shown below. In particular, we
 5534 have to package up the MCMC history for the activity centers and the data augmentation
 5535 variables z into a list. This also requires that we add those variables to the parameters-
 5536 to-be-monitored list when we pass things to **BUGS**.

5537 We used the posterior output from the wolverine model fitted previously to compute
 5538 a relatively coarse version of a density map, using 100 pixels in a 10×10 grid (Fig. 5.6
 5539 top panel) and using 900 pixels arranged in a 30×30 grid (Fig. 5.6 lower panel) for a
 5540 fine-scale map. The **R** commands for producing such a plot (for a short MCMC run) are
 5541 as follows:

```
5543 > library(scrbook)
5544 > data(wolverine)
5545 > traps <- wolverine$wtraps
5546 > y3d <- SCR23darray(wolverine$wcaps,wolverine$wtraps)
5547
5548 # This takes 341 seconds on a standard CPU circa 2011
5549 > out <- wolvSCRO(y3d,traps,nb=1000,ni=2000,buffer=1,M=100,keepz=TRUE)
5550
5551 > Sx <- out$sims.list$s[,,1]
5552 > Sy <- out$sims.list$s[,,2]
5553 > z <- out$sims.list$z
5554 > obj <- list(Sx=Sx,Sy=Sy,z=z)
5555 > tmp <- SCRdensity(obj,nx=10,ny=10,scalein=100,scaleout=100)
```

5556 In these figures density is expressed in units of individuals per 100 km^2 , while the area of
 5557 the pixels is about 103.7 km^2 and 11.5 km^2 , respectively. That calculation is based on:

```
5558 > total.area <- (ylim[2]-ylim[1])*(xlim[2]-xlim[1])*100
5559 > total.area/(10*10)
5560 [1] 103.7427
5561 > total.area/(30*30)
5562 [1] 11.52697
```

5563 A couple of things are worth noting: First is that as we move away from “where the
 5564 data live” – away from the trap array – we see that the density approaches the mean
 5565 density. This is a property of the estimator as long as the detection function decreases
 5566 sufficiently rapidly as a function of distance. Relatedly, it is also a property of statistical
 5567 smoothers such as splines, kernel smoothers, and regression smoothers – predictions tend
 5568 toward the global mean as the influence of data diminishes. Another way to think of it is
 5569 that it is a consequence of the prior, which imposes uniformity, and as you get far away
 5570 from the data, the predictions tend to the expected constant density under the prior.
 5571 Another thing to note about this map is that density is not 0 over water (although the
 5572 coastline is not shown). This might be perplexing to some who are fairly certain that
 5573 wolverines do not like water. However, there is nothing about the model that recognizes

5574 water from non-water and so the model predicts over water *as if* it were habitat similar to
 5575 that within which the array is nested. But, all of this is OK as far as estimating density
 5576 goes and, furthermore, we can compute valid estimates of N over any well-defined region
 5577 which presumably wouldn't include water if we so wished. Alternatively, areas covered by
 5578 water could be masked out, which we discuss in the next section.

5579 5.11.3 Predicting where an individual lives

5580 The density maps in the previous section show the expected number of individuals per
 5581 unit area. A closely related problem is that of producing a map of the probable location
 5582 of a specific individual's activity center. For any observed encounter history, we can easily
 5583 generate a posterior distribution of \mathbf{s}_i for individual i . In addition, for an individual that
 5584 is *not* captured, we can use the MCMC output to produce a corresponding plot of where
 5585 such an individual might live, say \mathbf{s}_{n+1} . Obviously, all such uncaptured individuals (for
 5586 $i = n + 1, \dots, N$) should have the same posterior distribution. To illustrate, we show the
 5587 posterior distribution of \mathbf{s}_1 , the activity center for the individual labeled 1 in the data
 5588 set, in Fig. 5.7. This individual was captured a single time at trap 30 which is circled
 5589 in Fig. 5.7. We see that the posterior distribution is affected by traps of capture *and*
 5590 traps of non-capture in fairly intuitive ways. In particular, because there are other traps
 5591 in close proximity to trap 30, in which individual 1 was *not* captured, the model pushes
 5592 its activity center away from the trap array. The help file for `SCRdensity` shows how to
 5593 calculate Fig. 5.7.

5.12 EFFECTIVE SAMPLE AREA

5594 One of the key issues in using ordinary capture recapture models which we've brought up
 5595 over and over again is this issue that the area which is sampled by a trapping array is
 5596 unknown – in other words, the N that is estimated by capture-recapture models does not
 5597 have an explicit region of space associated with it. Classically this has been addressed in
 5598 the ad hoc way of prescribing an area that contains the trap array, usually by adding a
 5599 buffer of some width, which is not estimated as part of the capture-recapture model. In
 5600 SCR models we avoid the problem of not having an explicit linkage between N and “area”,
 5601 by prescribing explicitly the area within which the underlying point process is defined – the
 5602 state-space of the point process. This state-space is *not* the effective sample (or sampled)
 5603 area (ESA) – it is desirable that it be somewhat larger than the ESA, whatever that may
 5604 be, in the sense that individuals at the edge of the state-space have no probability of being
 5605 captured, but as part of the SCR model we don't need to try to estimate or otherwise
 5606 characterize the ESA explicitly.

5607 However, it is possible to provide a characterization of effective sampled area under
 5608 any SCR model. This is directly analogous to the calculation of “effective strip width” in
 5609 distance sampling (Buckland et al., 2001; Borchers et al., 2002). The conceptual definition
 5610 of ESA follows from equating density to “apparent density” – ESA is the magic number
 5611 that satisfies that equivalence:

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

5612 In other words, the ratio of N to the area of the state-space should be equal to the ratio
 5613 of the observed sample size n to this number ESA. Both of these should equal density.

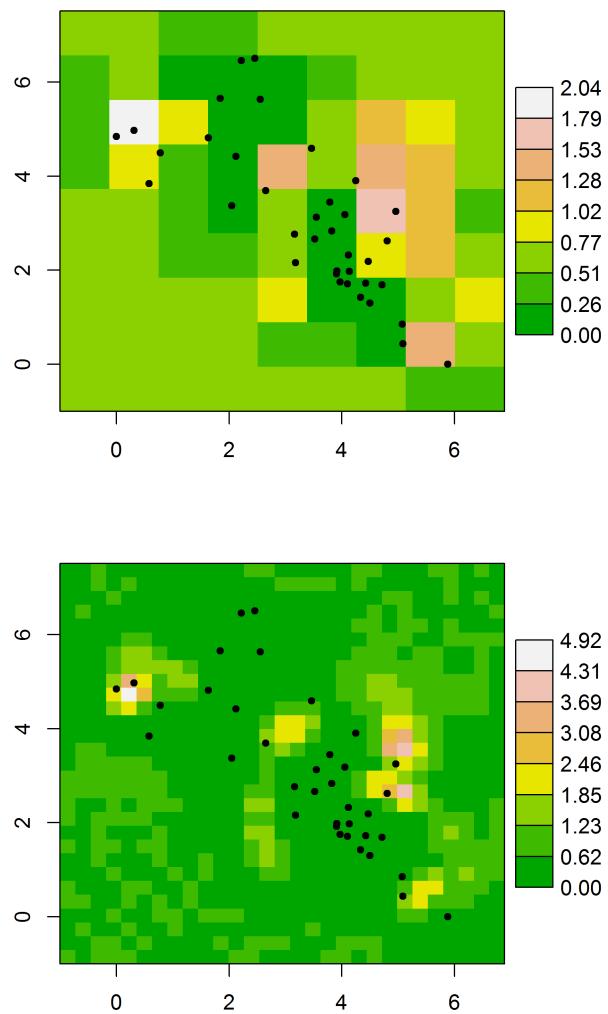


Figure 5.6. Density of wolverines (individuals per 100 km²) in SE Alaska in 2007 based on model SCR0. Map grid cells are about 103.7 km² (top panel) and 11.5 km² (bottom panel) in area. Dots are the trap locations.

5614 So, to compute ESA for a model, we substitute $\mathbb{E}(n)$ for n into the above equation, and
 5615 solve for ESA , to get:

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

5616 Our following development assumes that D is constant, but these calculations can be
 5617 generalized to allow for D to vary spatially. Imagine our habitat mask for the wolverine
 5618 data, or the bins we just used to produce a density map, then we can write $\mathbb{E}(n)$ according
 5619 to

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

5620 where if we prefer to think of this more conceptually we could replace the summation with
 5621 an integration (which, in practice, we would just replace with a summation, and so we
 5622 just begin there). In this expression note that $\mathbb{E}(N(\mathbf{s}))$ is the expected population size at
 5623 pixel \mathbf{s} which is the density times the area of the pixel, i.e., $\mathbb{E}(N(\mathbf{s})) = D \times a$. Therefore

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

5624 and (plugging this into the expression above for ESA)

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

5625 We see that D cancels and we have $ESA = a \times \sum_s \Pr(\text{encounter}|\mathbf{s})$ So what you have to
 5626 do here is substitute in $\Pr(\text{encounter}|\mathbf{s})$ and just sum them up over all pixels. For the
 5627 Bernoulli model of model SCR0

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

5628 with slight modifications when encounter probability depends on covariates. Thus,

$$ESA = a \sum_s 1 - (1 - p(\mathbf{s}))^K \tag{5.12.1}$$

5629 Clearly the calculation of ESA is affected by the use of a habitat mask, because the
 5630 summation in Eq. 5.12.1 only occurs over pixels that define the state-space.

5631 For the wolverine camera trapping data, we used the 2×2 km habitat mask and the
 5632 posterior means of p_0 and σ (see Sec. 5.10.2) to compute the probability of encounter for
 5633 each \mathbf{s} of the mask points. The result is shown graphically in Fig. 5.8. The ESA is the
 5634 sum of the values plotted in that figure multiplied by 4, the area of each pixel. For the
 5635 wolverine study, the result is 2507.152 km^2 . We note that the probability of encounter
 5636 declines rapidly to 0 as we move away from the periphery of the camera traps, indicating
 5637 the state-space constructed from a 40 km buffered trap array was indeed sufficient for the
 5638 analysis of these data. An **R** script for producing this figure is in the `wolvESA` function of
 5639 the `scrbook` package.

5.13 SUMMARY AND OUTLOOK

5640 In this chapter, we introduced the simplest SCR model – “model SCR0” – which is an ordi-
 5641 nary capture-recapture model like model M_0 , but augmented with a set of latent individual

5642 effects, s_i , which relate encounter probability to some sense of individual location using a
5643 covariate, “distance”, from s_i to each trap location. Thus, individuals in close proximity
5644 to a trap will have a higher probability of encounter, and *vice versa*. The explicit modeling
5645 of individual locations and distance in this fashion resolves classical problems related to
5646 estimating density: unknown sample area, and heterogeneous encounter probability due
5647 to variable exposure to traps.

5648 SCR models are closely related to classical individual covariate models (“model M_x ”,
5649 as introduced in Chapt. 4), but with imperfect information about the individual covari-
5650 ate. Therefore, they are also not too dissimilar from standard GLMMs used throughout
5651 statistics and, as a result, we find that they are easy to analyze using standard MCMC
5652 methods encased in black boxes such as **WinBUGS** or **JAGS**. We will also see that they
5653 are easy to analyze using likelihood methods, which we address in Chapt. 6.

5654 Formal consideration of the collection of individual locations (s_1, \dots, s_N) is funda-
5655 mental to all models considered in this book. In statistical terminology, we think of the
5656 collection of points $\{s_i\}$ as a realization of a point process. Because SCR models formally
5657 link individual encounter history data to an underlying point process, we can obtain for-
5658 mal inferences about the point process. For example, we showed how to produce a density
5659 map (Fig. 5.6), or even a probability map for an individual’s home range center (Fig.
5660 5.7). We can also use SCR models as the basis for doing more traditional point process
5661 analyses, such as testing for “complete spatial randomness” (CSR) (see Chapt. 8), and
5662 computing other point process summaries (Illian et al., 2008).

5663 Part of the promise, and ongoing challenge, of SCR models is to develop models that
5664 reflect interesting biological processes, for example interactions among points or temporal
5665 dynamics in point locations. In this chapter we considered the simplest possible point
5666 process model in which points are independent and uniformly (“randomly”) distributed
5667 over space. Despite the simplicity of this model, it should suffice in many applications of
5668 SCR models, although we do address generalizations in later chapters. Moreover, even
5669 though the *prior* distribution on the point locations is uniform, the realized pattern may
5670 deviate markedly from uniformity as the observed encounter data provide information to
5671 impart deviations from uniformity. Thus, estimated density maps will typically appear
5672 distinctly non-uniform (as we saw in the wolverine example). In applications of the basic
5673 SCR model, we find that this simple *a priori* model can effectively reflect or adapt to
5674 complex realizations of the underlying point process. For example, if individuals are
5675 highly territorial then the data should indicate this in the form of individuals not being
5676 encountered in the same trap – the resulting posterior distribution of point locations should
5677 therefore reflect non-independence. Obviously the complexity of posterior estimates of the
5678 point pattern will depend on the quantity of data, both number of individuals and captures
5679 per individual. Because the point process is such an integral component of SCR models,
5680 the state-space of the point process plays an important role in developing SCR models.
5681 As we emphasized in this chapter, the state-space is part of the model. It can have an
5682 influence on parameter estimates and other inferences, such as model selection (see chapter
5683 8).

5684 One concept we introduced in this chapter, which has not been discussed much in
5685 the literature on SCR models, is the manner in which the encounter probability model
5686 relates to a model of space usage by individuals. The standard SCR models of encounter
5687 probability can all be motivated as simplistic models of space usage and movement, in

5688 which individuals make random use decisions from a probability distribution proportional
5689 to the encounter probability model. This both clarifies the simplicity of the underlying
5690 model of space usage and also suggests a direct extension to produce more realistic models,
5691 which we discuss in Chapt. 13. We consider some other important extensions of the basic
5692 SCR model in later chapters. For example, we consider models that include covariates that
5693 vary by individual, trap, or over time (Chapt. 7), spatial covariates on density (Chapt.
5694 11), open populations (Chapt. 16), and methods for model assessment and selection
5695 (Chapt. 8) among other topics. We also consider technical details of maximum likelihood
5696 (Chapt. 6) and Bayesian (Chapt. 17) estimation, so that the interested reader can develop
5697 or extend methods to suit their own needs.

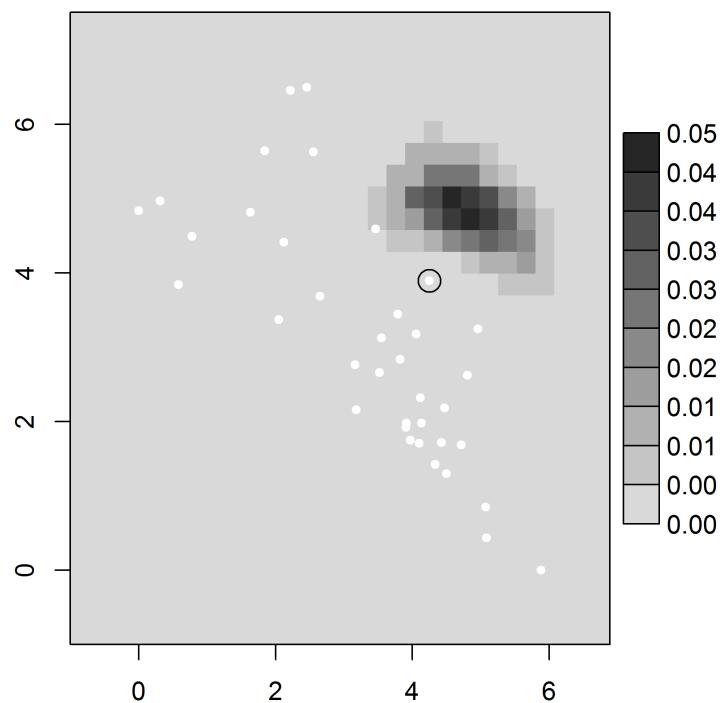


Figure 5.7. Posterior probability distribution of s_1 , the activity center for individual 1 in the wolverine data set. This individual was captured a single time in one trap (trap 30) which is circled. White dots are trap locations.

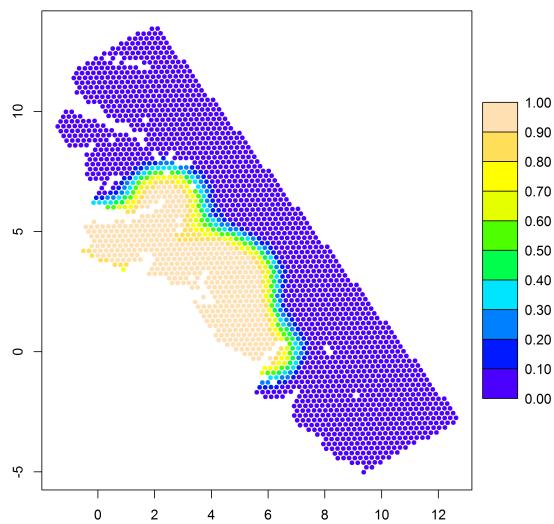


Figure 5.8. Probability of encounter used in computing effective sampled area for the wolverine camera trapping array, using the parameter estimates (posterior means) for the 2×2 km habitat mask.

5698

5699

5700

5701

6

LIKELIHOOD ANALYSIS OF SPATIAL CAPTURE-RECAPTURE MODELS

5702 We have so far mainly focused on Bayesian analysis of spatial capture-recapture models.
5703 And, in the previous chapters we learned how to fit some basic spatial capture-recapture
5704 models using a Bayesian formulation of the models analyzed in **BUGS** engines including
5705 **WinBUGS** and **JAGS**. Despite our focus on Bayesian analysis, it is instructive to de-
5706 velop the basic concepts and ideas behind classical analysis based on likelihood methods
5707 and frequentist inference for SCR models. We recognized earlier (Chapt. 5) that SCR
5708 models are versions of binomial (or other) GLMs, but with random effects (i.e., GLMMs).
5709 Throughout statistics, such models are routinely analyzed by likelihood methods. In par-
5710 ticular, likelihood analysis is based on the integrated or marginal likelihood in which the
5711 random effects are removed, by integration, from the conditional-on-s likelihood (s being
5712 the individual activity center). This has been the approach taken by Borchers and Ef-
5713 ford (2008); Dawson and Efford (2009) and related papers. Therefore, in this chapter, we
5714 provide some conceptual and technical foundation for likelihood-based analysis of spatial
5715 capture-recapture models.

5716 We will show here that it is straightforward to compute the maximum likelihood esti-
5717 mates (MLE) for SCR models by integrated likelihood. We develop the MLE framework
5718 using **R**, and we also provide a basic introduction to the **R** package **secr** (Efford, 2011)
5719 which does likelihood analysis of SCR models (see also the stand-alone program **DEN-**
5720 **SITY** (Efford et al., 2004)). To set the context for likelihood analysis of SCR models,
5721 we first analyze the SCR model when N is known because, in that case, analysis is no
5722 different at all than a standard GLMM. We generalize the model to allow for unknown N
5723 using both conventional ideas based on the “full likelihood” (e.g., Borchers et al., 2002)
5724 and also using a formulation based on data augmentation. We obtain the MLEs for the
5725 SCR model from the wolverine camera trapping study (Magoun et al., 2011) analyzed in
5726 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:

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

5756 We emphasize that two independence assumptions are explicit in this development: independence of trap-specific encounters within individuals and also independence among
 5757 individuals. In particular, this would only be valid when individuals are not physically
 5758 restrained or removed upon capture, and when traps do not “fill up.”

5759 The key operation for computing the likelihood is solving a 2-dimensional integration
 5760 problem. There are some general purpose **R** packages that implement a number of multi-
 5761 dimensional integration routines including **adapt** (Genz et al., 2007) and **R2cuba** (Hahn
 5762 et al., 2010). In practice, we won’t rely on these extraneous **R** packages (except see Chapt.
 5763 11 for an application of **R2cuba**) but instead will use perhaps less efficient methods in which
 5764 we replace the integral with a summation over an equal area mesh of points on the state-
 5765 space \mathcal{S} and explicitly evaluate the integrand at each point. We invoke the rectangular
 5766 rule for integration here¹ in which we evaluate the integrand on a regular grid of points
 5767 of equal area and compute the average of the integrand over that grid of points. Let
 5768 $u = 1, 2, \dots, nG$ index a grid of nG points, \mathbf{s}_u , where the area of grid cells is constant, say
 5769 A . In this case, the integrand, i.e., the marginal pmf of \mathbf{y}_i , is approximated by

$$[\mathbf{y}_i | \boldsymbol{\alpha}] = \frac{1}{nG} \sum_{u=1}^{nG} [\mathbf{y}_i | \mathbf{s}_u, \boldsymbol{\alpha}] \quad (6.1.2)$$

5770 This is a specific case of the general expression that could be used for approximating
 5771 the integral for any arbitrary distribution $[\mathbf{s}]$. The general case is

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

5772 Under the uniformity assumption, $[\mathbf{s}] = 1/A(\mathcal{S})$ and thus the grid-cell area cancels in the
 5773 above expression to yield Eq. 6.1.2. The rectangular rule for integration can be seen as
 5774 an application of the Law of Total Probability for a discrete random variable \mathbf{s} , having
 5775 nG unique values with equal probabilities $1/nG$.

5777 6.1.1 Implementation (simulated data)

5778 Here we will illustrate how to carry out this integration and optimization based on the
 5779 integrated likelihood using simulated data (i.e., see Sec. 5.5). Using **simSCR0** we simulate
 5780 data for 100 individuals and an array of 25 traps laid out in a 5×5 grid of traps having unit
 5781 spacing. The specific encounter model is the Gaussian model. The 100 activity centers
 5782 were simulated on a state-space defined by an 8×8 square within which the trap array was
 5783 centered (thus the trap array is buffered by 2 units). Therefore, the density of individuals
 5784 in this system is fixed at 100/64. In the following set of **R** commands we generate the
 5785 data and then harvest the required data objects:

```
5786 ## simulate a complete data set (perfect detection)
5787 > data <- simSCR0(discard0=FALSE, rnd=2013)
5788 ## extract the objects that we need for analysis
5789 > y <- data$Y
```

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

```

5790 > traplocs <- data$traplocs
5791 > nind <- nrow(y) ## in this case nind=N
5792 > J <- nrow(traplocs)
5793 > K <- data$K
5794 > xlim <- data$xlim
5795 > ylim <- data$ylim

```

5796 Now, we need to define the integration grid, say **G**, which we do with the following set of
 5797 **R** commands (here, **delta** is the grid spacing):

```

5798 > delta <- .2
5799 > xg <- seq(xlim[1]+delta/2,xlim[2]-delta/2,by=delta)
5800 > yg <- seq(ylim[1]+delta/2,ylim[2]-delta/2,by=delta)
5801 > npix <- length(xg)           # valid for square state-space only
5802 > G <- cbind(rep(xg,npix),sort(rep(yg,npix)))
5803 > nG <- nrow(G)

```

5804 In this case, the integration grid is set up as a grid with spacing $\delta = 0.2$ which produces,
 5805 for our example, a 40×40 grid of points for evaluating the integrand if the state-space
 5806 buffer is set at 2. We note that the integration grid is set-up here to correspond exactly
 5807 to the state-space used in simulating the data. However, in practice, we wouldn't know
 5808 this, and our estimate of N (for the unknown case, see below) would be sensitive to choice
 5809 of the extent of the integration grid. As we've discussed previously, density, which is N
 5810 standardized by the area of the state-space, will not be so sensitive in most cases.

5811 We are now ready to compute the conditional-on-s likelihood and carry out the
 5812 marginalization described by Eq. 6.1.2. We need to do this by defining an **R** function
 5813 that computes the likelihood for the integration grid, as a function of the data objects
 5814 **y** and **traplocs** which were created above. However, it is a bit untidy to store the grid
 5815 information in your workspace, and define the likelihood function in a way that depends
 5816 on these things that exist in your workspace. Therefore, we build the **R** function so that
 5817 it computes the integration grid *within* the function, thereby avoiding potential problems
 5818 if our trapping grid locations change, or if we want to modify the state-space buffer easily.
 5819 We therefore define the function, called **intlik1**, to which we pass the data objects and
 5820 other information necessary to compute the marginal likelihood. This function is available
 5821 in the **scrbook** package (use **?intlik1** at the **R** prompt). The code is reproduced here:

```

5822 intlik1 <- function(parm,y=y,X=traplocs, delta=.2, ssbuffer=2){
5823
5824   Xl <- min(X[,1]) - ssbuffer ## These lines of code are setting up the
5825   Xu <- max(X[,1]) + ssbuffer ## support for the integration which is
5826   Yu <- max(X[,2]) + ssbuffer ## the same as the state-space of "s"
5827   Yl <- min(X[,2]) - ssbuffer
5828   xg <- seq(Xl+delta/2,Xu-delta/2,,length=npix)
5829   yg <- seq(Yl+delta/2,Yu-delta/2,,length=npix)
5830   npix<- length(xg)
5831
5832   G <- cbind(rep(xg,npix),sort(rep(yg,npix)))

```

```

5833   nG <- nrow(G)
5834   D <- e2dist(X,G)
5835
5836   alpha0 <- parm[1]
5837   alpha1 <- exp(parm[2]) # alpha1 restricted to be positive here
5838
5839   probcap <- plogis(alpha0)*exp(-alpha1*D*D)
5840   Pm <- matrix(NA,nrow=nrow(probcap),ncol=ncol(probcap))
5841           # Frequency of all-zero encounter histories
5842   n0 <- sum(apply(y,1,sum)==0)
5843           # Encounter histories with at least 1 detection
5844   ymat <- y[apply(y,1,sum)>0,]
5845   ymat <- rbind(ymat,rep(0,ncol(ymat)))
5846   lik.marg <- rep(NA,nrow(ymat))
5847
5848   for(i in 1:nrow(ymat)){
5849       ## Next line: log conditional likelihood for ALL possible values of s
5850       Pm[1:length(Pm)] <- dbinom(rep(ymat[i,],nG),K,probcap[1:length(Pm)],
5851                                     log=TRUE)
5852       ## Next line: sum the log conditional likelihoods, exp() result
5853       ## same as taking the product
5854       lik.cond <- exp(colSums(Pm))
5855       ## Take the average value == computing marginal
5856       lik.marg[i] <- sum(lik.cond*(1/nG))
5857   }
5858   ## n0 = number of all-0 encounter histories
5859   nv <- c(rep(1,length(lik.marg)-1),n0)
5860   return( -1*(sum(nv*log(lik.marg)) ) )
5861 }
```

5862 We emphasize that this function (and subsequent) are not meant to be general-purpose
 5863 routines for solving all of your SCR problems but, rather, they are meant for illustrative
 5864 purposes – so you can see how the integrated likelihood is constructed and how we connect
 5865 it to data and other information that is needed.

5866 The function `intlik1` accepts as input the encounter history matrix, `y`, the trap locations,
 5867 `X`, and the state-space buffer. This allows us to vary the state-space buffer and easily
 5868 evaluate the sensitivity of the MLE to the size of the state-space. Note that we have a
 5869 peculiar handling of the encounter history matrix `y`. In particular, we remove the all-zero
 5870 encounter histories from the matrix and tack-on a single all-zero encounter history as the
 5871 last row which then gets weighted by the number of such encounter histories (`n0`). This is
 5872 a bit long-winded and strictly unnecessary when N is known, but we did it this way be-
 5873 cause the extension to the unknown- N case is now transparent (as we demonstrate in the
 5874 following section). The matrix `Pm` holds the log-likelihood contributions of each encounter
 5875 frequency for each possible state-space location of the individual. The log contribu-
 5876 tions are summed up and the result exponentiated on the next line, producing `lik.cond`, the
 5877 conditional-on-s likelihood (Eq. 6.1.1 above). The marginal likelihood (`lik.marg`) sums
 5878 up the conditional elements weighted by the probabilities [`s`] (Eq. 6.1.2 above).

5879 This is a fairly primitive function which doesn't allow much flexibility in the data
 5880 structure. For example, it assumes that K , the number of replicates, is constant for each
 5881 trap. Further, it assumes that the state-space is a square. We generalize this to some
 5882 extent later in this chapter.

5883 Here is the **R** command for maximizing the likelihood using **nlm** (the function **optim**
 5884 could also be used) and saving the results into an object called **frog**. The output is a list
 5885 of the following structure and these specific estimates are produced using the simulated
 5886 data set:

```
5887 # should take 15-30 seconds
5888
5889 > starts <- c(-2,2)
5890 > frog <- nlm(intlik1,starts,y=y,X=traplocs,delta=.1,ssbuffer=2,hessian=TRUE)
5891 > frog
5892
5893 $minimum
5894 [1] 297.1896
5895
5896 $estimate
5897 [1] -2.504824 2.373343
5898
5899 $gradient
5900 [1] -2.069654e-05 1.968754e-05
5901
5902 $hessian
5903 [,1]      [,2]
5904 [1,] 48.67898 -19.25750
5905 [2,] -19.25750 13.34114
5906
5907 $code
5908 [1] 1
5909
5910 $iterations
5911 [1] 11
```

5912 Details about this output can be found on the help page for **nlm**. We note briefly that
 5913 **frog\$minimum** is the negative log-likelihood value at the MLEs, which are stored in the
 5914 **frog\$estimate** component of the list. The order of the parameters is as they are defined
 5915 in the likelihood function so, in this case, the first element (value = -2.504824) is the
 5916 logit transform of p_0 and the second element (value = 2.373343) is the value of α_1 the
 5917 “coefficient” on distance-squared. The Hessian is the observed Fisher information matrix,
 5918 which can be inverted to obtain the variance-covariance matrix using the command:

```
5919 > solve(frog$hessian)
```

5920 It is worth drawing attention to the fact that the estimates are slightly different than
 5921 the Bayesian estimates reported previously in Sec. 5.6. There are several reasons for this.
 5922 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,

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

5960 (2) We compute the probability of an all-0 encounter history:

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

5961 (3) The combinatorial term: $\binom{N}{n}$. Then, the marginal likelihood has this form:

$$\mathcal{L}(\boldsymbol{\alpha}, n_0 | \mathbf{y}) = \frac{N!}{n! n_0!} \left\{ \prod_{i=1}^n [\mathbf{y}_i | \boldsymbol{\alpha}] \right\} \pi_0^{n_0}. \quad (6.2.1)$$

5962 This is discussed in Borchers and Efford (2008, p. 379) as the conditional-on- N form of the
 5963 likelihood – we also call it the “binomial form” of the likelihood because of its appearance.

5964 Operationally, things proceed much as before: We compute the marginal probability
 5965 of each observed \mathbf{y}_i , i.e., by removing the latent \mathbf{s}_i by integration. In addition, we com-
 5966 pute the marginal probability of the “all-zero” encounter history \mathbf{y}_{n+1} , and make sure to
 5967 weight it n_0 times. We accomplish this by “padding” the data set with a single encounter
 5968 history having $y_{n+1,j} = 0$ for all traps $j = 1, 2, \dots, J$. Then we be sure to include the
 5969 combinatorial term in the likelihood or log-likelihood computation. We demonstrate this
 5970 shortly. To analyze a specific case, we’ll simulate our fake data set (simulated using the
 5971 parameters given above). To set some things up in our workspace we do this:

```
5972 ## Obtain a simulated data set
5973 > data <- simSCRO(discard0=TRUE, rnd=2013)
5974
5975 ## Extract the items we need for analysis
5976 > y <- data$Y
5977 > nind <- nrow(y)
5978 > traplocs <- data$traplocs
5979 > J <- nrow(traplocs)
5980 > K <- data$K
```

5981 Recall that these data are simulated by default with $N = 100$, on an 8×8 unit state-
 5982 space representing the trap locations buffered by 2 units, although you can modify the
 5983 simulation script easily.

5984 As before, the likelihood is defined in the **R** workspace as an **R** function, **intlik2**,
 5985 which takes an argument being the unknown parameters of the model and additional
 5986 arguments as prescribed. In particular, we provide the encounter history matrix **y**, the
 5987 trap locations **traplocs**, the spacing of the integration grid (argument **delta**) and the
 5988 state-space buffer. Here is the new likelihood function:

```
5989 intlik2 <- function(parm,y=y,X=traplocs,delta=.3,ssbuffer=2){
5990
5991   Xl <- min(X[,1]) - ssbuffer
5992   Xu <- max(X[,1]) + ssbuffer
5993   Yu <- max(X[,2]) + ssbuffer
5994   Yl <- min(X[,2]) - ssbuffer
5995
5996   xg <- seq(Xl+delta/2,Xu-delta/2,delta)
```

```

5997   yg <- seq(Yl+delta/2,Yu-delta/2,delta)
5998   npix.x <- length(xg)
5999   npix.y <- plength(yg)
6000   area <- (Xu-Xl)*(Yu-Yl)/((npix.x)*(npix.y))
6001   G <- cbind(rep(xg,npix.y),sort(rep(yg,npix.x)))
6002   nG <- nrow(G)
6003   D <- e2dist(X,G)
6004   # extract the parameters from the input vector
6005   alpha0 <- parm[1]
6006   alpha1 <- exp(parm[2])
6007   n0 <- exp(parm[3]) # note parm[3] lives on the real line
6008   probcap <- plogis(alpha0)*exp(-alpha1*D*D)
6009   Pm <- matrix(NA,nrow=nrow(probcap),ncol=ncol(probcap))
6010   ymat <- rbind(y,rep(0,ncol(y)))
6011
6012   lik.marg <- rep(NA,nrow(ymat))
6013   for(i in 1:nrow(ymat)){
6014     Pm[1:length(Pm)] <- (dbinom(rep(ymat[i,],nG),K,probcap[1:length(Pm)],
6015                                     log=TRUE))
6016     lik.cond <- exp(colSums(Pm))
6017     lik.marg[i] <- sum(lik.cond*(1/nG) )
6018   }
6019   nv <- c(rep(1,length(lik.marg)-1),n0)
6020   ## part1 here is the combinatorial term.
6021   ## math: log(factorial(N)) = lgamma(N+1)
6022   part1 <- lgamma(nrow(y)+n0+1) - lgamma(n0+1)
6023   part2 <- sum(nv*log(lik.marg))
6024   return( -1*(part1+ part2) )
6025 }
```

6026 To execute this function for the data that we created with `simSCR0`, we execute the
 6027 following command (saving the result in our friend `frog`). This results in the usual output,
 6028 including the parameter estimates, the gradient, and the numerical Hessian which is useful
 6029 for obtaining asymptotic standard errors (see below):

```

6030 > starts <- c(-2.5,0,4)
6031 > frog <- nlm(intlik2,starts,hessian=TRUE,y=y,X=traplocs,delta=.2,ssbuffer=2)
6032
6033 Warning message:
6034 In nlm(intlik2, starts, hessian = TRUE, y = y, X = traplocs, delta = 0.2, :
6035 NA/Inf replaced by maximum positive value
6036
6037 > frog
6038 $minimum
6039 [1] 113.5004
6040
6041 $estimate
```

```

6042 [1] -2.538333 0.902807 4.232810
6043
6044 [... additional output deleted ...]

```

6045 Executing `nlm` here usually produces one or more **R** warnings due to numerical calculations
6046 happening on extremely small or large numbers (calculation of p near the edge of the
6047 state-space), and they also happen if a poor parameterization is used which produces
6048 evaluations of the objective function beyond the boundary of the parameter space (e.g.,
6049 $n_0 < 0$). Such numerical warnings can often be minimized or avoided altogether by picking
6050 judicious starting values of parameters or properly transforming or scaling the parameters
6051 but, in general, they can be ignored. You will see from the `nlm` output that the algorithm
6052 performed satisfactory in minimizing the objective function. The estimate of population
6053 size, \hat{N} , for the state-space (using the default state-space buffer) is

```

6054 > Nhat <- nrow(y) + exp(4.2328) #### This is n + MLE of n0
6055 > Nhat
6056 [1] 110.9099

```

6057 Which differs from the data-generating value ($N = 100$), as we might expect for a single
6058 realization. We usually will present an estimate of uncertainty associated with this MLE
6059 which we can obtain by inverting the Hessian. Note that $\text{Var}(\hat{N}) = n + \text{Var}(\hat{n}_0)$. Since
6060 we have parameterized the model in terms of $\log(n_0)$ we use the delta method² described
6061 in Williams et al. (2002, Appendix F4) (see also Ver Hoef, 2012) to obtain the variance
6062 on the scale of n_0 as follows:

```

6063 > (exp(4.2328)^2)*solve(frog$hessian)[3,3]
6064 [1] 260.2033
6065
6066 > sqrt(260)
6067 [1] 16.12452

```

6068 Therefore, the asymptotic “Wald-type” confidence interval for N is $110.91 \pm 1.96 \times 16.125 =$
6069 $(79.305, 142.515)$. To report this in terms of density, we scale appropriately by the area
6070 of the prescribed state-space which is 64 units of area (i.e., an 8×8 square). Our MLE
6071 of D is $\hat{D} = 110.91/64 = 1.733$ individuals per square unit. To get the standard error
6072 for \hat{D} we need to divide the SE for \hat{N} by the area of the state-space, and so $\text{SE}(\hat{D}) =$
6073 $(1/64) * 16.12452 = 0.252$.

6074 6.2.1 Integrated likelihood under data augmentation

6075 The likelihood analysis developed in the previous sections is based on the likelihood in
6076 which N (or n_0) is an explicit parameter. This is usually called the “full likelihood” or
6077 sometimes “unconditional likelihood” (Borchers et al., 2002) because it is the likelihood
6078 for all individuals in the population, not just those which have been captured, i.e., not that
6079 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>

6080 likelihood using data augmentation, replacing the parameter N with ψ (e.g., see Sec. 7.1.6
 6081 Royle and Dorazio, 2008, for an example). We don't go into detail here, but we note that
 6082 the likelihood under data augmentation is a zero-inflated binomial mixture – precisely an
 6083 occupancy type model (Royle, 2006). Thus, while it is possible to carry out likelihood
 6084 analysis of models under data augmentation, we primarily advocate data augmentation
 6085 for Bayesian analysis.

6086 6.2.2 Extensions

6087 We have only considered basic SCR models with no additional covariates. However,
 6088 in practice, we are interested in covariate effects including “behavioral response”, sex-
 6089 specificity of parameters, and potentially others. Some of these can be added directly to
 6090 the likelihood if the covariate is fixed and known for all individuals captured or not. An
 6091 example is a behavioral response, which amounts to having a covariate $x_{ik} = 1$ if individ-
 6092 ual i was captured prior to occasion k and $x_{ik} = 0$ otherwise. For uncaptured individuals,
 6093 $x_{ik} = 0$ for all k . Royle et al. (2011b) called this a global behavioral response because the
 6094 covariate is defined for all traps, no matter the trap in which an individual was captured.
 6095 We could also define a *local* behavioral response which occurs at the level of the trap, i.e.,
 6096 $x_{ijk} = 1$ if individual i was captured in trap j prior to occasion k , etc... Trap-specific
 6097 covariates such as trap type or status, or time-specific covariates such as date, are eas-
 6098 ily accommodated as well. As an example, Kéry et al. (2010) develop a model for the
 6099 European wildcat *Felis silvestris* in which traps are either baited or not (a trap-specific
 6100 covariate with only 2 values), and also encounter probability varies over time in the form
 6101 of a quadratic seasonal response. We consider models with behavioral response or fixed
 6102 covariates in Chapt. 7. The integrated likelihood routines we provided above can be
 6103 modified directly for such cases, which we leave to the interested reader to investigate.

6104 Sex-specificity is more difficult to deal with since sex is not known for uncaptured
 6105 individuals (and sometimes not even for all captured individuals). To analyze such models,
 6106 we do Bayesian analysis of the joint likelihood using data augmentation (Gardner et al.,
 6107 2010b; Russell et al., 2012), discussed further in Chapt. 7. For such covariates (i.e., that
 6108 are not fixed and known for all individuals), it is somewhat more challenging to do MLE
 6109 based on the joint likelihood as we have developed above. Instead it is more conventional
 6110 to use what is colloquially referred to as the “Huggins-Alho” type model which is one of
 6111 the approaches taken in the software package **secr** (Efford, 2011). We introduce the **secr**
 6112 package in Sec. 6.5 below.

6.3 CLASSICAL MODEL SELECTION AND ASSESSMENT

6113 In most analyses, one is interested in choosing from among various potential models, or
 6114 ranking models, or something else to do with assessing the relative merits of a set of
 6115 models. A good thing about classical analysis based on likelihood is we can apply Akaike
 6116 Information Criterion (AIC) methods (Burnham and Anderson, 2002) without difficulty.
 6117 AIC is convenient for assessing the relative merits of these different models although if
 6118 there are only a few models it is not objectionable to use hypothesis tests or confidence
 6119 intervals to determine importance of effects. A second model selection context has to
 6120 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)`):

```
6151 > library(scrbook)
6152 > data(wolverine)
6153
6154 > traps <- wolverine$traps
6155 > traplocs <- traps[,2:3]/10000
6156 > K.wolv <- apply(traps[,4:ncol(traps)],1,sum)
6157
6158 > y3d <- SCR23darray(wolverine$wcaps,traps)
6159 > y2d <- apply(y3d,c(1,2),sum)
6160
6161 > starts <- c(-1.5,0,3)
6162
```

```

6163 > wolv <- nlm(intlik3,starts,hessian=TRUE,y=y2d,K=K.wolv,X=traplocs,
6164           delta=.2,ssbuffer=2)
6165
6166 > wolv
6167 $minimum
6168 [1] 220.4313
6169
6170 $estimate
6171 [1] -2.8176120 0.2269395 3.5836875
6172
6173 [.... output deleted ....]

```

6174 Of course we're interested in obtaining an estimate of population size for the prescribed
 6175 state-space, or density, and associated measures of uncertainty which we do using the delta
 6176 method (Williams et al., 2002, Appendix F4). To do all of that we need to manipulate the
 6177 output of `nlm` since we have our estimate in terms of $\log(n_0)$. We execute the following
 6178 commands:

```

6179 > wolv <- nlm(intlik3,starts,hessian=TRUE,y=y2d,K=K.wolv,X=traplocs,delta=.2,
6180           ssbuffer=2)
6181 > Nhat <- nrow(y2d)+exp(wolv$estimate[3])
6182 > area <- attr(intlik3(starts,y=y2d,K=K.wolv,X=traplocs,delta=.2,ssbuffer=2),
6183           "SSarea")
6184 > Dhat <- Nhat/area
6185
6186 > Dhat
6187 [1] 0.5494947
6188
6189 > SE <- (1/area)*exp(wolv$estimate[3])*sqrt(solve(wolv$hessian)[3,3])
6190
6191 > SE
6192 [1] 0.1087073

```

6193 Our estimate of density is 0.55 individuals per “standardized unit” which is 100 km^2 ,
 6194 because we divided UTM coordinates by 10000. So this is about 5.5 individuals per 1000
 6195 km^2 , with a SE of around 1.09 individuals. This compares closely with 5.77 reported in
 6196 Sec. 5.9 based on Bayesian analysis of the model.

6197 6.4.1 Sensitivity to integration grid and state-space buffer

6198 The effect of approximating the integral by a discrete mesh of points is that it induces
 6199 some numerical error in evaluation of the integral and, further, that error increases as the
 6200 coarseness of the mesh increases. To evaluate the effect (or sensitivity) of the integration
 6201 grid spacing, we obtained the MLEs for a state-space buffer of 2 (standardized units) and
 6202 for integration grid with spacing $\delta = .3, .2, .1, .05$. The MLEs for these 4 cases including
 6203 the relative runtime are given in Table 6.1. We see the results change only slightly as the
 6204 integration grid changes. Conversely, the runtime on the platform of the day for the 4 cases

6205 increases rapidly. These runtimes could be regarded in relative terms, across platforms,
 6206 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 (s)	$\hat{\alpha}_0$	$\hat{\alpha}_1$	$\log(\hat{n}_0)$
0.30	9.9	-2.819786	1.258468	3.569731
0.20	32.3	-2.817610	1.254757	3.583690
0.10	115.1	-2.817570	1.255112	3.599040
0.05	407.3	-2.817559	1.255281	3.607158

6207 We studied the effect of the state-space buffer on the MLEs, using a fixed $\delta = .2$ for
 6208 all analyses. We used state-space buffers of 1 to 4 units stepped by .5. As we can see
 6209 (Table 6.2), the estimates of D stabilize rapidly and the incremental difference is within
 6210 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

6211 6.4.2 Using a habitat mask (Restricted state-space)

6212 In Sec. 5.10 we used a discrete representation of the state-space in order to have control
 6213 over its extent and shape. This makes it easy to do things like clip out non-habitat, or
 6214 create a *habitat mask* which defines suitable habitat. Clearly that formulation of the model
 6215 is relevant to the calculation of the marginal likelihood in the sense that the discrete state-
 6216 space is equivalent to the integration grid. Thus, for example, we could easily compute
 6217 the MLE of parameters under some model with a restricted state-space merely by creating
 6218 the required state-space at whatever grid resolution is desired, and then inputting that
 6219 state-space into the likelihood function above, instead of computing it within the function.
 6220 We can easily create an explicit state-space grid for integration from arbitrary polygons or
 6221 GIS shapefiles which we demonstrate here. Our approach is to create the integration grid
 6222 (or state-space grid) outside of the likelihood evaluation, and then determine which points
 6223 of the grid lie in the polygon defined by the shapefile using functions in the **R** packages **sp**
 6224 and **maptools**. For each point in the state-space grid (object **G** in the code below which is

6225 assumed to exist), we determine whether it is inside the polygon³, identifying such points
 6226 with a value of `mask=1` and `mask=0` for points that are *not* in the polygon. We load the
 6227 shapefile which originates by an application of the `readShapeSpatial` function. We have
 6228 saved the result into an **R** data object called `SSp` which is in the `scrbook` package. Here
 6229 are the **R** commands for doing this (see the helpfile `?intlik4`):

```
6230 > library(maptools)
6231 > library(sp)
6232 > library(scrbook)
6233
6234 ##### If we have the .shp file in place, we would use this command:
6235 ##### SSp <- readShapeSpatial('Sim_Polygon.shp')
6236 ##### The object SSp is in data(fakeshapefile)
6237 > data(fakeshapefile)
6238 > Pcoord <- SpatialPoints(G)
6239 > PinPoly <- over(Pcoord,SSp)  ### determine if each point is in polygon
6240 > mask <- as.numeric(!is.na(PinPoly[,1])) ## convert to binary 0/1
6241 > G <- G[mask==1,]
```

6242 We created the function `intlik4` which accepts the integration grid as an explicit argument,
 6243 and this function is also available in the package `scrbook`.

6244 We apply this modification to the wolverine camera trapping study. Royle et al.
 6245 (2011b) created 2, 4 and 8 km state-space grids so as to remove “non-habitat” (mostly
 6246 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
 6247 likelihood we execute the following commands:

```
6249 > library(scrbook)
6250 > data(wolverine)
6251
6252 > traps <- wolverine$wtraps
6253 > traplocs <- traps[,2:3]/10000
6254 > K.wolv <- apply(traps[,4:ncol(traps)],1,sum)
6255
6256 > y3d <- SCR23darray(wolverine$wcaps,traps)
6257 > y2d <- apply(y3d,c(1,2),sum)
6258 > G <- wolverine$grid2/10000
6259
6260 > starts <- c(-1.5,0,3)
6261 > 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. Maximum likelihood estimates (MLEs) and asymptotic standard errors (SE) for the wolverine camera trapping data using 2, 4 and 8 km state-space grids.

Grid	α_0	α_1	$\log(n_0)$	N	SE	D(1000)	SE
2	-3.00	1.27	4.11	81.98	16.31	8.31	1.65
4	-2.99	1.34	4.16	84.88	16.76	8.57	1.69
8	-3.05	1.08	4.06	78.89	15.31	7.85	1.52

```

6262
6263 > wolv
6264
6265 $minimum
6266 [1] 225.8355
6267
6268 $estimate
6269 [1] -2.9955424 0.2350885 4.1104757
6270
6271 [... some output deleted ...]

```

Next we convert the parameter estimates to estimates of total population size for the prescribed state-space, and then obtain an estimate of density (per 1000 km²) using the area computed as the number of pixels in the state-space grid, G, multiplied by the area per grid cell. In the present case (the calculation above) we used a state-space grid with 2 km × 2 km pixels. Finally, we compute a standard errors using the delta approximation:

```

6277 > area <- nrow(G)*4
6278 # Nhat = n (observed) + MLE of n0 (not observed)
6279 > Nhat <- 21 + exp(wolv$estimate[3])
6280 > SE <- exp(wolv$estimate[3])*sqrt(solve(wolv$hessian)[3,3])
6281 > D <- (Nhat/(nrow(G)*area))*1000
6282 > SE.D <- (SE/(nrow(G)*area))*1000

```

We did this for each the 2 km, 4 km and 8 km state-space grids which produced the estimates summarized in Table 6.3. These estimates compare with the 8.6 (2 km grid) and 8.2 (8 km grid) reported in Royle et al. (2011b) based on a clipped state-space as described in Sec. 5.10.

6.5 DENSITY AND THE R PACKAGE SECR

DENSITY is a software program developed by Efford (2004) for fitting spatial capture-recapture models based mostly on classical maximum likelihood estimation and related inference methods. Efford (2011) has also released an **R** package called **secr**, that contains much of the functionality of **DENSITY** but also incorporates new models and features. Here, we briefly introduce the **secr** package which we prefer to use over **DENSITY**, because it allows us to remain in the **R** environment for data processing and summarization. We provide a brief introduction to **secr** and some of its capabilities here, and we also use

6294 it for doing some analysis in other parts of this book. We believe that **secr** will be sufficient
 6295 for many (if not most) of the SCR problems that one might encounter. It provides
 6296 a flexible analysis platform, with a large number of summary features, and “publication
 6297 ready” output. Its user-interface is clean and intuitive to **R** users, and it has been stable,
 6298 efficient and reliable in the (fairly extensive) evaluations that we have done.

6299 To install and run models in **secr**, you must download the package and load it in **R**.

```
6300 > install.packages("secr")
6301 > library(secr)
```

6302 **secr** allows the user to simulate data and fit a suite of models with various detection functions
 6303 and covariate responses. It also contains a number of helpful constructor functions
 6304 for creating objects of the proper class that are recognized by other **secr** functions. We
 6305 provide a brief overview of the capabilities here, but the **secr** help manual can be accessed
 6306 with the command:

```
6307 > RShowDoc("secr-manual", package = "secr")
```

6308 We note that **secr** has many capabilities that we will not cover or do so only sparingly.
 6309 We encourage you to read through the manual, the extensive documentation, and the
 6310 vignettes, in order to get a better understanding of what the package is capable of. We
 6311 also cover certain capabilities of **secr** in other chapters.

6312 The main model-fitting function in **secr** is called **secr.fit**, which makes use of the
 6313 standard **R** model specification framework with tildes. As an example, the equivalent of
 6314 the basic model SCR0 is fitted as follows:

```
6315 > secr.fit(capturedata, model = list(D ~ 1, g0 ~ 1, sigma ~ 1),
6316           buffer = 20000)
```

6317 where **capturedata** is the object created by **secr** containing the encounter history data
 6318 and the trap information, and the model expression $g0 \sim 1$ indicates the intercept-only (i.e.,
 6319 constant) model. Note that we use p_0 for the baseline encounter probability parameter,
 6320 which is g_0 in **secr** notation. A number of possible models for encounter probability can
 6321 be fitted including both pre-defined variables (e.g., **t** and **b** corresponding to “time” and
 6322 “behavior”), and user-defined covariates of several kinds. For example, to include a global
 6323 behavioral response, this would be written as $g0 \sim b$. The discussion of this (global versus
 6324 local trap-specific behavioral response) and other covariates is developed more in Chapt.
 6325 7. We can also model covariates on density in **secr**, which we discuss in Chapt. 11. It
 6326 is important to note that **secr** requires the buffer distance to be defined in meters and
 6327 density will be returned as number of animals per hectare. Thus to make comparisons
 6328 between **secr** and output from other programs, we will often have to convert the density
 6329 to the same units.

6330 Before we can fit the models, the data must first be packaged properly for **secr**.
 6331 We require data files that contain two types of information: trap layout (location and
 6332 identification information for each trap), which is equivalent to the trap deployment file
 6333 (TDF) described in Sec. 5.9 and the capture data file containing sampling *session*, animal
 6334 identification, trap occasion, and trap location, equivalent in information content to the
 6335 encounter data file (EDF). Sample session can be thought of as primary period identifier

6336 in a robust design like framework – it could represent a yearly sample or multiple sample
 6337 periods within a year, each of them producing data on a closed population. We discuss
 6338 “multi-session” models in more detail below, in Sec. 6.5.4 and Chapt. 14.

6339 There are three important constructor functions that help package-up your data for
 6340 use in **secr**: **read.traps**, **make.capthist** and **read.mask**. We provide a brief description
 6341 of each here, but apply them to our wolverine camera trapping data in the next section:

6342 (1) **read.traps**: This function points to an external file *or* **R** data object containing the
 6343 trap coordinates, and other information, and also requires specification of the type of
 6344 encounter devices (described in the next section). A typical application of this function
 6345 looks like the following, invoking the **data=** option when there is an existing **R** object
 6346 containing the trap information:

```
6347 > trapfile <- read.traps(data=traps, detector="proximity")
```

6348 (2) **make.capthist**: This function takes the EDF and combines it with trap information,
 6349 and the number of sampling occasions. A typical application looks like this:

```
6350 > capturedata <- make.capthist(enc.data, trapfile, fmt="trapID",  

  6351 noccaasions=165)
```

6352 See **?make.capthist** for definition of distinct file formats. Specifying **fmt = trapID** is
 6353 equivalent to our EDF format.

6354 (3) **read.mask**: If there is a habitat mask available (as described in sec. 6.4.2), then this
 6355 function will organize it so that **secr.fit** knows what to do with it. The function
 6356 accepts either an external file name (see **?read.mask** for details of the structure) or a
 6357 $nG \times 2$ **R** object, say **mask.coords**, containing the coordinates of the mask. A typical
 6358 application looks like the following:

```
6359 > grid <- read.mask(data=mask.coords)
```

6360 These constructor functions produce output that can then be used in the fitting of models
 6361 using **secr.fit**.

6362 6.5.1 Encounter device types and detection models

6363 The **secr** package requires that you specify the type of encounter device. Instead of
 6364 describing models by their statistical distribution (Bernoulli, Poisson, etc..), **secr** uses
 6365 certain operational classifications of detector types including ‘proximity’, ‘multi’, ‘single’,
 6366 ‘polygon’ and ‘signal’. For camera trapping/hair snares we might consider ‘proximity’
 6367 detectors or ‘count’ detectors. The ‘proximity’ detector type allows, at most, one detection
 6368 of each individual at a particular detector on any occasion (i.e., it is equivalent to what
 6369 we call the Bernoulli or binomial encounter process model, or model SCR0). The ‘count’
 6370 detector designation allows repeat encounters of each individual at a particular detector
 6371 on any occasion. There are other detector types that one can select such as: ‘polygon’
 6372 detector type which allows for a trap to be a sampled polygon (Royle and Young, 2008)
 6373 which we discuss further in Chapt. 15, and ‘signal’ detector which allows for traps that
 6374 have a strength indicator, e.g., acoustic arrays (Dawson and Efford, 2009). The detector
 6375 types ‘single’ and ‘multi’ refer to traps that retain individuals, thus precluding the ability
 6376 for animals to be captured in other traps during the sampling occasion. The ‘single’ type

6377 indicates trap that can only catch one animal at a time (single-catch traps), while 'multi'
 6378 indicates traps that may catch more than one animal at a time (multi-catch). These are
 6379 both variations of the multinomial encounter models described in Chapt. 9.

6380 As with all SCR models, **secr** fits an encounter probability model ("detection function"
 6381 in **secr** terminology relating the probability of encounter to the distance of a detector from
 6382 an individual activity center. **secr** allows the user to specify one of a variety of detection
 6383 functions including the commonly used half-normal ("Gaussian"), hazard rate ("Gaussian
 6384 hazard"), and (negative) exponential models. There are 12 different functions as of version
 6385 2.3.1 (see Table 7.1 in Chapt. 7), but some are only available for simulating data. The
 6386 different detection functions are defined in the **secr** manual and can be found by calling
 6387 the help function for the detection function:

6388 > ?detectfn

6389 Most of the detection functions available in **secr** contain some kind of a scale parameter
 6390 which is usually labeled σ . The units of this parameter default to meters in the **secr**
 6391 output. We caution that the meaning of this parameter depends on the specific detection
 6392 model being used, and it should not be directly compared as a measure of home-range size
 6393 across models. Instead, as we noted in Sec. 5.4 most encounter probability models imply
 6394 a model of space-usage and fitted encounter models should be converted to a common
 6395 currency such as "area used."

6396 6.5.2 Analysis using the **secr** package

6397 To demonstrate the use of the **secr** package, we will show how to do the same analysis on
 6398 the wolverine study as shown in Sec. 5.9. To use the **secr** package, the data need to be
 6399 formatted in a similar but slightly different manner than we use in **WinBUGS**.

6400 For example, in Sec. 5.9 we introduced a standard data format for the encounter data
 6401 file (EDF) and trap deployment file (TDF). The EDF shares the same format as that used
 6402 by the **secr** package with 1 row for every encounter observation and 4 columns representing
 6403 trap session ('Session'), individual identity ('ID'), sample occasion ('Occasion'), and trap
 6404 identity ('trapID'). For a standard closed population study that takes place during a single
 6405 season, the 'Session' column in our case is all 1's, to indicate a single primary sampling
 6406 occasion. In addition to providing the encounter data file (EDF), we must tell **secr** infor-
 6407 mation about the traps, which is formated as a matrix with column labels 'trapID', 'x' and
 6408 'y', the last two being the coordinates of each trap, with additional columns representing
 6409 the operational state of each trap during each occasion (1=operational, 0=not).

6410 We demonstrate these differences now by walking through an analysis of the wolverine
 6411 camera trapping data using **secr**. To read in the trap locations and other related infor-
 6412 mation, we make use of the constructor function **read.traps** which also requires that we
 6413 specify the detector type. The detector type is important because it will determine the
 6414 likelihood that **secr** will use to fit the model. Here, we have selected "proximity" which
 6415 corresponds to the Bernoulli encounter model in which individuals are captured at most
 6416 once in each trap during each sampling occasion:

6417 > library(secr)
 6418 > library(scrbook)

```

6419 > data(wolverine)
6420
6421 > traps <- as.matrix(wolverine$wtraps)
6422 > dimnames(traps) <- list(NULL,c("trapID","x","y",paste("day",1:165,sep="")))
6423 > traps1 <- as.data.frame(traps[,1:3])
6424 > trapfile1 <- read.traps(data=traps1,detector="proximity")

```

6425 Here we note that trap coordinates are extracted from the wolverine data but we do
6426 *not* scale them. This is because **secr** defaults to coordinate scaling of meters which is
6427 the extant scaling of the wolverine trap coordinates. Note that we add a 'trapID' column
6428 to the trap coordinates and provide appropriate column labels to the 'traps' matrix. An
6429 important aspect of the wolverine study is that while the camera traps were operated over
6430 a 165 day period, each trap was operational during only a portion of that period. We need
6431 to provide the trap operation information which is contained in the columns to the right
6432 of the trap coordinates in our standard trap deployment file (TDF). Unfortunately, this is
6433 less easy to do in **secr**⁴, which requires an external file with a single long string of 1's and
6434 0's indicating the days in which each trap was operational (1) or not (0). The **read.traps**
6435 function will not allow for this information on trap operation if the data exists as an **R**
6436 object – instead, we can create this external file and then read it back in with **read.traps**
6437 using these commands:

```

6438 > hold <- rep(NA,nrow(traps))
6439 > for(i in 1:nrow(traps)){
6440 >   hold[i] <- paste(traps[i,4:ncol(traps)],collapse="")
6441 > }
6442 > traps1 <- cbind(traps[,1:3],"usage"=hold)
6443
6444 > write.table(traps1, "traps.txt", row.names=FALSE, col.names=FALSE)
6445 > trapfile2 <- read.traps("traps.txt",detector="proximity")

```

6446 These operations can be accomplished using the function **scr2secr** which is provided in
6447 the **R** package **scrbook**.

6448 After reading in the trap data, we now need to create the encounter matrix or array
6449 using the **make.capthist** command, where we provide the capture histories in EDF format,
6450 which is the existing format of the data input file **wcaps**. In creating the capture history,
6451 we provide also the trapfile created previously, the format (e.g., here EDF format is
6452 **fmt= "trapID"**), and finally, we provide the number of occasions.

```

6453 #
6454 # Grab the encounter data file and format it:
6455 #
6456 wolv.dat <- wolverine$wcaps
6457 dimnames(wolv.dat) <- list(NULL,c("Session","ID","Occasion","trapID"))
6458 wolv.dat <- as.data.frame(wolv.dat)
6459 wolvcapt2 <- make.capthist(wolv.dat,trapfile2,fmt="trapID",noccasions=165)

```

⁴as of v. 2.3.1

6460 We also set up a habitat mask using the 2×2 km grid which we used previously in the
 6461 analysis of the wolverine data and then pass the relevant objects to `secr.fit` as follows:

```

6462 #
6463 # Grab the habitat mask (2 x 2 km) and format it:
6464 #
6465 gr2 <- (as.matrix(wolverine$grid2))
6466 dimnames(gr2) <- list(NULL,c("x","y"))
6467 gr2 <- read.mask(data=gr2)
6468 #
6469 # To fit the model we use secr.fit:
6470 #
6471 wolv.secr2 <- secr.fit(wolvcapt2,model=list(D ~ 1, g0 ~ 1, sigma ~ 1),
6472                         buffer=20000,mask=gr2)
```

6473 We are using the “proximity detector” model (SCR0), so we do not need to make any
 6474 specifications in the command line because we have specified the detector type using the
 6475 constructor function `read.traps`, except to provide the buffer size (in meters). To specify
 6476 different models, you can change the default model $D \sim 1$, $g_0 \sim 1$, $\sigma \sim 1$. We provide all
 6477 of these commands and additional analyses in the `scrbook` package with the function called
 6478 `secr_wolverine`. Printing the output object produces the following (slightly edited):

```

6479 > wolv.secr2
6480
6481 secr 2.3.1, 15:52:45 29 Aug 2012
6482
6483 Detector type      proximity
6484 Detector number    37
6485 Average spacing     4415.693 m
6486 x-range             593498 652294 m
6487 y-range             6296796 6361803 m
6488 N animals           : 21
6489 N detections         : 115
6490 N occasions          : 165
6491 Mask area            : 987828.1 ha
6492
6493 Model                : D ~ 1 g0 ~ 1 sigma ~ 1
6494 Fixed (real)          : none
6495 Detection fn          : halfnormal
6496 Distribution           : poisson
6497 N parameters          : 3
6498 Log likelihood        : -602.9207
6499 AIC                   : 1211.841
6500 AICc                  : 1213.253
6501
6502 Beta parameters (coefficients)
6503          beta      SE.beta       lcl       ucl
```

```

6504 D      -9.390124 0.22636698 -9.833795 -8.946452
6505 g0     -2.995611 0.16891982 -3.326688 -2.664535
6506 sigma   8.745547 0.07664648  8.595323  8.895772
6507
6508 Variance-covariance matrix of beta parameters
6509          D           g0           sigma
6510 D      0.0512420110 -0.0004113326 -0.003945371
6511 g0     -0.0004113326  0.0285339045 -0.006269477
6512 sigma  -0.0039453711 -0.0062694767  0.005874683
6513
6514 Fitted (real) parameters evaluated at base levels of covariates
6515      link   estimate    SE.estimate      lcl      ucl
6516 D      log 8.354513e-05 1.915674e-05 5.360894e-05 1.301982e-04
6517 g0     logit 4.762453e-02 7.661601e-03 3.466689e-02 6.509881e-02
6518 sigma  log 6.282651e+03 4.822512e+02 5.406315e+03 7.301037e+03

```

6519 The object returned by `secr.fit` provides extensive default output when printed.
6520 Much of this is basic descriptive information about the model, the traps, or the encounter
6521 data. We focus here on the parameter estimates. Under the fitted (real) parameters, we
6522 find D , the density, given in units of individuals/hectare (1 hectare = 10000 m^2). To
6523 convert this into individuals/1000 km², we multiply by 100000, thus our density estimate
6524 is 8.35 individuals/1000 km². The parameter σ is given in units of meters, and so this
6525 corresponds to 6.283 km. Both of these estimates are very similar to those obtained in
6526 our likelihood analysis summarized in Table 6.3 which, for the 2 × 2 km grid, we obtained
6527 $\hat{D} = 8.31$ with a SE of $100000 \times 1.915674e - 05 = 1.9156$ and, accounting for the scale
6528 difference (1 unit = 10000 m in the previous analysis), $\hat{\sigma} = \sqrt{1/(2\hat{\alpha}_1)} * 10000 = 6.289$
6529 km. The difference in the MLE between Table 6.3 and those produced by `secr` could be
6530 due to subtle differences in internal tuning of optimization algorithms, starting values or
6531 other numerical settings. In addition, the likelihood is based on a Poisson prior for N (see
6532 the next section). On the other hand, the SE is slightly larger based on `secr` which is due
6533 to a subtle difference in the interpretation of D under the `secr` model (See below).

6534 6.5.3 Likelihood analysis in the `secr` package

6535 The `secr` package does likelihood analysis of SCR models for most classes of models
6536 as developed by Borchers and Efford (2008). Their formulation deviates slightly from
6537 the binomial form we presented in Sec. 6.2 above (though Borchers and Efford (2008)
6538 also mention the binomial form). Specifically, the likelihood that `secr` implements is that
6539 based on removing N from the likelihood by integrating the binomial likelihood (Eq. 6.2.1
6540 above) over a Poisson prior for N – what we will call the *Poisson-integrated likelihood* as
6541 opposed to the conditional-on- N (*binomial-form*) considered previously.

6542 To develop the Poisson-integrated likelihood we compute the marginal probability of
6543 each \mathbf{y}_i and the probability of an all-0 encounter history, π_0 , as before, to arrive at the
6544 marginal likelihood in the binomial-form:

$$\mathcal{L}(\boldsymbol{\alpha}, n_0 | \mathbf{y}) = \frac{N!}{n! n_0!} \left\{ \prod_i [\mathbf{y}_i | \boldsymbol{\alpha}] \right\} \pi_0^{n_0}$$

6545 Now, what Borchers and Efford (2008) do is assume that $N \sim \text{Poisson}(\Lambda)$ and they do a
 6546 further level of marginalization over this prior distribution:

$$\sum_{n_0=0}^{\infty} \frac{N!}{n_0! n_0!} \left\{ \prod_i [\mathbf{y}_i | \boldsymbol{\alpha}] \right\} \pi_0^{n_0} \frac{\exp(-\Lambda) \Lambda^N}{N!}$$

6547 In Chapt. 11 we write $\Lambda = \mu ||\mathcal{S}||$ where $||\mathcal{S}||$ is the area of the state-space, and μ is the
 6548 density (“intensity”) of the point process. Carrying out the summation above produces
 6549 exactly this marginal likelihood:

$$\mathcal{L}_2(\boldsymbol{\alpha}, \Lambda | \mathbf{y}) = \left\{ \prod_i [\mathbf{y}_i | \boldsymbol{\alpha}] \right\} \Lambda^n \exp(-\Lambda(1 - \pi_0))$$

6550 which is Eq. 2 of Borchers and Efford (2008) except for notational differences. It also
 6551 resembles the binomial-form of the likelihood in Eq. 6.2.1 with $\Lambda^n \exp(-\Lambda\pi_0)$ replacing
 6552 the combinatorial term and the $\pi_0^{n_0}$ term. We emphasize there are two marginalizations
 6553 going on here: (1) the integration to remove the latent variables \mathbf{s} ; and, (2) summation
 6554 to remove the parameter N . We provide a function for computing this in the **scrbook**
 6555 package called **intlik3Poisson**. The help file for that function shows how to conduct a
 6556 small simulation study to compare the MLE under the Poisson-integrated likelihood with
 6557 that from the binomial form.

6558 The essential distinction between our MLE and Borchers and Efford as implemented in
 6559 **secr** is whether you keep N in the model or remove it by integration over a Poisson prior.
 6560 If you have prescribed a state-space explicitly with a sufficiently large buffer, then we
 6561 imagine there should be hardly any difference at all between the MLEs obtained by either
 6562 the Poisson-integrated likelihood or the binomial-form of the likelihood which retains N .
 6563 There is a subtle distinction in the sense that under the binomial form, we estimate the
 6564 realized population size N for the state-space whereas, for the Poisson-integrated form we
 6565 estimate the *prior* expected value which would apply to a hypothetical new study of a
 6566 similar population (see Sec. 5.7.3).

6567 Both models (likelihoods) assume \mathbf{s} is uniformly distributed over space, but for the
 6568 binomial model we make no additional assumption about N whereas we assume N is
 6569 Poisson using the formulation in **secr** from (Borchers and Efford, 2008). Using data
 6570 augmentation we could do a similar kind of integration but integrate N over a binomial
 6571 (M, ψ) prior – which we referred to as the binomial-integrated likelihood in Sec. 4.2.4.
 6572 So obviously the two approaches (data augmentation and Poisson-integrated likelihood)
 6573 are approximately the same as M gets large. However, doing a Bayesian analysis by
 6574 MCMC, we obtain an estimate of both N , the *realized population size*, and the parameter
 6575 controlling its expected value ψ which are, in fact, both identifiable from the data even
 6576 using likelihood analysis (Royle et al., 2007). That said we can integrate N out completely
 6577 and just estimate ψ as we noted in Sec. 6.2.1 above.

6578 6.5.4 Multi-session models in **secr**

6579 In practice we will often deal with SCR data that have some meaningful stratification or
 6580 group structure. For example, we might conduct mist-netting of birds on K consecutive
 6581 days, repeated, say, T times during a year, or perhaps over T years. Or we might collect

6582 data from R distinct trapping grids. In these cases, we have T or R groups which we might
 6583 reasonably regard as being samples of independent populations. While the groups might
 6584 be distinct sites, year, or periods within years, they could also be other biological groups
 6585 such as sex or age. Conveniently, **secr** fits a specific model for stratified populations –
 6586 referred to as *multi-session* models. These models build on the Poisson assumption which
 6587 underlies the integrated likelihood used in **secr** (as described in the previous section). To
 6588 understand the technical framework, let N_g be the population size of group g and *assume*

$$N_g \sim \text{Poisson}(\Lambda_g).$$

6589 Naturally, we model group-specific covariates on Λ_g :

$$\log(\Lambda_g) = \beta_0 + \beta_1 z_g$$

6590 where z_g is some group-specific covariate such as a categorical index to the group, or a
 6591 trend variable, or a spatial covariate, such as treatment effect or habitat structure, if the
 6592 groups represent spatial units. Under this model, we can marginalize *all* N_g parameters
 6593 out of the likelihood to concentrate the likelihood on the parameters β_0 and β_1 precisely
 6594 as discussed in the previous section. This Poisson hierarchical model is the basis of the
 6595 multi-session models in **secr**.

6596 To implement a multi-session model (or stratified population model) in **secr**, we pro-
 6597 vide the relevant stratification information in the ‘Session’ variable of the input encounter
 6598 data file (EDF). If ‘Session’ has multiple values then a “multi-session” object is created
 6599 by default and session-specific variables can be described in the model. For example, if
 6600 the session has 2 values for males and females then we have sex-specific densities , and
 6601 baseline encounter probability p_0 (g_0 in **secr**) by just doing this (see Chapt. 8 for the **R**
 6602 code to set this up):

```
6603 > out <- secr.fit(capdata, model=list(D ~ session, g0 ~ session, sigma^~ 1),  

  6604   buffer=20000)
```

6605 More detailed analysis is given in Sec. 8.1 where we fit a number of different models and
 6606 apply methods of model selection to obtain model-averaged estimates of density.

6607 We can also easily implement stratified population models in the various **BUGS** en-
 6608 gines using data augmentation (Converse and Royle, 2012; Royle and Converse, in review)
 6609 which we discuss, with examples, in Chapt. 14.

6610 6.5.5 Some additional capabilities of **secr**

6611 The **secr** package has capabilities to do a complete analysis of SCR data sets, including
 6612 model fitting, selection, and many summary analyses. In the previous sections, we’ve
 6613 given a basic overview, and we do more in later chapters of this book. Here we mention a
 6614 few of these other capabilities that you should know about as you use **secr**. Of course, you
 6615 should skim through the associated documentation (`?secr`) to see more of what’s available.

6616 Alternative observation models

6617 **secr** fits a wide range of alternative observation models besides the Bernoulli encounter
 6618 model, including multinomial encounter models for “multi-catch” and “single catch” traps,
 6619 models for sound attenuation from acoustic detection devices, and many others. We
 6620 discuss many of these other methods in Chapt. 9 and elsewhere in the book.

Summary statistics

6621 **secr** provides a useful default summary of the data, but it also has summary statistics
 6622 about animal movement including mean-maximum distance moved (the function **MMDM**).
 6623 For example, see the help page **?MMDM** which lists a number of other summary functions
 6624 which take a **capthist** object:

```
6626 > moves(capthist)
6627 > dbar(capthist)
6628 > RPSV(capthist)
6629 > MMDM(capthist, min.recapt = 1, full = FALSE)
6630 > ARL(capthist, min.recapt = 1, plt = FALSE, full = FALSE)
```

6631 The function **moves** returns the observed distances moved, **dbar** returns the average dis-
 6632 tance moved, **RPSV** produces a measure of dispersion about the home-range center, and
 6633 **ARL** gives the *Asymptotic Range Length* which is the asymptote of an exponential model
 6634 fit to the observed range length vs. the number of detections of each individual (Jett and
 6635 Nichols, 1987).

State-space buffer

6636 **secr** will produce a warning if the state-space buffer is chosen too small. For example,
 6637 in fitting the wolverine data as in Sec. 6.5.2 but with a 1000 m buffer, and we see the
 6638 following warning message:

```
6640 Warning message:
6641 In secr.fit(wolvcapt2, model=list(D ~ 1, g0 ~ 1, sigma ~ 1), buffer=1000):
6642   predicted relative bias exceeds 0.01 with buffer = 1000
```

6643 This should cause you to contemplate modifying the state-space buffer if that is a reason-
 6644 able thing to do in the specific application.

Model selection and averaging

6645 **secr** does likelihood ratio tests to compare nested models using the function **LR.test**.
 6646 You can create model selection tables based on AIC or AICc, using the function **AIC**,
 6647 and obtain model-averaged parameter estimates using the function **model.average** (See
 6648 Chapt. 8 for examples).

Population closure test

6649 **secr** has a population closure test with the function **closure.test** which implements the
 6650 tests of Stanley and Burnham (1999) or Otis et al. (1978). The function is used like this:
 6651 **closure.test(object, SB = FALSE)**. Here **object** is a **capthist** object and **SB** is a logical
 6652 variable that, if TRUE, produces the Stanley and Burnham (1999) test.

Density mapping and effective sample area

6653 **secr** produces likelihood versions of the various summaries of posterior density and effec-
 6654 tive sample area that we discussed in Chapt. 5. For example, while **secr** reports estimates
 6655 of the expected value of N or density directly in the summary output from fitting a model,
 6656 you can use the function **region.N** to produce estimates of N for any given region. In
 6657 addition, **secr** has functions for creating maps of detection contours for individuals traps,
 6658 or for the entire trap array. See the function **pdot.contour**, and also **fxi.contour** for

6662 computing the 2-dimensional pdf of the locations of one or more individual activity cen-
 6663 ters (as in Sec. 5.11.3). In the context of likelihood analysis, estimation of a random effect
 6664 **s** is based on a plug-in application of Bayes' Rule. When **s** has a uniform distribution, and
 6665 we use a discrete evaluation of the integral, it can be computed simply by renormalizing
 6666 the likelihood:

$$[s|y, \theta] = \frac{[y|s, \theta]}{\sum_s [y|s, \theta]}.$$

6667 Any of the **intlik** functions given previously in this chapter can be easily modified to
 6668 return the posterior distribution of **s** for any, or all, individuals, or an individual that is
 6669 not encountered.

6670 Effective sample area (see Sec. 5.12) can be calculated in **secr** using the functions **esa**
 6671 and **esa.plot**).

6672 Covariate models

6673 **secr** has many capabilities for modeling covariates. It has a number of built-in models
 6674 that allow certain covariates on encounter probability, which we cover to a large extent
 6675 in Chapt. 7, and also see Chapt. 8 for more examples. **secr** also allows covariates to be
 6676 built into the density model (see Chapt. 11). It has some built in response surface models,
 6677 allowing for the fitting of linear or quadratic response surfaces. This is done by modifying
 6678 the density model in **secr.fit**. For example, $D \sim 1$ is a constant density surface, and
 6679 $D \sim x + y$ fits a linear response surface, etc.. See the manual **secr-densitysurfaces.pdf**
 6680 for the details.

6681 There are a number of ways to model your own "custom" covariates (as opposed to
 6682 pre-specified models). One way is to use the **addCovariates** function and supply it a
 6683 **mask** or **traps** object along with some "spatialdata." Or, if you have covariates at each
 6684 trap location then it will extrapolate to all points on the habitat mask. There's also a
 6685 method by which the user can create a function of geographic coordinates, **userDfn**, which
 6686 seems to provide additional flexibility, although we haven't used this method. There is a
 6687 handy function **predictDsurface** for producing density maps under the specified model
 6688 for density.

6.6 SUMMARY AND OUTLOOK

6689 In this chapter, we discussed basic concepts related to classical analysis of SCR models
 6690 based on likelihood methods. Analysis is based on the so-called integrated or marginal
 6691 likelihood in which the individual activity centers (random effects) are removed from the
 6692 conditional-on-**s** likelihood by integration. We showed how to construct the integrated
 6693 likelihood and fit some simple models in the **R** programming language. In addition,
 6694 likelihood analysis for some broad classes of SCR models can be accomplished using the
 6695 **R** library **secr** (Efford, 2011) which we provided a brief introduction to. In later chapters
 6696 we provide more detailed analyses of SCR data using likelihood methods and the **secr**
 6697 package.

6698 Why or why not use likelihood inference exclusively? For certain specific models, it
 6699 is may be more computationally efficient to produce MLEs (for an example see Chapt.
 6700 12). And, likelihood analysis makes it easy to do model-selection by AIC and compute
 6701 standard errors or confidence intervals. However, **BUGS** is extremely flexible in terms
 6702 of describing models and we can devise models in the **BUGS** language easily that we

6703 cannot fit in **secr**. For example, in Chapt 16 we consider open population models which
6704 are straightforward to develop in **BUGS** but, so far, there is no available platform for
6705 doing MLE of such models. We can also fit models in **BUGS** that accommodate missing
6706 covariates in complete generality (e.g., unobserved sex of individuals), and we can adopt
6707 SCR models to include auxiliary data types. For example, we might have camera trapping
6708 and genetic data and we can describe the models directly in **BUGS** and fit a joint model
6709 (Gopalaswamy et al., 2012b). To do maximum likelihood estimation, we have to write a
6710 custom new piece of code for each model⁵ or hope someone has done it for us. You should
6711 have some capability to develop your own MLE routines with the tools we provided in
6712 this chapter.

⁵Although we may be able to handle multiple survey methods together in **secr** using the multi-session models.

6713
6714

7

6715
6716

MODELING VARIATION IN 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 meth-
6719 ods (Chapt. 6 or using **secr**). We mostly focused on a specific observation model, the
6720 Bernoulli or binomial model for devices such as “proximity detectors” (although we extend
6721 this model to Poisson and multinomial type observation models in Chapt. 9). We have
6722 not, however, described a general framework for modeling covariates that might influence
6723 encounter probability of individuals, traps or over time. In practice, investigators are
6724 invariably concerned with explicit factors or covariates that might influence variation in
6725 parameters. Such covariates include time (e.g., day of year, or season), behavior (e.g., is
6726 there an effect of trapping on subsequent capture probabilities), sex of the individual, and
6727 trap type (e.g., various camera types, or different constructions for hair snares). Tradition-
6728 ally, in the non-spatial capture recapture literature, such models were called “model M_t ”,
6729 “model M_h ”, or “model M_b ”, identifying models that account for variation in detection
6730 probability as a function of time, “individual heterogeneity” or “behavior”, where behav-
6731 ior describes whether or not an individual had been previously captured. In SCR models,
6732 more complex covariate models are possible because we might also have trap-specific co-
6733 variates, or covariates that vary spatially over the landscape, and because we generally
6734 have more than one parameter describing the detection function: Most encounter proba-
6735 bility functions include a baseline encounter rate (λ_0) or probability (p_0) parameter, and
6736 a scale parameter (σ), which takes on different interpretations depending on the specific
6737 encounter probability function under consideration.

6738 In this chapter, we generalize the basic SCR model to accommodate both alternative
6739 detection functions as well as many different kinds of covariates. We focus on the binomial
6740 observation model used throughout Chaps. 5 and 6 and the Gaussian encounter model
6741 (also called the “half-normal” model in the distance sampling literature), but the extension
6742 to other observation models is straightforward (and other encounter probability models
6743 with different functions of distance are considered in Sec. 7.1). Specifically, we consider

6744 three distinct types of covariates – those which are fixed, partially observed or completely
 6745 unobserved (latent). Fixed covariates are those that are fully observed; for example, the
 6746 date of all sampling occasions. Partially observed covariates are those which are not known
 6747 for all observations; for example, the sex of an individual cannot always be determined
 6748 from photos taken during camera trapping. Even if we are able to observe the sex of all
 6749 individuals sampled, we cannot know it for those individuals never observed during the
 6750 study. And finally, unobserved covariates are those which we cannot observe at all, for
 6751 example, the home range size of individuals, or unstructured random “individual effects”.

6752 We will see that models containing these different types of covariates are relatively easy
 6753 to describe in **WinBUGS** or **JAGS**, and therefore to analyze using Bayesian analysis
 6754 of the joint likelihood based on data augmentation thus providing a coherent and flexible
 6755 framework for inference for all classes of SCR models. Throughout the chapter, we will
 6756 continue to develop the analysis of the black bear study introduced in Chapt. 4, using the
 6757 software **JAGS**. We also consider the likelihood analysis of many of these models; to do so,
 6758 we continue to use the **R** package **secr**, and we introduce some ideas of model comparison
 6759 using AIC (Sec. 7.4 at the end of the chapter). There are other types of covariates that
 6760 we do *not* cover in this chapter; for example, covariates that vary across the landscape
 6761 might affect density, and we consider these covariates in Chapt. 11. Alternatively, these
 6762 landscape covariates might affect the way individuals use space. There are probably very
 6763 few circumstances under which animals use all space uniformly and we develop more
 6764 realistic models of encounter probability in which covariates affect space usage in Chapt.
 6765 12.

7.1 ENCOUNTER PROBABILITY MODELS

6766 In Chapt. 5, we developed a basic spatial capture recapture model using a standard
 6767 encounter probability function based on the kernel of a normal (Gaussian) probability
 6768 distribution:

$$p_{ij} = p_0 \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

6769 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
 6770 Sec. 5.4) that one can view this model as corresponding to an explicit model of space
 6771 usage – namely, that individual locations are draws from a bivariate normal distribution.
 6772 We also mentioned that other detection models are possible, including a logit model of
 6773 the form:

$$\text{logit}(p_{ij}) = \alpha_0 + \alpha_1 ||\mathbf{x}_j - \mathbf{s}_i||. \quad (7.1.1)$$

6774 However, there's nothing preventing us from constructing a myriad of other models for
 6775 encounter probability as a function of distance. The most commonly used detection prob-
 6776 ability models are also those used in the distance sampling literature: the half-normal
 6777 (Gaussian), the hazard, and the negative exponential. The negative exponential model is:

$$p_{ij} = p_0 * \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||)$$

6778 where we define $\alpha_1 = 1/\sigma$. We could use the general power model (Russell et al., 2012):

$$p_{ij} = p_0 * \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^\theta)$$

6779 of which the Gaussian and exponential models are special cases. Another model that could
 6780 be considered is the Gaussian hazard rate model (Hayes and Buckland, 1983):

$$p_{ij} = 1 - \exp(-\lambda_0 * \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2))$$

6781 which was previously discussed in Sec. .

6782 In each of the cases, the relationship of α_1 to σ varies and must be properly spec-
 6783 ified. The **R** package **secr** allows the user to access 12 different encounter probability
 6784 models (termed “distance functions” in **secr**), of which some are only used for simulating
 6785 data (see Table 7.1). These encounter probability models can also be implemented in **R**,
 6786 **WinBUGS**, **JAGS** etc..

Table 7.1. Basic encounter probability models (“distance functions”) available in **secr**. (Table taken from the **secr** help files). Notation deviates from that used in the text. In this table g_0 is the baseline encounter rate or probability parameter used in **secr** which is equivalent to our p_0 or λ_0 depending on context. d is distance defined as we have done throughout, as the distance between the activity center and the trap. One can read more on this specific table by loading the **secr** package and using the **help** command in **R** (**?detectfn**).

	Name	Params	Function
0	half-normal	g_0, σ	$g(d) = g_0 e^{-d^2/(2\sigma^2)}$
1	hazard rate	g_0, σ, z	$g(d) = g_0(1 - e^{-(d/\sigma)^{-z}})$
2	exponential	g_0, σ	$g(d) = g_0 e^{-d/\sigma}$
3	compound half-normal	g_0, σ, z	$g(d) = g_0[1 - \{1 - e^{-d^2/(2\sigma^2)}\}^z]$
4	uniform	g_0, σ	$g(d) = g_0, d \leq \sigma;$ $g(d) = 0, \text{ otherwise}$
5	w exponential	g_0, σ, w	$g(d) = g_0, d < w;$ $g(d) = g_0 e^{(-(d-w)/\sigma)}, \text{ otherwise}$
6	annular normal	g_0, σ, w	$g(d) = g_0 e^{(-(d-w)^2/(2\sigma^2))}$
7	cumulative lognormal	g_0, σ, z	$g(d) = g_0[1 - F(d - \mu)/s)]$
8	cumulative gamma	g_0, σ, z	$g(d) = g_0\{1 - G(d; k, \theta)\}$
9	binary signal strength	b_0, b_1	$g(d) = 1 - F\{-(b_0 + b_1 d)\}$
10	signal strength	β_0, β_1, S	$g(d) = 1 - F[\{c - (\beta_0 + \beta_1 d)\}/S]$
11	signal strength spherical	β_0, β_1, S	$g(d) = 1 - F[\{c - (\beta_0 + \beta_1(d-1) - 10 * \log_{10}(d^2))\}/S]$

6787 Insofar as all these encounter probability models are symmetric and stationary, they
 6788 are pretty crude descriptions of space usage by real animals. This is not to say they are
 6789 inadequate descriptions of the data and, as we discuss in Chaps. 13 and 12, we can use
 6790 them as the basis for producing more realistic models of space usage.

6791 By changing the encounter probability model and the specification of α_1 , we can
 6792 basically create any function of distance for the data. It is important to note that σ is not
 6793 comparable under these different encounter probability models and should not be regarded
 6794 as “home range radius” in general. While there is generally a relationship between σ and
 6795 home range size, that relationship varies depending on the model under consideration. We
 6796 demonstrate how to fit different encounter probability models in the Bayesian framework
 6797 here, and then provide information on the likelihood analysis (in **secr**) in a separate
 6798 section below.

6799 **7.1.1 Bayesian analysis with bear.JAGS**

6800 To demonstrate how to incorporate various types of covariates into models for encounter
 6801 probability using **JAGS**, we return to the data collected during the Fort Drum bear study.
 6802 This data set was first introduced in Chapt. 4, but, to refresh your memory, there were
 6803 38 baited hair snares that were operated between June and July 2006. The snares were
 6804 checked each week for a total for $K = 8$ sample occasions and $n = 47$ individual bears
 6805 were encountered at least once. The data are provided in the **R** package **scrbook** and an
 6806 **R** function called **bear.JAGS** allows the user to easily pick which model to analyze. The
 6807 function **bear.JAGS** will set up the data, write the model, define the MCMC specifications
 6808 (e.g., initial values, etc.) and, finally, run the selected model in **JAGS**. In addition to
 6809 choosing which model to run, the user can also specify the number of chains, iterations and
 6810 length of the burn-in phase. Calling the function will provide all the code to implement
 6811 the models independently as well. In the following sections we will present the model code
 6812 and output for the most commonly employed models; for all analyses we ran 3 chains with
 6813 a burn-in of 500 iterations and 20000 saved iterations.

6814 **7.1.2 Bayesian analysis of encounter probability models**

6815 In Panel 7.1, we present the basic SCR model and show how to specify the negative exponential
 6816 encounter probability model. To call each of these from the function **bear.JAGS** set
 6817 **model='SCRO'** or **model='SCRep'** in the function call, respectively. To reduce repetition
 6818 of the R coding, we include the basic code here and then only show modifications when
 6819 necessary throughout the chapter. All of the R coding can be found within the **bear.JAGS**
 6820 function as well. The function begins by loading the required **R** libraries as well as the
 6821 Ft. Drum bear data set. This data set includes a 3-d data array (called **bearArray** in our
 6822 code), with dimensions **nind** \times **ntraps** \times **nreps** representing the capture histories of **nind**
 6823 captured individuals at **ntraps** trap locations. In the Bayesian analysis, data augmentation
 6824 is used to estimate N and therefore the **bearArray** data must be augmented with
 6825 $M - nind$ all zero encounter histories. In models without time dependence, the augmented
 6826 **bearArray** (called **Yaug** in the code) will be reduced to a 2 dimensional array (denoted **y**
 6827 in the code) that has dimensions **M** \times **ntraps**.

```
6828 > library(rjags) # Load the necessary libraries
6829 > library(scrbook)
6830
6831 > data(beardata) # Attach the bear data for Ft. Drum
6832 > ymat <- beardata$bearArray
6833 > trapmat <- beardata$trapmat
6834 > nind <- dim(beardata$bearArray)[1]
6835 > K <- dim(beardata$bearArray)[3]
6836 > ntraps <- dim(beardata$bearArray)[2]
6837 > M <- 650
6838 > nz <- M-nind
6839
6840 # Create augmented array
6841 > Yaug <- array(0, dim=c(M,ntraps,K))
```

```

6842 > Yaug[1:nind,,] <- ymat
6843 > y <- apply(Yaug,1:2, sum)

```

6844 The function `bear.JAGS` also establishes the upper and lower limits on the state space
 6845 by centering the trap array coordinates (which are imported with the `beardata` and saved
 6846 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.

6847 Applying the SCR model with Gaussian encounter probability model provides an
 6848 estimate (posterior mean) of $D = 0.167$ bears per km^2 and with the negative exponential
 6849 encounter probability model the posterior mean is virtually the same $D = 0.167$. In
 6850 distance sampling, the use of different encounter probability models often results in very
 6851 different estimates of density (especially when using the negative exponential model).
 6852 There are two main reasons why the different models may have less of an impact on the
 6853 density estimates under the SCR models. First, we can estimate the baseline encounter
 6854 probability parameter (p_0). In most distance sampling models, detection at distance 0
 6855 is set to 1. In Table 7.2, the posterior mean of p_0 is 0.11 under the Gaussian model
 6856 and 0.34 under the negative exponential model. The larger baseline encounter probability

under the negative exponential model reduces the impact of the quick decline in detection as a function of distance. Secondly, the detection probability function here is governing 'movement' of individuals (which we have more information on than in distance sampling), not the whole detection process, so the shape of the detection probability function does not impact the density estimation as much.

In all analyses it is important to check that the size of the augmented data set (M) is sufficiently large and does not impact the estimate of N . Here, the 97.5% percentile for N is 628 (Table 7.2), thus not reaching our $M = 650$ value. We could also increase M and compare the posterior of N under the different scenarios as another check that the data augmentation is sufficient.

Table 7.2. Posterior summaries of SCR model parameters having different encounter probability models, for the Fort Drum black bear data.

Parameter	Mean	SD	2.5%	97.5%
Gaussian				
N	500.63	66.652	371.000	628.000
D	0.17	0.022	0.122	0.207
p_0	0.11	0.014	0.081	0.135
σ	1.99	0.131	1.762	2.275
ψ	0.77	0.104	0.566	0.966
Exponential				
N	512.06	65.771	382.000	634.000
D	0.17	0.022	0.130	0.210
p_0	0.34	0.056	0.246	0.465
σ	1.12	0.095	0.951	1.323
ψ	0.79	0.102	0.584	0.974

A very important consideration when using different detection probability functions is the interpretation of σ . The estimate (posterior mean) of σ under the negative exponential model is 1.12, which is distinct from our estimate of σ under the Gaussian model, $\sigma = 1.996$. The interpretation of σ in the two models is really quite distinct. In the normal model it can be interpreted as the standard deviation of a bivariate normal movement model whereas the manner in which σ relates to "area used" for the negative exponential model has nothing to do with a bivariate normal model of movement. This highlights that it is important for the user to know what detection probability function is used and what the interpretation of σ might be in relation to the home range size. This relationship was discussed in Sec. 5.4.

We now move onto incorporating covariates into the model using the **JAGS** language. For this part, we will stick with the Gaussian encounter probability model shown in Panel 7.1 above.

7.2 MODELING COVARIATE EFFECTS

The basic strategy for modeling covariate effects is to include them on the baseline encounter rate or probability parameter, p_0 (or λ_0), or the scale parameter of the encounter model, σ , or in some cases, both parameters.

6883 Broadly speaking, we recognize (here) 3 types of covariates. Fixed covariates are fully
 6884 observable and might vary by trap alone (e.g., type of trap, baited or not, disturbance
 6885 regime, even habitat), sample occasion (e.g., day of season or weather conditions), or both
 6886 (e.g., behavior, weather - if over a large region). Another class of covariates are those
 6887 which vary at the level of the individual (and possibly also over time). As a technical
 6888 matter, and as noted before, these are different from fixed covariates because we cannot
 6889 see all of the individuals and the covariates are almost always incompletely observed (if
 6890 at all). The lone exception is the effect of previous capture, used to model a behavioral
 6891 response to capture, which is known for all individuals, captured or not (an animal never
 6892 captured/observed has never been captured before). We noted many times before that
 6893 space itself (i.e., the activity centers) is a type of individual covariate and this notion
 6894 actually helped us derive the fully spatial capture-recapture model from the traditional,
 6895 non-spatial model (Chapt. 4). We do not get to observe the activity center for any
 6896 individuals, but for individuals that are encountered we get to observe some information
 6897 about it in the form of which traps the individual was encountered in. And finally, we have
 6898 completely unobserved covariates such as heterogeneity in home range size. We consider
 6899 heterogeneity in a separate section below since there are a suite of models for describing
 6900 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

6901 To develop covariate models, we assume a standard sampling design in which an array
 6902 of J traps is operated for K sample occasions, which produces encounter histories for n
 6903 individuals. For the null model, there are no time-varying covariates that influence en-
 6904 counter, there are no explicit individual-specific covariates, and there are no covariates
 6905 that influence density. For fixed effects, those which we observe fully, we can easily incor-
 6906 porate these into the encounter probability model, just as we would do in any standard
 6907 GLM or GLMM, on some suitable scale for the encounter probability, p_{ijk} . For example,

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * C_{ijk}$$

$$p_{ijk} = p_{0,ijk} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

6908 where C_{ijk} is some covariate that varies (potentially) by individual (i), trap (j) and
 6909 occasions (k), and α_2 is the coefficient to be estimated. How we define specific covariates
 6910 (e.g., trap specific versus individual specific) will influence exactly how we include them
 6911 in the model. Table 7.3 shows examples of covariates by type – trap, individual, and time
 6912 – and also gives examples of some combined types. These are the types of covariates we
 6913 will specifically address in this chapter, demonstrating how to analyze the different types
 6914 in the following sections.

6915 **7.2.1 Date and time**

6916 Often, researchers are interested in modeling the effect of date or chronological time on
 6917 encounter probability. For example, in a long term hair snare study, we may expect that
 6918 seasonal shedding (Wegan et al., 2012) will influence encounter probabilities directly. Or,
 6919 we may expect behaviors such as denning, mating, etc., to influence the encounter of
 6920 certain species at certain times of year (Kéry et al., 2011). There are two common ways
 6921 to incorporate date or time information into a model for encounter probability. For cases
 6922 with a small number of sampling occasions we can fit a time-specific intercept (analogous
 6923 to “model M_t ” in classical capture-recapture (Otis et al., 1978)). In this model, there are
 6924 K sampling occasion-specific parameters to reflect potential variation in sampling effort
 6925 or other factors that might vary across samples. Alternatively, we can model parametric
 6926 functions of date or time such as polynomial or sinusoidal functions.

6927 In the first case, we allow each sampling occasion, k , to have its own baseline encounter
 6928 probability, e.g.,

$$\text{logit}(p_{0,k}) = \alpha_{0,k}$$

6929 so that

$$p_{ijk} = p_{0,k} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2).$$

6930 This description of the model includes k occasion-specific baseline encounter probabilities.
 6931 Thus, if there are 4 sampling occasions, then there are 4 different baseline encounter
 6932 probabilities. We imagine that complete time-specificity of p_0 (i.e., one distinct value
 6933 for each sample occasion) would be most useful in situations where there are just a few
 6934 sampling occasions (if there are many, this formulation will dramatically increase the
 6935 number of parameters to be estimated) or we do not expect systematic patterns over time
 6936 (e.g., explainable by a polynomial function or time-varying covariates).

6937 To implement this in **JAGS**, α_0 has to be estimated for each time period k either
 6938 using an index vector or dummy variables (as described in Chapt. 2 and Sec. 4.3) and this
 6939 can be done by only changing only a few lines in Panel 7.1:

```
6940 alpha0[k] ~ dnorm(0,.1)
6941 logit(p0[k]) <- alpha0[k]
6942 .....
6943 .....
6944 y[i,j,k] ~ dbin(p[i,j,k],K)
6945 p[i,j,k] <- z[i]*p0[k]*exp(- alpha1*d[i,j]*d[i,j])
```

6946 Since the model contains a parameter for each time period, the encounter histories
 6947 must be time-dependent. Thus, a 3-d data array (called **bearArray** in our code), with
 6948 dimensions **nind** × **ntraps** × **nreps** is required (recall that we use the 3-d augmented array
 6949 called **Yaug** with dimensions **M** × **ntraps** × **nreps** for the Bayesian analysis). In addition
 6950 to using the 3-d data array, the initial values must be updated so that there are K values
 6951 generated for α_0 . And finally, this means that another nested *for loop* is needed in the
 6952 code to account for the K sample occasions. A side note: the computation time will
 6953 increase quite a bit (this model for the bear data may take up to 15 hours or more on
 6954 your machine to obtain a sufficient posterior sample).

6955 Running this model with the function **bear.JAGS** by setting **model=SCRt**, returns esti-
 6956 mates of density similar to those from the model without covariates (see Table 7.4), but

now we have a characterization of variation in encounter probability over time. Encounter probability seems to increase for the first few time periods before stabilizing around 0.14, dropping off again at the end of the study. The differences in encounter probability from the first time periods to the others might actually be due to something like a behavioral response (see below) or possibly seasonal differences in the efficiency of the sampling technique. Researchers have found that hair snares are more effective at different times of the year (even within season) due to shedding (Wegan et al., 2012). In this particular example, our density estimates (posterior means) are similar to the base model, likely because the differences in encounter probability between occasion were not that large. In a longer term study or in one with greater variation in the encounter probability, the implication of such differences might have a bigger impact on the estimates of density and σ .

Table 7.4. Posterior summaries of parameter estimates from a SCR model with time-dependent baseline encounter probability for the Ft. Drum black bear data set.

Parameter	Mean	SD	2.5%	97.5%
N	509.24	66.13	381	632
D	0.17	0.02	0.13	0.21
$p_0(t = 1)$	0.06	0.02	0.03	0.10
$p_0(t = 2)$	0.05	0.02	0.02	0.09
$p_0(t = 3)$	0.15	0.03	0.09	0.22
$p_0(t = 4)$	0.14	0.03	0.09	0.21
$p_0(t = 5)$	0.15	0.03	0.09	0.22
$p_0(t = 6)$	0.12	0.03	0.07	0.19
$p_0(t = 7)$	0.15	0.03	0.09	0.22
$p_0(t = 8)$	0.08	0.02	0.04	0.13
σ	1.96	0.12	1.73	2.22
ψ	0.78	0.10	0.58	0.97

The occasion specific intercepts (baseline encounter probability) model might not be the most appropriate for all scenarios and could require the estimation of many parameters if we had many sampling occasions, take the wolverine example from Chapt. 5.9 where there were 165 daily sampling occasions. Particularly in such a case as the wolverine study, variation in the encounter process over time is to be expected. For example, if a camera trap study is conducted for an entire year, it is expected that there would be behavioral patterns in individuals due to mating or denning. Instead of fitting a model with K baseline encounter probabilities, we can include date as a linear (or quadratic, ...) effect. An example can be found in Kéry et al. (2011) who incorporated a day-of-year covariate, both as a linear and a quadratic effect, into their SCR model of European wildcats; the data had been collected over a year-long period and cat behavior was expected to vary seasonally thus influencing the probability of encounter. In these cases, we would specifically incorporate day of year (variable “Date”) as a numeric covariate as:

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * \text{Date}_k$$

$$p_{ijk} = p_{0,ijk} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

6981 or a quadratic effect of day-of-year:

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * \text{Date}_k + \alpha_3 * \text{Date}_k^2$$

$$p_{ijk} = p_{0,ijk} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

6982 where the variable **Date** is an integer coding of day-of-year, indexed to some arbitrary
 6983 start point in time.

6984 **7.2.2 Trap-specific covariates**

6985 In some studies it makes sense to model encounter probability as a function of local or trap-
 6986 specific covariates. These can be one of two types: genuine trap covariates that describe
 6987 the trap or encounter site, such as whether a trap is baited or not, or how many traps were
 6988 set at a sampling location, or what kind of bait was used, etc., or local covariates that
 6989 describe the likelihood that an animal would use the habitat in the vicinity of the trap
 6990 (see Chapt. 13 for more on this situation). We imagine that these covariates, of either
 6991 type, should affect baseline encounter probability. For example, Sollmann et al. (2011)
 6992 found a large difference in the encounter probability of jaguars due to traps being located
 6993 on roads, which the animals were using to travel along, as opposed to traps placed off
 6994 of roads. In this case, the trap type is a binary variable – on/off road, (another binary
 6995 variable could be baited/non-baited). We can write this as:

$$\text{logit}(p_{0,j}) = \alpha_{0,type_j}$$

$$p_{ijk} = p_{0,j} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2).$$

6996 Here, we use an index variable, “type”, an integer value for the trap-specific covariate.
 6997 Thus for our example of on/off road, we would have $type_j = 1$ if trap j is on a road and
 6998 $type_j = 2$ otherwise, and we would estimate two separate α_0 parameters – one for on-road
 6999 and one for off-road cameras. An alternative way to express the 2-category model, using
 7000 dummy variables, requires that we specify our “type” vector as $Type_j = 0$ if trap j is on
 7001 a road and $Type_j = 1$ otherwise, and write the model as

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * Type_j.$$

7002 Now, α_0 is the baseline encounter probability (on the logit scale) for traps on a road
 7003 ($Type_j = 0$) and α_2 is the effect on baseline encounter probability of a trap being of
 7004 $Type = 1$. This general set up also allows for more than 2 categories, say if 4 different
 7005 camera models were used in a study, we would use a set of 3 binary dummy variables
 7006 to allow for estimation of the different encounter rates (i.e., the intercept). While these
 7007 models are equivalent, and should yield identical results, sometimes one parameterization
 7008 might work better than the other in **WinBUGS** or **JAGS** (Kéry, 2010).

7009 **7.2.3 Behavior or trap response by individual**

7010 One of the most basic of encounter models is that which accommodates a change in
 7011 encounter probability as a result of initial encounter. This is colloquially referred to as
 7012 “trap happiness” or “trap shyness”, or in other words, a behavioral response of individuals

7013 to being captured (Otis et al., 1978). If a trap is baited with a food source, an individual
 7014 might come back for more. On the other hand, if being captured is traumatic then an
 7015 individual might learn to avoid traps. Both of these types of responses can occur in
 7016 most species depending on the type of encounter mechanisms being employed. Moreover,
 7017 behavioral response can be either global (Gardner et al., 2010b) or local (Royle et al.,
 7018 2011b). The local response is a trap-specific response while a global response suggests that
 7019 initial capture provides a net increase or decrease in subsequent probabilities of capture
 7020 (across all traps). A behavioral response does not need to be enduring (i.e., persist for
 7021 the entire study after the individual has been captured/observed for the first time) but
 7022 can also be ephemeral, if, for example, an animal only avoids a trap on the occasion
 7023 immediately after it was captured (Yang and Chao, 2005; Royle, 2008). While we will
 7024 focus the examples in this chapter on enduring behavioral effects, extending such a model
 7025 to the case of an ephemeral response should not pose any difficulties.

7026 To describe these behavioral models we need to create a binary matrix that indicates
 7027 if an individual has been captured previously. For the global behavioral response, define
 7028 the $n \times K$ matrix, \mathbf{C} , where $C_{ik} = 1$ if individual i was captured at least once prior to
 7029 session k , otherwise $C_{ik} = 0$.

$$\text{logit}(p_{0,ik}) = \alpha_0 + \alpha_2 * C_{ik}$$

$$p_{ijk} = p_{0,ik} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2).$$

7030 For the local behavioral response, which is trap specific, we create an array, C_{ijk} , that
 7031 indicates if an individual i has been previously captured in trap j at time k . (For the
 7032 augmented individuals, the entries are all 0 since the animals were never captured.) We
 7033 then include this in the model in the exact same form as above (with the sole difference
 7034 that both C and p are now also indexed by k):

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * C_{i,j,k}$$

$$p_{ijk} = p_{0,ijk} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2).$$

7035 Since the behavioral response is occasion specific, to implement either the local or
 7036 global response model in **JAGS**, we will have to use the 3-d array of the augmented
 7037 capture histories ($M \times ntraps \times nreps$) as we did for the time-varying encounter probability
 7038 model above. The code must loop over each sampling occasion, but otherwise, the model
 7039 varies only a little from the basic SCR model shown in Panel 7.1. Here is the specification
 7040 of the the occasion specific (k) loop:

```
7041 for(k in 1:K){
 7042   logit(p0[i,j,k]) <- alpha0 + alpha2*C[i,j,k]
 7043   y[i,j,k] ~ dbin(p[i,j,k],1)
 7044   p[i,j,k] <- z[i]*p0[i,j,k]*exp(- alpha1*d[i,j]*d[i,j]).
```

7045 }
 7046 Despite only minor changes to the **BUGS** code, this model can require quite a bit
 7047 of time and computational effort. Implementing the behavioral models with the function
 7048 **bear.JAGS** by setting **model=SCRb** or **model=SCRb** for the local or global model respec-
 7049 tively, returns the results shown in Table 7.5. There is a strong global behavioral response
 7050 suggested by the posterior mean of $\alpha_2 = 0.90$. The estimate of N and subsequently D are

7051 larger than under the model without a behavioral response; here we estimate the posterior
 7052 mean of $N = 577.56$, whereas in the SCR0 model, we estimated the posterior mean as
 7053 $N = 500$. This makes sense given the large estimate of α_2 , which suggests that bears
 7054 are trap happy. In situations where animals are trap happy, the null model tends to over
 7055 estimate encounter probability (i.e., the bears that are never observed have a lower en-
 7056 counter probability than those that have been captured in the study) and thereby reduce
 7057 the estimate of N . We do not include the results here, but the estimates were similar
 7058 under the local behavioral response model.

Table 7.5. Posterior summaries of parameter estimates from the SCR model with a global behavioral response in encounter for the Fort Drum black bear data set.

Parameter	Mean	SD	2.5%	97.5%
N	577.56	54.30	452	648
D	0.19	0.02	0.15	0.21
α_0	-2.81	0.24	-2.91	-2.36
α_2	0.90	0.23	0.45	1.35
σ	2.00	0.13	1.77	2.28
ψ	0.88	0.08	0.69	0.99

7059 7.2.4 Individual covariates

7060 Individual covariates are those which are measured (or measurable) on individuals, so
 7061 we get to observe them only for the captured individuals. Sex is a simple example of an
 7062 individual covariate, but one of the most commonly used in capture-recapture studies. The
 7063 sex of an individual can influence many aspects of its ecology and behavior, including for
 7064 example, the frequency of movement, seasonal behavior, and its home range size. This is
 7065 common in studies of carnivores where females often have smaller home ranges than males
 7066 (Gardner et al., 2010b; Sollmann et al., 2011). Additionally, we may find differences in
 7067 the baseline encounter probability between males and females because females may move
 7068 around less frequently, or possibly because they are less likely to use landscape structures
 7069 that researchers may target with sampling devices in order to increase sample size, such
 7070 as roads (e.g. Salom-Pérez et al., 2007). Therefore, we can imagine that sex may impact
 7071 both the baseline encounter probability α_0 and the typical home range size, so that α_1
 7072 might also be sex-specific also. The fully sex-specific model is:

$$\text{logit}(p_{0,i}) = \alpha_{0,sex_i}$$

$$p_{ijk} = p_{0,i} \exp(-\alpha_{1,sex_i} * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

7073 where sex_i is a vector indicating the sex of each individual (1 = male, 2 = female). While
 7074 we might know the sex of all individuals observed in the study, we will never know the sex
 7075 of individuals that are not observed (Gardner et al., 2010b). It is also possible that we
 7076 may not be able to determine the sex of individuals that are observed during the study.
 7077 For example photographic captures do not necessarily result in pictures that allow the sex
 7078 to be absolutely determined, thus sometimes resulting in missing values of this covariate
 7079 for animals captured in the study. We deal with this slightly differently depending on

7080 the inference framework that we adopt (Bayesian or likelihood). Here we demonstrate
 7081 the Bayesian implementation and we discuss the likelihood approach using **secr** in detail
 7082 below in Sec. 7.4.2. Before proceeding with that, we note that it would be possible also to
 7083 model covariates directly on the parameter σ (or its logarithm), e.g., $\log(\sigma_i) = \theta_1 + \theta_2 \text{sex}_i$
 7084 (see Sec. 8.1). One or the other (or perhaps *some* other) parameterization may yield a
 7085 better performing MCMC algorithm or provide a more natural or preferred interpretation.
 7086 In the context of Bayesian analysis, given that priors are not invariant to transformation of
 7087 the parameters, this may be a consideration in choosing the particular parameterization.

7088 Specifying a fully sex-specific model for **JAGS** is similar to the time-specific model
 7089 shown above. We need to use an index or dummy variable to let α_0 and/or α_1 be defined
 7090 separately for males and females. The main difference in this specification is that we do
 7091 not observe sex for the augmented individuals. Therefore, we have missing observations
 7092 of the covariate for those individuals. As a result, sex is regarded as a random variable
 7093 and so the missing values can be estimated along with the other structural parameters of
 7094 the model.

7095 Because we are regarding sex as a random variable, we have to specify a distribution for
 7096 it. With only two possible outcomes, it is natural to suppose that $\text{Sex}_i \sim \text{Bernoulli}(\psi_{\text{sex}})$
 7097 where the parameter ψ_{sex} is the sex ratio of the population. We assume our default non-
 7098 informative prior for this parameter: $\psi_{\text{sex}} \sim \text{Uniform}(0, 1)$. The model specification in
 7099 Panel 7.2 demonstrates how to incorporate a partially observed covariate (i.e., “sex”). It
 7100 is important to note that in the previous equation, sex_i is a vector with two categories
 7101 indicating the sex of each individual (e.g., 1 = male, 2 = female). This corresponds
 7102 directly to having a binary indicator of sex (e.g., $\text{Sex}_i = 1$ if individual i is female, and 0
 7103 otherwise). In the Bayesian formulation of the model, we use both the binary indicator
 7104 (**Sex**) and a categorical indicator ($\text{Sex2} = \text{Sex} + 1$). The former (termed **Sex** in Panel
 7105 7.2) allows us to specify the Bernoulli distribution for the random variable, and the latter
 7106 (termed **Sex2**) allows us to use the dummy or indicator variable specification in the model.

7107 In both **JAGS** or **BUGS** missing data are indicated by **NA** in the data objects passed
 7108 to the program through the **bugs** or **jags** functions in **R**. To set up the data, we need to
 7109 create a vector of length M with the first n elements being 0 if individual i is a female, or
 7110 1 if i is a male (for the Fort Drum black bear data the function **bear.JAGS** extracts this
 7111 information automatically from the **beardata** object), and the subsequent $M - n$ elements
 7112 being **NA**. It is generally a good idea to provide starting values for the missing data, but we
 7113 cannot provide starting values for observed data; in this case where one vector (or other
 7114 object) contains both observed and missing data, initial values for the observed data have
 7115 to be specified as **NA**. The code snippet below shows you how to set up the data including
 7116 the **Sex** vector and the initial values function (the remainder of the code is identical to
 7117 what we've shown before).

```
7118 > sex <- beardata$sex #the sex data for captured individual
7119 > Sex <- c(sex-1, rep(NA, nz)) #sex enters as 1/2, this recodes it to 0/1
7120                                #so we can use Bernoulli distribution
7121
7122 > data <- list(y=y,Sex=Sex, M=M,K=K, J=ntraps, xlim=xlim, ylim=ylim,area=areaX)
7123 > params <- c('psi','p0','N', 'D', 'sigma', 'psi.sex')
7124 > inits <- function() { list(z=c(rep(1,nind), rbinom(nz,1,0.5)),psi=rnorm(1),
7125                                s=cbind(rnorm(M, xlim[1],xlim[2]), rnorm(M,ylim[1],ylim[2])),
```

```
7126     psi.sex=runif(1,Sex=c(rep(NA, nind), rbinom(nz,1,0.5)),
7127           sigma=runif(2,2,3),alpha0=runif(2)) }
```

7128 The **BUGS** model specification is shown in Panel 7.2.

```
model{

psi ~ dunif(0,1)                                # Prior distributions
psi.sex ~ dunif(0,1)
for(t in 1:2){
  alpha0[t] ~ dnorm(0,.1)
  logit(p0[t]) <- alpha0[t]
  alphai[t] <- 1/(2*sigma[t]*sigma[t])
  sigma[t] ~ dunif(0, 15)
}

for(i in 1:M){
  z[i] ~ dbern(psi)
  Sex[i] ~ dbern(psi.sex)                      # Sex is binary
  Sex2[i] <- Sex[i] + 1                         # Convert to categorical
  s[i,1] ~ dunif(xlim[1],xlim[2])
  s[i,2] ~ dunif(ylim[1],ylim[2])

  for(j in 1:J){
    d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
    y[i,j] ~ dbin(p[i,j],K)
    p[i,j] <- z[i]*p0[Sex2[i]]*exp(-alphai[Sex2[i]]*d[i,j]*d[i,j])
  }
}
N <- sum(z[])
D <- N/area
}
```

Panel 7.2: **JAGS** model specification for an SCR model with sex-specific encounter probability parameters.

7129 Our estimate of density under the fully sex-specific model is still very similar to the
 7130 previous models (Table 7.6), and while the baseline detection was not very different be-
 7131 tween males and females, we can see that they had very different σ estimates (note that
 7132 the BCIs do not overlap). As usual, you can reproduce this analysis by calling the function
 7133 `bear.JAGS` and set `model='SCRsex'`.

Table 7.6. Posterior summaries of parameter estimates from sex-specific SCR models for the Fort Drum black bear data set.

Parameter	Mean	SD	2.5%	97.5%
N	509.982	66.355	376	631
D	0.168	0.022	0.12	0.21
$p_{0,female}$	0.136	0.025	0.09	0.19
$p_{0,male}$	0.092	0.017	0.06	0.13
σ_{female}	1.542	0.132	1.31	1.83
σ_{male}	2.682	0.389	2.09	3.62
ψ_{sex}	0.310	0.068	0.19	0.45
ψ	0.784	0.103	0.58	0.97

7.3 INDIVIDUAL HETEROGENEITY

7134 Here we consider SCR models with individual heterogeneity. Capture-recapture models
 7135 with individual heterogeneity in detection probability, so-called model M_h , have a long
 7136 history in classical capture recapture models and they have special relevance to SCR (Sec.
 7137 4.4). While the advent of SCR models may appear to have rendered the use of classical
 7138 model M_h obsolete (because one major source of heterogeneity, namely exposure to the
 7139 trap array is being accounted for explicitly) we may still wish to consider heterogeneity
 7140 models for other biological reasons. It is reasonable to expect in real populations that there
 7141 exists heterogeneity in home range size and so we think that α_1 could exhibit heterogeneity
 7142 among individuals. As we noted previously, it may be advantageous or desirable in some
 7143 cases to model heterogeneity directly in terms of the scale parameter of the encounter
 7144 probability function, σ , or some other transformation of the “distance coefficient”, perhaps
 7145 even 95% home range area.

7146 In this section, we describe a class of spatial capture-recapture models to allow for
 7147 individual heterogeneity in encounter probability. In particular, one class of models we
 7148 propose explicitly admits individual heterogeneity in home range *size*. In addition, we con-
 7149 sider a standard representation for heterogeneity in which an additive individual-specific
 7150 random effect is included in the linear predictor for baseline encounter probability.

7151 7.3.1 Models of heterogeneity

7152 An obvious extension to the SCR model is to include an additive individual effect, analo-
 7153 gous to classical “model M_h ”. We’ll call this model “SCR+Mh”:

$$\begin{aligned} \text{logit}(p_{0,i}) &= \alpha_0 + \eta_i \\ p_{ijk} &= p_{0,i} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2) \end{aligned}$$

7154 where η_i is an individual random effect having distribution $[\eta|\sigma_p]$. A popular class of
 7155 models arises by assuming $\eta_i \sim \text{Normal}(0, \sigma_p^2)$ (Coull and Agresti, 1999; Dorazio and
 7156 Royle, 2003). We show how to implement this specific SCR + Mh model in Panel 7.3,
 7157 and this model can be used to analyze the Ft. Drum bear data by calling the function
 7158 `bear.JAGS` and setting `model='SCRh'`. While we show one possible implementation here,
 7159 many other random effects distributions are possible. A popular one is the finite-mixture

7160 of point masses (Norris and Pollock, 1996; Pledger, 2004) which we demonstrate how to
 7161 fit using **secr** in Sec. 7.4.3.

```
model{

  alpha0 ~ dnorm(0,.1)                                # Prior distributions
  alpha1 <- 1/(2*sigma*sigma)
  sigma ~ dunif(0, 15)
  psi ~ dunif(0,1)
  tau_p ~ dgamma(.001,.001)

  for(i in 1:M){
    eta[i] ~ dnorm(0, tau_p)                         # Individual level variables
    z[i] ~ dbern(psi)
    s[i,1] ~ dunif(xlim[1],xlim[2])
    s[i,2] ~ dunif(ylim[1],ylim[2])

    for(j in 1:J){                                     # The "likelihood" etc..
      d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
      y[i,j] ~ dbin(p[i,j],K)
      logit(p0[i,j]) <- alpha0 + eta[i]
      p[i,j] <- z[i]*p0[i,j]*exp(-alpha1*d[i,j]*d[i,j])
    }
  }
  N <- sum(z[])
  # N, D are derived
  D <- N/area
}
```

Panel 7.3: **JAGS** model specification for the SCR + Mh model with Gaussian encounter probability model and additive normal random effect.

7162 7.3.2 Heterogeneity induced by variation in home range size

7163 An alternative heterogeneity model, one that has more of a direct biological motivation and
 7164 interpretation, describes heterogeneity in home range size among individuals. To model
 7165 heterogeneity in home range area, we can assume a distribution for a transformation of
 7166 the scale parameter of the encounter probability model such as σ^2 , or $\log(\sigma^2)$, etc.. We
 7167 call this “model SCR + Ah” (Ah here for area-induced heterogeneity).

7168 Consider the following log-normal model for the individual scale parameter of the
 7169 Gaussian encounter probability model, σ_i^2 :

$$\log(\sigma_i^2) \sim \text{Normal}(\mu_{hra}, \tau_{hra}^2)$$

7170 then the 95% home range area has a scaled log-normal distribution with mean

$$6\pi \exp(\mu_{hra} + \tau_{hra}^2/2).$$

7171 The variance is slightly more complicated, but you can look up the variance of a log-normal
 7172 distribution and combine it with the 95% home range area calculation in Sec. 5.4 to work
 7173 out the implied variance of home range area under this model. We show two examples of
 7174 the implied *population* distribution of home range area under this log-normal model that
 7175 indicates a mean home range area of about 6.9 area units (Figure 7.1). The left panel
 7176 shows a standard deviation in home range area of 2.88 units and the right panel shows
 7177 a standard deviation in home range area of 0.70 units. The two cases were generated by
 7178 tweaking the μ_{hra} and τ_{hra}^2 parameters of the log-normal distribution to achieve a constant
 7179 expected value of home range area, but modify the standard deviation.

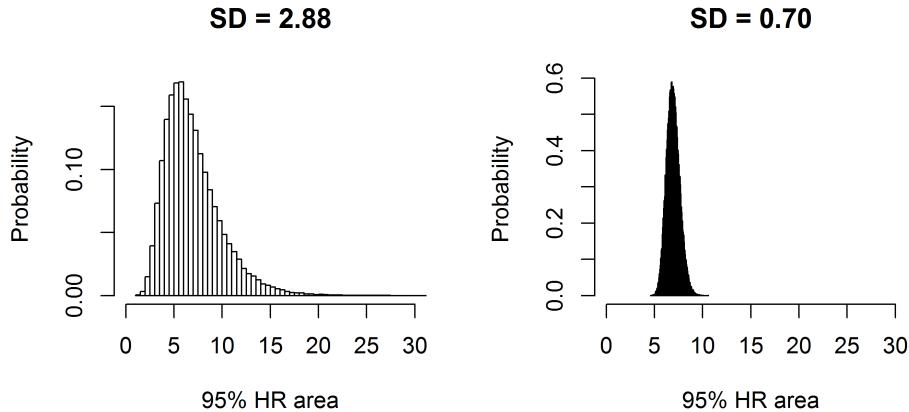


Figure 7.1. Population distribution of home range area for a model in which $\log(\sigma^2)$ has a normal distribution with mean μ_{hra} and variance τ_{hra}^2 . The parameters were chosen to yield a constant expected value of about 6.9 units of area, but to produce two different levels of heterogeneity: A population standard deviation of 2.88 units (left panel) and 0.70 units (right panel).

7.4 LIKELIHOOD ANALYSIS IN SECR

7180 Previously, in Chapt. 6, we introduced the R package **secr** and described the likelihood
 7181 based inference approach taken by that package (see Sec. 6.5.3). Here we discuss how
 7182 to implement some standard covariate models in **secr** and provide an example of model
 7183 selection using AIC. As we saw in Chapt. 6, **secr** uses the standard R model specifi-
 7184 cation syntax, defining the dependent and independent variable relationship using tildes

(e.g., $y \sim x$). Thus, in **secr** we might have $g0 \sim behavior$ or $sigma \sim time$; when left unspecified or set to 1 (e.g., $g0 \sim 1$), this will default to a model with no covariates (i.e., constant parameter values). A number of default model formulas for the baseline and scale parameter of the encounter probability model are available in **secr**. Additionally, **secr** allows us to specify covariates on density (we cover this in Chapt. 11), which are set for example as $D \sim habitat$.

To demonstrate models with various types of covariates using **secr**, we continue using the Fort Drum black bear data. We include in the **scrbook** package a function called **secr.bear** that will format the data (see Chapt. 6 for the **secr** data format) and then fit and compare 8 models (details shown in Panel 7.4). We have described all of these models in the previous sections, so we only briefly comment here on how to fit certain models in **secr** and compare them using AIC, and give a few helpful notes.

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 2.3.1 or newer of **secr**).

7.4.2 Sex effects

Incorporating sex effects into models with **secr** can be done a few different ways, but there are not pre-defined models for this. A limitation of fitting models with sex effects in **secr** is that it does not accommodate missing values of the sex variable. Thus, in all cases, individuals that are of unknown sex must be removed from the data set (recall that in a Bayesian framework we can keep these individuals in the data set by specifying a distribution for the individual covariate “sex”). In **secr**, the easiest way to include sex effects is to code sex as a “session” variable using the multi-session models (see Sec. 6.5.4 for a description of the multi-session models), providing two sessions, one representing

```
1. null model with a bivariate normal encounter probability model  
bear_0=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ 1, sigma ~ 1))  
  
2. null model with an exponential encounter probability model  
bear_0exp=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ 1, sigma ~ 1),  
                    detectfn=2)  
  
3. model with fixed time effects  
bear_t=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ t, sigma ~ 1))  
  
4. global behavioral model  
bear_B=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ b, sigma ~ 1))  
  
5. trap specific behavioral response  
bear_b=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ bk, sigma ~ 1))  
  
6. global behavior model with fixed time effects  
bear_bt=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ b+t, sigma ~ 1))  
  
7. sex-specific model  
bear_sex=secr.fit(bear.cap, model=list(D ~ session, g0 ~ session,  
                                         sigma ~ session))  
  
8. heterogeneity model  
bear_h2=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ h2, sigma ~ h2))
```

Panel 7.4: Models called from **secr.bear** function. All models use **buffer = 20000**

7227 males and one for females (see model 7 in Panel 7.4). This method provides two separate
 7228 density estimates, which can then be combined into a total density.

7229 **7.4.3 Individual heterogeneity**

7230 To incorporate heterogeneity, **secr** fits a set of finite mixture models (Norris and Pollock,
 7231 1996; Pledger, 2004). These are expensive in terms of parameters but they have been
 7232 widely adopted because they are easy to analyze using likelihood methods, as the marginal
 7233 distribution of the data is just a sum of a small number of components. Using **secr**,
 7234 individual heterogeneity can be incorporated into the encounter probability model using
 7235 default models for either a 2- or 3-component finite mixture model using the “**h2**” or “**h3**”
 7236 model terms. The 2-part mixture is shown in model 8 of panel 7.4 and the 3-part mixture
 7237 can easily be fit by substituting **h3** for **h2**. We only showed the SCR + Mh logit-normal
 7238 mixture in the version above (see Sec. 7.3.1), but finite-mixture models can also be fit in
 7239 **JAGS** or **BUGS**.

7240 **7.4.4 Model selection in **secr** using AIC**

7241 One practical advantage to using the **secr** package, or likelihood inference in general, is
 7242 the convenience of automatic model selection using AIC (Burnham and Anderson, 2002).
 7243 The **secr** package has a number of convenient functions for computing AIC and producing
 7244 model selection tables, or doing model-averaging (as described in Chapt. 8). Running the
 7245 function **secr.bear**, which calls all of the models we have described, will return, in addition
 7246 to all model results, an AIC table with all of the summarized results including the AIC
 7247 values, delta AIC, and model weights (see Table 7.7 or reproduce results in R using **out<-**
 7248 **secr.bear()**; **out\$AIC.tab**).

7249 It is important to note that AIC is not comparable between a multi-session model and
 7250 a model that is not a multi-session model. Therefore, to compare the sex-specific model
 7251 (which uses “sessions”) with all the other models including the null, time, and behavioral
 7252 models, we coded the data set as a multi-session design when first loading it to **secr**. This
 7253 results in all the model outputs listing separate parameter estimates for each session, even
 7254 the null model with no covariates; however, the estimates are the same for both “sessions”
 7255 in all but the sex-specific model (in other words, we don’t specify any effect of session on
 7256 parameters, except in the sex specific model).

7257 The results from this AIC analysis are straightforward to interpret; the model with
 7258 a local trap response of encounter probability, “**bk**”, has a model weight of 1 and thus,
 7259 according to AIC, 100% support compared with the other models in this model set. The
 7260 2-part finite mixture model for g_0 and σ has the second lowest AIC, but considering the
 7261 large dAICc compared to the local trap response model we would probably not consider
 7262 it any further.

7.5 SUMMARY AND OUTLOOK

7263 There are endless covariates and encounter probability models that can be defined and our
 7264 goal in this chapter was to introduce basic types of covariate models and demonstrate how
 7265 to implement them in **BUGS** and **secr**. Essentially, SCR’s are GLMMs and therefore

Table 7.7. Log-likelihood, AIC, deltaAIC and AIC weight for several models run in secr for the Fort Drum black bear data set.

model	logLik	AIC	AICc	dAICc	AICwt
bear.b	-641.7215	1291.443	1292.395	0.000	1
bear.h2	-653.8382	1319.676	1321.776	29.381	0
bear.0exp	-663.9152	1333.830	1334.389	41.994	0
bear.B	-677.6175	1363.235	1364.187	71.792	0
bear.bt	-668.3044	1358.609	1366.152	73.757	0
bear.sex	-677.7151	1367.430	1369.530	77.135	0
bear.t	-674.4134	1368.827	1374.938	82.543	0
bear.0	-686.2455	1378.491	1379.049	86.654	0

we develop covariate models in much the same way, using a suitable transformation (link function) of the parameter(s). In SCR models, we typically have 2 parameters of the encounter probability model for which we might specify covariate models – the baseline encounter probability (or rate) parameter, and a scale parameter that is related in many cases to the home range size of the species. A few examples of different covariate models are given in Table 7.3. We can also consider covariates by their classification as fixed, partially observed, or unobserved (see Table 7.8). This classification of covariate types can be important because the MLE and Bayesian approaches to dealing with partially and unobserved covariates is often different. This was seen above in how the covariate **Sex** was handled in the two frameworks.

Table 7.8. Examples of different covariate classifications.

Covariate class	Examples
Fixed	baited, weather, habitat
Partially observed	sex, age,
Unobserved	home range size, ind. effects

While the move to spatially explicit models in capture-recapture studies has largely rendered the basic CR models (Otis et al., 1978) obsolete, we continue to find this classification useful for categorizing the *spatial* extensions of these standard CR models. The extended models include the standard M_0 , M_t , M_b , and M_h , but also new models that allow for trap-specific information such as "baited/not-baited" or "on/off road". In addition, in Chaps. 12, 13 and 11, we explore models for explaining variation in encounter probability and density based on spatial covariates that describe variation in landscape or habitat conditions.

7284
7285

8

7286

MODEL SELECTION AND ASSESSMENT

7287 Our purpose in life is to analyze models. By that, we mean one or more of the following
7288 basic 4 tasks: (1) estimate parameters, (2) make predictions of unobserved random vari-
7289 ables, (3) evaluate the relative merits of different models or choosing a best model (model
7290 selection), and (4) checking whether a specific model appears to provide a reasonable de-
7291 scription of the data or not (model checking, assessment, or “goodness-of-fit”). In previous
7292 chapters we addressed the problems of estimation of model parameters, and also making
7293 predictions of latent variables, s or z , or functions of these variables such as density or
7294 population size. In this chapter, we focus on the last two of these basic inference tasks:
7295 model selection (which model or models should be favored), and model assessment (do
7296 the data appear to be consistent with a particular model).

7297 In this chapter we review basic strategies of model selection using both likelihood
7298 methods (as implemented in the `secr` package) and Bayesian analysis. Specifically, we
7299 review a number of standard methods of model selection that apply to “variable selection”
7300 problems, when our set of models consists of distinct covariate effects and they represent
7301 constraints of some larger model. For classical analysis based on likelihood, model selection
7302 by Akaike Information Criterion (AIC) is the standard approach (Burnham and Anderson,
7303 2002). For Bayesian analysis we rely on a number of different methods. We demonstrate
7304 the use of the deviance information criterion (DIC) (Spiegelhalter et al., 2002) for variable
7305 selection problems although it has deficiencies when applied to hierarchical models in some
7306 cases (Millar, 2009). We use the Kuo and Mallick indicator variable selection approach
7307 (Kuo and Mallick, 1998) which produces direct statements of posterior model probabilities
7308 which we think are the most useful, and leads directly to model-averaged estimates of
7309 density. There is a good review paper recently by O’Hara and Sillanpää (2009) that
7310 discusses these and many other related ideas for variable selection. In addition to O’Hara
7311 and Sillanpää (2009) we also recommend Link and Barker (2010, Chapt. 7) for general
7312 information on model selection and assessment.

7313 To check model adequacy in a Bayesian framework, or whether a specific model pro-
7314 vides a satisfactory description of our data set, we rely exclusively on the Bayesian p-value
7315 framework (Gelman et al., 1996). For assessing fit of SCR models, part of the challenge

7316 is coming up with good measures of model fit, and there does not appear much definitive
 7317 guidance in the literature on this point. Following Royle et al. (2011a), we break the prob-
 7318 lem up into 2 components which we attack separately: (1) Conditional on the underlying
 7319 point process, does the encounter model fit? (2) Do the uniformity and independence
 7320 assumptions appear adequate for the point process model of activity centers? The latter
 7321 component of model fit has a considerable precedence in the ecological literature as it
 7322 is analogous to the classical problem of testing “complete spatial randomness” (Cressie,
 7323 1991; Illian et al., 2008).

7324 We apply some of these methods to the wolverine camera trapping data first introduced
 7325 in Chapt. 5 to investigate sex specificity of model parameters and whether there is a
 7326 behavioral response to encounter. We note that individuals are drawn to the camera
 7327 trap devices by bait and therefore it stands to reason that once an individual discovers a
 7328 trap, it might be more likely to return subsequently, a response termed “trap happiness”.
 7329 We evaluate whether certain models for encounter probability appear to be adequate
 7330 descriptions of the data, and we evaluate the uniformity assumption for the underlying
 7331 point process.

8.1 MODEL SELECTION BY AIC

7332 Using classical analysis based on likelihood, model selection is easily accomplished using
 7333 AIC (Burnham and Anderson, 2002) which we demonstrate below. The AIC of a model is
 7334 simply twice the negative log-likelihood evaluated at the MLE, penalized by the number
 7335 of parameters (np) in the model:

$$\text{AIC} = -2\log L(\hat{\theta}|\mathbf{y}) + 2np$$

7336 Models with small values of AIC are preferred. It is common to use a modified (“cor-
 7337 rected”) AIC referred to as AIC_c for small sample sizes which is

$$AIC_c = -2\log L(\hat{\theta}|\mathbf{y}) + \frac{2np(np+1)}{n-np-1}$$

7338 where n is the sample size. Two important problems with the use of AIC and AIC_c are
 7339 that they don’t apply directly to hierarchical models that contain random effects, unless
 7340 they are computed directly from the marginal likelihood (for SCR models we can do this,
 7341 see Chapt. 6). Moreover, it is not clear what should be the effective sample size n in
 7342 calculation of AIC_c , as there can be covariates that affect individuals, that vary over
 7343 time, or space. We do not offer strict guidelines as to when to use a small sample size
 7344 adjustment.

7345 The R package **secr** computes and outputs AIC automatically for each model fitted
 7346 and it provides some capabilities for producing a model selection table (function **AIC**) and
 7347 also doing model-averaging (function **model.average**), which we recommend for obtaining
 7348 estimates of density from multiple models.

8.1.1 AIC analysis of the wolverine data

7349 We provide an example of model selection for the wolverine camera trapping data using
 7350 **secr**. We consider a model set with distinct models to accommodate various types of sex
 7351 specificity of model parameters:

7353 Model 0: model SCR0 with constant density and constant encounter model parameters;
 7354 Model 1: model SCR0 with constant parameter values for both male and female wolverines but with sex-specific density only;
 7355 Model 2: Sex-specific density, sex-specific p_0 but constant σ ;
 7356 Model 3: Sex-specific density, sex-specific σ but constant p_0 ;
 7357 Model 4: Sex-specific density, sex-specific p_0 and sex-specific σ .

7359 To model sex-specific abundance (density), we use the multi-session models provided
 7360 by **secr** (introduced in Sec. 6.5.4), which allow one to model session-specific effects on
 7361 density, baseline encounter probability, p_0 (labeled g_0 in **secr**), and also the scale parameter
 7362 σ of the encounter probability model. Using this formulation, we define the “Session”
 7363 variable to be a *categorical* sex code having value 1 or 2 (demonstrated below) and thus
 7364 *session*-specific parameters represent *sex*-specific parameters. For example, if we model
 7365 session-specific density, D , then this corresponds to Model 1 in our list above. We note
 7366 that “Model 0” in our list corresponds to a model where all of the encounter histories
 7367 have the same session ID. This model is one of constant density, which implies that the
 7368 population sex ratio is fixed at 0.5, i.e., $\psi_{\text{sex}} = 0.5$.

7369 Although **secr** also uses the logit/log linear predictors as the default for modeling
 7370 covariates on baseline encounter probability and the scale parameter, respectively, **secr**
 7371 does something different with the multi-session models. It reports estimates in a *session*
 7372 *mean* parameterization (equivalent to, in **BUGS**, using an index variable instead of a set
 7373 of dummy variables), and not the *session effect* (i.e., deviation from the intercept) which
 7374 arises from the use of dummy variables. We show this **BUGS** model description in Sec.
 7375 8.2.2.

7376 To fit these models using **secr**, we load the wolverine data and do a slight bit of
 7377 formatting to prepare the data objects for analysis by **secr**. The key difference from our
 7378 analysis in Chapt. 6 is, here, we use the wolverine sex information (**wolverine\$wsex**)
 7379 which is a binary 0/1 variable (1=male) and we add 1 so that we can define a categorical
 7380 “Session” variable (having values 1 or 2). We also have a function **scr2secr** which converts
 7381 a standard trap-deployment file (TDF) matrix into a **secr** object of class “traps.” The
 7382 **R** commands are as follows (contained in the help file **?secr_wolverine**):

```

7383
7384 > library(secr)
7385 > library(scrbook)
7386 > data(wolverine)
7387 > traps <- as.matrix(wolverine$wtraps)

7388 ## Name variables as required by secr
7389 > dimnames(traps) <- list(NULL,c("trapID","x","y",paste("day",1:165,sep="")))
7390 ## Convert trap information to a secr "traps" object
7391 > trapfile <- scr2secr(scrtraps=traps,type="proximity")

7392 ## Grab the wolverine state-space grid (2km here)
7393 > gr <- as.matrix(wolverine$grid2)
7394 > dimnames(gr) <- list(NULL,c("x","y"))
7395 > gr2 <- read.mask(data=gr)
7396
7397

```

```

7398
7399 ## Grab the encounter data, and re-name variables
7400 > wolv.dat <- wolverine$wcaps
7401 > dimnames(wolv.dat) <- list(NULL,c("Session","ID","Occasion","trapID"))
7402
7403 ## Convert binary 0/1 sex variable to categorical 1/2 for "session"
7404 > wolv.dat[,1] <- wolverine$wsex[wolv.dat[,2]]+1
7405 > wolv.dat <- as.data.frame(wolv.dat)
7406
7407 ## Convert to capthist object
7408 > wolvcapt <- make.capthist(wolv.dat,trapfile,fmt="trapID",noccasions=165)

```

7409 Once the data have been prepared in this way, we use the `secr` model fitting function
 7410 `secr.fit` to fit the different models, and then the function `AIC` to package the models
 7411 together and summarize them in the form of an AIC table, with rows of the table ordered
 7412 from best to worst. The function `model.average` performs AIC-based model-averaging of
 7413 the parameters specified by the `realnames` variable (below this is demonstrated for the
 7414 parameter density, D). Because this function defaults to averaging by AIC_c , we slightly
 7415 modified this function (called `model.average2`) to do model averaging by either AIC or
 7416 AIC_c as specified by the user. The model fitting commands look like this (for Model 0
 7417 and Model 1):

```

7418 > model0 <- secr.fit(wolvcapt, model=list(D~1, g0~1, sigma~1),
7419           buffer=20000)
7420 > model1 <- secr.fit(wolvcapt, model=list(D~session, g0~1, sigma~1),
7421           buffer=20000)

```

7422 Next we use the function `AIC`, passing the fit objects from all 5 models, and that
 7423 produces the following output (abbreviated horizontally to fit on the page):

```

7424 > AIC (model0,model1,model2,model3,model4)
7425           model      ... npar logLik   AIC    AICc dAICc  AICwt
7426 model0  D~1 g0~1 sigma~1  ...  3 -627.2603 1260.521 1261.932 0.000 0.5831
7427 model2      ..      ...  5 -624.9051 1259.810 1263.810 1.878 0.2280
7428 model1      ..      ...  4 -627.2365 1262.473 1264.973 3.041 0.1275
7429 model4      ..      ...  6 -624.6632 1261.326 1267.326 5.394 0.0393
7430 model3      ..      ...  5 -627.2358 1264.472 1268.472 6.540 0.0222

```

7431 Model averaging the results is done as follows:

```

7432 > model.average (model0,model1,model2,model3,model4,realnames="D")
7433           estimate  SE.estimate      lcl      ucl
7434 session=1 2.707190e-05 7.913577e-06 1.544474e-05 4.745224e-05
7435 session=2 2.927423e-05 8.270402e-06 1.700631e-05 5.039193e-05

```

7436 As usual, estimates and standard errors of the individual model parameters can be
 7437 obtained from the `secr.fit` summary output of any of the `modelX` objects shown above.
 7438 The default output of estimated density is in individuals per ha, so we have to scale this
 7439 up to something more reasonable. To get into units of per 1000 km², we need to first

7440 multiply by 100 to get to units of km^2 and then multiply by 1000. This produces an
 7441 estimated density of about 2.71 for `session=1` (females) and 2.93 for `session=2` (males).
 7442 We can use the generic **R** function `predict` applied to the `secr.fit` output to obtain
 7443 specific information about the MLEs on the natural scale.

7444 We don't necessarily agree with the use of AIC_c here and think its better to use AIC,
 7445 in general. This is because, as noted previously, it is not clear what the effective sample
 7446 size is for most capture-recapture problems. While we have 21 individuals in the data
 7447 set, most of the model structure has to do with encounter probability samples and for
 7448 that there are hundreds of observations. We do note that the AIC and AIC_c results are
 7449 not entirely consistent. By looking at the best model by AIC (Table 8.1), we find that
 7450 the model with sex specific density and sex-specific baseline encounter probability, p_0 , is
 7451 preferred (Model 2). This is just slightly better than the null model (Model 0) with no
 7452 sex effects at all and hence an implied fixed sex ratio of $\psi_{\text{sex}} = 0.50$.

Table 8.1. Model selection results for the wolverine models of sex specificity, with/without habitat mask. Fitting was done using `secr` with a half-normal (Gaussian) encounter probability model. Models are ordered by *AIC*. Density, *D*, is reported in units of individuals per 1000 km^2 . Model abbreviations indicate which parameters are sex-specific in order $D/p_0/\sigma$.

NO HABITAT MASK										
model	npar	Female			Male			D	p_0	σ
		AIC	AICc	D	p_0	σ				
2: sex/sex/1	5	1259.8	1263.8	2.45	0.08	6435.51	3.16	0.04	6435.51	
0: 1/1/1	3	1260.5	1261.9	2.83	0.06	6298.66	2.83	0.06	6298.66	
4: sex/sex/sex	6	1261.3	1267.3	2.59	0.08	6080.70	2.99	0.04	6833.16	
1: sex/1/1	4	1262.5	1265.0	2.69	0.06	6298.69	2.96	0.06	6298.69	
3: sex/1/sex	5	1264.5	1268.5	2.70	0.06	6280.49	2.95	0.06	6319.03	
WITH HABITAT MASK										
model	npar	Female			Male			D	p_0	σ
		AIC	AICc	D	p_0	σ				
2: sex/sex/1	5	1268.1	1272.1	3.64	0.07	6382.88	4.73	0.03	6382.88	
4: sex/sex/sex	6	1268.7	1274.7	3.87	0.07	5859.40	4.41	0.03	7039.09	
0: 1/1/1	3	1271.2	1272.6	4.18	0.05	6282.62	4.18	0.05	6282.62	
1: sex/1/1	4	1273.1	1275.6	3.98	0.05	6282.65	4.38	0.05	6282.65	
3: sex/1/sex	5	1275.1	1279.1	3.93	0.05	6357.26	4.41	0.05	6220.22	

7453 We fit the same models but now using a modified state-space which excludes the ocean
 7454 (this is a habitat mask in `secr`). Results are shown in Table 8.1 along with the previous
 7455 models without a mask. We see AIC values are smaller for the model without the mask.
 7456 It is probably acceptable to compare these different fits (with and without habitat mask)
 7457 by AIC because we recognize the mask as having the effect of modifying the random
 7458 effects distribution (i.e., of the activity centers, *s*) and the results should be sensitive to
 7459 choice of the distribution for *s*. That said, we tend to prefer the mask model because it
 7460 makes sense to exclude the areas of open water from the state-space of *s*. For females the
 7461 model-averaged density is 3.88 individuals per 1000 km^2 and for males the model-averaged
 7462 density estimate is 4.46 individuals per 1000 km^2 as we see here:

7463 > `model.average (model0b,model1b,model2b,model3b,model4b,realnames="D")`

```

7464
7465      estimate   SE.estimate      lcl      ucl
7466 session=1 3.876615e-05 1.189102e-05 2.153795e-05 6.977518e-05
7467 session=2 4.459658e-05 1.323696e-05 2.523280e-05 7.882022e-05

```

7468 This is quite a bit higher than that based on the rectangular state-space (i.e., not
 7469 specifying a habitat mask). This is not surprising given that **the state-space is part**
 7470 **of the model** and the specific state-space modification we made here, which reduces the
 7471 area from the rectangular state-space, should be extremely important from a biological
 7472 standpoint (i.e., wolverines are not actively using open ocean).

8.2 BAYESIAN MODEL SELECTION

7473 Model selection is somewhat less straightforward as a Bayesian, and there is no canned
 7474 all-purpose method like AIC. As such we recommend a pragmatic approach, in general,
 7475 for all problems, based on a number of basic considerations:

- 7476 (1) For a small number of fixed effects we think it is reasonable to adopt a conventional
 7477 “hypothesis testing” approach – i.e., if the posterior for a parameter overlaps zero
 7478 substantially, then it is probably reasonable to discard that effect from the model.
- 7479 (2) Calculation of posterior model probabilities: In some cases we can implement methods
 7480 which allow calculation of posterior model probabilities. One such idea is the indicator
 7481 variable selection method from Kuo and Mallick (1998). For this, we introduce a latent
 7482 variable $w \sim \text{Bern}(.5)$ and expand the model to include the variable w as follows:

$$\text{logit}(p_{ijk}) = \alpha_0 + w * \alpha_1 * C_{ijk}.$$

7483 The importance of the covariate C is then measured by the posterior probability that
 7484 $w = 1$.

7485 (3) The Deviance Information Criterion (DIC): Bayesian model selection is now routinely
 7486 carried out using DIC ((Spiegelhalter et al., 2002)), although its effectiveness in hier-
 7487 archical models depends very much on the manner in which it is constructed (Millar,
 7488 2009). We recommend using it if it leads to sensible results, but we think it should be
 7489 calibrated to the extent possible for specific classes of models. This has not yet been
 7490 done in the literature for SCR models, to our knowledge.

7491 (4) Logical argument: For something like sex specificity of certain parameters, it seems
 7492 to make sense to leave an extra parameter in the model no matter what because, bio-
 7493 logically, we might expect a difference (e.g., home range size). In some cases failure to
 7494 apply logical argument leads to meaningless tests of gratuitous hypotheses (Johnson,
 7495 1999).

7496 In all modeling activities, as in life itself, the use of logical argument should not be under-
 7497 utilized.

8.2.1 Model selection by DIC

7499 The availability of AIC makes the use of likelihood methods convenient for problems where
 7500 likelihood estimation is achievable. For Bayesian analysis, DIC seemed like a general-
 7501 purpose equivalent, at least for a brief period of time after its invention. However, there

7502 seem to be many variations of DIC, and a consistent version is not always reported across
 7503 computing platforms. Even statisticians don't have general agreement on practical issues
 7504 related to the use of DIC (Millar, 2009). Despite this, it is still widely reported. We think
 7505 DIC is probably reasonable for certain classes of models that contain only fixed effects,
 7506 or for which the latent variable structure is the same across models so that only the fixed
 7507 effects are varied (this covers many SCR model selection problems). However, it would be
 7508 useful to see some calibration of DIC for some standardized model selection problems.

7509 Model deviance is defined as negative twice the log-likelihood; i.e., for a given model
 7510 with parameters θ : $\text{Dev}(\theta) = -2 * \log L(\theta|\mathbf{y})$. The DIC is defined as the posterior mean
 7511 of the deviance, $\overline{\text{Dev}}(\theta)$, plus a measure of model complexity, p_D :

$$\text{DIC} = \overline{\text{Dev}}(\theta) + p_D$$

7512 The standard definition of p_D is

$$p_D = \overline{\text{Dev}}(\theta) - \text{Dev}(\bar{\theta})$$

7513 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.
 7514 Gelman et al. (2004) suggest a different version of p_D based on one-half the posterior
 7515 variance of the deviance:
 7516

$$p_V = \text{Var}(\text{Dev}(\theta)|\mathbf{y})/2.$$

7517 This is what is produced from **WinBUGS** and **JAGS** if they are run from **R2WinBUGS** or
 7518 **R2jags**, respectively. It is less easy to get DIC summaries from **rjags**, so we used **R2jags**
 7519 in our analyses below.

7520 8.2.2 DIC analysis of the wolverine data

7521 We repeated the analysis of the wolverine models with sex specificity, but this time doing
 7522 a Bayesian analysis paralleling the likelihood analysis we did above in **secr**, using the
 7523 logit/log parameterization of the model parameters. To do so in **BUGS**, we used dummy
 7524 variables. Thus, we can express models allowing for sex specificity using a dummy variable
 7525 **Sex** and new parameters (α_{sex} , β_{sex}) which represent the effect of **Sex** at level 1:

$$\text{logit}(p_{0,i}) = \alpha_0 + \alpha_{sex} \mathbf{Sex}_i$$

7526 and

$$\log(\sigma_i) = \log(\sigma_0) + \beta_{sex} \mathbf{Sex}_i.$$

7527 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.

7528 Unlike the multi-session model in **secr**, we carry out the analysis of the sex-specific
 7529 model here by putting all of the data into a single data set, and explicitly accounting for
 7530 the covariate 'sex' in the model by assigning it a Bernoulli prior distribution with ψ_{sex}
 7531 being the proportion of males in the population. In this case, we produce "Model 0" above,
 7532 the model with no sex effect on density, by setting the population proportion of males at
 7533 one-half: $\psi_{sex} = 0.5$ (see also Sec. 7.2.4). As usual, handling of missing values of the
 7534 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:

```
7548 > wolv0 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=0)
7549 > wolv1 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=1)
7550 > wolv2 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=2)
7551 > wolv3 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=3)
7552 > 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.

Model	Meandev	p_D	DIC	Rank
Model 0	441.01	77.09	518.10	1
Model 1	441.78	77.504	519.28	3
Model 2	440.12	78.440	518.56	2
Model 3	443.31	79.478	522.79	5
Model 4	441.24	80.078	521.32	4

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 (2009b) 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, activity centers, etc..
  s[i,1] ~ dunif(Xl,Xu)
  s[i,2] ~ dunif(Yl,Yu)
  logit(p0[i]) <- alpha0 + alpha.sex*wsex[i]
  log(sigma.vec[i]) <- log(sigma0) + beta.sex*wsex[i]
  alpha1[i] <- 1/(2*sigma.vec[i]*sigma.vec[i])
  for(j in 1:ntraps){
    mu[i,j] <- z[i]*p[i,j]
    y[i,j] ~ dbin(mu[i,j],K[j])
    dd[i,j] <- pow(s[i,1] - traplocs[j,1],2) + pow(s[i,2] - traplocs[j,2],2)
    p[i,j] <- p0[i]*exp( - alpha1[i]*dd[i,j] )
  }
}

```

Panel 8.1: Part of the **BUGS** specification for a complete sex specificity of model parameters. This is a simplified version of the model contained in the **wolvSCR0ms** script, because it does not contain the on/off switches for creating the various sub-models.

Table 8.3. Posterior summaries of model parameters for models with varying sex specificity of model parameters. Model 0 = no sex specificity, model 4 = fully sex-specific (see text). Models are based on the Gaussian encounter probability model, each with 21000 iterations, 1000 burn-in, 3 chains for a total of 60000 posterior samples.

Parameter	model 0		model 1		model 2		model 3		model 4	
	Mean	SD								
N	60.02	11.91	60.24	11.93	59.37	11.97	59.67	11.97	58.77	11.75
D	5.79	1.15	5.81	1.15	5.72	1.15	5.75	1.15	5.66	1.13
α_0	-2.81	0.18	-2.82	0.17	-2.44	0.25	-2.82	0.18	-2.43	0.25
α_{sex}	0.00	1.73	0.00	1.73	-0.75	0.34	0.00	1.73	-0.79	0.36
σ_0	0.64	0.06	0.64	0.05	0.66	0.06	0.65	0.08	0.63	0.09
β_{sex}	0.00	1.73	-0.01	1.73	0.01	1.74	-0.01	0.17	0.10	0.18
ψ_{sex}	0.50	0.29	0.52	0.10	0.56	0.10	0.52	0.11	0.54	0.11
ψ	0.30	0.07	0.30	0.07	0.30	0.07	0.30	0.07	0.30	0.07
deviance	441.01	12.42	441.78	12.45	440.12	12.53	443.31	12.61	441.24	12.66
	$pD = 77.1$		$pD = 77.5$		$pD = 78.4$		$pD = 79.5$		$pD = 80.1$	
	$DIC = 518.1$		$DIC = 519.3$		$DIC = 518.6$		$DIC = 522.8$		$DIC = 521.3$	

the different models because they are meaningless (averaging apples and oranges....). For example, if a regression parameter is in a specific model then the posterior is informed by the data and a specific MCMC draw is from the appropriate posterior distribution. On the other hand, if the regression parameter is not in the model then the MCMC draw is obtained directly from the prior distribution, and so we need to think carefully about whether it makes sense to report an average of such a thing (in the vast majority of cases the answer is no). But some parameters like N or density, D , do have a consistent interpretation and we support producing model-averaged results of those parameters.

To implement the Kuo and Mallick approach, we expand the model to include the latent indicator variables, say w_m , for variable m in the model, such that

$$w_m = \begin{cases} 1 & \text{linear predictor includes covariate } m \\ 0 & \text{linear predictor does not include covariate } m \end{cases}$$

We assume that the indicator variables w_m are mutually independent with

$$w_m \sim \text{Bernoulli}(0.5)$$

for each variable $m = 1, 2, \dots$, in the model. For example, with 2 variables, the expanded model has the linear predictor:

$$\text{logit}(p_{ijk}) = \alpha_0 + \alpha_1 w_1 C_{1,i} + \alpha_2 w_2 C_{2,ijk}$$

where, let's suppose, $C_{1,i}$ is an individual covariate such as sex, and $C_{2,ijk}$ is a behavioral response covariate which is individual-, trap-, and occasion-specific. We can assume a parallel model specification on the parameter σ which is liable to vary by individual level covariates such as sex:

$$\log(\sigma_i) = \beta_0 + \beta_1 w_3 C_{1,i}.$$

Using this indicator variable formulation of the model selection problem we can characterize unique models by the sequence of w variables. In this case, each unique sequence (w_1, w_2, w_3) represents a model, and we can tabulate the posterior frequencies of each model by post-processing the MCMC histories of (w_1, w_2, w_3) , as we demonstrate shortly. This method then evaluates all possible combinations of covariates or 2^m models.

Conceptually, analysis of this expanded model within the data augmentation framework does not pose any additional difficulty. One broader, technical consideration is that posterior model probabilities are well known to be sensitive to priors on parameters (Aitkin, 1991; Link and Barker, 2006). See also Royle and Dorazio (2008, Sec. 3.4.3) and Link and Barker (2010, Sec. 7.2.5). What might normally be viewed as vague or non-informative priors, are not usually innocuous or uninformative when evaluating posterior model probabilities. The use of AIC seems to avoid this problem largely by imposing a specific and perhaps undesirable prior that is a function of the sample size (Kadane and Lazar, 2004). One solution is to compute posterior model probabilities under a model in which the prior for parameters is fixed at the posterior distribution under the full model (Aitkin, 1991). At a minimum, one should evaluate the sensitivity of posterior model probabilities to different prior specifications.

Analysis of the wolverine data

The **R** script `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:

```

7618 > mod <- wolv5$BUGSoutput$sims.list$mod
7619 > mod <- paste(mod[,1],mod[,2],mod[,3],sep="")
7620 >
7621 > table(mod)
7622 mod
7623   000   001   010   011   100   101   110   111
7624 17181 4935 1057 296 25211 8337 2275 708
7625
7626 > round( table(mod)/length(mod) , 3)
7627 mod
7628   000   001   010   011   100   101   110   111
7629 0.286 0.082 0.018 0.005 0.420 0.139 0.038 0.012

```

7630 This results in a comparison of all 8 possible models (based on $m = 3$ covariates) instead
 7631 of just the 5 models we originally proposed. We see that the best model is that labeled
 7632 100 which, according to our construction above, has `mod[1]=1, mod[2]=0` and `mod[3]=0`.
 7633 This is the model having sex-specific baseline encounter probability p_0 , and $\psi_{sex} = 0.5$.
 7634 This model has posterior model probability 0.420. The model with no sex specificity at
 7635 all (the model with label 000) has posterior probability 0.286 and the remaining posterior
 7636 mass is distributed over the other six models. We could arrive at a qualitatively similar
 7637 conclusion using a more ad hoc approach based on looking at the posterior mass for each
 7638 parameter under the full model (model 4; see Table 8.3, in part). Considering the sex-
 7639 specific intercept, it appears to be very important as its posterior mass is mostly away
 7640 from 0. On the other hand, the coefficient on log-sigma is concentrated around 0, and
 7641 the estimated ψ_{sex} (probability that an individual is a male) is 0.54 with a large posterior
 7642 standard deviation. We might therefore be inclined to discard the sex effect on $\log(\sigma)$
 7643 based on classical thinking-like-a-hypothesis-testing-person and settle for the model with
 7644 a sex-specific intercept in the encounter probability model. This is consistent with our
 7645 indicator variable approach which found that model (1,0,0) has posterior probability of
 7646 0.420. Looking at the posteriors for each parameter to thin the model down is consistent
 7647 with these results. We can obtain model-averaged estimates from the indicator variable
 7648 approach, which produces direct model-averaged estimates of N and D :

```
7649   mu.vect sd.vect  2.5%   25%   50%   75% 97.5% Rhat n.eff
7650 D     5.695   1.133  3.759  4.916  5.591  6.362  8.193 1.002 3600
7651 N    59.077  11.758 39.000 51.000 58.000 66.000 85.000 1.002 3600
```

7652 We obtain a model-averaged estimate (posterior mean) for density of $D = 5.695$ which
 7653 is hardly any different from our model specific estimates (Table 8.3) and, in particular,
 7654 from model 2 which has only a sex-specific intercept.

7655 8.2.4 Choosing among detection functions

7656 Another approach to implementing model indicator variables is to introduce a categorical
 7657 “model identity” variable which is itself a parameter of the model. Using this approach,
 7658 then each distinct model is associated with a unique set of covariates or other set of model
 7659 features. This is convenient especially when we cannot specify the linear predictor as
 7660 some general model that reduces to various alternative sub-models simply by switching
 7661 binary variables on or off. In the context of SCR models, choosing among different en-
 7662 counter probability models would be an example. For this case we do something like this
 7663 `mod ~ dcat(probs[])` where `probs` is a vector with elements $1/(\#models)$, and the en-
 7664 counter probability matrix is filled in depending on the value of `mod`. In particular, instead
 7665 of a 2-dimensional array `p[i,j]`, we build `p[i,j,m]` for each of $m = 1, 2, \dots, M$ models.
 7666 An example with 3 distinct models is:

```
7667   mod ~ dcat(probs[])
7668   ##
7669   ## Using a double loop construction fill-in p[,] for each model:
7670   ##
7671   p[i,j,1] <- p0[1]*exp( - alpha1[1]*dist2[i,j] )
```

```

7672 p[i,j,2] <- 1-exp(-p0[2]*exp( - alpha1[2]*dist2[i,j] ) )
7673 logit(p[i,j,3]) <- p0[3] - alpha1[3]*dist2[i,j]
7674
7675 mu[i,j] <- z[i]*p[i,j,mod]
7676 y[i,j] ~ dbin(mu[i,j],K[j])

```

7677 As before the posterior probabilities can be highly sensitive to priors on the different
 7678 model parameters and sometimes mixing is really poor and, in general, we've experienced
 7679 mixed success trying to carry out model selection using this construction. We do provide
 7680 a template **R/JAGS** script (`wolvSCR0ms2`) in the `scrbook` package which has an example
 7681 of choosing among 3 different encounter probability models: The Gaussian encounter
 7682 probability, Gaussian hazard, and logistic model with the square of distance (defined
 7683 in Sec. 7.1). The key things to note are that there are 3 intercepts and 3 different
 7684 ‘`alpha1`’ parameters (the coefficient on distance). The parameters should not be regarded
 7685 as equivalent across the models, so it is important to have them separately defined (and
 7686 estimated) for each model. In our analysis we used a vague normal prior (precision = 0.1)
 7687 for the intercept parameter (either log or logit-scale of baseline encounter probability p_0)
 7688 and a `Uniform(0,5)` prior for one-half the inverse of the coefficient on distance-squared. In
 7689 the **BUGS** model specification the priors look like this:

```

7690 for(i in 1:3){
7691   alpha0[i] ~ dnorm(0,.1)
7692   sigma[i] ~ dunif(0,5)
7693   alpha1[i] <- 1/(2*sigma[i]*sigma[i])
7694 }

```

7695 Then, we create a probability of encounter for each individual, trap *and* model so that
 7696 the holder object “`p`” in the model description is a 3-dimensional array (sometimes this
 7697 would have to be a 4 or 5-d array in more complex models with time effects, etc..), so that
 7698 construction of the encounter probability models look like this:

```

7699 p[i,j,1] <- p0[1]*exp( - alpha1[1]*dist2[i,j] )
7700 p[i,j,2] <- 1-exp(-p0[2]*exp( - alpha1[2]*dist2[i,j] ) )
7701 logit(p[i,j,3]) <- p0[3] - alpha1[3]*dist2[i,j]

```

7702 where

```

7703 logit(p0[1]) <- alpha0[1]
7704 log(p0[2]) <- alpha0[2]
7705 p0[3] <- alpha0[3]

```

7706 You can experiment with the `wolvSCR0ms2` script to investigate the importance of different
 7707 models of encounter probability and whether they have an affect on the inferences.

8.3 EVALUATING GOODNESS-OF-FIT

7708 In practical settings, we estimate parameters of a desirable model, or maybe fit a bunch
 7709 of models and report estimates from all of them or a model-averaged summary of density.

7710 An important question is: Is our model worth anything? In other words, does the model
7711 appear to be an adequate description of our data? Formal assessment of model adequacy or
7712 goodness-of-fit is a challenging problem and there are no all-purpose algorithms for doing
7713 this in either frequentist or Bayesian paradigms. Moreover, there are some philosophical
7714 challenges to evaluating model fit, such as, if we do model averaging then should all of
7715 the models have to fit? Or should the averaged model have to fit? What if none of the
7716 models fit? We don't know the answers to these questions and we won't try to answer
7717 them. Instead, we will provide what guidance we can on taking the first steps to evaluating
7718 fit, of a single model, as if it were a cherished family heirloom of great importance. We
7719 suggest that if you have a model that you really like, a single model, then it is a sensible
7720 thing to check that the model is a good fit to your data. If it is not, we do not imagine
7721 that the model is useless but just that some thought should be put into why the model
7722 doesn't fit so that, perhaps, some remediation might happen as future data are collected.
7723 After all, you may have spent 2, 3 or many more years of your life collecting that data set,
7724 perhaps thousands of hours, and therefore it seems a reasonable proposition to expect to
7725 do some estimation and analysis of the model regardless of model fit. You can still learn
7726 something from a model that does not pass some technical litmus test of model fit.

7727 Conceptually, we can think of evaluation of model fit as follows: if we simulate data
7728 under the model in question, do the simulated realizations resemble the data set that we
7729 actually have? For either Bayesian or classical inference, the basic strategy to assessing
7730 model fit is to come up with a fit statistic that depends on the parameters and the data
7731 set, which we denote by $T(\mathbf{y}, \theta)$, and then we compute this for the observed data set, and
7732 compare its value to that computed for perfect data sets simulated under the correct model.
7733 In the case of classical inference, we will often rely on the standard practice of parametric
7734 bootstrapping (Dixon, 2002), where we simulate data sets conditional on the MLE $\hat{\theta}$ and
7735 compare realizations with what we've observed. The R package **unmarked** (Fiske and
7736 Chandler, 2011) contains generic bootstrapping methods for certain hierarchical models,
7737 including distance sampling (e.g., see Sillett et al., 2012, for an application). In simple
7738 cases, using classical inference methods, it is sometimes possible to identify a test statistic
7739 of theoretical merit, perhaps with a known asymptotic distribution. For examples from
7740 capture-recapture see Burnham et al. (1987), Lebreton et al. (1992), and Chapt. 5 of
7741 Cooch and White (2006). For Bayesian analysis we use the Bayesian p-value method
7742 (Gelman et al., 1996) (we introduced the Bayesian p-value in sec. 3.9.1). Using this
7743 approach, data sets are simulated based on a posterior sample of the model parameters
7744 θ and some fit statistic for the simulated data sets, usually based on the discrepancy of
7745 the observed data from its expected values, is compared to that for the actual data. In
7746 most cases, whether Bayesian or frequentist, the main idea for assessing model fit is the
7747 same: We compare data sets from the model we're interested in with the data set we have
7748 in hand. If they appear to be consistent with one another, then our faith in the model
7749 increases, at least to some extent, and we say "the model fits."

7750 To date, we are unaware of any goodness-of-fit applications based on likelihood analysis
7751 of SCR models. For Bayesian analysis of SCR models, there has not been a definitive or
7752 general proposal for a fit statistic or even a class of fit statistics, although a few specialized
7753 implementations of Bayesian p-values have been provided (Royle, 2009b; Gardner et al.,
7754 2010a; Royle et al., 2011a; Gopalaswamy et al., 2012a,b; Russell et al., 2012). While
7755 we universally adopt the Bayesian p-value approach, and suggest some fit statistics in

the following text, we caution that there is no general expectation to support how well they should do. As such, one might consider doing some kind of custom evaluation or calibration when using such methods, if the power of the test (ability to reject under specific departures from the model) is of paramount interest. We note that this uncertain power or performance of the Bayesian p-value is not a weakness of the Bayesian approach because the same issue applies in using bootstrap approaches applied to classical analysis of models, if we were to devise such methods.

8.4 THE TWO COMPONENTS OF MODEL FIT

For most SCR models, there are at least two distinct components of model fit, and we propose to evaluate these two distinct components individually. First, we can ask, are the data consistent with the *observation* model, conditional on the underlying point process? We can evaluate this based on the encounter frequencies of individuals *conditional* on (posterior samples of) the underlying point process $\mathbf{s}_1, \dots, \mathbf{s}_N$. We discuss some potential fit statistics for addressing this in the next section. Second, we can evaluate whether the data appear consistent with the *state* process model (i.e., the “uniformity” assumption of the point process). For the simple model of independence and uniformity, this is similar to the assumption of *complete spatial randomness* (CSR) which we consider in Sec. 8.4.1 below. Actually, this is not strictly the assumption of CSR because of the binomial assumption on N under data augmentation, so we instead use the term *spatial randomness*.

8.4.1 Testing uniformity or spatial randomness

Historically, especially in ecology, there has been an extraordinary amount of interest in whether a realization of a point process indicates “complete spatial randomness,” i.e., that the points are distributed uniformly and independently in space. Two good references for such things are Cressie (1991, Ch. 8) and Illian et al. (2008)¹. In the context of animal capture-recapture studies, the spatial randomness hypothesis is manifestly false, purely on biological grounds. Typically individuals will be clustered, or more regular (for territorial species), than expected under spatial randomness and heterogeneous habitat will generate the appearance of clustering even if individuals are distributed independently of one another. While we recommend modeling spatial structure explicitly when possible (Chapts. 11, 12, 13), the uniformity assumption may be an adequate description of data sets in some situations. Further, we find that it is generally flexible enough to reflect non-uniform patterns in the data, because we do observe some direct information about some of the point locations.

The basic technical framework for evaluating the spatial randomness hypothesis is based on counts of activity centers in cells or bins. For that we use any standard goodness-of-fit test statistic, based on gridding (i.e., binning) the state-space of the point process into $g = 1, 2, \dots, G$ cells or bins, and we tabulate $N_g \equiv N(\mathbf{x}_g)$ the number of activity centers in bin g , centered at coordinate \mathbf{x}_g . Specifically, let $B(\mathbf{x})$ indicate a bin centered at coordinate

¹We also like Tony Smith’s lecture notes (Univ. of Penn. ESE 502), which can be found at http://www.seas.upenn.edu/~ese502/NOTEBOOK/Part_I/3_Testing_Spatial_Randomness.pdf, accessed January 24, 2013.

7793 \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,
 7794 we used the summaries $N(\mathbf{x})$ for producing density maps from MCMC output. Here, we
 7795 use them for constructing a fit statistic. We have used the Freeman-Tukey statistic of this
 7796 form:

$$T(\mathbf{N}, \theta) = \sum_g (\sqrt{N_g} - \sqrt{\mathbb{E}(N_g)})^2$$

7797 where $\mathbb{E}(N_g)$ is estimated by the mean bin count. An alternative conventional assessment
 7798 of fit is based on the following statistic: Conditional on N , the total number of activity
 7799 centers in the state-space \mathcal{S} , the bin counts N_g should have a binomial distribution. It will
 7800 usually suffice to approximate the binomial cell counts by Poisson cell counts, in which
 7801 case we can use the classical “index-of-dispersion” test (Illian et al., 2008, p. 87), based
 7802 on the variance-to-mean ratio:

$$ID = (G - 1) * s^2 / \bar{N}$$

7803 where s^2 is the sample variance of the bin counts and \bar{N} is the sample mean. When the
 7804 point process realization is *observed*, as in classical point pattern modeling (but not in
 7805 SCR), this statistic has approximately a Chi-square distribution on $(G - 1)$ degrees-of-
 7806 freedom under the spatial randomness hypothesis. If $s^2 / \bar{N} > 1$, clustering is suggested
 7807 whereas, $s^2 / \bar{N} < 1$ suggests the point process is too regular.

7808 Whatever statistic we choose as our basis for assessing spatial randomness, *the im-*
 7809 *portant technical issue is that we don’t observe the point process and so the standard*
 7810 *statistics for evaluating spatial randomness cannot be computed directly. However, using*
 7811 *Bayesian analysis, we do have a posterior sample of the underlying point process and*
 7812 *so we suggest computing the posterior distribution of any statistic in a Bayesian p-value*
 7813 *framework. For a given posterior draw of all model parameters, N is known, based on the*
 7814 *value of the data augmentation variables z_i , and so we can obtain a posterior sample of*
 7815 *$N(\mathbf{x})$ by taking all of the output for MCMC iterations $m = 1, 2, \dots$, and doing this:*

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

7816 Thus, $N(\mathbf{x})^{(1)}, N(\mathbf{x})^{(2)}, \dots$, is the Markov chain for the derived parameter $N(\mathbf{x})$.

7817 In addition to computing the bin counts for each iteration of the MCMC algorithm,
 7818 at the same time we generate a realization of the activity centers \mathbf{s}_i under the spatial
 7819 randomness model, and we obtain bin counts for these “new” data, $\tilde{N}(\mathbf{x})$. For each of
 7820 the posterior samples – that of the real data, and that of the posterior simulated data, we
 7821 compute the fit-statistic. The fit statistic based on the actual data is:

$$T(\mathbf{N}, \theta) = \sum_x (\sqrt{N(\mathbf{x})} - \sqrt{\tilde{N}(\mathbf{x})})^2$$

7822 whereas the fit statistic based on a simulated realization of points under the spatial ran-
 7823 domness hypothesis is:

$$T(\tilde{\mathbf{N}}, \theta) = \sum_x (\sqrt{\tilde{N}(\mathbf{x})} - \sqrt{\tilde{N}(\mathbf{x})})^2$$

² $I(arg)$ is the indicator function which evaluates to 1 if arg is true, otherwise 0

7824 And we compute the Bayesian p-value by tallying up the proportion of times that $T(\tilde{\mathbf{N}}, \theta)$
 7825 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
 7826 **SCRgof** in our package **scrbook** will do this, given the output from **JAGS** (see below).

7827 Sensitivity to bin size

7828 Evaluating fit based on bin counts in point process models are sensitive to the number of
 7829 bins (Illian et al., 2008, p. 87-88). This is related to the classical problem of fit testing
 7830 for binary regression because in a point process model, as the number of grid cells gets
 7831 small, the grid cell counts go to 0 or 1 and standard fit statistics (e.g., based on deviance
 7832 or Pearson residuals) are known not to be very useful. There is some good discussion of
 7833 this in McCullagh and Nelder (1989, Sec. 4.4.5). What it boils down to is, using the
 7834 example of the Pearson residual statistic considered by McCullagh and Nelder (1989), the
 7835 fit statistic is exactly a deterministic function of the sample size only, which clearly should
 7836 not be regarded as useful for model fit. This is why, in order to do a check of model fit
 7837 when you have a binary response, one must always aggregate the data in some fashion. In
 7838 the context of testing spatial randomness, computing the test statistic we described above
 7839 has us chop up the region \mathcal{S} into bins, and tally up N_g , the frequency of activity centers
 7840 in each bin g . Suppose that we choose the bin size to be extremely small such that $\mathbb{E}(N_g)$
 7841 tends to N/G (N being the number of activity centers). Further, N_g tends to a binary
 7842 outcome. Therefore the fit statistic has N components that have value $N_g = 1$, and it has
 7843 $G - N$ components that have value $N_g = 0$. Therefore, the fit statistic resembles:

$$T(\mathbf{N}, \theta) = \sum_{g \ni N_g=1}^N (1 - \sqrt{N/G})^2 + \sum_{g \ni N_g=0}^{G-N} (N/G)^2 = N(1 + (G - N)/G)$$

7844 (here \ni means “such that”). If G is huge relative to N , then we see that this tends to
 7845 about $2 * N$, which does not provide any meaningful assessment of model fit. So if you
 7846 look at this in the limit in which the bin counts become binary, the fit statistic loses all
 7847 its variability to the specific model used and is just a deterministic function of N . As a
 7848 practical matter, it probably makes sense to restrict the number of bins to *fewer* than the
 7849 number of observed individuals in the sample size. In typical SCR applications this will
 7850 therefore result, usually, in very large (and few) bins, and presumably not much power.

7851 There are some extensions that help resolve the issue of sensitivity to bin size. We can
 7852 construct fit statistics based not just on quadrat counts but also the neighboring quadrat
 7853 counts – this is the Greig-Smith method (Greig-Smith, 1964). In addition, there are a
 7854 myriad of “distance methods” for evaluating point process models, and we believe that
 7855 many of these can (and will) be adapted to SCR models. Again the main feature is that
 7856 the point process on which inference is focused is completely latent in SCR models – so
 7857 this makes the fit assessment slightly different than in classical point processes. That said,
 7858 the methods should be adaptable, e.g., in a Bayesian p-value kind of way.

7859 Sensitivity to state-space extent

7860 An issue that we have not investigated is that any model assessment that applies to a *latent*
 7861 point process is probably sensitive to the size of the state-space. As the size of the state-
 7862 space increases then the cell counts (far away from the data) *are* independent binomial
 7863 counts with constant density, and so we can overwhelm the fit statistic with extraneous
 7864 “data” simulated from the posterior, which is equal to the prior as we move away from the

7865 data, and therefore uninformed by the observed data that live in the vicinity of the trap
 7866 array. Therefore we recommend computing these goodness-of-fit statistics in the vicinity
 7867 of the trap array only. Perhaps, as an ad hoc rule-of-thumb, less than the average trap
 7868 spacing from the rectangle enclosing the trap array. For example, if the average trap
 7869 spacing is, say, 10 km, then the bins used to obtain the observed and predicted activity
 7870 centers should not extend any further from the traps than 5 km. This should be a matter
 7871 of future research.

7872 **8.4.2 Assessing fit of the observation model**

7873 In evaluating the spatial randomness hypothesis, we could draw on well-established ideas
 7874 from point process modeling. On the other hand, it is less clear how to approach goodness-
 7875 of-fit evaluation of the observation model. For most SCR problems, we have a 3-dimensional
 7876 data array of *binary* observations, y_{ijk} for individual i , trap j and sample occasion k . As
 7877 discussed in the previous section, we need to construct fit statistics based on observed and
 7878 expected frequencies that are aggregated in some fashion. In practice, the data will be
 7879 too sparse to have much power, unless the data are highly aggregated. We recommend
 7880 focusing on summary statistics that represent aggregated versions of y_{ijk} over 1 or 2 of
 7881 the dimensions. We describe 3 such fit statistics below. We recognize that, depending on
 7882 the model, some information about model fit will be lost by summarizing the data in this
 7883 way. For example if there is a behavioral response and we aggregate over time to focus
 7884 on the individual and trap level summaries then some information about lack of fit due
 7885 to temporal structure in the data is lost.

7886 **Fit statistic 1: individual x trap frequencies** We summarize the data by indi-
 7887 vidual and trap-specific counts y_{ijk} aggregated over all sample occasions. Using standard
 7888 “dot notation” to represent summed quantities, we express that as: $y_{ij\cdot} = \sum_{k=1}^K y_{ijk}$.
 7889 Conditional on \mathbf{s}_i , the expected value under any encounter model is:

$$\mathbb{E}(y_{ij\cdot}) = p_{ij} K$$

7890 (or K_j if the traps are operational for variable periods). If there is time-varying structure
 7891 to the model, then expected values would have to be computed according to $\mathbb{E}(y_{ij\cdot}) =$
 7892 $\sum_k p_{ijk}$. Then we can define a fit statistic from the Freeman-Tukey residuals according
 7893 to:

$$T_1(\mathbf{y}, \theta) = \sum_i \sum_j (\sqrt{y_{ij\cdot}} - \sqrt{\mathbb{E}(y_{ij\cdot})})^2$$

7894 where we use θ here to represent the collection of all parameters in the model. This is
 7895 conditional on \mathbf{s} as well as on the data augmentation variables \mathbf{z} . We compute this statistic
 7896 for *each* iteration of the MCMC algorithm for the observed data set and also for a new
 7897 data set simulated from the posterior distribution, say $\hat{\mathbf{y}}$.

7898 We could also use a similar fit statistic derived from summarizing over traps to obtain
 7899 an $n_{ind} \times K$ matrix of count statistics. We imagine that either summary of the data will
 7900 probably be too disaggregated (have mostly values of 0) in most practical settings to have
 7901 much power.

7902 **Fit statistic 2: Individual encounter frequencies.** SCR models represent a
 7903 type of model for heterogeneous encounter probability, like model M_h , but with an ex-
 7904 plicit factor (space) that explains part of the heterogeneity. For model M_h , the individual

7905 encounter frequencies are the sufficient statistic for model parameters, and so it makes in-
 7906 tuitive sense to provide some kind of omnibus fit assessment of the core heuristic that SCR
 7907 model is adequately explaining the heterogeneity using a model M_h -like statistic based
 7908 on individual encounter frequencies. So, we build a fit statistic based on the individual
 7909 total encounters (Russell et al., 2012), $y_{i..} = \sum_j \sum_k y_{ijk}$. In addition, the expected value
 7910 is a similar summary over traps and occasions: $\mathbb{E}(y_{i..}) = \sum_j \sum_k p_{ijk}$. Then, we define
 7911 statistic T_2 according to:

$$T_2(\mathbf{y}, \theta) = \sum_i (\sqrt{y_{i..}} - \sqrt{\mathbb{E}(y_{i..})})^2$$

7912 We imagine this test statistic should provide an omnibus test of extra-binomial variation
 7913 and should therefore capture some effect of variable exposure to encounter of individuals,
 7914 although we have not carried out any evaluations of power under specific alternatives.
 7915 Obviously, in using this statistic, we lose information on departures from the model that
 7916 might only be trap- or time-specific.

7917 **Fit Statistic 3: Trap frequencies.** We construct an analogous statistic based
 7918 on aggregating over individuals and replicates to form trap encounter frequencies: $y_{.j} =$
 7919 $\sum_i \sum_k y_{ijk}$ (Gopalaswamy et al., 2012b) and the expected value is a similar summary
 7920 over individuals and occasions: $\mathbb{E}(y_{.j}) = \sum_i \sum_k p_{ijk}$. Then statistic T_3 is:

$$T_3(\mathbf{y}, \theta) = \sum_j (\sqrt{y_{.j}} - \sqrt{\mathbb{E}(y_{.j})})^2$$

7921 This seems like a sensible fit statistic because we can think of SCR models as spatial
 7922 models for counts (Chandler and Royle, In press). Therefore, we should seek models that
 7923 provide good predictions of the observable spatial data, which are the trap totals. In this
 7924 context, it might even make sense to pursue cross-validation based methods for model
 7925 selection. Cross-validation is a standard method of evaluating models such as in kriging
 7926 or spline smoothing, so we could as well develop such ideas based on the trap-specific
 7927 frequencies.

7928 8.4.3 Does the SCR model fit the wolverine data?

7929 We use the ideas described in the previous section to evaluate goodness-of-fit of the SCR
 7930 model to the wolverine camera trapping data.

7931 We consider first whether the simple model of spatial randomness of the activity
 7932 centers is adequate. We think that the encounter model shouldn't have a large effect
 7933 on whether the spatial randomness assumption is adequate or not, so we fit "Model 0"
 7934 (in which parameters are *not* sex-specific) using an **R** script provided in the function
 7935 **wolvSCR0gof** which will default to fitting the model in **JAGS**. This is the same script as
 7936 **wolvSCR0ms** except that it saves the MCMC output for the activity centers **s** and the data
 7937 augmentation variables **z**, which are required in order to compute the Bayesian p-value
 7938 test of spatial randomness.

7939 The MCMC output is processed with the **R** function **SCRgof** which computes the test
 7940 of spatial randomness based on bin counts, using the Bayesian p-value calculation. The
 7941 function **SCRgof** requires a few things as inputs: (1) the output from a **BUGS** run (in
 7942 particular, the activity center coordinates and the data augmentation variables); (2) the

7943 number of bins to create for computing spatial frequencies of activity centers; (3) the trap
 7944 locations and, (4) the buffer around the trap array to use in computing the bin counts.
 7945 This buffer could be that used in defining the state-space for the model fitting, but we
 7946 think it should be relatively tighter to the trap array than the state-space used in model-
 7947 fitting. For the wolverine analysis, where we're using 10-km grid cells (1 unit = 10 km)
 7948 and a 20 km buffer for model fitting, we'll use a state-space buffer of 0.4 units (4 km) for
 7949 computing the fit statistic. The **R** code to fit the model and obtain the goodness-of-fit
 7950 result is as follows:

```
7951 > wolv1 <- wolvSCR0gof(nb=1000,ni=6000,buffer=2,M=200,model=0)
7952
7953 > bugsout <- wolv1$BUGSoutput$sims.list
7954
7955 > traplocs <- wolverine$wtraps[,2:3]
7956 > traplocs[,1] <- traplocs[,1] - min(traplocs[,1])
7957 > traplocs[,2] <- traplocs[,2] - min(traplocs[,2])
7958 > traplocs <- traplocs/10000
7959
7960 > set.seed(2013) # set seed so Bayesian p-value is the same each time
7961
7962 > SCRgof(bugsout,5,5,traplocs=traplocs,buffer=.4)
7963
7964 Cluster index observed: 1.099822
7965 Cluster index simulated: 1.000453
7966 P-value index of dispersion: 0.408
7967 P-value2 freeman-tukey: 0.6842667
```

7968 The output produced by **SCRgof** is the index of dispersion based on the ratio of the variance to the mean (see above), which is computed as the posterior mean index of dispersion for the latent point process, and also the average value for simulated data. If this value is > 1 then clustering is suggested, which we see a (very) minor amount of evidence for here. Two Bayesian p-values are produced: the first is based on the cluster index, and the 2nd is based on the Freeman-Tukey statistic calculated as described in Sec. 8.4.1. Because our p-values aren't close to 0 or 1, we judge that the model of spatial randomness provides an adequate fit to the data. You can verify that a similar result is obtained if we use the model with fully sex-specific parameters (Model 4).

7977 Next, we did a Bayesian p-value analysis of the observation component of the model,
 7978 using the 3 fit statistics described in Sec. 8.4.2. These statistics can be calculated as
 7979 part of the **BUGS** model specification or by post-processing the MCMC output returned
 7980 from a **BUGS** run. The **R** script **wolvSCR0gof** contains the relevant calculations. For
 7981 example, to compute fit statistic 1, we have to add some commands to the **BUGS** model
 7982 specification such as this (note: this is only a fraction of the model specification):

```
7983 .....
7984 for(j in 1:ntraps){
7985   mu[i,j] <- w[i]*p[i,j]
7986   y[i,j] ~ dbin(mu[i,j],K[j])
```

```

7988 ynew[i,j] ~ dbin(mu[i,j],K[j])
7989
7990 err[i,j] <- pow(pow(y[i,j],.5) - pow(K[j]*mu[i,j],.5),2)
7991 errnew[i,j] <- pow(pow(ynew[i,j],.5) - pow(K[j]*mu[i,j],.5),2)
7992 }
7993
7994 Tlobs <- sum(err[,])
7995 Tnew <- sum(errnew[,])
7996 .....
7997 Similar calculations are carried out to obtain the posterior samples of test statistics 2
7998 (individual totals) and 3 (trap totals). For the wolverine data, the Bayesian p-value
7999 calculations produce:
```

```

8000 > mean(wolv1$BUGSoutput$sims.list$T1new>wolv1$BUGSoutput$sims.list$T1obs)
8001 [1] 0
8002
8003 > mean(wolv1$BUGSoutput$sims.list$T2new>wolv1$BUGSoutput$sims.list$T2obs)
8004 [1] 0.17
8005
8006 > mean(wolv1$BUGSoutput$sims.list$T3new>wolv1$BUGSoutput$sims.list$T3obs)
8007 [1] 0.02066667
```

8008 Based on statistic T_2 , we might conclude that the model is adequate for explaining
8009 individual heterogeneity although the other two statistics suggest a general lack of fit of
8010 the observation model. A similar result is obtained using the fully sex-specific model. We
8011 note that one individual was captured 8 times in one trap, which is pretty extreme under
8012 a model which assumes independent Bernoulli trials. We summarize that the trap-counts
8013 simply are not well-explained by this model.

8014 In attempt to resolve this problem, we extended the model to include a local (trap-
8015 specific) behavioral response (following Royle et al. (2011b)) which can be fitted using
8016 the sample **R** script **wolvSCRMb**. To fit a model using **WinBUGS**, and then compute the
8017 Bayesian p-values we do this:

```

8018 > wolv.Mb <- wolvSCRMb(nb=1000,ni=6000,buffer=2,M=200)
8019
8020 > mean(wolv.Mb$sims.list$T1new>wolv.Mb$sims.list$T1obs)
8021 [1] 0.9666667
8022
8023 > mean(wolv.Mb$sims.list$T2new>wolv.Mb$sims.list$T2obs)
8024 [1] 0.3644667
8025
8026 > mean(wolv.Mb$sims.list$T3new>wolv.Mb$sims.list$T3obs)
8027 [1] 0.4990667
```

8028 Given that this model seems to fit better, we might prefer reporting estimates under
8029 this model, which we do in Table 8.4. (the behavioral response parameter is labeled α_2
8030 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

8052 In this chapter, we offered some general strategies for model selection and model checking,
8053 or assessment of model fit. We think the strategies we outlined for model selection are fairly
8054 standard and can be effectively applied to many SCR modeling problems. Some technical
8055 issues of Bayesian analysis need to be addressed (in general) before Bayesian methods
8056 are more generally useful and accessible. For one thing, Bayesian model selection based
8057 on the indicator variable approach of Kuo and Mallick (1998) can be tediously slow even
8058 for small data sets, and so improved computation will improve our ability to do Bayesian
8059 model selection in practical situations. Also, and most importantly, sensitivity to prior
8060 distributions is an important issue. Further research and practice might identify preferred
8061 prior configurations for SCR that provide a good calibration in relevant model selection
8062 problems. Finally, we believe that cross-validation should prove to be a useful method
8063 in model assessment and selection, as SCR models are a form of spatial model of counts,
8064 and so it is natural to pick models that predict the observable spatial counts (i.e., at trap
8065 locations) well.

8066 For Bayesian model assessment, or goodness-of-fit checking, we suggested a framework
8067 based on independent testing of the spatial model of independence and uniformity, and
8068 testing fit of the observation model conditional on the underlying point process. These
8069 ideas are based on mostly *ad hoc* attempts in a number of published applications (Royle
8070 et al., 2009a, 2011a; Gopalaswamy et al., 2012b; Russell et al., 2012, e.g.). While we think
8071 this general strategy should be fruitful, we know of no studies on the power to detect
8072 various model departures, and so the ideas should be viewed as experimental. We have
8073 not discussed assessment of model fit for SCR models using likelihood methods, although
8074 we imagine that standard bootstrapping ideas should be effective, perhaps based on the
8075 fit statistics (or similar ones) we suggested here for computing Bayesian p-values.

8076 Clearly there is much research to be done on assessment of model fit in SCR models.
8077 For testing the spatial randomness hypothesis, we used a classical approach based on
8078 count frequencies, in which point locations are put into spatial bins. Other approaches
8079 from spatial point process modeling should be pursued including nearest-neighbor methods
8080 or distance-based methods. In addition, studies to evaluate the power to detect relevant
8081 departures from the standard assumptions, and the robustness of inferences about N or
8082 density, need to be conducted. If the spatial randomness model appears inadequate, it
8083 is possible to fit models that allow for a non-uniform distribution of points (see Chapt.
8084 11) and even point process models that allow for interactions among points (Reich et al.,
8085 2012). On the other hand, we expect that most of these Bayesian p-value tests will have
8086 low power in typical data sets consisting of a few to a few dozen individuals. As such,
8087 failure to detect a lack of fit may not be that meaningful. But, on the other hand, it
8088 may not make a difference in terms of density estimates either. We think inference about
8089 density should be relatively insensitive to departures from spatial randomness, because
8090 we get to observe direct information on some component of the population, component
8091 of density is *observed*. For those activity centers, the assumed model of the point process
8092 should exert little influence on the placement of the activity centers. Conversely, as is
8093 the case with classical closed population models (Otis et al., 1978; Dorazio and Royle,
8094 2003; Link, 2003), inferences may be somewhat more sensitive to bad-fitting models for
8095 the observation process.

8096
8097

9

8098

ALTERNATIVE OBSERVATION MODELS

8099 In previous chapters we considered various models of *encounter probability*, both in terms
8100 of parametric functions of distance and also a myriad of covariate models (Chapt. 7 and
8101 elsewhere). However, we have so far only considered a specific probability model for the
8102 observations (we'll call this the "observation model") – the Bernoulli encounter process
8103 model which, in **secr**, is the *proximity detector* model. This assumes that individual and
8104 trap-specific encounters are independent Bernoulli trials.

8105 In this chapter, we focus on developing additional observation models. The observation
8106 model could be thought of as being determined by the type of device – or the type of "de-
8107 tector" using the terminology of **secr** (Efford, 2011). We consider models that apply when
8108 observations are not binary and, in some cases, that do not require independence of the
8109 observations. We present models when the data are encounter *frequencies*, based on the
8110 Poisson distribution, and observation models based on the multinomial distribution. For
8111 example, if sampling devices can detect an individual some arbitrary number of times dur-
8112 ing an interval, then it is natural to consider observation models for encounter frequencies,
8113 such as the Poisson model. Another type of encounter device is the "multi-catch" device
8114 (Efford et al., 2009a) which is a physical device that can capture and hold an arbitrary
8115 number of individuals. A typical example is a mist-net for birds (Borchers and Efford,
8116 2008). It is natural to regard observations from these kinds of studies as independent
8117 multinomial observations. A related type of device that produces *dependent* multinomial
8118 observations are the so-called *single-catch* traps (Efford, 2004; Efford et al., 2009a). The
8119 canonical example are small-mammal live traps which catch and hold a single individual.
8120 Competition among individuals for traps induces a complex dependence structure among
8121 individual encounters. To date, no formal inference framework has been devised for this
8122 method although it stands to reason that the independent multinomial model should be
8123 a good approximation in some situations (Efford et al., 2009a). We analyze a number of
8124 examples of these different observation models using **JAGS** and also the **R** package **secr**
8125 (Efford, 2011).

9.1 POISSON OBSERVATION MODEL

The models we analyze in Chapt. 5 assumed binary observations – i.e., standard encounter history data – so that individuals are captured at most one time in a trap on any given sample occasion. This makes sense for many types of DNA sampling (e.g., based on hair snares) because distinct visits to sampled locations or devices cannot be differentiated. However, for some encounter devices, or methods, the potential number of encounters is *not* fixed, and so it is possible to encounter an individual some arbitrary number of times during any particular sampling episode. That is, we might observe encounter frequencies $y_{ijk} > 1$ for individual i , trap j and sampling interval k . As an example, if a camera device is functioning properly it may be programmed to take photos every few seconds if triggered. For a second example, suppose we are searching a quadrat or length of trail for scat, we may find multiple samples from the same individual. Therefore, we seek observation models that accommodate such encounter frequency data. In general, any discrete probability mass function could be used for this purpose, including the standard models for count data used throughout ecology, the Poisson and negative binomial. Here we focus on using the Poisson model only although other count frequency models are possible for SCR models (Efford et al., 2009b).

Let y_{ijk} be the frequency of encounter for individual i , in trap j , during occasion k , then assume:

$$y_{ijk} \sim \text{Poisson}(\lambda_{ij})$$

where the expected encounter frequency λ_{ij} depends on both individual and trap. As we did in the binary model of Chapt. 5, we now seek to model the expected value of the observation (which was p_{ij} in Chapt 5) as a function of the individual activity center \mathbf{s}_i . We propose

$$\lambda_{ij} = \lambda_0 k(\mathbf{x}_j, \mathbf{s}_i)$$

Where $k(\mathbf{x}, \mathbf{s})$ is any positive valued function, such as the negative exponential or the bivariate Gaussian kernel, and λ_0 is the baseline encounter rate – the expected number of encounters if a trap is placed precisely at an individuals home range center (note: in `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 \mathbf{x}_j for an individual having activity center \mathbf{s}_i . Note that

$$\log(\lambda_{ij}) = \log(\lambda_0) + \log(k(\mathbf{x}_j, \mathbf{s}_i)).$$

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), then:

$$\log(\lambda_{ij}) = \alpha_0 - \alpha_1 d(\mathbf{x}_j, \mathbf{s}_i)^2 \quad (9.1.1)$$

where $\alpha_1 = 1/(2\sigma^2)$, which is the same linear predictor as we have seen for the Bernoulli model in Chapt. 5. This Poisson SCR model is therefore a type of Poisson generalized linear mixed model (GLMM).

We can accommodate covariates at the level of individual-, trap- or sample occasion by including them on the baseline encounter rate parameter λ_0 . For example, if C_j is some covariate that depends on trap only, then we express the relationship between λ_0 and C_j as:

$$\log(\lambda_{0,ijk}) = \alpha_0 + \alpha_2 C_j$$

and therefore covariates on the logarithm of baseline encounter probability appear also as linear effects on λ_{ij} . In general, covariates might also affect the coefficient on the distance

8164 term (α_1) (e.g., sex of individual). We don't get into too much discussion of general
 8165 covariate models here, but we covered them in some detail in both Chaps. 7 and 8.

8166 For models in which we do not have covariates that vary across the sample occasions
 8167 k , we can aggregate the observed data by the property of compound additivity of the
 8168 Poisson distribution (if x and y are *iid* Poisson with mean λ then $x + y$ is Poisson with
 8169 mean 2λ). Therefore,

$$y_{ij} = \left(\sum_{k=1}^K y_{ijk} \right) = \text{Poisson}(K\lambda_0 k(\mathbf{x}_j, \mathbf{s}_i))$$

8170 We see that K and λ_0 serve the same role as affecting the base encounter rate. Since the
 8171 observation model is the same, probabilistically speaking, for all values of K , evidently
 8172 we need only $K = 1$ "survey" from which to estimate model parameters (Efford et al.,
 8173 2009b). We know this intuitively, as sampling by multiple traps serves as replication
 8174 in SCR models. This has great practical relevance to the conduct of capture-recapture
 8175 studies and the use of SCR models. For example, if individuality is obtained by genetic
 8176 information from scat sampling, one should only have to carry out a single spatial sampling
 8177 of the study area. However, one must be certain that sufficient spatial recaptures will be
 8178 obtained so that effective estimation is possible.

8179 9.1.1 Poisson model of space usage

8180 It is natural to interpret the Poisson encounter model as a model of space usage resulting
 8181 from movement of individuals about their home range (Sec. 5.4). Imagine we have perfect
 8182 samplers in every pixel of the landscape so that whenever an individual moves from one
 8183 pixel to another, we can record it. Let m_{ij} be the number of times individual i was
 8184 recorded in pixel j (i.e., it selected or used pixel j). Then, we might think of the Poisson
 8185 model for the observed *use* frequencies:

$$m_{ij} \sim \text{Poisson}(\lambda_0 k(\mathbf{x}_j, \mathbf{s}_i))$$

8186 where λ_0 is related to the baseline movement rate of the animal (how often it moves). This
 8187 model of space usage gives rise to the standard resource selection function (RSF) models
 8188 (see Chapt. 13). But now suppose our samplers are not perfect but, rather, record only
 8189 a fraction of the resulting visits. A sensible model is

$$y_{ij}|m_{ij} \sim \text{Binomial}(m_{ij}, p).$$

8190 The marginal distribution of y_{ij} is:

$$y_{ij} \sim \text{Poisson}(p_0 k(\mathbf{x}_j, \mathbf{s}_i)).$$

8191 where p_0 is a composite of the movement rate and conditional detection probability p .
 8192 Therefore, we see that encounters accumulate in proportion to the frequency of outcomes
 8193 of an individual using space (or "selecting resources").

8194 We introduced an interpretation of SCR models in terms of movement and space usage
 8195 in Sec. 5.4, and it is one of the main underlying concepts of SCR models that is not present
 8196 in ordinary capture-recapture models. As we noted there, the underlying model of space
 8197 usage is only as complex as the encounter probability model which has been, so far in this
 8198 book, only symmetric and stationary (does not vary in space). We generalize this model
 8199 of space usage substantially in Chapt. 13.

9.1.2 Poisson relationship to the Bernoulli model

8201 There is a sense in which the Poisson and Bernoulli models can be viewed as consistent with
 8202 one another. Note that under the Poisson model, the relationship between the expected
 8203 count and the probability of counting “at least 1”, is given by

$$\Pr(y > 0) = 1 - \exp(-\lambda) \quad (9.1.2)$$

8204 where $\mathbb{E}(y) = \lambda$. Therefore, if we equate the event “encountered” with the event that the
 8205 individual was captured at least 1 time under the Poisson model, i.e., $y > 0$, then it would
 8206 be natural to set $p_{ij} = \Pr(y > 0)$ according to Eq. 9.1.2. That is, we can use Eq. 9.1.2
 8207 as the model for encounter probability for binary observations. This is the “hazard rate”
 8208 model in distance sampling.

8209 In fact, as λ gets small, the Poisson model is a close approximation to the Bernoulli
 8210 model in the sense that outcomes concentrate on $\{0, 1\}$, i.e., $\Pr(y \in \{0, 1\}) \rightarrow 1$ as $\lambda \rightarrow 0$.
 8211 Indeed, under the Poisson model, $\Pr(y > 0) \rightarrow \lambda$ for small values of λ . This phenomenon
 8212 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
 8213 superimposed on that is a plot of $p_{ij} = 1 - \exp(-\lambda_{ij})$ vs. distance, for values $\lambda_0 = 0.1$
 8214 and $\sigma = 1$, and the right panel shows a plot of $\Pr(y > 0)$ vs. $\mathbb{E}(y)$. We see that the two
 8215 quantities are practically indistinguishable. This is convenient in some cases because the
 8216 Poisson model might be more tractable to fit (or even vice versa). For an example, see
 8217 the models described in Chapt. 18, and we also consider another case in Sec. 9.3 below.
 8218 To evaluate the closeness of the approximation, you can use the following R commands
 8219 which we used to produce Fig. 9.1:

```
8220 > x <- seq(0.001, 5, , 200)
8221 > lam0 <- .1
8222 > sigma <- 1
8223 > lam <- lam0*exp(-x**/(2*sigma*sigma))
8224
8225 > par(mfrow=c(1,2))
8226 > p1 <- 1-exp(-lam)
8227 > plot(x, lam, ylab="E[y] or Pr(y>0)", xlab="distance", type="l", lwd=2)
8228 > lines(x,p1,lwd=2,col="red")
8229 > plot(lam, p1, xlab="E[y]", ylab="Pr(y>0)", type="l", lwd=2)
8230 > abline(0,1,col="red")
```

8231 To summarize, if y is Poisson then, as λ gets small,

$$\begin{aligned} \Pr(y > 0) &\approx \mathbb{E}(y) \\ 1 - \exp(-\lambda_0 k(\mathbf{x}, \mathbf{s})) &\approx \lambda_0 k(\mathbf{x}, \mathbf{s}) \end{aligned} \quad (9.1.3)$$

8232 What all of this suggests it that if we have very few observations > 1 in our SCR data
 8233 set, then we won’t lose much information by using the Bernoulli model. On the other
 8234 hand, the Poisson model may have some advantages in terms of analytic or numerical
 8235 tractability in some cases. Further, this approximation explains the close correspondence
 8236 we have found between these two versions of the Gaussian encounter probability model
 8237 (Sec. 5.4). Namely, the Gaussian hazard model and the Gaussian encounter probability
 8238 model are close approximations because $1 - \exp(-\lambda) \approx \lambda$ if λ is small.

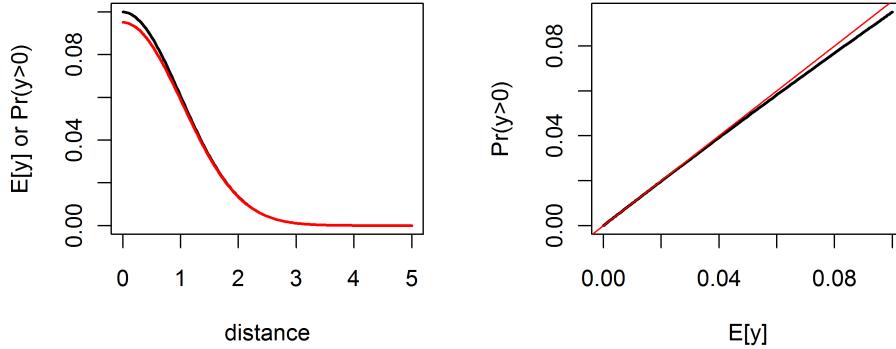


Figure 9.1. Poisson approximation to the binomial. As the Poisson mean approaches 0, then $\Pr(y > 0)$ under the Poisson model approaches λ and therefore $y \sim \text{Poisson}(\lambda)$ is well-approximated by a Bernoulli model with parameter λ .

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

8259 produces encounter frequencies. The main reason is that, with frequency data, we are
 8260 forced to confront a model choice problem (i.e., Poisson, negative binomial, log-normal
 8261 mixture) that is wholly unrelated to the fundamental space usage process that underlies
 8262 the genesis of many types of SCR data. Repeated encounters over short time intervals are
 8263 not likely to be the result of independent encounter events. E.g., an individual moving back
 8264 and forth in front of a camera yields a cluster of observations that is not informative about
 8265 the underlying spatial structure of the population. Similarly in scat surveys dogs are used
 8266 to locate scats which are processed in the lab for individuality (Kohn et al., 1999; MacKay
 8267 et al., 2008; Thompson et al., 2012). The process of local scat deposition is not strictly
 8268 the outcome of movement or space usage but rather the outcome of complex behavioral
 8269 considerations as well as dependence in detection of scat by dogs. For example, dogs find
 8270 (or smell) one scat and then are more likely to find one or more nearby ones, if present, or
 8271 they get into a den or latrine area and find many scats. The additional assumption required
 8272 to model variation in observed frequencies (i.e., conditional on location) provides relatively
 8273 no information about space usage and density, and we feel that the model selection issue
 8274 should therefore be avoided.

8275 To elaborate on this, we suppose that an individual with activity center \mathbf{s} visits
 8276 a particular pixel \mathbf{x} with some probability $p(\mathbf{x}, \mathbf{s})$, and then, once there, deposits a
 8277 number of scat, or visits a camera some number of times with frequency $y(\mathbf{x}, \mathbf{s}) \geq 0$.
 8278 We describe the outcome of this movement/usage process with a two-level hierarchical
 8279 model of the form: $[y|w][w|p(\mathbf{x}, \mathbf{s})]$ where $w(\mathbf{x}, \mathbf{s})$ is a binary variable that indicates
 8280 whether the individual with activity center \mathbf{s} used pixel \mathbf{x} during some interval, and let
 8281 $w(\mathbf{x}, \mathbf{s}) \sim \text{Bernoulli}(p(\mathbf{x}, \mathbf{s}))$. If we suppose encounter frequency y is independent of \mathbf{x} and
 8282 \mathbf{s} conditional on the use variable w , then we see that the model for y (amount of use) does
 8283 not depend on \mathbf{s} .

8284 9.1.4 Analysis of the Poisson SCR model in BUGS

8285 We consider the simplest possible model here in which we have no covariates that vary
 8286 over sample occasions $k = 1, 2, \dots, K$ so that we work with the aggregated individual-
 8287 and trap-specific encounters:

$$8288 y_{ij} = (\sum_{k=1}^K y_{ijk}) = \text{Poisson}(K\lambda_0 k(\mathbf{x}_j, \mathbf{s}_i))$$

and we consider the bivariate normal form of $k(\mathbf{x}, \mathbf{s})$:

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

so that

$$\log(\lambda_{ij}) = \alpha_0 - \alpha_1 d(\mathbf{x}_j, \mathbf{s}_i)^2$$

where $\alpha_0 = \log(\lambda_0)$ and $\alpha_1 = 1/(2\sigma^2)$.

As usual, we approach Bayesian analysis of these models using data augmentation (Sec. 4.2). Under data augmentation, we introduce a collection of all-zero encounter histories to bring the total size of the data set up to M , and a corresponding set of data augmentation variables $z_i \sim \text{Bern}(\psi)$. Then the observation model is specified conditional on z according to:

$$y_{ij} \sim \text{Poisson}(z_i K \lambda_{ij})$$

which evaluates to a point mass at $y = 0$ if $z = 0$. In other words, the observation model under data augmentation is a zero-inflated Poisson model which is easily analyzed by Bayesian methods, e.g., in one of the **BUGS** dialects or, alternatively, using likelihood methods, which we neglect here although the same principles as in Chapt. 6 apply.

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`:

```

8305 ## 
8306 ## S =activity centers and traplocs defined as in simSCR0()
8307 ##
8308 ## Compute distance between activity centers and traps:
8309 > D <- e2dist(S,traplocs)
8310
8311 ## Define parameter values:
8312 > alpha0 <- -2.5
8313 > sigma <- 0.5
8314 > alpha1 <- 1/(2*sigma*sigma)
8315
8316 ## Encounter probability model:
8317 > muy <- exp(alpha0)*exp(-alpha1*D*D)
8318
8319 ## Now generate the encounters of every individual in every trap
8320 > Y <-matrix(NA,nrow=N,ncol=ntraps)
8321 > for(i in 1:nrow(Y)){
8322   Y[i,] <- rpois(ntraps,K*muy[i,])
8323 }
```

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:

```

8330 ## Simulate data and extract data elements
8331 ##
8332 > data <- simPoissonSCR(discard0=TRUE,rnd=2013)
8333 > y <- data$Y
8334 > nind <- nrow(y)
8335 > X <- data$traplocs
8336 > K <- data$K
8337 > J <- nrow(X)
```

```

8338 > xlim <- data$xlim
8339 > ylim <- data$ylim
8340
8341 ## Data augmentation
8342 > M <- 200
8343 > y <- rbind(y,matrix(0,nrow=M-nind,ncol=ncol(y)))
8344 > z <- c(rep(1,nind),rep(0,M-nind))

```

8345 The process for fitting the model in **WinBUGS** or **JAGS** is identical to what we've
 8346 done previously in Chapt. 5. In particular, we set up some starting values, package
 8347 the data and inits, identify the parameters to be monitored, and then send everything
 8348 off to our MCMC engine. Here it all is for fitting the Poisson observation model (these
 8349 commands are shown in the help file for `simPoissonSCR`):

```

8350 ## Starting values for activity centers
8351 ##
8352 > sst <- X[sample(1:J,M,replace=TRUE),]
8353 > for(i in 1:nind){
8354   if(sum(y[i,])==0) next
8355   sst[i,1] <- mean( X[y[i,>0,1] )
8356   sst[i,2] <- mean( X[y[i,>0,2] )
8357 }
8358 ## Dithered a little bit from trap locations
8359 > sst <- sst + runif(nrow(sst)*2,0,1)/8
8360 > data <- list (y=y,X=X,K=K,M=M,J=J,xlim=xlim,ylim=ylim)
8361 > inits <- function(){
8362   list (alpha0=rnorm(1,-2,.4),alpha1=runif(1,1,2),s=sst,z=z,psi=.5)
8363 }
8364 > parameters <- c("alpha0","alpha1","N","D")

```

8365 Next, we write the **BUGS** model to an external file:

```

8366 > cat("
8367 model{
8368   alpha0 ~ dnorm(0,.1)
8369   alpha1 ~ dnorm(0,.1)
8370   psi ~ dunif(0,1)
8371
8372   for(i in 1:M){
8373     z[i] ~ dbern(psi)
8374     s[i,1] ~ dunif(xlim[1],xlim[2])
8375     s[i,2] ~ dunif(ylim[1],ylim[2])
8376     for(j in 1:J){
8377       d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
8378       y[i,j] ~ dpois(lam[i,j])
8379       lam[i,j] <- z[i]*K*exp(alpha0)*exp(- alpha1*d[i,j]*d[i,j])
8380     }
8381   }

```

```

8382   N <- sum(z[])
8383   D <- N/64
8384 }
8385 ",file = "SCR-Poisson.txt")

```

8386 To fit the model we execute **bugs** in the usual way:

```

8387 > library(R2WinBUGS)
8388 > out1 <- bugs (data, inits, parameters, "SCR-Poisson.txt", n.thin=1,
8389           n.chains=3,n.burnin=1000,n.iter=2000,working.dir=getwd(),
8390           debug=TRUE)

```

8391 Or, using **JAGS** via **rjags** we would do something like this:

```

8392 > library(rjags)
8393 > jm <- jags.model("SCR-Poisson.txt", data=data, inits=inits,
8394   n.chains=3, n.adapt=1000)
8395 > out2 <- coda.samples(jm, parameters, n.iter=1000, thin=1)

```

8396 Summarizing the output from the **WinBUGS** run produces the following:

```

8397 > print(out1,digits=2)
8398 Inference for Bugs model at "SCR-Poisson.txt", fit using WinBUGS,
8399 3 chains, each with 2000 iterations (first 1000 discarded)
8400 n.sims = 3000 iterations saved
8401      mean    sd   2.5%   25%   50%   75% 97.5% Rhat n.eff
8402 alpha0   -2.57  0.19  -2.95  -2.69  -2.57  -2.44  -2.19 1.00  2600
8403 alpha1    2.34  0.36   1.69   2.08   2.32   2.57   3.12 1.00  3000
8404 N        114.13 15.25  87.97 103.00 113.00 124.00 147.00 1.01  370
8405 D        1.78  0.24   1.37   1.61   1.77   1.94   2.30 1.01  370
8406 deviance 329.95 21.92 290.00 314.20 329.50 344.40 375.80 1.00 1700
8407 ...
8408 [..some output deleted..]
8409 ...

```

8410 9.1.6 Analysis of the wolverine study data

8411 We reanalyzed the data from the wolverine camera trapping study that were first introduced in Sec. 5.9. We modified the **R** script from the function **wolvSCR0** to fit the Poisson model (see the help file for **wolvSCR0pois**). Executing this function produces the results shown in Table 9.1. The results are almost indistinguishable from the Bernoulli model fitted previously, where we had a posterior mean for N of 59.84 and σ was 0.64. You can edit the script **wolvSCR0pois** to obtain more posterior samples, or modify the model in some way.

Table 9.1. Results of fitting the SCR model with Poisson encounter frequencies to the wolverine camera trapping data. Posterior summaries were obtained using **WinBUGS** with 3 chains, each with 6000 iterations, discarding the first 1000 as burn-in, to yield a total of 15000 posterior samples.

Parameter	Mean	SD	2.5%	50%	97.5%	Rhat	n.eff
N	60.12	11.91	40.00	59.00	87.00	1	630
D	5.80	1.15	3.86	5.69	8.39	1	630
$\log(p_0)$	-2.89	0.17	-3.22	-2.89	-2.57	1	5000
λ_0	0.06	0.01	0.04	0.06	0.08	1	5000
σ	0.64	0.06	0.54	0.64	0.76	1	730
ψ	0.30	0.07	0.19	0.30	0.45	1	650

9.1.7 Count detector models in the secr package

The R package **secr** will fit Poisson or negative binomial encounter frequency models. The formatting of data and structure of the analysis proceeds in a similar fashion to the Bernoulli model described in Sec. 6.5, except that we specify the `detector='count'` option when the traps object is created. The set-up proceeds as follows:

```

8423 > library(secr)
8424 > library(scrbook)
8425 > data(wolverine)
8426
8427 > traps <- as.matrix(wolverine$wtraps)
8428 > dimnames(traps) <- list(NULL,c("trapID","x","y",paste("day",1:165,sep="")))
8429 > traps1 <- as.data.frame(traps[,1:3])
8430 > trapfile1 <- read.traps(data=traps1,detector="count")

```

You can proceed with analysis of these data and compare/contrast with the Bayesian analysis given above, or the results of the Bernoulli model fitted in Chapt. 6.

9.2 INDEPENDENT MULTINOMIAL OBSERVATIONS

Several types of encounter devices yield multinomial observations in which an individual can be caught in a single trap during a particular encounter occasion, but traps might catch any number of individuals. Mist netting is the canonical example of such a “multi-catch” device (Efford et al., 2009a). Also some kinds of bird or mammal cage-traps hold multiple animals, as do pit-fall traps which are commonly used for many species of herptiles. Another type of sample method that might be viewed (in some cases) as a multi-catch device are area-searches of, for example, reptiles where we think of a small polygon as the “trap” – we could get multiple individuals (turtles, lizards) in the same plot but not, in the same sample occasion, at different plots. The key features of this independent multinomial or multi-catch model are: (1) capture of an individual in a trap is *not* independent of its capture in other traps, because initial capture precludes capture in any other trap and (2) individuals behave independently of one another, so whether a trap captures some individual doesn’t have an affect on whether it captures another. A

8446 type of model in which the 2nd assumption is violated are the “single catch” trap systems
 8447 which we address in Sec. 9.3 below.

8448 In this case we assume the observation \mathbf{y}_{ik} for individual i during sample occasion k is
 8449 a multinomial observation which consists of a sequence of 0’s and a single 1 indicating the
 8450 trap of capture, or “not captured”. For the “not captured” event we define an additional
 8451 outcome, by convention element $J + 1$ of the vector. As an example, if we capture an
 8452 individual in trap 2 during some occasion of a study involving $J = 6$ traps. Then, the
 8453 multinomial observation has length $J+1 = 7$, and the observation is $\mathbf{y}_i = (0, 1, 0, 0, 0, 0, 0)$.
 8454 An individual not captured at all would have the observation vector $(0, 0, 0, 0, 0, 0, 1)$. If
 8455 we sample for 5 occasions in all and the individual is also caught in trap 4 during occasion
 8456 3, but otherwise uncaptured, then the 5 encounter observations for that individual are as
 8457 follows:

8458	occassion	trap						"not captured"
		1	2	3	4	5	6	
8459								
8460								
8461	1	0	1	0	0	0	0	0
8462	2	0	0	0	0	0	0	1
8463	3	0	0	0	1	0	0	0
8464	4	0	0	0	0	0	0	1
8465	5	0	0	0	0	0	0	1

8466 Statistically we regard the *rows* of this data matrix as *independent* multinomial trials.

8467 Analogous to our previous Bernoulli and Poisson models, we seek to construct the
 8468 multinomial cell probabilities for each individual, as a function of *where* that individual
 8469 lives, through its center of activity \mathbf{s} . Thus we suppose that

$$\mathbf{y}_{ik} | \mathbf{s}_i \sim \text{Multinomial}(1, \boldsymbol{\pi}(\mathbf{s}_i)) \quad (9.2.1)$$

8470 where $\boldsymbol{\pi}(\mathbf{s}_i)$ is a vector of length $J + 1$, where $\pi_{i,J+1}$, the last cell, corresponds to the
 8471 probability of the event “not captured”. Now we have to construct these cell probabili-
 8472 ties in some meaningful way that depends on each individual’s \mathbf{s} . We use the standard
 8473 multinomial logit with distance as a covariate:

$$\pi_{ij} = \frac{\exp(\alpha_0 - \alpha_1 d_{ij})}{1 + \sum_j \exp(\alpha_0 - \alpha_1 d_{ij})}$$

8474 for $j = 1, 2, \dots, J$ and, for $J + 1$, i.e., “not captured”,

$$\pi_{i,(J+1)} = \frac{\exp(0)}{1 + \sum_j \exp(\alpha_0 - \alpha_1 d_{ij})}$$

8475 or, more commonly, we use d_{ij}^2 to correspond to our Gaussian kernel model for encounter
 8476 probability. Whatever function of distance we use in the construction of multinomial prob-
 8477 abilities will have a direct correspondence to the standard encounter probability models
 8478 we used in the Bernoulli or Poisson models as well (see Sec. 5.4).

8479 It is convenient to express these multinomial models short-hand as follows, e.g., for
 8480 the Gaussian encounter probability model:

$$\text{mlogit}(\pi_{ij}) = \alpha_0 - \alpha_1 d_{ij}^2$$

8481 In this way we can refer to models with covariates in a more concise way. For example, a
 8482 model with a trap-specific covariate, say C_j , is:

$$\text{mlogit}(\pi_{ij}) = \alpha_0 - \alpha_1 d_{ij}^2 + \alpha_2 C_j$$

8483 or we could include occasion-specific covariates too, such as behavioral response.

8484 A statistically equivalent distribution to the multinomial is the *categorical* distribution.

8485 If \mathbf{y} is a multinomial trial with probabilities $\boldsymbol{\pi}$ than the *position* of the non-zero element of
 8486 \mathbf{y} is a categorical random variable with probabilities $\boldsymbol{\pi}$. We express this for SCR models
 8487 as

$$\mathbf{y}|\mathbf{s} \sim \text{Categorical}(\boldsymbol{\pi}(\mathbf{s}))$$

8488 In the SCR context, the categorical version of the multinomial trial corresponds to the
 8489 *trap of capture*. Using our example above with 6 traps then we could as well say y_{ik} is a
 8490 categorical random variable with possible outcomes $(1, 2, 3, 4, 5, 6, 7)$ where outcome $y = 7$
 8491 corresponds to “not captured.” Obviously, how this is organized or labeled is completely
 8492 irrelevant, although it is convenient to use the integers 1 to $(J + 1)$ where $J + 1$ is the
 8493 event not captured. Therefore, for our illustration in the previous table, $y_{i1} = 2$, $y_{i2} = 7$,
 8494 $y_{i3} = 4$ and so on.

8495 For simulating and fitting data in the **BUGS** engines we will typically use the cat-
 8496 egorical representation of the model because it is somewhat more convenient. We have
 8497 found that fitting multinomial models in **WinBUGS** is less efficient than **JAGS** (Royle
 8498 and Converse, in review), which we use in the subsequent examples involving multinomial
 8499 observation models.

8500 9.2.1 Multinomial resource selection models

8501 The multinomial probabilities in Eq. 9.2.2 look similar to the multinomial resource selec-
 8502 tion function (RSF) model for telemetry data (Manly et al., 2002; Lele and Keim, 2006).
 8503 This suggests how we might model landscape or habitat covariates using such methods
 8504 – i.e., by including them as explicit covariates in a larger multinomial model for “use” –
 8505 which, if we take the product of use with encounter, produces a model for the observable
 8506 encounter data. This leads naturally to the development of models that integrate RSF
 8507 data from telemetry studies with SCR data (Royle et al., 2012b), which is the topic of
 8508 Chapt. 13.

8509 9.2.2 Simulating data and analysis using JAGS

8510 We’re going to show the nugget of a simulation function which is used in the function
 8511 **simMnSCR** found in the **R** package **scrbook**. The first lines of the following **R** code make
 8512 use of some things that you need to define, but we omit them here (e.g., **xlim**, **ylim** are
 8513 the boundaries of the state-space, **N** is the population size, etc.):

```
8514 ##
8515 ## Simulate random activity centers:
8516 ##      (first define N, xlim, ylim, etc...)
8517 ##
8518 > S <- cbind(runif(N,xlim[1],xlim[2]),runif(N,ylim[1],ylim[2]))
```

```

8519
8520 ## Distance from each individual to each trap
8521 > D <- e2dist(S,traplocs)
8522
8523 ## Set parameter values
8524 > sigma <- 0.5
8525 > alpha0 <- -1
8526 > alpha1 <- -1/(2*sigma*sigma)
8527
8528 ## make an empty data matrix and fill it up with data
8529 > Ycat <- matrix(NA,nrow=N,ncol=K)
8530 > for(i in 1:N){
8531   for(k in 1:K){
8532     lp <- alpha0 + alpha1*D[i,]*D[i,]
8533     cp <- exp(c(lp,0))
8534     cp <- cp/sum(cp)
8535     Ycat[i,k] <- sample(1:(ntraps+1),1,prob=cp)
8536   }
8537 }
```

8538 We save the data in the matrix `Ycat` to clarify that it is the categorical observation
 8539 representing “trap of capture”. The matrix `Ycat` here has the maximal dimension N
 8540 and so, to do an analysis that mimics a real situation, we would have to discard the
 8541 uncaptured individuals. The function `simMnSCR` in the package `scrbook` will also simulate
 8542 data that includes a behavioral response which will be the typical situation in small-
 8543 mammal trapping problems (see Converse and Royle, 2012, for details).

8544 Here we use our function `simMnSCR` to simulate a data set with $K = 7$ occasions. We’ll
 8545 run the model using `JAGS` which we have found is much more effective for this class of
 8546 models. We get the data set-up for analysis by augmenting the size of the data set to
 8547 $M = 200$. In addition we choose starting values for s and the data augmentation variables
 8548 z . For starting values of s we cheat a little bit here and use the true values for the observed
 8549 individuals and then augment the $M \times 2$ matrix \mathbf{S} with $M - n$ randomly selected activity
 8550 centers. Our function `spiderplot` returns the mean observed location of individuals for
 8551 use as starting values for the `nind` encountered individuals. The parameters input to
 8552 `simMnSCR` are the intercept α_0 , $\sigma = \sqrt{1/(2\alpha_1)}$ for the Gaussian encounter probability
 8553 model, and α_2 is the behavioral response parameter. The data simulation and set-up
 8554 proceeds as follows:

```

8555 > set.seed(2013)
8556 > parms <- list(N=100,alpha0= -.40, sigma=0.5, alpha2= 0)
8557 > data <- simMnSCR(parms, K=7, ssbuff=2)
8558 > nind <- nrow(data$Ycat)
8559
8560 > M <- 200
8561 > Ycat <- rbind(data$Ycat,matrix(nrow(data$X)+1,nrow=(M-nind),ncol=data$K))
8562 > Sst <- rbind(data$S,cbind(runif(M-nind,data$xlim[1],data$xlim[2]),
8563                           runif(M-nind,data$ylim[1],data$ylim[2])))
8564
```

```
8564 > zst <- c(rep(1,160),rep(0,40))
```

8565 The model specification is not much more complicated than the binomial or Poisson
 8566 models given previously. The main consideration is that we define the cell probabilities for
 8567 each trap $j = 1, 2, \dots, J$ and then define the last cell probability, $J+1$, for “not captured”,
 8568 to be the complement of the sum of the others. The code is shown in Panel 9.1. In the
 8569 last lines of code here we specify N and density, D , as derived parameters.

8570 To fit the model, we need to package everything up (inits, parameters, data) and send
 8571 it off to **JAGS** to build an MCMC simulator for us (these commands are executed in
 8572 the help file for `simMnSCR`). In addition to the usual data objects, we also pass the limits
 8573 of the assumed rectangular state-space (`ylim`, `xlim`, both 1×2 vectors) and the scale of
 8574 the standardized units, called `trap.space` here because we typically will define the trap
 8575 coordinates to be an integer grid. If the trap spacing is 10 m and we want units of density
 8576 computed in terms of individuals per meter-squared, then we input `trap.space=10`. The
 8577 analysis is carried out as follows:

```
8578 > inits <- function(){ list (z=zst,sigma=rnorm(1,.5,1) ,S=Sst) }  

8579  

8580 # Parameters to monitor  

8581 > parameters <- c("psi","alpha0","alpha1","sigma","N","D")  

8582  

8583 # Bundle the data. Note this reuses "data"  

8584 > data <- list (X=data$X,K=data$K, trap.space=1,Ycat=Ycat,M=M,  

8585   ntraps=nrow(data$X),ylim=data$ylim,xlim=data$xlim)  

8586  

8587 > library(R2jags)  

8588 > out <- jags (data, inits, parameters, "model.txt", n.thin=1,  

8589   n.chains=3, n.burnin=1000, n.iter=2000)
```

8590 The posterior summaries are provided in the following **R** output (recall that $N = 100$,
 8591 $\alpha_0 = -.40$, and $\sigma = 0.5$):

```
8592 > out  

8593 Inference for Bugs model at "model.txt", fit using jags,  

8594 3 chains, each with 2000 iterations (first 1000 discarded)  

8595 n.sims = 3000 iterations saved  

8596      mu.vect sd.vect    2.5%     25%     50%     75%   97.5% Rhat n.eff  

8597 D        1.873   0.189   1.531   1.750   1.859   2.000   2.250 1.006 1300  

8598 N       119.867  12.107  98.000 112.000 119.000 128.000 144.000 1.006 1300  

8599 alpha0   -0.435   0.151  -0.738  -0.535  -0.439  -0.331  -0.146 1.004  580  

8600 alpha1   2.195   0.286   1.658   2.004   2.180   2.372   2.785 1.003 2400  

8601 psi      0.599   0.069   0.465   0.552   0.599   0.645   0.739 1.006 1400  

8602 sigma    0.480   0.032   0.424   0.459   0.479   0.500   0.549 1.003 2400  

8603 deviance 892.164 21.988 850.922 877.417 891.561 906.246 937.728 1.003  950  

8604  

8605 [... 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

8606 The multinomial is related to the Poisson encounter rate model by a conditioning argument.
 8607 Let y_{ij} be the number of encounters for individual i in trap j . If $y_{ij} \sim \text{Poisson}(\lambda_{ij})$,
 8608 then, conditional on the *total* number of captures (i.e., across all traps), $y_i = \sum_j y_{ij}$, the
 8610 trap encounter frequencies are multinomial with probabilities

$$\pi_{ij} = \frac{\lambda_{ij}}{\sum_j \lambda_{ij}}$$

8611 for $j = 1, 2, \dots, J$. Or equivalently the *trap of capture* is categorical with probabilities π_{ij}
 8612 as given above. Under the Gaussian kernel model, these probabilities are:

$$\pi_{ij} = \frac{\exp(-\alpha_1 d(\mathbf{x}_i, \mathbf{s}_i)^2)}{\sum_j \exp(-\alpha_1 d(\mathbf{x}_i, \mathbf{s}_j)^2)} \quad (9.2.2)$$

8613 where, we note, the intercept α_0 has canceled from both the numerator and denominator.
 8614 This makes sense because, here, these probabilities describe the trap-specific capture probabilities
 8615 *conditional on capture*. Therefore, the model is not completely specified, absent
 8616 a model for the “overall” probability of encounter or the expected frequency of captures,
 8617 say ϕ_i . Depending on how we specify a model for this quantity ϕ_i , we can reconcile it
 8618 directly with the Poisson model. Let y_i be the total number of encounters for individual
 8619 i and suppose y_i has a Poisson distribution with mean ϕ_i . Then, marginalizing Eq. 9.2.1
 8620 over the Poisson distribution for y_i produces the original set of *iid* Poisson frequencies
 8621 with probabilities:

$$\lambda_{ij} = \phi_i \pi_{ij}$$

8622 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
 8623 the marginal distribution of y_{ij} is Poisson with mean $\exp(\alpha_0 - \alpha_1 d(\mathbf{x}, \mathbf{s})^2)$, equivalent to
 8624 Eq. 9.1.1.

8625 In summary, the Poisson and multinomial models are equivalent in how they model
 8626 the distribution of captures among traps. It stands to reason that, if the encounter
 8627 rate of individuals is low, we could use the Poisson and multinomial models interchangeably.
 8628 In fact, based on our discussion in Sec. 9.1.2 above we could use any of the binomial/Poisson/multinomial models with little ill-effect when encounter rate is low.

9.2.4 Avian mist-netting example

8630 We analyze data from a mist-netting study of ovenbirds, conducted at the Patuxent
 8631 Wildlife Research Center, Laurel MD, by D.K. Dawson and M.G. Efford. The data from
 8632 this study are available in the **secr** package, and have been analyzed previously by Efford
 8633 et al. (2004), see also Borchers and Efford (2008). Forty-four mist nets spaced 30 m apart
 8634 on the perimeter of a 600-m x 100-m rectangle were operated on 9 or 10 non-consecutive
 8635 days in late May and June for 5 years from 2005-2009. The ovenbird data can be loaded
 8636 as follows:

```
8638 > library(secr)
8639 > data(ovenbird)
```

8640 The data set consists of adult ovenbirds caught during sampling in each of 5 years, 2005-
8641 2009. (one ovenbird was killed in 2009, indicated by a negative net number in the encounter
8642 data file). As with most mist-netting studies, nets are checked multiple times during a
8643 day (e.g., every hour during a morning session). However, for this data set, the within-day
8644 recaptures are not included so each bird has at most a single capture per day. Therefore
8645 the multinomial model (detector type ‘multi’ in **secr**) is appropriate. Although several
8646 individuals were captured in more than one year, this information is not used in the models
8647 presently offered in **secr**, but we do make use of it in the development of open models in
8648 Chapt. 16.

8649 **Multiple sample sessions**

8650 Up to this point we have only dealt with a basic closed population sampling situation
8651 consisting of repeated sample occasions on a single population of individuals using a single
8652 array of traps. In practice, many studies produce repeated samples over longer periods
8653 of time over which demographic closure isn’t valid, or at different locations where the
8654 populations are completely distinct. We adopt the **secr** terminology of *session* for such
8655 replication by groups of time or space, and the models are *multi-session* models, although
8656 we think of such models as being relevant to any stratified population (see Chapt. 14).
8657 We introduced **secr**’s multi-session models in Sec. 6.5.4. In the case of the ovenbird data,
8658 sampling was carried out in multiple years, with a number of sample occasions within
8659 each year (9 or 10), a type of data structure commonly referred to as “the robust design”
8660 (Pollock, 1982). In this context, it stands to reason that there is recruitment and mortality
8661 happening across years. In Chapt. 16 we model these processes explicitly but, here, we
8662 provide an analysis of the data that does not require explicit models for recruitment and
8663 survival, regarding the yearly populations as independent strata, and fitting a multi-session
8664 model.

8665 When the sessions represent explicit time periods, the multi-session model of **secr** can
8666 be thought of as a type of open population model. In particular, a special case of open
8667 models arises when we assume N_t (time-specific population sizes) are independent from
8668 one time period or session to the next – this can be thought of as a “random temporary
8669 emigration” model of the Kendall et al. (1997) variety, and this is the multi-session model
8670 implemented in **secr**. In particular, by assuming that N_t is Poisson with mean Λ_t , one can
8671 model variation in abundance among sessions based on the Poisson-integrated likelihood
8672 in which parameters of Λ_t appear directly in the likelihood as we noted in Sec. 6.5.4.
8673 We provide an analysis (below) of the ovenbird data here using the multi-session models
8674 in **secr**. We formalize the multi-session model approach from a Bayesian perspective
8675 using data augmentation in Chapt. 14 (Converse and Royle, 2012; Royle and Converse,
8676 in review).

8677 A 3rd way to develop models for stratified or grouped populations, not based on
8678 multi-session models, but that is convenient in **BUGS**, is to regard the data from each
8679 session as an independent data set with its own N_t parameter, and do T distinct data
8680 augmentations. Because each N_t is regarded as a free parameter, independent of the
8681 other parameters, we’ll call this the nonparametric multi-session model to distinguish it
8682 from the multi-session model which assumes the N_t are related to one another by having
8683 been generated from a common Poisson distribution. We can analyze this model in the
8684 normal context of data augmentation by augmenting each year separately in the same
8685 **BUGS** model specification. This approach avoids making explicit model assumptions

8686 about the N_t parameters. This is distinct from the model implemented in **secr** in that
 8687 **secr** is removing the N_t parameters by integrating the conditional-on- N_t likelihood over
 8688 the Poisson prior for N_t ¹

8689 We demonstrate these 3 approaches to analyzing grouped/stratified data using the
 8690 ovenbird data: (1) In the following section, we provide the nonparametric multi-session
 8691 model with unconstrained N_t ; (2) we demonstrate the Poisson model-based multi-session
 8692 models from **secr** both here (following section) and in Chapt. 14 from a Bayesian stand-
 8693 point; (3) later, in Chapt. 16, we provide a fully dynamic “spatial Jolly-Seber” model and
 8694 apply it to the ovenbird data.

8695 Analysis in JAGS

8696 The ovenbird data are provided as a multi-session **capthist** object **ovenCH** which, by
 8697 regarding years as independent strata, or sessions, allows for the fitting of the multi-
 8698 session model. For doing a Bayesian analysis in one of the **BUGS** engines (we use **JAGS**
 8699 here) there are a number of ways to structure the data and describe the model. We can
 8700 analyze either a 2-d data set with all years (data augmented) “stacked” into a data set of
 8701 dimension $(5 * M) \times 10$ (5 years, M = size of the augmented data set, K = 10 replicate
 8702 sample occasions). Or, we could produce a 3-d array $(M \times J \times K)$. We adopted the
 8703 former approach, analyzing the data as a 2-d array and creating an additional categorical
 8704 variable for “year” to indicate which stratum (year) each record goes with.

8705 Data on individual sex is included with **secr**, but we provide an analysis of a single
 8706 model for all adults, constant σ across years, constant p_0 , and year-specific values of N_t
 8707 (and hence D_t). There is a habitat mask provided with the data but the mask appears
 8708 to just be a modified rectangle around the net locations, clipped to have rounded corners,
 8709 and so we don’t use it here. Instead, we used a rectangular state-space buffer of 200 meters
 8710 for our analysis. There was a single loss-on-capture which we accounted for by fixing $p = 0$
 8711 for all subsequent encounters of that individual (indicated by the binary variable **dead**,
 8712 as shown in Panel 9.2). We have an **R** script in **scrbook** package called **SCRovenbird**, so
 8713 you can see how to set-up the data and run the model. Executing the script **SCRovenbird**
 8714 produces the posterior summaries given in Table 9.2. Here, density is in units of birds per
 8715 ha. The posterior mean of σ is about 76 meters, and there is considerable variability in
 8716 density over the 5 year period with density peaking at 1.2 birds/ha in year 3, although
 8717 there is considerable posterior uncertainty. The R-hat’s look a little bit peaked and so we
 8718 might consider running the MCMC analysis longer.

8719 Analysis in secr

8720 Included with the ovenbird data are a number of models fitted as examples. Those include:

```
8721 ovenbird.model.1    fitted secr model -- null
8722 ovenbird.model.1b   fitted secr model -- g0 net shyness
8723 ovenbird.model.1T   fitted secr model -- g0 time trend within years
8724 ovenbird.model.h2   fitted secr model -- g0 finite mixture
8725 ovenbird.model.D   fitted secr model -- trend in density across years
```

¹We do not know of **secr** documentation that states this (or contradicts it). We think this is what is being done, based partially on conversations or emails with M.G. Efford, D.L. Borchers, the various publications on **secr**, and our own thinking about it.

```

model{
  alpha0 ~ dnorm(0,.1)
  sigma ~ dunif(0,200)
  alpha1 <- 1/(2*sigma*sigma)

  A <- ((xlim[2]-xlim[1]))*((ylim[2]-ylim[1]))
  for(t in 1:5){
    N[t] <- inprod(z[1:bigM],yrdummy[,t])
    D[t] <- (N[t]/A)*10000 # Put in units of per ha
    psi[t] ~ dunif(0,1)
  }

  for(i in 1:bigM){ # bigM = total size of jointly augmented data set
    z[i] ~ dbern(psi[year[i]])
    S[i,1] ~ dunif(xlim[1],xlim[2])
    S[i,2] ~ dunif(ylim[1],ylim[2])

    for(j in 1:ntraps){ # X = trap locations, S = activity centers
      d2[i,j] <- pow(pow(S[i,1]-X[j,1],2) + pow(S[i,2]-X[j,2],2),1)
    }
    for(k in 1:K){
      Ycat[i,k] ~ dcat(cp[i,k,])
      for(j in 1:ntraps){
        lp[i,k,j] <- exp(alpha0 - alpha1*d2[i,j])*z[i]*(1-dead[i,k])
        cp[i,k,j] <- lp[i,k,j]/(1+sum(lp[i,k,1:ntraps]))
      }
      cp[i,k,ntraps+1] <- 1-sum(cp[i,k,1:ntraps]) # Last cell = not captured
    }
  }
}

```

Panel 9.2: **BUGS** model specification for the non-parametric multi-session model in which each N_t is independent of the other. The implied prior (by data augmentation) is that $N_t \sim \text{Uniform}(0, 100)$. To fit this model to the ovenbird data, see `?SCRovenbird` in the **R** package `scrbook`.

Table 9.2. Posterior summary statistics for the ovenbird mist-netting data based on the independent multinomial (“multi-catch”) encounter process model. Parameters ψ , N and D are indexed by year. MCMC was done using jags with 3 chains, each with 11000 iterations, discarding the first 1000, for a total of 30000 posterior samples.

Parameter	Mean	SD	2.5%	50%	97.5%	Rhat	n.eff
$D[1]$	0.983	0.211	0.636	0.966	1.455	1.002	1900
$D[2]$	1.023	0.209	0.673	1.003	1.492	1.001	7100
$D[3]$	1.208	0.238	0.807	1.186	1.749	1.004	740
$D[4]$	0.896	0.195	0.575	0.880	1.333	1.002	3000
$D[5]$	0.753	0.177	0.465	0.734	1.149	1.001	4000
α_0	-3.479	0.160	-3.797	-3.477	-3.171	1.005	490
α_1	0.000	0.000	0.000	0.000	0.000	1.003	1100
σ	76.214	6.125	65.569	75.758	89.360	1.003	1100
$N[1]$	80.423	17.283	52.000	79.000	119.000	1.002	1900
$N[2]$	83.685	17.077	55.000	82.000	122.000	1.001	7100
$N[3]$	98.822	19.483	66.000	97.000	143.000	1.004	740
$N[4]$	73.288	15.962	47.000	72.000	109.000	1.002	3000
$N[5]$	61.589	14.468	38.000	60.000	94.000	1.001	4000
$\psi[1]$	0.403	0.092	0.246	0.395	0.606	1.002	1600
$\psi[2]$	0.419	0.091	0.260	0.412	0.620	1.001	6400
$\psi[3]$	0.494	0.102	0.315	0.486	0.723	1.004	760
$\psi[4]$	0.368	0.086	0.221	0.361	0.555	1.002	3200
$\psi[5]$	0.310	0.079	0.178	0.302	0.485	1.002	3500

8726 The model fit objects provided in `secr` are based on the use of the habitat mask.
 8727 To make the analyses consistent with our previous analysis in **JAGS**, we refit all of the
 8728 models here without the habitat mask. The re-analysis proceeds as follows, changing the
 8729 “trend in density across years” model to allow for year-specific density:

```
8730 ## Fit constant-density model
8731 > ovenbird.model.1 <- secr.fit(ovenCH)
8732 ## Fit net avoidance model
8733 > ovenbird.model.1b <- secr.fit(ovenCH, model = list(g0 ~ b))
8734 ## Fit model with time trend in detection
8735 > ovenbird.model.1T <- secr.fit(ovenCH, model = list(g0 ~ T))
8736 ## Fit model with 2-class mixture for g0
8737 > ovenbird.model.h2 <- secr.fit(ovenCH, model = list(g0 ~ h2))
8738 ## Fit a model with session (year)-specific Density
8739 > ovenbird.model.DT <- secr.fit(ovenCH, model = list(D ~ session))
```

8740 All of these can be fitted easily in **JAGS** but the model we fitted previously is roughly
 8741 equivalent to the last model, `ovenbird.model.DT`, because we allowed for year-specific
 8742 population sizes (and hence density). So, we’ll compare our results from **JAGS** to that
 8743 model. The `secr` output is extensive and so we do not reproduce it completely here. By

8744 default, it summarizes the trap information for each year, encounter information, and then
 8745 output for each year. Here is an abbreviated version for `ovenbird.model.DT`:

```

8746 > print(ovenbird.model.DT,digits=2)
8747
8748 secr.fit( capthist = ovenCH, model = list(D ~ session), buffer = 300 )
8749 secr 2.3.1, 14:46:52 23 Jan 2013
8750
8751 $`2005`
8752 Object class      traps
8753 Detector type    multi
8754 Detector number   44
8755 Average spacing   30.27273 m
8756 x-range          -50 49 m
8757 y-range          -285 285 m
8758
8759 [... deleted ...]
8760
8761      2005 2006 2007 2008 2009
8762 Occasions     9   10   10   10   10
8763 Detections    35   42   52   30   33
8764 Animals       20   22   26   19   16
8765 Detectors     44   44   44   44   44
8766
8767 Model          : D~session g0~1 sigma~1
8768 Fixed (real)   : none
8769 Detection fn   : halfnormal
8770 Distribution   : poisson
8771 N parameters   : 7
8772 Log likelihood : -1119.845
8773 AIC            : 2253.689
8774 AICc           : 2254.868
8775
8776 [... deleted ...]
```

8777 To do model selection we use the handy helper-function `AIC` as follows (output edited
 8778 to fit on the page):

```

8779 AIC (ovenbird.model.1, ovenbird.model.1b, ovenbird.model.1T,
8780          ovenbird.model.h2, ovenbird.model.DT)
8781
8782             model detectfn npar logLik     AIC     AICc     dAICc
8783 ovenbird.model.1T [edited output]  4 -1111.850 2231.700 2232.109  0.000
8784 ovenbird.model.1b        ....      4 -1117.615 2243.229 2243.637 11.528
8785 ovenbird.model.h2        ....      3 -1121.164 2248.327 2248.570 16.461
8786 ovenbird.model.1         ....      5 -1119.762 2249.524 2250.143 18.034
8787 ovenbird.model.DT        ....      7 -1119.845 2253.689 2254.868 22.759
```

8788 We see that our DT model is way down at the bottom of the list. Instead, the model with
 8789 a time-trend (within-season) in detection probability is preferred, followed by a behavioral
 8790 response. We encourage you to adapt the **JAGS** model specification for such models which
 8791 is easily done (see Chapt. 7 for many examples). We provide the summary results for the
 8792 model having $D \sim \text{session}$ as follows:

```

8793 > print(ovenbird.model.DT,digits=2)
8794
8795 secr.fit( capthist = ovenCH, model = list(D ~ session), buffer = 300 )
8796 secr 2.3.1, 14:46:52 23 Jan 2013
8797
8798 [...deleted....]
8799
8800 Fitted (real) parameters evaluated at base levels of covariates
8801
8802 session = 2005
8803      link estimate SE.estimate    lcl    ucl
8804 D      log     0.920       0.228  0.571  1.484
8805 g0     logit    0.028       0.004  0.021  0.037
8806 sigma   log    78.566      6.379 67.025 92.095
8807
8808 session = 2006
8809      link estimate SE.estimate    lcl    ucl
8810 D      log     0.963       0.238  0.598  1.553
8811 g0     logit    0.028       0.004  0.021  0.037
8812 sigma   log    78.566      6.379 67.025 92.095
8813
8814 session = 2007
8815      link estimate SE.estimate    lcl    ucl
8816 D      log     1.139       0.282  0.706  1.836
8817 g0     logit    0.028       0.004  0.021  0.037
8818 sigma   log    78.566      6.379 67.025 92.095
8819
8820 session = 2008
8821      link estimate SE.estimate    lcl    ucl
8822 D      log     0.832       0.206  0.516  1.341
8823 g0     logit    0.028       0.004  0.021  0.037
8824 sigma   log    78.566      6.379 67.025 92.095
8825
8826 session = 2009
8827      link estimate SE.estimate    lcl    ucl
8828 D      log     0.701       0.173  0.435  1.130
8829 g0     logit    0.028       0.004  0.021  0.037
8830 sigma   log    78.566      6.379 67.025 92.095

```

8831 The point estimates (MLEs) of density are uniformly lower than the Bayesian estimates
 8832 (posterior means) shown in Table 9.2. We expect some difference in this direction due

8833 to small-sample skew of the posterior. In addition, there may be slight differences due
 8834 to the fact that **secr** multi-session model assumes that the N_t have a Poisson prior, but
 8835 the implementation in **JAGS** using data augmentation is based on a binomial prior. The
 8836 estimated σ is very similar between the **JAGS** analysis and **secr**.

9.3 SINGLE-CATCH TRAPS

8837 The classical animal trapping experiment is based on a physical trap which captures a
 8838 single animal and holds that individual until subsequent molestation by a biologist. This
 8839 type of observation model – the “single-catch” trap – was the original situation considered
 8840 in the context of spatial capture-recapture by Efford (2004). Nowadays, capture-recapture
 8841 data are more often obtained by other methods (DNA from hair snares, or scat sampling,
 8842 camera traps etc...) but nevertheless the single-catch traps are still widely used in small
 8843 mammal studies (Converse et al., 2006b; Converse and Royle, 2012) and other situations.

8844 The single-catch model is basically a multinomial model but one in which the number
 8845 of available traps is reduced as each individual is captured. As such, the constraints on the
 8846 joint likelihood for the sample of n encounter histories are very complicated. As a result,
 8847 at the time of this writing, there has not been a formal development of either likelihood or
 8848 Bayesian analysis of this model and applications of SCR models to single-catch systems
 8849 have used the independent multinomial model as an approximation (see below).

8850 Nevertheless, we can make some progress to describing the basic observation model
 8851 formally. In particular, if we imagine that all of the individuals captured queued up at
 8852 the beginning of the capture session to draw a number indicating their order of capture,
 8853 then there is a nice conditional structure resulting from a “removal process” operating on
 8854 the traps. The first individual captured has the multinomial observation model:

$$\mathbf{y}_1 \sim \text{Multinomial}(\boldsymbol{\pi}_1)$$

8855 whereas the 2nd individual captured also has a multinomial encounter probability model
 8856 but with the trap which captured the first individual removed. We might express this as:

$$\mathbf{y}_2 \sim \text{Multinomial}(\boldsymbol{\pi}_2)$$

8857 where

$$\pi_{2j} = \frac{(1 - y_{1j}) * \exp(\alpha_0 - \alpha_1 d_{ij}^2)}{\sum_j (1 - y_{1j}) * \exp(\alpha_0 - \alpha_1 d_{ij}^2)}$$

8858 and so on for $i = 3, 4, \dots, n$. In a certain way, this model is a type of local behavioral
 8859 response model but where the response is to other individuals being captured. Evidently,
 8860 the **order of capture** is relevant to the construction of these multinomial cell probabilities.
 8861 More generally, the *time* of capture of an individual in any trapping interval will
 8862 affect the encounter probability of subsequently captured individuals, but we think that
 8863 order of capture might lead to a practical approximation to the single-catch process (this
 8864 is how we simulate the data in our function **simScSCR**). In the simulation of single catch
 8865 data, we randomly ordered the population of individuals for each sample occasion, and
 8866 then cycled through them, turning off each trap if an individual was captured in it.

8867 **9.3.1 Inference for single-catch systems**

8868 For the single-catch model, we argued that the observations have a multinomial type of
 8869 observation model, but the multinomial observations have a unique conditional dependence
 8870 structure among them owing to the “removal” of traps as they fill-up with individuals.
 8871 Thus, competition for single-catch traps renders the independence assumptions for the
 8872 independent multinomial model invalid. However, as Efford et al. (2009a) noted, we
 8873 expect “bias to be small when trap saturation (the proportion of traps occupied) is low.
 8874 Trap saturation will be higher when population density is high...” relative to trap density,
 8875 or when net encounter probability is high. Efford et al. (2009a) did a limited simulation
 8876 study and found essentially no effective bias and concluded that estimators of density
 8877 from the misspecified independent multinomial model are robust to the mild dependence
 8878 induced when trap saturation is low. Naturally then, we expect that the Poisson model
 8879 could also be an effective approximation under the same set of circumstances.

8880 In the **R** package **scrbook** we provide a function for simulating data from a single-catch
 8881 system (function **simScSCR**) and fitting the misspecified model (**example(simScSCR)**) in
 8882 **JAGS** so that you can evaluate the effectiveness of this misspecified model for situations
 8883 that interest you.

8884 **9.3.2 Analysis of Efford's possum trapping data**

8885 We provide an analysis here of data from a study of brushtail possums in New Zealand.
 8886 The data are available with the **R** package **secr** (Efford et al., 2009a); see the help file
 8887 **?possum** after loading the **secr** package. Originally the data were analyzed by Efford et al.
 8888 (2005), and a detailed description of the data set is available in the help file, from which
 8889 we summarize:

8890 *Brushtail possums (*Trichosurus vulpecula*) are an unwanted invasive species in New
 8891 Zealand. Although most abundant in forests, where they occasionally exceed densities
 8892 of 15/ha, possums live wherever there are palatable food plants and shelter.*

8893 To load the possum data, execute the following commands:

```
8894 > library(secr)
8895 > data(possum)
```

8896 The study area encompasses approximately 300 ha, and 180 live traps were organized in 5
 8897 distinct grids, shown in Fig. 9.2. Each square arrangement of traps consisted of 36 traps
 8898 with a spacing of 20 m. Thus the squares are 180 m on a side. Individuals were captured,
 8899 tagged, and released over 5 days during April, 2002. A noteworthy aspect of this study is
 8900 that it involves replicated grids selected in some fashion from within a prescribed region.
 8901 From an analysis standpoint, we could adopt the use of the multi-session models which we
 8902 used previously to analyze the ovenbird data. This would be useful if we had covariates
 8903 at the trapping grid level that we wanted to model. Alternatively, we could pool the data
 8904 from all of the grids and analyze them jointly as if they were based on a single trapping
 8905 grid (with 180 traps) which is clearly a reasonable view in this case. In doing this sort of
 8906 pooling, there is an implicit assumption that N_t (t indexing trapping grid in this case) is
 8907 Poisson distributed, with constant mean (Royle, 2004a; Royle et al., 2012c) which we also
 8908 address in Chapt. 14.

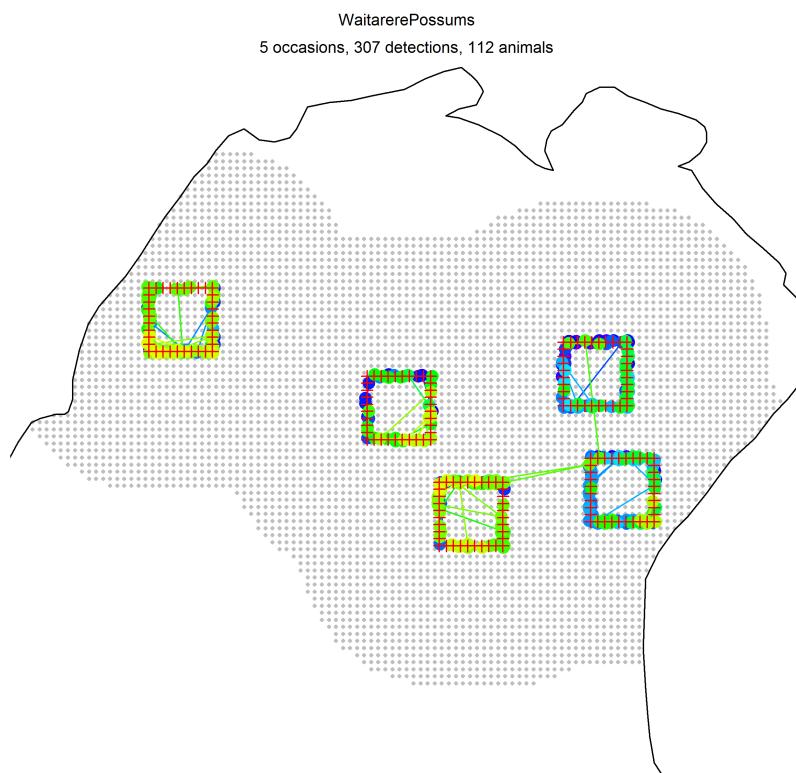


Figure 9.2. Trapping grids used in possum study from Efford et al. (2005), data are contained in the R package `secr` (Efford, 2011), refer to the help file `?possum` for additional details of this study.

8909 The data file **possumCH** contains 112 encounter histories, and we analyze those here
 8910 although the last 8 of those are recaptures treated as new individuals². The encounter
 8911 process is not strictly a single-catch multinomial process because, as noted in the **possum**
 8912 help file “One female possum was twice captured at two sites on one day, having entered
 8913 a second trap after being released; one record in each pair was selected arbitrarily and
 8914 discarded.” which is a similar situation to what might happen in bird mist net studies, as
 8915 a bird might fly into a net upon release from another. By discarding the two extra-capture
 8916 events, we can satisfactorily view these data as single-catch data, for which **secr** uses the
 8917 independent multinomial likelihood (M. Efford, pers. comm.). If multiple, same-session
 8918 captures were common, then it might be worth developing a model for n_{ik} = the number
 8919 of captures of individual i during sample occasion k , in order to make use of all captures.
 8920

8921 For our Bayesian analysis here, we used a rectangular state-space which doesn’t ac-
 8922 count for any geographic boundaries of the survey region, but we note that a habitat mask
 8923 is included in **secr** and it could be used in a Bayesian analysis. Whether or not we use the
 8924 mask is probably immaterial as long as we understand the predictions of N or D over the
 8925 water don’t mean anything biological and we probably wouldn’t report such predictions.
 8926 The **JAGS** model specification is based on that of the ovenbird analysis given previously,
 8927 and so we don’t reproduce the model here. The **R/JAGS** script is called **SCRpossum**,
 8928 which is in the **scrbook** package. The results are summarized in Table 9.3.

Table 9.3. Results of fitting the independent multinomial observation model to the possum trapping data. Strictly speaking, the trapping device is a “single-catch” trap, and the model represents an intentional misspecification. Density is reported in individuals per ha (D_{ha}). Posterior summaries were obtained using **JAGS** with 3 chains, each with 2000 iterations, discarding the first 1000 as burn-in, to yield a total of 3000 posterior samples.

Parameter	Mean	SD	2.5%	50%	97.5%	Rhat	n.eff
N	235.407	17.435	204.000	235.000	270.000	1.009	340
D_{ha}	1.549	0.115	1.343	1.547	1.777	1.009	340
α_0	-0.935	0.167	-1.270	-0.934	-0.605	1.007	870
α_1	0.000	0.000	0.000	0.000	0.000	1.001	2800
σ	52.020	2.675	47.067	51.933	57.585	1.001	2800
ψ	0.783	0.062	0.666	0.782	0.903	1.008	340

8928 The estimated density (posterior mean) is about 1.53 possums/ha. To obtain the **secr**
 8929 results for the equivalent null model, we execute the following command

```
8930 > secr.fit( capthist = possumCH, trace = F )
8931 which produces (edited) summary output:
8932 [... some output deleted ...]
8933
8934 Fitted (real) parameters evaluated at base levels of covariates
8935   link estimate SE.estimate      lcl      ucl
8936 D      log    1.6988930  0.17352645  1.3913904  2.0743547
```

²M. Efford, personal communication

```

8937 g0    logit  0.1968542  0.02256272  0.1563319  0.2448321
8938 sigma   log 51.4689114  2.59981905  46.6204139  56.8216500
8939
8940 [... some output deleted ...]

```

8941 As we've discussed previously, there are many reasons for why there might be differences
8942 between Bayesian and likelihood estimates. But even among likelihood estimates – any
8943 time you run a model there is some numerical integration going on which requires some
8944 specific choices of how to do the integration (see Chapt. 6). For now we just observe that
8945 the estimated density is certainly in the ballpark (compared to those in Table. 9.3), and
8946 so too is the estimated σ .

9.4 ACOUSTIC SAMPLING

8947 The last decade has seen an explosion of technology that benefits the study of animal
8948 populations. This includes DNA sampling methods that allow for identification from
8949 hair or scat, camera trapping and identification software that allow efficient sampling
8950 of many mammals, and the resulting statistical technology that helps us to make sense
8951 of such data (Borchers and Efford, 2008; Royle and Young, 2008; Efford et al., 2009b;
8952 Gopalaswamy et al., 2012b; Sollmann et al., 2012b; Chandler and Royle, In press). One
8953 other extremely promising technology area is that of acoustic sampling using microphones
8954 or recording devices. That is, instead of having cameras record encounters, or humans pick
8955 up scat, we can establish an array of (usually) electronic recording devices which, instead of
8956 establishing a visual identity of individuals, record a vocal expression of each individual. In
8957 this context, Efford et al. (2009b) referred to audio recorders as “signal strength proximity
8958 detectors” to distinguish them from other types of proximity detections, including camera
8959 traps, which are *visual* proximity detector. Using audio records, the spatial pattern of the
8960 *signal strength* at the different audio recorders or microphones can be used for inference
8961 about density (Dawson and Efford, 2009; Efford et al., 2009b) in the same way as the
8962 spatial pattern of detections is used in the types of SCR models we have discussed so far.
8963 The basic technical formulation of these models comes from Efford et al. (2009b), and it
8964 was applied to field study of birds by Dawson and Efford (2009). In that study, recording
8965 devices were organized in groups of 4 (in a square pattern), with an array of 5×15 such
8966 clusters of 4, separated by 100 m (300 total recorder locations). This data set, called
8967 **signalCH**, is provided with the **secr** package along with some sample analyses and help
8968 files. See Efford and Dawson (2010), a version of the document **secr-sound.pdf** (that
8969 also comes with the **secr** package) which you can access directly from the main help file
8970 (**?secr**).

8971 Our development here mostly follows Efford et al. (2009b), but we change some nota-
8972 tion to be consistent with our previous material. Let $S(\mathbf{x}, \mathbf{u})$ be the strength of a signal
8973 emanating from signal location \mathbf{u} , as recorded by a device at location \mathbf{x} . Just as ordinary
8974 SCR models represent a model of *encounter frequency* as a function of distance, in acoustic
8975 models, the acoustic SCR model is a model of sound attenuation as a function of distance.
8976 In particular, the acoustic models assumes that S (or a suitable transformation) declines
8977 with distance d from the origin of the sound, to the recording device. In the context of
8978 spatial sampling of animals, the origin is the actual location of some individual animal,

8979 and the recording device is something we nailed to a tree, or mounted on a post. For example, a model of sound attenuation used by Dawson and Efford (2009) is the following:

$$8981 \quad S(\mathbf{x}, \mathbf{u}) = \alpha_0 + \alpha_1 d(\mathbf{x}, \mathbf{u}) + \epsilon \quad (9.4.1)$$

8982 where $\epsilon \sim \text{Normal}(0, \sigma_s^2)$. In many standard situations, S will be measured in decibels, which can be any value on the real line. In the conduct of acoustic sampling and the development of custom models for your own situation, it would probably be helpful to know something about sound dynamics and signal processing. In this model, the parameters α_0 , α_1 and σ_s^2 are to be estimated. We abbreviate the set of parameters by $\boldsymbol{\theta}$ for short.

8983 The basic structure of an acoustic SCR study is not really much different from ordinary
 8984 SCR studies. Just as ordinary SCR models require that individuals be encountered at > 1
 8985 trap, these acoustic models require that individuals be heard at > 1 recorder. Therefore,
 8986 the acoustic signals (calls or vocalizations) must be reconcilable and, in fact, reconciled
 8987 successfully by the investigator. In practice, this would require associating signals that
 8988 occur at the same instant with the same individual (or making a decision one way or the
 8989 other). Further, if individuals are actively moving during the sample period (that recorders
 8990 are functioning) then individuals might be double-counted, thereby biasing estimates of
 8991 density. In general, the models produce an estimate of density of sources, and how that is
 8992 interpreted depends on whether individuals are stationary or mobile, and other things. In
 8993 particular, if multiple survey occasions are used (e.g., on different days), then modeling
 8994 movement of individuals would be essential in order to interpret estimates of density
 8995 meaningfully. Models that allow some movement should be possible (see Sec. 9.4.3 below,
 8996 and Chaps. 15 and 16).

9001 9.4.1 The signal strength model

9002 We assert that an individual is detected if S exceeds a threshold, c . The reason for introducing
 9003 this threshold c is that sound recorders will always record some background sound,
 9004 and so effective use of the acoustic SCR models requires specification of the threshold of
 9005 measured signal below which the record is censored (non-detection occurs) because the
 9006 recorded sound is assumed to be background noise. So we assert that an individual is
 9007 detected if $S > c$ which occurs with probability $\Pr(S > c)$, the encounter probability. To
 9008 expand on and formalize this, let S_{ij} be the observed value of S for animal i at detector
 9009 j . The encounter probability is $\Pr(S_{ij} > c)$ which is $\Pr(S_{ij} > c) = 1 - \Pr(S_{ij} < c)$, so
 9010 that, if we standardize the variate we have

$$1 - \Pr\left(\frac{(S_{ij} - \mathbb{E}(S))}{\sigma_s} < \frac{(c - \mathbb{E}(S))}{\sigma_s}\right)$$

9011 This probability calculation requires evaluation of the CDF of a standard normal variate
 9012 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
 9013 the parameters α_0 , α_1 , σ_s^2 and also the individual location \mathbf{u} and trap location \mathbf{x} . We'll
 9014 identify it by $\gamma(\boldsymbol{\theta}, \mathbf{x}, \mathbf{u})$ when we need to be explicit about those things. We can compute
 9015 $\Pr(S_{ij} > c) = 1 - \Pr(\eta < \gamma(\boldsymbol{\theta}, \mathbf{x}, \mathbf{u}))$ easily using any software package including **R** which
 9016 has a standard function, **pnorm**, for computing the normal cdf. To be more precise, we'll
 9017 use the **Phi()** to represent the normal cdf. Therefore, an individual is encountered whenever
 9018 $S_{ij} > c$ which happens with probability $\Pr(S_{ij} > c) = 1 - \Phi(\gamma(\boldsymbol{\theta}, \mathbf{x}, \mathbf{u}))$.

9019 Naturally this quantity should depend on *where* an individual is located at the time
 9020 of recording – what we call it’s instantaneous location, say \mathbf{u} , to distinguish it from it’s
 9021 home-range center \mathbf{s} (but we outline a model below that contains both \mathbf{u} and \mathbf{s}), and
 9022 also the trap \mathbf{x} , so we index the quantity γ by those two quantities, in addition to the
 9023 parameters α_0 , α_1 and σ_s . The probability of detection is therefore

$$p_{ij} = p(\alpha_0, \alpha_1, \sigma | \mathbf{x}_j, \mathbf{u}_i) = 1 - \Phi(\gamma(\cdot))$$

9024 where \mathbf{u}_i is the instantaneous location of individual i and \mathbf{x}_j is the location of trap j .
 9025 We’ll suppose here that the random variables \mathbf{u}_i have state-space \mathcal{U} ³.

9026 How do we interpret this probability? Well, two things have to happen for an individual
 9027 to be encountered by a trap: (1) it has to vocalize; (2) the microphone has to record a
 9028 signal $> c$. These two things together are a product of biological and environmental factors
 9029 which could include time of day, wind direction and speed, or maybe rain, humidity and
 9030 other things. The bottom line is a lot of factors are balled up in whether or not the
 9031 microphone records a sound greater than the threshold.

9032 The observations from an acoustic survey are the signal strength measurements, and
 9033 the likelihood of the observed signal strength from individual i at detection device j can
 9034 be specified by noting that the likelihood is the normal pdf for the observed signal *if* the
 9035 signal strength is $> c$ and, otherwise, the contribution to the likelihood is $\Phi(\gamma(\cdot))$ (see Eq.
 9036 8 of Efford et al. (2009b)):

$$\Pr(S_{ij} | \mathbf{u}_i) = \Phi(\gamma(\cdot))^{1-I(S_{ij}>c)} \text{Normal}(S_{ij}; \alpha_0, \alpha_1, \sigma_s, \mathbf{x}_j, \mathbf{u}_i)^{I(S_{ij}>c)}$$

9037 We can use this as the basis for constructing the binomial-form of the likelihood as
 9038 we did in Chapt. 6, which involves the number of individuals not encountered, n_0 . The
 9039 probability that an individual is *not* captured is equal to the probability that its signal
 9040 strength doesn’t exceed c at any microphone. The probability of not being captured at a
 9041 microphone \mathbf{x}_j is:

$$1 - p_{\mathbf{u},j} = \Phi(\gamma(\cdot))$$

9042 and therefore the probability of not being captured at any microphone is:

$$\Pr(\text{all } S_{\mathbf{u},j} < c | \mathbf{u}) = \prod_{j=1}^J (1 - p_{\mathbf{u},j}) = \prod_{j=1}^J \Phi(\gamma(\cdot, \mathbf{x}_j, \mathbf{u}))$$

9043 and therefore the marginal probability of not being captured is

$$\pi_0 = [\text{all } S_{\mathbf{u},j} < c | \boldsymbol{\alpha}] = \int_{\mathcal{U}} \left\{ \prod_{j=1}^J \Phi(\gamma(\boldsymbol{\theta}, \mathbf{x}_j, \mathbf{u})) \right\} d\mathbf{u}$$

9044 which can be used to construct the binomial form of the likelihood as we did in Chapt. 6
 9045 (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

9046 **9.4.2 Implementation in secr**

9047 Fitting acoustic encounter models in **secr** is no more difficult than other SCR models.
 9048 There is a handy manual (**secr-sound.pdf**) with examples (Efford and Dawson, 2010)
 9049 which comes with the **secr** package. The basic process is that **make.capthist** will make a
 9050 **capthist** object from a 3-dimensional encounter array – which is a binary array indicating
 9051 whether each individual was detected or not at each recorder/microphone. In the case
 9052 of signal strength data, **secr** handles the case where # occasions = 1, i.e., the recorders
 9053 obtained data for a single sample occasion, but this is not a general requirement of the
 9054 model for signal strength data (see next section). The “signal” attribute of the **capthist**
 9055 object contains the signal strength in decibels. The best way to include the signal attribute
 9056 is to use **make.capthist** in the usual way, providing it with the encounter data and
 9057 trap data and, in addition, the variable “*c*utval” (which is *c* in our notation above) and
 9058 then provide the signal strength data as an extra column of the **capthist** object. See
 9059 **?make.capthist** for details.

9060 **9.4.3 Implementation in BUGS**

9061 We don’t know of any Bayesian applications of acoustic SCR models, although we imagine
 9062 that implementation of such models in the **BUGS** engines should be achievable. It seems
 9063 easy enough to write down a general hierarchical model that would accommodate sampling
 9064 on repeated occasions. Let \mathbf{s}_i be the home range center, and let \mathbf{u}_{ik} the instantaneous
 9065 location of individual i during sample occasion k (see Chapt. 15 for similar models). The
 9066 model for \mathbf{u}_{ik} can be specified conditional on \mathbf{s}_i . For example, we could assume that \mathbf{u}_{ik}
 9067 are bivariate normal draws with mean \mathbf{s}_i and some variance σ_u^2 . Then, conditional on \mathbf{u}_{ik}
 9068 an individual produces a signal according to the signal attenuation model (Eq. 9.4.1), or
 9069 perhaps some other model. Then we generate the binary encounter data by truncating the
 9070 observed signal at c . This general model then is an example of an SCR model in which
 9071 parameters of a movement model are identifiable (see Sec. 2.6) because there is direct
 9072 information about movement outcomes from the sampling method, unlike other types of
 9073 encounter methods (e.g., camera traps) for which animal locations are restricted to a set of
 9074 fixed, pre-determined points where traps are located. Other types of SCR methods allow
 9075 for movement information too, including some of the search-encounter models (Chapt.
 9076 15).

9077 Instead of developing a Bayesian version of this model here, we leave it to the reader
 9078 to explore simulating data and devising a Bayesian implementation of the acoustic model
 9079 in one of the **BUGS** engines. Note that for a single occasion, you can simulate the data
 9080 using the two stage model (having both \mathbf{s} and \mathbf{u}) or you can simulate \mathbf{u} uniformly without
 9081 dealing with \mathbf{s} in the model. The kernel of the **BUGS** model specification should resemble
 9082 the following snippet:

```
9083 model {
  9084   # Ignoring loops and data augmentation
  9085   u[i,1] ~ dunif(xlim[1], xlim[2])
  9086   u[i,2] ~ dunif(ylim[1], ylim[2])
  9087   mu[i,j] <- alpha0 + alpha1*d[i,j]
  9088   ####
```

```

9089  ##### JAGS has this T() truncation feature
9090  S[i,j] ~ dnorm(mu[i,j], 1/sigma^2)T(c,Inf)
9091  #####
9092  gamma[i,j] <- (c - mu[i,j])/sigma
9093  p[i,j] <- 1 - pnorm(gamma[i,j], 0, 1) # JAGS has pnorm() function
9094  y[i,j] ~ dbern(p[i,j])
9095 }
```

9096 9.4.4 Other types of acoustic data

9097 Efford and Dawson (2010) noted that various other types of acoustic data might arise
 9098 for which SCR-like models would be useful⁴. For example, we could measure the *time of*
 9099 *arrival* of a vocal queue of some sort at multiple recorders to estimate the number and
 9100 origin of N queues. Another example is that where we measure *direction* to a queue from
 9101 multiple devices and do, effectively, a type of statistical triangulation to the multiple but
 9102 unknown number of sources. This has direct relevance to types of double or multiple-
 9103 observer sampling that people do in field studies of birds. Normally 2 observers stand
 9104 in close proximity and record birds, reconciling their detections after data collection.
 9105 An SCR-based formulation of the double-observer method has two observers (or more)
 9106 standing some distance apart, e.g., 50 or 100 meters, and marking individual birds on a
 9107 map (or at least a direction) and a time of detection. The SCR/double-observer method
 9108 could be applied to such data.

9.5 SUMMARY AND OUTLOOK

9109 In this chapter we extended SCR models to accommodate alternative models for the
 9110 observation process, including Poisson and multinomial models. Along with the binomial
 9111 model described in Chapt. 5, this sequence of models will accommodate a substantial
 9112 majority of contemporary spatial capture-recapture problems, including the 4 main types
 9113 of encounter data: binary encounters, multinomial trials from “multi-catch” and “single-
 9114 catch” (Efford, 2004, 2011; Royle and Gardner, 2011) trap systems, and Poisson encounter
 9115 frequency data from devices that can record multiple encounters of the same individual
 9116 at a device. We summarize the standard observation models and the corresponding **secr**
 9117 terminology in Table 9.4. What we refer to as search-encounter (or area-search) models
 9118 (see Chapt. 15) are distinct from most of the other classes in that the observation location
 9119 can also be random (in contrast to traps, where the location is fixed by design). This
 9120 auxiliary data is informative about an intermediate process related to movement (Royle
 9121 and Young, 2008).

9122 There is a need for other types of encounter models that arise in practice. We identify
 9123 a few of them here, although we neglect a detailed development of them at the present
 9124 time or, in some cases, put that off until later chapters: (1) Removal systems – Sometimes
 9125 traps kill individuals and SCR models can handle that. This can be viewed as a kind of
 9126 open model, with mortality only, and we handle such models (in part) in Chapt. 16; (2)
 9127 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)

9128 Royle, In press; Sollmann et al., 2012b) which we cover in Chaps. 18 - 19; (3) We can
 9129 have multiple observation methods working together as in Gopalaswamy et al. (2012b).

9130 There remains much research to be done to formalize models for certain observation
 9131 systems. For example, while we think one will usually be able to analyze single-catch
 9132 systems using the multi-catch model, or even the Bernoulli model if encounter probability
 9133 is sufficiently low, a formalization of the single-catch model would be a useful development
 9134 and, we believe, it should be achievable using one or another of the **BUGS** engines. In
 9135 addition, classical “trapping webs” (Anderson et al., 1983; Wilson and Anderson, 1985a;
 9136 Jett and Nichols, 1987; Parmenter and MacMahon, 1989; Link and Barker, 1994) have
 9137 been around for quite some time and it seems like they are amenable to formulation as
 9138 a type of SCR model although we have not pursued that development simply because
 9139 trapping webs are rarely used in practice.

9140
9141

10

9142

SAMPLING DESIGN

9143 Statistical design is recognized as an important component of animal population studies
9144 (Morrison et al., 2008; Williams et al., 2002). There are probably few to no field biologists
9145 who have never been in the situation where a problem with data could be traced back
9146 to some flaw in study design. Commonly, design is thought of in terms of number of
9147 samples to take, when to sample, methods of capture, desired sample size (of individuals),
9148 power of tests, and related considerations. In the context of spatial sampling problems,
9149 where populations of mobile animals are sampled by an array of traps or devices, there
9150 are a number of critical design elements. Two of the most important ones are the spacing
9151 and configuration of traps (or sampling devices) within the array. While conceptual and
9152 heuristic design considerations have been addressed by a number of authors (e.g., Nichols
9153 and Karanth, 2002, Chapt. 11), little formal analysis focused on spatial design of arrays
9154 has been carried out. Bondrup-Nielsen (1983) investigated the effect of trapping grid size
9155 (relative to animal home range area) on density estimates using a simulation study and
9156 some authors have addressed trap spacing and configuration by sensitivity “re-analysis”
9157 (deleting traps and reanalyzing; Wegge et al., 2004; Tobler et al., 2008). The scarcity of
9158 simulation-based studies looking at study design issues is surprising, as it seems natural
9159 to evaluate prescribed designs by Monte Carlo simulation in terms of their accuracy and
9160 precision.

9161 Theoretically optimal (or extremely good) designs are useful to study in order to
9162 understand how factors influence the design problem. However, often field studies face
9163 logistic difficulties, especially when dealing with wide-ranging, rare and cryptic species
9164 like large carnivores. The need to sample large areas with limited resources often forces
9165 researchers to compromise between a study design that is optimized for data analysis and
9166 a study design that is logically viable. In such studies, traps are often placed along
9167 roads or rivers where a large number of traps can be accessed with relative ease. Unless
9168 the study site is crossed by a network of roads or waterways, this will lead to a sampling
9169 design that is much more ‘linear’ compared to an area-based design where traps are spread
9170 out uniformly over the study site.

9171 In this chapter we recommend a general framework for evaluating specific design

9172 choices for SCR studies based on Monte Carlo simulation of specific design scenarios
9173 based on trade-offs between available effort, funding, logistics and other practical consider-
9174 ations – what we call *scenario analysis*. Many study design-related issues such as how
9175 long to survey in order to obtain sufficient data, can be addressed with preliminary field
9176 studies that will give you an idea of how much data you can expect to collect within a unit
9177 of effort (a camera trap day or a point count survey, for example). But it is also always
9178 useful to perform scenario analysis based on simulation before conducting the actual field
9179 survey not only to evaluate the design in terms of its ability to generate useful estimates
9180 of things, but also so that you have an expectation of what the data will look like as they
9181 are being collected. This gives you the ability to recognize some pathologies and possi-
9182 bly intervene to resolve issues before they render a whole study worthless. Suppose you
9183 design a study to place 40 camera traps based on your expectations of parameter values
9184 you obtained from a careful review of the literature, and simulation studies suggest that
9185 you should get 3-5 captures of individuals per night of sampling. In the field you find
9186 that you're realizing 0 or 1 captures per night and therefore you have the ability to sit
9187 down and immediately question your initial assumptions and possibly take some remedial
9188 action. Simulation evaluation of design *a priori* is therefore a critical element of any field
9189 study.

9190 While we recommend scenario analysis as a general tool to understand your *expected*
9191 *data* before carrying out a spatial capture-recapture study, it is possible to develop some
9192 heuristics and even analytic results related to the broader problem of model-based spatial
9193 design (Muller, 2007) using an explicit objective function based on the inference objective.

10.1 GENERAL CONSIDERATIONS

9194 Many biologists have experience with the design of natural resource surveys from a classical
9195 perspective (Thompson, 2002; Cochran, 2007), a key feature of which involves sampling
9196 space. That is, we identify a sample frame comprised of spatial units and we sample
9197 randomly (or by some other method, such as generalized random tessellation stratified
9198 (GRTS) sampling (Stevens Jr and Olsen, 2004)) those units and measure some attribute.
9199 The resulting inference applies to the attribute of the sample frame. There are some
9200 distinct aspects of the design of SCR studies which many people struggle with in their
9201 attempts to reconcile SCR design with classical survey design problems.

9202 10.1.1 Model-based not design-based

9203 **XXX ANDY: I find this little subsection hard to understand; maybe it needs**
9204 **some more elaborating. I really like the following one though. XXXXXX**

9205 Classical finite-population sampling is often “design-based” which means properties
9206 of estimators (bias, variance) are evaluated over repeated realizations of the *sample*. The
9207 sample is random, but the attribute being observed is not. However, in the SCR modeling
9208 framework properties of our estimators are distinctly model-based. We evaluate estimators
9209 (usually) or care only about a *fixed* sample, averaged over realizations of the underlying
9210 process and data we might generate. This is a classical parametric frequentist idea which
9211 we think makes as much sense as a Bayesian too.

9212 10.1.2 Sampling space or sampling individuals?

9213 A fundamental question in any sampling problem is what is the sample frame – or the
9214 population we are hoping to extrapolate too? In the context of capture-recapture studies,
9215 it is tempting to think of the sample frame as being spatial (the space within “the study
9216 area”, tiled into quadrats perhaps). Clearly SCR models involve a type of spatial sampling
9217 – we have to identify spatial locations for traps, or arrays of traps. However, unlike
9218 conventional natural resource sampling the attribute we measure is *not* directly relevant
9219 to the *sample location*, such as where we place a trap and, therefore, it may not be
9220 sensible to think of the sample frame as being comprised of spatial units. On the other
9221 hand, capture-recapture studies clearly obtain a sample of *individuals* and SCR models are
9222 models of *individual* encounter and space use. Therefore, it is more natural to think of the
9223 sample frame as a list of N individuals, determined by the definition of the state-space,
9224 or a subset of the state-space, i.e., the study-area, but the number N is unknown.

9225 Spatial sampling in SCR studies is important, but only as a device for accumulating
9226 individuals in the sample from which we can learn about their inclusion probability. That
9227 is, we’re not interested in any sample unit attribute directly but, rather, we use spatial
9228 units as a means for sampling individuals and obtaining individual level encounter histories
9229 that indicate the different sample locations at which each individual is encountered. It
9230 makes sense in this context that we should want to choose a set of spatial sample units
9231 that provides an adequate sample size of individuals, perhaps as many as possible. The
9232 key technical consideration as it relates to spatial sampling and SCR is that arbitrary
9233 selection of sample units has a side-effect that it induces unequal probabilities of inclusion
9234 into the sample and so we must also learn about these unequal probabilities of sample
9235 inclusion as we obtain our sample.

9236 The fact that SCR sampling induces unequal probabilities of sampling is consis-
9237 tent with the classical sampling idea of Horvitz-Thompson estimation (see http://en.wikipedia.org/wiki/Horvitz-Thompson_estimator) which has motivated capture-
9238 recapture models similar to SCR (Huggins, 1989; Alho, 1990). In the Horvitz-Thompson
9239 framework, the sample inclusion probabilities are usually fixed and known. However, in
9240 all real animal sampling problems they are unknown because we never know precisely
9241 where each individual lives and therefore cannot characterize its encounter probability.
9242 Therefore, we have to estimate the sample inclusion probabilities using a model. SCR
9243 models achieve this effect formally, using a fully model based approach based on a model
9244 that accounts for the organization of individual activity centers and trap locations. This
9245 notion of Horvitz-Thompson estimation suggests that perhaps we should consider designing
9246 SCR studies based on the H-T variance estimator as a design criterion. We discuss
9247 this a little bit later in this chapter.

9249 10.1.3 Scope of inference vs. state-space

9250 In SCR models we make a distinction between the scope of inference – the population we
9251 care about – and the state-space of the point process, which we are required to prescribe
9252 in order to fit models. These are not the same thing. The geographic scope of inference
9253 is the region within which animals live that you care about in your study – let’s call this
9254 “the study area”. This is often prescribed for political reasons or legal reasons (e.g. a
9255 National Park). To initiate a study, or perhaps motivating the study, you have to draw

9256 a line on a map to delineate a study area, although often it is difficult to draw this line,
9257 and where you draw it is not so much a statistical/SCR issue. On the other hand, you
9258 need to prescribe the state-space to define and fit an SCR model. This is the region that
9259 contains individuals that you *might* capture. This is different from the study area in most
9260 cases.

9261 It is helpful to think about this distinction operationally. We define our study area a
9262 priori. As a conceptual device, we might think of this as the area that, given an infinite
9263 amount of resources, we might wall-off so that we can study a real closed population.
9264 This 'study area' should exist independent of any model or estimator of some population
9265 quantity. i.e., the subject-matter context should determine what the study area is. Given
9266 a well-defined study area, we use some method to arrange data collecting devices within
9267 this study area. The method of arrangement can be completely arbitrary but, naturally,
9268 we want to choose arrangements of traps that are better in terms of obtaining statistical
9269 information from the data we wind up collecting.

9270 Lets face it – Its quite a nuisance that animals move around and this makes the idea of
9271 a spatial study area kind of meaningless in terms of management in most cases. Wherever
9272 you draw a line on a map, there will be animals who live mostly beyond that line that
9273 will sometimes be subjected to your study. One of the benefits of SCR models is they
9274 formalize the exposure and contribution of these individuals to your study. That is a good
9275 thing. Thus, you can probably be a bit sloppy or practical in your definition of "the study
9276 area" and not worry too much.

10.2 STUDY DESIGN FOR (SPATIAL) CAPTURE-RECAPTURE

9277 The importance of adequate trap spacing and overall configuration of the trapping array
9278 has long been discussed in the capture-recapture literature. A heuristic based on recog-
9279 nizing the importance of typical home range sizes (Dice, 1938, 1941) and thus being able
9280 to obtain information about home range size is that traps should be spaced such that the
9281 array of available traps exposes as many individuals as possible but, at the same time,
9282 individuals should be captureable in multiple traps. Thus, good designs should generate
9283 a high sample size n and a large number of spatial recaptures. These two considerations
9284 trade-off in building designs. On one hand, having a lot of traps very close together should
9285 produce the most spatial recaptures but produce very few unique individuals captured (as-
9286 suming that studies are limited in the total number of sampling devices they can deploy).
9287 On the other hand, spreading the traps out as much as possible, in a nearly systematic or
9288 regular design, should yield the most unique individuals. We will formalize this trade-off
9289 later, when we consider formal model-based design of SCR studies.

9290 Traditional CR models require that all individuals in the study area have a probability
9291 > 0 of being captured, which means that the trap array must not contain holes large
9292 enough to contain an animal's entire home range (Otis et al., 1978). As a consequence,
9293 trap spacing should be on the same order as the radius of a typical home range (e.g.,
9294 Dillon and Kelly, 2007). For example, imagine a camera trap study implemented in South
9295 America with the objective to survey populations of both jaguars and the much smaller
9296 ocelots. Ocelots also have much smaller home ranges and therefore should require closer
9297 trap spacing than the large wide-ranging jaguars. Where approaches such as MMDM are
9298 used in combination with traditional CR models to obtain density estimates (see Chapt.

9299 4), trap spacing also has a major effect on movement estimates, since it determines the
9300 resolution of the information on individual movement (Parmenter et al., 2003; Wilson and
9301 Anderson, 1985a). If trap spacing is too wide, there is little to no information on animal
9302 movement because most animals will only be captured at one trap (Dillon and Kelly,
9303 2007). In addition, only a trapping grid that is large relative to individual movement can
9304 capture the full extent of such movements, and researchers have suggested that the grid
9305 size should be at least four times that of individual home ranges to avoid positive bias in
9306 estimates of density (Bondrup-Nielsen, 1983). This recommendation originated in small
9307 mammal trapping, and it should be relatively easy to follow when dealing with species
9308 covering home ranges < 1ha. However, translated to large mammal research, this can
9309 entail having to cover several thousands of square kilometers – a logistical and financial
9310 challenge probably few projects could realistically tackle.

9311 Holes in the study area are of no concern in SCR studies. As a practical matter, some
9312 animals within the study area might have vanishingly small probability of being included
9313 in the sample, i.e., $p \approx 0$, the nice thing about SCR models is that N is explicitly tied to
9314 the state-space, and not the traps which expose them to encounter. Within an SCR model,
9315 extending inference from the sample to individuals that live in these holes represents an
9316 extrapolation (prediction of the model outside the range of the data), but one that the
9317 model is capable of producing because we have explicit declarations, in the model, that it
9318 applies to any area within the state-space (the state-space is a part of the model!), even
9319 to areas where we can't capture individuals because we happened to not put a trap near
9320 them. Conversely, classical capture-recapture models only apply to individuals that have
9321 encounter probability that is consistent with the model being considered. Presumably,
9322 the existence of a hole in the trap array would introduce individuals with $p = 0$, which is
9323 not accommodated in those models.

9324 Whereas traditional CR studies are concerned with the number of individuals and
9325 recaptures and with satisfying the model assumption of all individuals having some prob-
9326 ability of being captured, in spatial capture-recapture we are looking at an additional
9327 level of information: We need spatially spread out captures and recaptures. That means,
9328 it is not enough to recapture an individual, but we need to recapture at least some in-
9329 dividuals at several traps. Therefore, in general, design of SCR studies boils down to
9330 obtaining three bits of information: total captures of unique individuals, gross recaptures
9331 informative about baseline encounter rate, and spatial recaptures, informative about σ .
9332 Most SCR design choices wind up trading these three things against each other to achieve
9333 some optimal (or good) mix. So for example if we sample a very small number of sites
9334 a huge number of times then we can get a lot of recaptures but only very few spatial
9335 ones, and few unique individuals etc. This need for spatial recaptures may appear as an
9336 additional constraint on study design, but actually, SCR studies are much less restricted
9337 than traditional CR studies, because of the way animal movement is incorporated into the
9338 model: σ is estimated as a specified function of the ancillary spatial information collected
9339 in the survey and the capture frequencies at those locations and this function is able to
9340 make a prediction across distances even when these are latent, including distances larger
9341 than the extent of the trap array. When there is enough data across at least some range
9342 of distances, the model will do well at making predictions at unobserved distances. The
9343 key here is that there needs to be 'enough data across some range of distances', which
9344 induces some constraint on how large our overall trap array must be to provide this range

9345 of distances (e.g., Marques et al., 2011). We will review the flexibility of SCR models in
 9346 terms of trap spacing and trapping grid size in the following section.

10.3 TRAP SPACING AND ARRAY SIZE RELATIVE TO ANIMAL MOVEMENT

9347 **XXXX Does Efford have anything out there with discussions of trap spacing?**
 9348 **XXXXX**

9349 Using a simulation study, Sollmann et al. (2012a) investigated how trap spacing and
 9350 array size relative to animal movement influence SCR parameter estimates and we will
 9351 summarize this study here. They simulated encounter histories on an 8×8 trap array
 9352 with regular spacing of 2 units, using a Binomial encounter model with Gaussian hazard
 9353 encounter model (complementary log-log link), across a range of values for the movement
 9354 parameter σ^* . We refer to the movement parameter as σ^* here, because Sollmann et al.
 9355 (2012a) use a slightly different parametrization of SCR models, in which σ^* corresponds
 9356 to $\sigma \times \sqrt{2}$.

9357 **XXXX ANDY:** I am totally blanking here: We formulated the model so that the
 9358 distance function was $d2/\text{sig2}$. That means $\sigma^* = \sigma \times \sqrt{2}$, right? It was different be-
 9359 fore but I think now it's right. If I got it wrong, let me know and I'll fix it. Sorry!!
 9360 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX In Sec. 5.4 we pointed out that
 9361 under the bivariate normal (or half-normal) detection model σ can be converted into an
 9362 estimate of the 95 % home range or "use area" around s_i . Based on this transformation,
 9363 values for σ^* were chosen so that there was a scenario where the trap array was smaller
 9364 than a single individual's home range, i.e. trap spacing was narrow relative to individual
 9365 movements ($\sigma^* = 5$), a scenario where spaces between traps were large enough to con-
 9366 tain entire home ranges ($\sigma^* = 0.5$), and two intermediate scenarios and where sigma was
 9367 smaller ($\sigma^* = 1$ unit) and larger ($\sigma^* = 2.5$ units) than the trap spacing, respectively. N
 9368 was 100 and the baseline trap encounter rate λ_0 was 0.5 for all four scenarios, and trap
 9369 encounters were generated over 4 occasions. Table 10.1 shows the results as the average
 9370 over 100 simulations.

9371 All model parameters were identifiable and estimated with relatively low bias (< 10
 9372 %) and high to moderate precision (rrmse < 25 %) for all scenarios of σ^* , except $\sigma^* =$
 9373 0.5 units (therefore excluded from Table 10.1). Data for the latter case mostly differed
 9374 from the other scenarios in that fewer animals were captured and very few of the captured
 9375 animals were recorded at more than 1 trap (Table 10.2). For $\sigma^* = 0.5$, abundance (N)
 9376 was not identifiable in 88 % of the simulations, and when identifiable, was underestimated
 9377 by approximately 50 %. This shows that a trap spacing that is considerably too large may
 9378 be problematic in SCR studies.

9379 Estimates of N were least biased and most precise under the $\sigma^* = 2.5$ scenario, and in
 9380 general, all parameters were estimated best under the $\sigma^* = 2.5$ or the $\sigma^* = 5$ scenario. All
 9381 estimates had the highest relative bias and the lowest precision under the $\sigma^* = 1$ scenario.
 9382 These results clearly demonstrate that SCR models can successfully handle a range of trap
 9383 spacing to animal movement ratios, and even when using a trapping array smaller than
 9384 an average home range: at $\sigma^* = 5$, the home range of an individual was approximately
 9385 235 units^2 , while the trapping grid only covered 196 units^2 . Still, the model performed
 9386 very well.

Table 10.1. Mean, relative root mean squared error (rrmse) of the mean, mode, 2.5 % and 97.5 % quantiles, relative bias of mean (RB) and 95BCI coverage (BCI) for spatial capture-recapture parameters across 100 simulations for four simulation scenarios, define by the input value of movement parameter σ^* . N = number of individuals in the state space; λ_0 = baseline trap encounter rate

Scenario	Mean	rrmse	Mode	2.5%	97.5%	RB	BCI
$\sigma^* = 1$ ($\sigma = 0.71$)							
N	108.497	0.172	104.099	78.977	143.406	0.085	96
λ_0	0.518	0.248	0.477	0.303	0.752	0.035	94
σ^*	1.008	0.093	0.990	0.857	1.195	0.008	94
$\sigma^* = 2.5$ ($\sigma = 1.77$)							
N	100.267	0.105	98.456	82.086	121.878	0.003	97
λ_0	0.507	0.118	0.500	0.409	0.623	0.014	92
σ^*	2.501	0.046	2.491	2.267	2.690	< 0.001	92
$\sigma^* = 5$ ($\sigma = 3.54$)							
N	102.859	0.137	100.756	77.399	130.020	0.029	88
λ_0	0.505	0.075	0.501	0.435	0.580	0.011	93
σ^*	5.023	0.039	5.001	4.687	5.431	0.005	97

Table 10.2. Summary statistics of 100 simulated data sets for four simulation scenarios, defined by the input value of movement parameter σ . Individual detection histories were simulated on an 8×8 trap array with regular trap spacing of 2 units.

Scenario	Inds. captured	Total captures	Inds. recaptured	Inds. captured at > 1 trap
$\sigma^* = 0.5$	18.29 (3.84)	25.38 (5.86)	5.52 (2.03)	0.72 (0.95)
$\sigma^* = 1.0$	37.70 (13.44)	69.35 (26.05)	19.48 (7.68)	11.87 (5.43)
$\sigma^* = 2.5$	44.19 (4.67)	231.78 (33.98)	36.60 (4.76)	35.21 (4.73)
$\sigma^* = 5.0$	40.51 (5.15)	427.77 (79.09)	33.09 (4.63)	32.60 (4.76)

An important consideration in this simulation study is that all but the $\sigma^* = 0.5$ units scenarios provided reasonably large amounts of data, including 20 + individuals being captured on the trapping grid. When dealing with real-life animals that are often territorial and may have lower trap encounter rates, a very small grid compared to an individual's home range may result in the capture of few to no individuals. In that case, the sparse data will limit the ability of the model to estimate parameters (Marques et al. 2011), which is true of most models.

In summary, SCR models performed best when σ^* was slightly larger than trap spacing (or in other words, when σ was slightly smaller) and did well as long as σ^* was at least 0.5 times the average distance between traps (which corresponds to σ being 0.35 times the average distance between traps). Although at this trap spacing to movement ratio, most individuals are captured at one trap only (see Tab. 10.2), parameter estimates exhibited low bias and remained relatively precise (see simulation results for $\sigma^* = 1$ in Tab. 10.1). Below this trap spacing to movement ratio the spatial information in the simulated data apparently was not sufficient to inform SCR model parameters.

10.3.1 Example: Black bears from Pictured Rocks National Lakeshore:

To see how trap array size influences parameter estimates from spatial capture-recapture models in the real world, Sollmann et al. (2012a) also looked at a black bear data set from Pictured Rocks National Lakeshore, Michigan, collected using 123 hair snares distributed over an area of 440 km^2 along the shore of Lake Superior in May-July 2005 (Belant et al., 2005). The SCR model for the bear data allowed for differences in the baseline trap encounter rate and σ^* between males and females, and λ_0 varied across occasions. This was motivated by a) the lower average number of detections for male bears and b) the decreasing number of detections over time in the raw data, and c) the fact that male black bears are known to move over larger areas than females (e.g., Gardner et al., 2010b; Koehler and Pierce, 2003).

To address the impact of a smaller trap array on the parameter estimates, the full data set and data subsets were analyzed with SCR models. The first subset retained only those 50 % of the traps closest to the grid center. In the second, only the southern 20 % of the traps were retained 10.3.

Table 10.3. Posterior summaries of SCR model parameters for black bears.

	Mean (SE)	Mode	2.5%	97.5%
Full data set				
D	10.556 (1.076)	10.448	8.594	12.792
σ^* (males)	7.451 (0.496)	7.323	6.579	8.495
σ^* (females)	2.935 (0.143)	2.939	2.671	3.226
50% of traps				
D	12.648 (1.838)	12.205	9.307	16.713
σ^* (males)	5.354 (0.511)	5.248	4.472	6.473
σ^* (females)	3.318 (0.277)	3.262	2.841	3.910
20% of traps				
D	6.752 (1.611)	5.953	4.000	10.218
σ^* (males)	9.881 (3.572)	7.566	5.121	18.447
σ^* (females)	2.686 (0.391)	2.657	2.121	3.404

Reducing the area of the trap array by 50 % created a grid polygon of 144 km^2 , which was smaller than an estimated male black bear home range and only 50 % larger than a female black bear home range - approximately 260 km^2 and 100 km^2 , respectively, when converting estimates of σ^* to home range size. Table 10.3 shows that this did not greatly influence model results, compared to the full data set. The observed smaller differences in parameter estimates may be due to individual differences in detection and movement that manifest themselves when only a smaller portion of the overall population is sampled. By reducing the number of traps we effectively reduced the size of the overall data set estimates were based on (both in terms of individuals captured and recaptures). This was reflected in overall higher SE and wider confidence intervals. In spite of these differences, density estimates - the main objective of applying SCR models - remained largely constant. Removing 80 % of the traps and thereby reducing the area of the trap array to 64 km^2 - well below the average black bear home range - had a great effect on sample size (only 25 of the original 83 individuals sampled) and parameter estimates. Particularly, male black bear movement was overestimated and imprecise. The combination of the low baseline trap encounter rate of males and the considerable reduction in sample size led to a low

level of information on male movement: 5 of the 12 males were captured at one trap only. Although they moved over smaller areas, owing to their higher trap encounter rate females were, on average, captured at more traps (3.4 traps per individual compared to 2.6 for males) so that their movement estimate remained relatively accurate. Overestimated male movements and female trap encounter rates resulted in an underestimate of density of almost 40 %. This effect is contrary to what we would expect to see in non-spatial CR models, where too small an area leads to underestimated movement and overestimated density (Bondrup-Nielsen, 1983; Dillon and Kelly, 2007; Maffei and Noss, 2008). While this example again demonstrates the ability of SCR models to deal with a range of trapping grid sizes, it also clearly shows that your study design needs to consider the amount of data you can expect to collect.

10.3.2 Final musings: SCR models, trap spacing and array size

When designing a capture-recapture study for a single species, trap spacing and the size of the array can (and should) be tailored to the spatial behavior of that species to ensure adequate data collection. However, some trapping devices like camera traps may collect data on more than one species and researchers may want to analyze these data, too. Independent of the trapping device used, study design will in most cases face a limit in terms of the number of traps available or logically manageable. As a consequence, researchers need to find the best compromise between trap spacing and the overall grid area and the following section will develop an approach to derive such an optimal study design.

Particularly for large mammal research SCR models have much more realistic requirements in terms of area coverage than non-spatial CR models, under which density estimates can be largely inflated with small trapping grids relative to individual movement (Maffei and Noss, 2008). How large the spatial survey effort needs to be does not only depend on the extent of movement of the target species, but also on the temporal effort, density and detection probability (Marques et al., 2011) – in summary, the amount of data that can be collected with any given trap array. For low-density species, like the black bears in the above example, small trapping grids bear the risk of not collecting enough data for parameter estimation. Simulation studies can help you assess how effective a certain study design is given a set of parameters. Alternatively, Efford et al. (2009b) provide a mathematical procedure to determine the expected number of individuals captured and recaptures for a given detector array and set of model parameters.

Overall, while there are limits to the flexibility in spatial trap array design for SCR modeling, the method is fairly robust to changes in trap array size and spacing relative to animal movement. Trapping grids with an extent of approximately a home range diameter can in theory - adequately estimated density and home range size. However, these results should not encourage researchers to design non-invasive trap arrays based on minimum area and spacing requirements. Study design should still strive to expose as many individuals as possible to sampling and obtain adequate data on individual movement. Large amounts of data do not only improve precision of parameter estimates (the density estimate for the full black bear data set has narrower confidence intervals than estimates from the reduced data sets), they also allow including potentially important covariates (such as gender or time effects in the black bear example) into SCR models to obtain

9477 density estimates that reflect the actual state of the studied population.

10.4 SPACING OF TRAPS WITH TELEMETERED INDIVIDUALS

9478 In Chapt. 13 we discussed SCR models that integrate auxiliary information on resource se-
9479 lection obtained by telemetry. Telemetry data are directly informative about the distance
9480 parameter (σ or α_1 as the case may be). It stands to reason that, when telemetry data are
9481 available, it should affect considerations related to trap spacing. Conceivably even, one
9482 might be able to build SCR designs that don't yield any formal spatial recaptures because
9483 all of the information about σ is provided by the telemetry data.

10.5 SAMPLING OVER LARGE SCALES

9484 Trap spacing is an essential aspect of design of SCR studies. However, it is only the most
9485 important aspect if one can uniformly cover a study area with traps. In many practical
9486 situations where the study area is large relative to effort that can be expended, one has to
9487 consider other strategies which deviate from a strict focus on trap spacing. There are two
9488 general strategies that have been suggested which we think are useful in practice, either
9489 by themselves or combined: Sampling based on *clusters* of traps and sampling based on
9490 *rotating* groups of traps over the landscape.

9491 Karanth and Nichols talk about moving traps around in the green book.....
9492 Efford (unpublished) looked at clusters....

9493 In practice, employing both of these strategies might be necessary.
9494 Work on formalizing and generalizing these ideas is needed. We believe the model-
9495 based spatial design approach, which we introduce below, is the way to do that.

10.6 MODEL-BASED SPATIAL DESIGN

9496 A point we have stressed in previous chapters is that SCR models are basically glorified
9497 versions of generalized linear models (GLMs) with a random effect that represents a latent
9498 spatial attribute of individuals, the activity center or home range center. This formula-
9499 tion makes analysis of the models readily accessible in freely available software and also
9500 allows us to adapt and use concepts from this broad class of models to solve problems
9501 in spatial capture recapture. In particular, we can exploit well-established model-based
9502 design concepts (Kiefer 1959; Box and Draper 1959, 1987; Fedorov 1972; Sacks et al. 1989;
9503 Hardin and Sloane 1993; Fedorov and Hackl 1997) to develop a framework for designing
9504 spatial trapping arrays for capture-recapture studies. Müller (2007) provides a recent
9505 book treatment of the subject that is very readable.

9506 In the following sections, we adapt these classical methods for constructing optimal
9507 designs to obtain the configuration of traps (or sampling devices) in some region (the design
9508 space, \mathcal{X}), that minimizes some appropriate objective function based on a compromise
9509 between the variance of estimating N for a prescribed state-space. We show that this
9510 criterion – based on the variance of an estimator of N – represents a formal compromise
9511 between minimizing the variance of the MLEs of the detection model parameters and
9512 obtaining a *high* expected probability of capture. Intuitively, if our only objective was
9513 to minimize the variance of parameter estimates than all of our traps should be in one or

9514 a small number of clusters where we can recapture a small number of individuals many
 9515 times each. Conversely, if our objective was only to maximize the expected probability
 9516 of encounter then the array should be highly uniform so as to maximize the number
 9517 of individuals being exposed to capture. By seeking to minimize the variance of an
 9518 estimator of N , our objective function is, formally, a compromise between these two
 9519 objectives and the resulting designs are not always highly regular nor clustered.

9520 10.6.1 Formalization of the Design Problem for SCR Studies

9521 Let \mathcal{X} , the *design space*, denote some region within which sampling could occur and let
 9522 $\mathbf{X} = \mathbf{x}_1, \dots, \mathbf{x}_J$ denote the *design*, the set of sample locations (e.g., of camera traps)
 9523 which henceforth will be referenced as “traps.” Operationally, we could equate \mathcal{X} to the
 9524 study area itself (which is of management interest) but, in practical cases, there will be
 9525 parts of the study area that we cannot sample. Those areas need to be excluded from
 9526 \mathcal{X} . The technical problem addressed in the following is how to choose the locations \mathbf{X} in
 9527 a manner that is statistically efficient for estimating abundance or density. The design
 9528 space, \mathcal{X} , which determines potential design points, will have to be prescribed. This could
 9529 be some polygon describing a park or forest unit from which we may choose trap locations.
 9530 Further, while \mathcal{X} maybe be continuous, in practice it will be sufficient to represent \mathcal{X} by
 9531 a discrete collection of points. This is especially convenient when the geometry of \mathcal{X} is
 9532 complicated and irregular (which would be in most practical applications).

9533 We regard the population of N such individual “activity centers” as the outcome of a
 9534 point process. Denote the home range center of an individual by the coordinate \mathbf{s} which is
 9535 regarded as the outcome of a random variable uniformly distributed over the state-space
 9536 \mathcal{S} , some 2-dimensional region. The importance of \mathcal{S} is obvious as it defines a population of
 9537 individuals (i.e., activity centers) and, in practice, it is not usually the same as \mathcal{X} due to
 9538 the fact that animals move freely over the landscape and the location of traps is typically
 9539 restricted by policies, ownership and other considerations. That \mathcal{X} and \mathcal{S} are not the same
 9540 is the basic problem of geographic non-closure of the population for which spatial capture-
 9541 recapture models have been devised (Efford 2004; Borchers and Efford 2008; Royle and
 9542 Young 2008; Royle and Gardner 2009).

9543 The basic strategy is: Given (1) \mathcal{X} , (2) a number of design points; (3) state-space \mathcal{S} ,
 9544 and (4) an SCR model, and (5) a design criterion $Q(\mathbf{X})$, we want to choose *which* design
 9545 points we should select in order to obtain the *optimal* design under the chosen model,
 9546 where the optimality is with respect to $Q(\mathbf{X})$.

9547 To see how this goes in a simplified situation, suppose you know \mathbf{s} for an individual.
 9548 In this case, its vector of counts of encounter in each trap \mathbf{y} are either binomial or Poisson
 9549 counts, i.e., just a GLM, with

$$g(\mathbb{E}(\mathbf{y})) = \alpha_0 + \alpha_1 \|\mathbf{x} - \mathbf{s}\|$$

9550 Lets think about this in the context of a normal linear model then:

$$\mathbf{y} = \mathbf{M}(\mathbf{X}, \mathbf{s})' \boldsymbol{\alpha} + \text{error}$$

9551 We could analyze the design problem for the binomial or Poisson case but to establish
 9552 basic ideas here lets just look at the normal model. The variance-covariance matrix of $\hat{\boldsymbol{\alpha}}$

9553 is, supressing the dependence on \mathbf{X} , is:

$$\text{Var}(\boldsymbol{\alpha}) = (\mathbf{M}(\mathbf{s})' \mathbf{M}(\mathbf{s}))^{-1}$$

9554 Therefore if we know *all* N values of \mathbf{s} we could now easily find the design \mathbf{X} that optimizes
 9555 some function of the variance-covariance matrix, whatever function we want. If we don't
 9556 know \mathbf{s} then we might as well minimize the expected variance:

$$E_{\mathbf{s}} \{ \text{Var}(\boldsymbol{\alpha}) \} = \sum_{s \in S} (\mathbf{M}'(\mathbf{s}) \mathbf{M}(\mathbf{s}))^{-1}$$

9557 This can be done for any number of design points $\mathbf{x}_1, \dots, \mathbf{x}_J$ using imperfect exchange
 9558 algorithms (Sec. 10.6.3) which always improve the criterion but will not necessarily yield
 9559 the optimal design. But it is usually good enough for practice.

9560 It is worth noting that asymptotic formulae for $\text{Var}(\boldsymbol{\alpha})$ can be cooked up fro any type
 9561 of GLM (e.g., see McCullagh and Nelder. .XXXXXX p. XXXXX) and we will see that
 9562 design for SCR models is closely related to the basic GLMs such as binomial or poisson
 9563 regression.

9564 Interestingly, if you minimize obvious functions of the variance of the encounter pa-
 9565 rameter estimates, then this produces strongly clustered designs. For example if I pick a
 9566 design of size 11 then it puts 2 or 3 points in each corner of the square and 1 or 2 points
 9567 in the center. I think this makes a lot of sense if your objective function is simply to
 9568 minimize the variance of your estimates of $\boldsymbol{\alpha}$. This suggests to us that maybe minimiz-
 9569 ing this variance isn't really the right thing to do. In fact, it is not sufficient to make a
 9570 design that is optimal for estimating regression parameters – we also want to produce a
 9571 low variance for estimating N . since $n \sim \text{Bin}(N, pbar)$ we want n to be as close to N as
 9572 possible, generally speaking. This suggests that we should find a design that maximizes
 9573 "pbar" – i.e., generates the highest expected sample size.

9574 If we just design networks to maximize \bar{p} then, as you expect, these designs are highly
 9575 regular.

9576 Ok, so what we really would like to do is optimize some function of these three things.
 9577 We want to minimize the variance of (α_0 , α_1 and n_0). The problem with this is
 9578 that there aren't easy formuals for this but we devise an approximation in the following
 9579 section and then we build designs for that criterion.

9580 10.6.2 An Optimal Design Criterion for SCR

9581 Can derive based on the H-A estimator. We tried that too hard.
 9582 We take an approach based on standard conditional estimator....
 9583 (how about data augmentation?)

9584 Consider a conditional estimator of N of the form

$$\tilde{N} = \frac{n}{\bar{p}}$$

9585 where \bar{p} is the probability that an individual appears in the sample of n unique individuals.
 9586 In SCR models an individual with activity center \mathbf{s}_i is captured if it is captured in *any*

trap and therefore, under the Bernoulli model,

$$\bar{p}(\mathbf{s}_i, \mathbf{X}) = 1 - \prod_{j=1}^J (1 - p_{ij}(\mathbf{x}_j, \mathbf{s}_i))$$

and, under the Poisson model, we have:

$$\bar{p}(\mathbf{s}_i, \mathbf{X}) = 1 - \exp(-\lambda_0 \sum_j \exp(\beta * d(\mathbf{x}_j, \mathbf{s}_i)))$$

where here we emphasized that this is conditional on \mathbf{s}_i and also the design – the trap locations \mathbf{x}_j . The *marginal* probability of encounter, averaging over all possible locations of \mathbf{s} is:

$$\bar{p}(\mathbf{X}) = 1 - \int_{\mathbf{s}} \bar{p}(\mathbf{s}_i, \mathbf{X}) d\mathbf{s}.$$

It is important to note that this can be calculated directly *given* the design \mathbf{X} . This is handy because we see that it is used in the variance formulae given subsequently.

The approach we take here is we develop the variance of \tilde{N} conditional on knowing the locations of all N individuals and then we suggest to unconditon on the realized point process by taking a Monte Carlo average over realizations of \mathbf{s} under a suitable model for \mathbf{s} . The variance criterion we propose here is based on a delta approximation $Var(n/\bar{p})$:

$$Var(\tilde{N}(\alpha) | \{\mathbf{s}_i\}_{i=1}^N) = \frac{N^2 Var(\bar{p})}{\bar{p}^2} + N \frac{(1-\bar{p})}{\bar{p}} \quad (10.6.1)$$

It is important to note that this is the sum of two parts which are essentially those due to (1) estimation of \bar{p} from the sample and (2) the variance of n . We see that generally this criterion is improved (decreases) as we do a better job estimating \bar{p} and also as n approaches N , i.e., as \bar{p} increases to 1. Thus, good designs should generate information about detection probability *and* produce large samples of individuals.

In order to work with this experssion we will have to do some analysis of $Var(\bar{p})$ which we take up now. We note that \bar{p} is itself a deterministic function of the parameters that we need to estimate, $\boldsymbol{\alpha}$. Therefore, we use a delta approximation to express $Var(\bar{p})$ in terms of the variance of the MLE $\hat{\boldsymbol{\alpha}}$. This produces:

$$Var(\bar{p}) = \left(\frac{\delta \bar{p}}{\delta \alpha_0}, \frac{\delta \bar{p}}{\delta \alpha_1} \right) Var(\hat{\boldsymbol{\alpha}}) \begin{pmatrix} \frac{\delta \bar{p}}{\delta \alpha_0} \\ \frac{\delta \bar{p}}{\delta \alpha_1} \end{pmatrix} \quad (10.6.2)$$

We need to break this down into its constituent pieces:

(1) $Var(\hat{\boldsymbol{\alpha}})$. It is not actually so obvious what the form of this matrix is. Some calculus would have to be done on the conditional likelihood (e.g., from Borchers and Efford 2008) to figure out the asymptotic form of this matrix. For now, a good heuristic is to use the analogous result from a Poisson or Binomial GLM to approximate it, since we have formulas for those. If we knew the activity centers of all individuals then the resulting data $y(x, s)$ are Poisson counts. The asymptotic variance-covariance matrix of $\boldsymbol{\alpha}$ in that case is:

$$Var(\hat{\boldsymbol{\alpha}} | \mathbf{X}, \mathbf{s}) = (\mathbf{M}(\mathbf{s})' \mathbf{D}(\boldsymbol{\alpha}, \mathbf{s}) \mathbf{M}(\mathbf{s}))^{-1}. \quad (10.6.3)$$

9615 where \mathbf{M} is a matrix which has a column of 1's and a column of $N \times J$ entries that are the
 9616 distances between each individual and each trap and the matrix \mathbf{D} is a diagonal matrix
 9617 having elements $\text{Var}(y_j|\mathbf{s}) = \exp(\mathbf{m}'\boldsymbol{\alpha})$ for y_{ij} the frequency of encounter in trap j . Thus,
 9618 the variance is a function of the design \mathbf{X} as well as \mathbf{s} both of which are balled-up in \mathbf{M}
 9619 – the regression design matrix and the matrix \mathbf{D} .

9620 (2) The derivative terms: multiple applications of the chain rule can be used (see Huggins
 9621 (1989) and Alho (1990) for relevant examples). Under the Poisson model we have

$$\bar{p} = 1 - \sum_s \exp(-\lambda_0 \sum_j \exp(\beta * d(x_j, s)))$$

9622 where the summation over \mathbf{s} arises as a result of approximating the integral in Eq. XXXXX
 9623 by a summation. If we differentiate this with respect to λ_0 and β we have the following:

$$\frac{\delta \bar{p}}{\delta \lambda_0} = 1 - \sum_s \left\{ \left(- \sum_j \exp(\beta d_{ij}^2) \right) \exp(-\lambda_0 \sum_j \exp(\beta d_{ij}^2)) \right\}$$

9624 and

$$\frac{\delta \bar{p}}{\delta \beta} = \left\{ \sum_s -\lambda_0 \left(\sum_j \exp(\beta d_{ij}^2) \right) \right\} \left\{ 1 - \sum_s \left(-\lambda_0 \sum_j \exp(\beta d_{ij}^2) \right) \exp \left(-\lambda_0 \sum_j \exp(\beta d_{ij}^2) \right) \right\}$$

9625 Therefore we have a design criterion which is obtained by plugging \bar{p} from Eq. XXXX
 9626 and the variance expression Eq. 10.6.2 into Eq. 10.6.1. This is a function of the design
 9627 \mathbf{X} . Furthermore, we emphasize that the above variance expression is *conditional* on the
 9628 realization $\mathbf{s}_1, \dots, \mathbf{s}_N$ which is, in the context of design, not observable. We will therefore
 9629 develop design criteria which are unconditional on $\{\mathbf{s}\}$. The total variance expression is
 9630 unconditional on \mathbf{s} is:

$$\text{Var}(\tilde{N}) = E_s \text{Var}(\tilde{N}|\mathbf{s}) + \text{Var}_s E(\tilde{N}|\mathbf{s})$$

9631 if we assert that sample sizes will be large enough so that our estimator is unbiased,
 9632 then the 2nd term will be close to 0 and we can ignore it. Therefore to evaluate the
 9633 unconditional variance we need to solve an N -fold integration to average over $\mathbf{s}_1, \dots, \mathbf{s}_N$,
 9634 or we can do this by taking a monte carlo average.

9635 10.6.3 Optimization of the criterion

9636 We need to come up with a ballpark guess of the model parameters. i.e., what is $\boldsymbol{\alpha}$ and
 9637 N ? If we do that, and specify the state-space \mathcal{S} and the number of traps to place, then
 9638 we can optimize the variance criterion.

9639 In formulating the optimization problem note that we have J sample locations corre-
 9640 sponding to rows of \mathbf{X} . The problem is a $2J$ dimensional optimization problem which,
 9641 for J small, could be solved using standard numerical optimization algorithms as exist in
 9642 almost every statistical computation environment. However, J will almost always be large
 9643 enough so as to preclude effective use of such algorithms. This is a common problem in ex-
 9644 perimental design, design for response surface estimation, computer experiments, spatial

9645 sampling designs and other disciplines for which sequential exchange or swapping algo-
 9646 rithms can be used (e.g., Wynn 1970; Fedorov 1972; Mitchell 1974; Meyer and Nachtsheim
 9647 1995). The basic idea is to pose the problem as a sequence of 1-dimensional optimization
 9648 problems in which the objective function is optimized over 1 or several coordinates at a
 9649 time.

9650 In the present case, we consider swapping out \mathbf{x}_j for some point in \mathcal{X} that is nearby
 9651 \mathbf{x}_j (e.g., a 1st order neighbor). The objective function is evaluated for all possible swaps
 9652 (at most 4 in the case of 1st order neighbors) and whichever point yields the biggest
 9653 improvement is swapped for the current value. The algorithm is iterated over all J design
 9654 points and this continues until convergence is achieved. Such algorithms may yield local
 9655 optima and optimization for a number of random initial designs can yield incremental
 9656 improvements. We implemented this swapping algorithm in **R**, using the basic strategy
 9657 employed elsewhere (e.g., Nychka et al. 1997; Royle and Nychka 1998). A version of
 9658 a swapping algorithm used to optimize a space-filling criterion is implemented in the
 9659 **R** package **fields** (Fields Development Team 2006). I developed an implementation that
 9660 requires a discrete representation of \mathcal{S} (an arbitrary matrix of coordinates) and an indicator
 9661 of which elements of \mathcal{S} are members of the design space \mathcal{X} . For each point in \mathbf{X} , only the
 9662 nearest neighbors (the number is specified) are considered for swapping into the design
 9663 during each iteration.

9664 While swapping algorithms are convenient to implement, and efficient at reducing
 9665 the criterion in very high dimensional problems, they do not always yield the global
 9666 optimum. In practice, as in the examples below, it is advisable to apply the algorithm to
 9667 a large number of random starting designs. My experience is that essentially meaningless
 9668 improvements are realized after searching through a few dozen random starts.

9669 10.6.4 Illustration

9670 Consider designing a study for camera traps in a square region defined by the square
 9671 $[10, 20] \times [10, 20]$ and with $\mathcal{X} = \mathcal{S}$. For this illustration I assumed $\beta_0 = \log(\lambda_0) = -2.7$
 9672 and $\beta_1 = 1/(\sigma^2) = 1/4, 1/9$ and $1/16$, so $\sigma = 2, 3, 4$. (this was dumb - note that σ is really
 9673 2 times the standard deviation of a normal distribution. Oh well!). Designs of size 9 and
 9674 10 were computed for each value of σ using many random starting designs. The putative
 9675 optimal designs (henceforth “best”) are shown¹ in Figure 14.2. For $J=9$, $\sigma = 2$, the best
 9676 design was produced in 180 out of 1000 random starts. For $\sigma = 3$ (row 2, left panel) the
 9677 best design was produced in about 88% of all optimizations from random starting values.
 9678 For $J = 10$, and $\sigma = 2$ (row 1, right panel), the best design was found about 24% of the
 9679 time (from random starts). The $\sigma = 3$ best design (row 2, right panel; 14% of random
 9680 starts) clusters 2 points in the center. Finally, consider the $\sigma = 4$ case (last row of Fig.
 9681 14.2). We have two irregular looking designs and the design points cluster in various ways.

9682 I computed the best designs using the same settings but increasing the size of \mathcal{S} relative
 9683 to \mathcal{X} . In particular, I nested \mathcal{X} into $[9, 21] \times [9, 21]$ (Figure ??) and then $[8, 22]^2$ (Figure
 9684 ??). The obvious effect of this is that the best designs move points toward the edge of
 9685 the design space \mathcal{X} so as to provide more exposure to points in \mathcal{S} . The effect is more

¹My intention is to provide many of these results in an Appendix in order to reduce the length of the paper.

9686 pronounced, obviously, as you provide more area outside of \mathcal{X} that is allowed to influence
9687 the design.

9688 As a final example, consider placing 20 camera traps in this region. Where do they
9689 go? Look at the 3 buffers, 3 values of sigma, that's 9 total designs (use a single panel).
9690 An interesting feature of the designs is that they are not regular. Traps occur in clusters
9691 of several traps close together with the clusters more widely spaced.

10.7 COVARIATE MODELS

9692 if the objective is to estimate a covariate effect on density then you should build this into
9693 the criterion..... In this case we can think of the captures in a trap being a Poisson r.v.
9694 with mean $\lambda(\text{trap}, s) * D(s)$ and the problem is slightly more complicated but it can
9695 be done... the calculus to work out the var-cov matrix of density parameters needs to be
9696 done,

9697 Intuitively, model-based approaches in this case should favor areas of higher density....

10.8 SUMMARY AND OUTLOOK

9698 Heuristics: recaptures vs sample size of individuals. The two objectives trade-off. We
9699 need designs that are good for estimating \bar{p} and also designs that obtain a high sample
9700 size of n . Designs that are only good for one or the other will produce bad SCR designs, or
9701 designs in which N is not estimable. One exception is when telemetry is available. These
9702 provide information on σ or other parameters of the detection model and this changes the
9703 whole situation so that trap arrays should be more spread out.

9704 In general though, for basic situations, trap spacing should be about XXXX σ .
9705 Clustering is important too.

9706 We should always do a simulation study. This allows us to learn what to expect as we
9707 start collecting real data. Plus we can simulate for any complex situation that we desire.

9708 However formal model-based design of SCR models has great potential and we think
9709 this is where things will be going. SCR models are amenable to some degree of analytic
9710 study using classical spatial design ideas. We have just barely scratched the surface here,
9711 showing how to formulate a criterion that is a function of the design, and then optimizing
9712 the criterion over all possible designs. We believe this approach merits more attention.

9713

Part III

9714

9715

Advanced SCR Models

9716
9717

11

9718
9719

MODELING SPATIAL VARIATION IN DENSITY

9720 Underlying every spatial capture-recapture models is a point process that describes the
9721 number and distribution of animal activity centers within the state-space (\mathcal{S}). A spatial
9722 point process is characterized by an intensity parameter defined at each location in \mathcal{S} ;
9723 and in the case of SCR models, this intensity parameter is population density. If the
9724 intensity is constant, density is constant throughout \mathcal{S} and the point process is said to be
9725 homogeneous. Thus far we have focused our attention on homogeneous point processes
9726 whose realized values are the locations of the N activity centers within the state-space.
9727 When a Poisson prior is placed on N , we have a homogeneous Poisson point process,
9728 which is referred to as a model of “complete spatial randomness.” A similar model, that
9729 we often use in conjunction with data augmentation and MCMC, places a binomial prior
9730 on N . This is also a model of spatial randomness, and in this chapter we will compare
9731 and contrast the two.

9732 The spatial randomness assumption is often viewed as restrictive because ecological
9733 processes such as territoriality and habitat selection can result in non-uniform distributions
9734 of organisms. We have argued, however, that this assumption is less restrictive than may
9735 be recognized because a homogeneous point process actually allows for infinite possible
9736 “point patterns”, or realized configurations of activity centers. Furthermore, given enough
9737 data, the uniform prior will have very little influence on the estimated locations of activity
9738 centers. Nonetheless, a homogeneous point process does not allow one to model population
9739 density using covariates, which is an important objective in much ecological research. For
9740 example, even when assuming a homogeneous point process model for the activity centers,
9741 an estimated density surface may strongly suggest that individuals are more abundant in
9742 one habitat than another; however, such results do not provide the basis for formally
9743 testing hypotheses about spatial variation in density, and they could not be used to make
9744 predictions about habitat-specific abundance in other regions. A more direct approach
9745 is to replace the homogeneous model with an inhomogeneous model in which the point
9746 process intensity is allowed to vary spatially.

9747 In this chapter, we cover methods for fitting inhomogeneous Poisson and binomial

9748 spatial point process models by treating the intensity parameter as a function of covariates,
 9749 in much the same way as is done in generalized linear models. The covariates we consider
 9750 differ from those covered in previous chapters, which were typically attributes of the
 9751 animal (e.g. sex or age) or the trap (e.g. baited or not) and were used to model movement
 9752 or encounter rate. In contrast, here we wish to model covariates that are defined at all
 9753 points in \mathcal{S} , which we will refer to as state-space covariates or density covariates. These
 9754 may include continuous covariates such as elevation, or discrete covariates such as habitat
 9755 type. Such covariates are often formatted as raster images with a prescribed resolution
 9756 and extent.

9757 Inhomogeneous Poisson point process models were discussed in the original formulation
 9758 of SCR models by Efford (2004) and were described in more detail by Borchers and Efford
 9759 (2008). We will show that an inhomogeneous binomial point process is quite similar to
 9760 the Poisson model, but is more easily implemented in MCMC algorithms. To do so, we
 9761 will define the data augmentation parameter ψ in terms of the point process intensity
 9762 function, and we will replace the uniform prior on the activity centers with a prior that is
 9763 also derived from the intensity function. Development of this prior, which does not have
 9764 a standard form, is a central component of this chapter. First we begin with a review of
 9765 homogeneous point process models.

11.1 HOMOGENEOUS POINT PROCESS REVISITED

9766 The homogeneous Poisson point process is *the* model of complete spatial randomness and
 9767 is often used in ecology as a null model to test for departures from randomness (Cressie,
 9768 1991; Diggle, 2003; Illian et al., 2008). The Poisson model asserts that the number of points
 9769 in \mathcal{S} is Poisson distributed: $N \sim \text{Poisson}(\mu|\mathcal{S}|)$ where $\mu > 0$ is the intensity parameter
 9770 and $|\mathcal{S}|$ is the area of the state-space. The intensity parameter μ is the density of points,
 9771 and thus multiplying the intensity by the area of some region yields the expected number
 9772 of points in that region. As with all homogeneous point process models, the N points
 9773 are distributed uniformly, which implies that they do not interact with each other in any
 9774 way—for example, they neither attract nor repel one another.

9775 Unlike the Poisson point process, the binomial point process assumes that N is fixed,
 9776 not random. The distinction is illustrated by this simple R code that generates realizations
 9777 from Poisson and binomial point processes in the unit square ($\mathcal{S} = [0, 1] \times [0, 1]$):

```
9778 Area <- 1                      # Area of unit square
9779 muP <- 4                        # intensity
9780 nP <- rpois(1, muP*Area)        # number of points: random
9781 PPP <- cbind(runif(nP), runif(nP)) # Poisson point pattern
9782
9783 nB <- 4                        # number of points: fixed
9784 muB <- nB/Area                  # intensity
9785 BPP <- cbind(runif(nB), runif(nB)) # binomial point pattern
```

9786 Both of these models are homogeneous because the intensity parameter is constant ($\mu = 4$
 9787 in both cases) and the N points do not interact with each other. This results from the
 9788 fact that the locations of the points follow a uniform distribution on the plane. The key
 9789 distinction is that N is random in the former and fixed in the latter.

Another difference between the Poisson and binomial models is that if the state-space is divided into K disjunct regions, the number of points in each region $n(B_k) : k = 1, \dots, K$; are independent and identically distributed (i.i.d.) under the Poisson model but not under the binomial model. In the Poisson case, the counts are simply distributed as $n(B_k) \sim \text{Poisson}(\mu|B_k|)$, where $|B_k|$ is the area of the region B_k . For the binomial case, $n(B_k) \sim \text{Binomial}(N, \pi(B_k))$ where $\pi(B_k)$ is the proportion of the state-space in B_k ; however, these counts are not i.i.d. because the number of points in one region is informative about the number of points in another region. For example, if $N = 10$, which would be known for a binomial point process, and if we know that there are 7 points outside the region B_1 , then we can say with certainty that $B_1 = 10 - 7 = 3$.

Fig. 11.1 is meant to further illustrate the characteristics of the binomial model. The left panel shows a point pattern realized from a homogeneous binomial point process with $N = 50$. The right panel shows the same realization, except that the state-space has been discretized into 25 equally-sized disjunct regions, or pixels, and the counts in each pixel are shown. Since the pixels are the same size, $\pi(B_k) = 1/25$, the expected number of point in each pixel is 2: $\mathbb{E}(n(B_k)) = N\pi(B_k) = 50/25$, which happens to be the empirical mean in this instance. However, as previously stated, these counts are not independent realizations from a binomial distribution since $\sum_k n(B_k) = N$. Rather, the model for the entire vector is multinomial: $\{n(B_1), n(B_2), \dots, n(B_K)\} \sim \text{Multinomial}(N, \{p(B_1), p(B_2), \dots, p(B_K)\})$ (Illian et al., 2008). If you need a refresher on the multinomial distribution, refer to Sec. 2.2.3, and consider the following R code, which generates counts such as those seen in Fig. 11.1:

```
9812 n.Bk <- rmultinom(1, size=50, prob=rep(1/25, 25))
9813 matrix(n.Bk, 5, 5)
```

The dependence among counts has virtually no practical consequence when the number of pixels is large. For example, if there are 100 pixels, the number of points in one pixels carries very little information about the expected number of points in another pixel. However, if there are only 2 pixels, then clearly the number of points in one pixel allows one to determine how many points will occur in the remaining pixel.

The discrete representation of space shown in Fig. 11.1 is not only helpful for understanding the properties of a point process, it is also of practical importance when fitting SCR models because spatial covariates are almost always represented as rasters, i.e. grids with predetermined extent and resolution. In such cases, the definition of the prior for the point locations can be changed from the probability that a point occurs at some location in space to the probability that it occurs in some pixel of the raster. As we will explain in Sec. 11.4.2, this typically involves changing the prior from a uniform distribution to a multinomial or categorical distribution.

Up to this point in this chapter we have sketched out the basic characteristics of homogeneous Poisson and binomial point process models. Now we need to speak more specifically about their relevance to SCR models before we move on to the inhomogeneous models. In a SCR model with a homogeneous point process, the intensity parameter μ is interpreted as population density, and N is interpreted as population size¹. These interpretations are true regardless of whether we consider the Poisson model or the binomial

¹Strictly speaking, N is the number of activity centers in \mathcal{S}

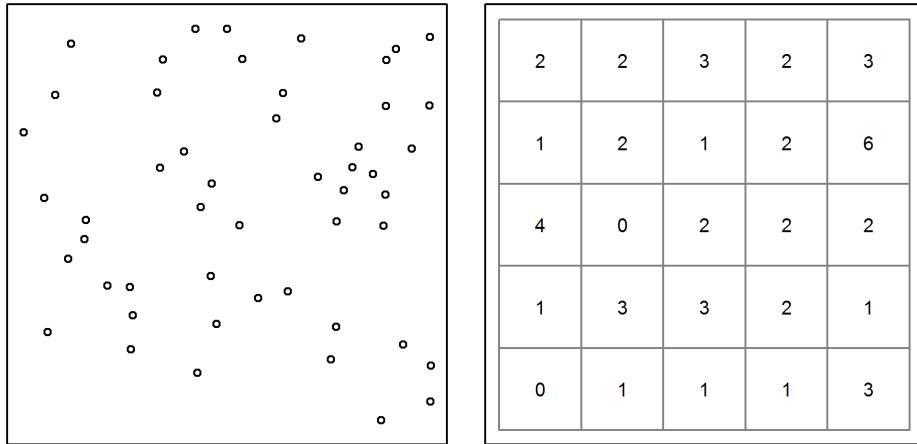


Figure 11.1. Homogeneous binomial point process with $N=50$ points represented in continuous and discrete space.

model, but since N is always unknown, one might wonder why we are discussing the binomial model at all.

In our work, we typically adopt the binomial model simply because it is easy to implement using MCMC and data augmentation. And while N is truly unknown, we use an upper bound M which is fixed. Thus, the standard point process we use Bayesian in analyses can be regarded in two ways. First, it is a binomial point process with M points. Second, in terms of N , it is a thinned binomial point process, where ψ is the thinning parameter. XXXX Is this thinned point process also binomial, even though N is no longer fixed? XXXX With this in mind, the only real difference between the Poisson and binomial models, as implemented in SCR contexts, is that in the former, we have $N \sim \text{Poisson}(\mu|\mathcal{S})$, and in the latter we have $N \sim \text{Binomial}(M, \psi)$. In other words, we just have a different prior on N , and when using MCMC, the binomial prior is much more convenient because it fixes the size of the parameter space and makes it easy to extend the model in each of the ways discussed in this book. It is also worth remembering that the Poisson distribution is the limit of the binomial distribution when M is very high and ψ is very low (Chapt. 2), and thus the two models are much more similar than may appear.

You might have noticed that the intensity parameter μ was not shown for the binomial prior $N \sim \text{Binomial}(M, \psi)$. Instead, we see the data augmentation parameter ψ , which has been used throughout this book, but without much mention of the point process intensity. What then is the relationship between ψ and μ ? As first discussed in Chapt.5, under data augmentation, the expected value of N is $\mathbb{E}[N] = M\psi$. But, from this chapter, we also know that the expected value of N can be written in terms of μ as $\mathbb{E}[N] = \mu|\mathcal{S}|$. Therefore, $\psi = \mu|\mathcal{S}|/M$ and hence we can directly estimate μ rather than ψ if we so desire—and we will so desire in the next section where the objective is to model μ as

9857 a function of spatially-referenced covariates. First, as an exercise, execute the following
 9858 **R** commands to familiarize yourself with some of the concepts we just covered:

```
9859 Area <- 1          # Area of state-space
9860 M <- 100           # Data augmentation size
9861 mu <- 10           # Intensity (points per area)
9862 psi <- (mu*Area)/M # Data augmentation parameter (thinning rate)
9863 N <- rbinom(M, 1, psi) # Realized value of N under binomial prior
9864 cbind(runif(N), runif(N)) # Point pattern from thinned binomial model
```

11.2 INHOMOGENEOUS POINT PROCESSES

9865 The principal difference between homogeneous and inhomogeneous point processes is that
 9866 the intensity parameter μ is allowed to vary spatially in the latter. Thus, rather than μ
 9867 being a fixed constant, it is now a function defined at each point $\mathbf{s} \in \mathcal{S}$. A vast number
 9868 of options exist for modeling spatial variation in the intensity of a point process (Cox,
 9869 1955; Stoyan and Penttinen, 2000; Illian et al., 2008), but here we focus on modeling μ
 9870 as a function of spatially-referenced covariates and a vector of regression coefficients β ; a
 9871 function we will denote $\mu(\mathbf{s}, \beta)$. To be clear, $\mu(\mathbf{s}, \beta)$, is a function that returns the expected
 9872 density of activity centers at location \mathbf{s} , given the covariate values at \mathbf{s} . Since the intensity
 9873 must be positive, and because the natural logarithm is the canonical link function of the
 9874 Poisson generalized linear model (McCullagh and Nelder, 1989), it is natural to consider
 9875 the following model:

$$\log(\mu(\mathbf{s}, \beta)) = \beta_0 + \sum_{v=1}^V \beta_v z_v(\mathbf{s}) \quad (11.2.1)$$

9876 which says that there are V covariates and β_v is the regression coefficient for covariate
 9877 $z_v(\mathbf{s})$. This covariate, $z_v(\mathbf{s})$, could be any variable defined at all points in the state-
 9878 space, such as habitat type or elevation. Eq. 11.2.1 should look familiar because it is
 9879 the standard linear predictor used in Poisson regression. As with other GLMs, one could
 9880 consider alternative link functions.

9881 Recall from the previous section that for a homogeneous point process, the expected
 9882 number of points in the state-space was simply the intensity parameter multiplied by area:
 9883 $\mathbb{E}[N] = \mu|\mathcal{S}|$. But now that we are regarding the intensity as a function, rather than a
 9884 scalar, this equation is not very useful. So what is $\mathbb{E}[N]$ for an inhomogeneous point
 9885 process? Contemplating a discrete state-space is useful for figuring this out. Imagine
 9886 that the state-space is represented as a raster with many tiny pixels. In this case, we
 9887 will associate \mathbf{s} with pixel ID, i.e. \mathbf{s} just references some pixel with V covariates values
 9888 associated with it. The expected number of individuals in this pixel, say $\mathbb{E}[n(\mathbf{s})]$, can
 9889 intuitively be found by evaluating the intensity function (Eq. 11.2.1) and multiplying it
 9890 by the area of the pixel. In other words, we compute the expected number of individuals
 9891 in a pixel by multiplying the expected value of density for that pixel by the area of the
 9892 pixel. If we do this for each pixel in the state-space, then summing up these values gives
 9893 us what we are after, the expected value of N . Specifically, $\mathbb{E}[N] = \sum_{\mathbf{s} \in \mathcal{S}} \mathbb{E}[n(\mathbf{s})]$. As
 9894 the area of the pixels approaches zero, such that we move from discrete space back to

9895 continuous space, the summation must be replaced with an integration of the form:

$$\mathbb{E}[N] = \int_{\mathcal{S}} \mu(\mathbf{s}, \boldsymbol{\beta}) d\mathbf{s}. \quad (11.2.2)$$

9896 Together, Eqs. 11.2.1 and 11.2.2 describe a model for spatial variation in density as well
 9897 as population size. The key task in fitting such inhomogeneous point process models is to
 9898 estimate the $\boldsymbol{\beta}$ parameters.

9899 We have now described an approach for modeling the point process intensity, yet in
 9900 order to define the likelihood or to develop an MCMC algorithm for the inhomogeneous
 9901 model, we need to specify the prior distribution for the activity centers. Recall that under
 9902 the homogeneous point process, the prior was $\mathbf{s}_i \sim \text{Uniform}(\mathcal{S})$, for $i = 1, \dots, N$, or
 9903 equivalently:

$$[\mathbf{s}_i] = 1/|\mathcal{S}| \quad (11.2.3)$$

9904 where once again $|\mathcal{S}|$ denotes the area of the state-space. This simply indicates that an
 9905 activity center is equally likely to occur at any location in the state-space. However, if
 9906 animals exhibit habitat selection or simply occur in one region more often than another,
 9907 it would be preferable to replace this prior with one describing the spatial variation in
 9908 density. Clearly this prior should be determined in some way by the spatially-varying
 9909 intensity function $\mu(s, \boldsymbol{\beta})$. Since the integral of a probability density function (pdf) must
 9910 be unity, we can convert $\mu(\mathbf{s}, \boldsymbol{\beta})$ into a pdf by dividing it by a normalizing constant. In this
 9911 case, the normalizing constant is found by integrating $\mu(s, \boldsymbol{\beta})$ over the entire state-space.
 9912 The probability density function of the new prior is therefore:

$$[\mathbf{s}_i | \boldsymbol{\beta}] = \frac{\mu(\mathbf{s}_i, \boldsymbol{\beta})}{\int_{\mathcal{S}} \mu(\mathbf{s}, \boldsymbol{\beta}) d\mathbf{s}} \quad (11.2.4)$$

9913 Substituting the uniform prior with this new distribution allows us to fit inhomogeneous
 9914 binomial point process models to spatial capture-recapture data.

9915 As a practical matter, note that the integral in the denominator of Eq. 11.2.4 is
 9916 evaluated over space, and since we always regard space as two-dimensional (the state-
 9917 space is planar), this is a two-dimensional integral that can be approximated using the
 9918 methods discussed in Chapter 9, which include Monte Carlo integration and Gaussian
 9919 quadrature. Alternatively, if our state-space covariates are in raster format, i.e. they are
 9920 in discrete space, the integral can be replaced with a summation over all the pixels in the
 9921 raster,

$$[\mathbf{s}_i | \boldsymbol{\beta}] = \frac{\mu(\mathbf{s}_i, \boldsymbol{\beta})}{\sum_{\mathbf{s} \in \mathcal{S}} \mu(\mathbf{s}, \boldsymbol{\beta})} \quad (11.2.5)$$

9922 where \mathbf{s} is now defined as “pixel ID” rather than a point in space.

9923 Although the discrete space approach is standard practice, it is technically unjustified
 9924 because covariate values must be known for all points in space. This same problem is
 9925 present anytime that we have a sample of the spatial covariates, rather than a function
 9926 defining their value for all points in space. In such cases, it may be necessary to interpolate
 9927 the values of the covariates for points in space where they were not measured. One option
 9928 would be to use a Kriging interpolator, as demonstrated by Rathbun (1996). Another
 9929 option is to sample the spatial covariates using probabilistic sampling methods, which
 9930 allow for design-based estimators of their values for the entire study area (Rathbun et al.,

9931 2007). Either option could be implemented within maximum likelihood or MCMC esti-
 9932 mation methods; however, we do not demonstrate them here because it seems likely that
 9933 they will be inconsequential in most cases where the raster data are of high resolution,
 9934 such that the loss of information is negligible when going from continuous space to discrete
 9935 space. Furthermore, the validity of this assertion, and the level of resolution required to
 9936 adequately approximate continuous space can often be assessed by checking the consis-
 9937 tency of the parameter estimates among varying levels of resolution, as was demonstrated
 9938 in Chapt. 5.

9939 We now have all the tools needed to fit inhomogeneous point process models. Likelihood-
 9940 based inference for inhomogeneous Poisson point process models was described by Borchers
 9941 and Efford (2008) and reviewed in Chapt. 6. Another example is demonstrated in the next
 9942 section, but first we focus on the binomial model that we favor when conducting Bayesian
 9943 inference. In the previous section we noted that the data augmentation parameter ψ can
 9944 be expressed in terms of the intensity parameter μ . The same is true for inhomogeneous
 9945 models. Specifically, rather than $\mathbb{E}[N] = \psi M$ as before, we use the expected value of N
 9946 shown in Eq. 11.2.2 which results in

$$\psi = \frac{\int_S \mu(\mathbf{s}, \boldsymbol{\beta}) d\mathbf{s}}{M} \quad (11.2.6)$$

9947 Note that the data augmentation limit M must be high enough so that it is greater than
 9948 the numerator—i.e. the expected value of N must be less than M .

If we refer to the distribution $[\mathbf{s}_i | \boldsymbol{\beta}]$ as “IPP”, we can write a hierarchical description of
 a SCR model with a Binomial encounter process and a half-normal, or Gaussian, detection
 function as

$$\begin{aligned} w_i &\sim \text{Bernoulli}(\psi) \\ \mathbf{s}_i &\sim \text{IPP}(\mu(\mathbf{s}, \boldsymbol{\beta})) \\ p_{ij} &= p_0 \exp(-\|\mathbf{s}_i - \mathbf{x}_j\|^2 / (2\sigma^2)) \\ y_{ij} &\sim \text{Binomial}(K, p_{ij}w_i) \end{aligned}$$

9949 The new prior for \mathbf{s}_i and Eq. 11.2.6 are the key differences between homogeneous an
 9950 inhomogeneous models.

9951 In the next sections we walk through a few examples, building up from the simplest
 9952 case where we actually observe the activity centers as though they were data. In the
 9953 second example, we fit our new model to simulated data in which density is a function of
 9954 a single continuous covariate. To build upon the developments in the previous chapter,
 9955 we further consider the plausible case where a state-space covariate is also a covariate of
 9956 ecological distance. A small simulation study indicates that both effects can be estimated.
 9957 A fourth example shows an analysis in discrete space using both **secr** (Efford, 2011) and
 9958 **JAGS** (Plummer, 2003). In the fifth and final example, we model the intensity of activity
 9959 centers for a real dataset collected on jaguars (*Panthera onca*) in Argentina.

11.3 OBSERVED POINT PROCESSES

9960 In SCR models, the points (activity centers) are not directly observed, but in other con-
 9961 texts they are. Examples include the locations of disease outbreaks, the locations of trees

9962 in a forest, or the locations of radio-tracked animals. In such cases, it is straightforward
 9963 to fit inhomogeneous point process models and estimate the the parameters β from
 9964 Eq. 11.2.1, as we will do in the following example.

9965 Suppose we knew the locations of N animal activity centers, perhaps as the result of
 9966 an extensive telemetry study. If we assume N is Poisson distributed and the points are
 9967 mutually independent of one another, we can fit the inhomogeneous Poisson point process
 9968 model whose likelihood is the product of N densities given by Eq. 11.2.4 (Diggle, 2003,
 9969 pg. 104). The log-likelihood is thus:

$$\ell(\beta | \{s_i\}) = \sum_{i=1}^N \log(\mu(s_i, \beta)) - \int_S \mu(s, \beta) ds.$$

9970 Having defined the likelihood we could choose a prior distribution for β and obtain the
 9971 posterior distribution using Bayesian methods, or we can find the maximum likelihood
 9972 estimates (MLEs) using standard numerical methods as is demonstrated below.

9973 First, we simulate some data under the model $\mu(s, \beta) = \beta_0 + \beta_1 ELEV(s)$, where
 9974 $ELEV(s)$ is a spatial covariate, say elevation, and $\beta_0 = 5$ and $\beta_1 = 2$. It is worth emphasizing
 9975 that a spatial covariate must be defined at any location s , which is demonstrated
 9976 by the following R code.

```
9977 elev.fn <- function(s) {           # spatial covariate
  9978   s <- matrix(s, ncol=2)        # Force s to be a matrix
  9979   (s[,1] + s[,2] - 100) / 40.8 # Returns (standardized) "elevation"
}
9980
9981 # intensity function
9982 mu <- function(s, beta0, beta1) exp(beta0 + beta1*elev.fn(s=s))
9983 beta0 <- -6 # intercept of intensity function
9984 beta1 <- 1 # effect of elevation on intensity
9985 # Next line computes integral
9986 EN <- cuhre(2, 1, mu, beta0=beta0, beta1=beta1,
  9987   lower=c(0,0), upper=c(100,100))$value
```

9988 The function `elev.fn` returns the value of elevation at any location, which can be either
 9989 a two-dimensional vector for the coordinates of a single point, or it can be matrix with
 9990 two columns for a collection of points. The standardization bit is not necessary, but helps
 9991 with the model fitting below. The next lines of the code define the intensity function
 9992 $\mu(s, \beta)$ in terms of elevation and the regression coefficients. The last line uses the `cuhre`
 9993 function in the `R2Cuba` package (Hahn et al., 2010) to compute the expected value of N in
 9994 a $[0, 100] \times [0, 100]$ square state-space, which is the two-dimensional integral of Eq. 11.2.4.
 9995 This integral could also be computed using a fine grid of points as we have done in previous
 9996 chapters, but it is useful to gain familiarity with more efficient integration functions in R.

9997 The R code above demonstrates how to obtain the expected value of N given a spatial
 9998 covariate and the coefficients defining the intensity function. Now we need to generate a
 9999 realized value of N and distribute the N points in proportion to the intensity function.
 10000 This is not as simple as it was to simulate data from a homogeneous point process because
 10001 the points are no longer uniformly distributed within the state-space. Instead one must
 10002 resort to methods such as rejection sampling, which involves simulating data from a stan-
 10003 dard distribution and then accepting or rejecting each point using probabilities defined

10004 by the distribution of interest. For more information, readers should consult an accessible
 10005 text such as Robert and Casella (2010). In our example, we simulate from a uniform dis-
 10006 tribution and then accept or reject using the (scaled) probability density function $[s_i|\beta]$
 10007 (Eq. 11.2.4). The following **R** commands demonstrate the use of rejection sampling to
 10008 simulate an inhomogeneous point process for the elevation covariate depicted in Fig. 11.3.

```
10009 set.seed(31025)
10010 N <- rpois(1, EN)      # Realized N
10011 s <- matrix(NA, N, 2) # This matrix will hold the coordinates
10012 elev.min <- elev.fn(c(0,0))
10013 elev.max <- elev.fn(c(100, 100))
10014 Q <- max(c(exp(beta0 + beta1*elev.min), # max of intensity function
10015             exp(beta0 + beta1*elev.max)))
10016 counter <- 1
10017 while(counter <= N) { # begin rejection sampling
10018   x.c <- runif(1, 0, 100); y.c <- runif(1, 0, 100)
10019   s.cand <- c(x.c,y.c) # proposed activity center
10020   pr <- mu(s.cand, beta0, beta1)
10021   if(runif(1) < pr/Q) { # Typically rejected if pr is low
10022     s[counter,] <- s.cand
10023     counter <- counter+1
10024   }
10025 }
```

10026 Similar methods are also implemented in the **R** package **spatstat** (Baddeley and Turner,
 10027 2005).

10028 The 41 simulated points are shown in Fig 11.3. High elevations are represented by
 10029 light gray and low elevations are darker. The density of points in apparently higher in
 10030 lighter regions suggesting that these simulated animals prefer high elevations. Given these
 10031 points, we will now estimate β_0 and β_1 by minimizing the negative-log-likelihood using
 10032 **R**'s **optim** function.

```
10033 nll <- function(beta) { # negative log-likelihood
10034   beta0 <- beta[1]
10035   beta1 <- beta[2]
10036   EN <- cuhre(2, 1, mu, beta0=beta0, beta1=beta1)$value
10037   -(sum(beta0 + beta1*elev.fn(s)) - EN)
10038 }
10039 starting.values <- c(0, 0)
10040 fm <- optim(starting.values, nll, hessian=TRUE)
10041 cbind(Est=fm$par, SE=sqrt(diag(solve(fm$hessian)))) # estimates and SEs
```

10042 Maximizing the likelihood took a fraction of a second, and we obtained estimates of
 10043 $\hat{\beta}_0 = -5.93$ and $\hat{\beta}_1 = 0.95$, which are very close to the data generating values. The
 10044 95% confidence interval for $\hat{\beta}_1$ is [0.61, 1.3] and since it does not include zero, the null
 10045 hypothesis that $\beta_1 = 0$, i.e. that there is no effect of elevation on density, can be rejected.
 10046 In addition to testing hypotheses, these results can be used to predict population size in

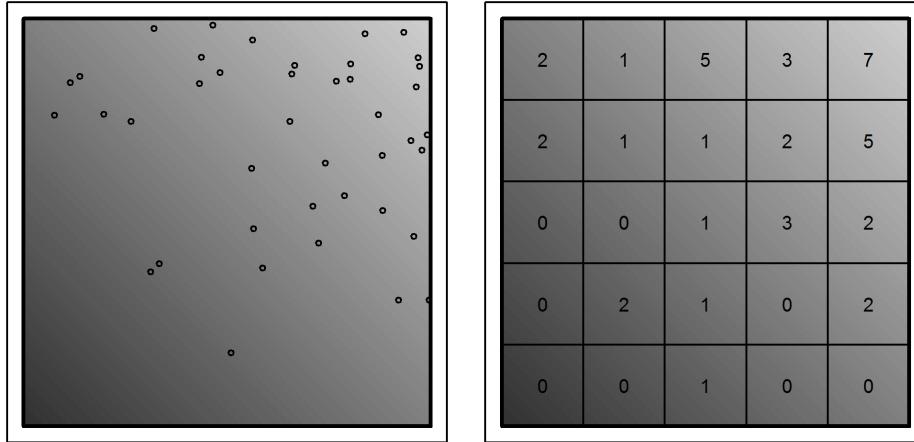


Figure 11.2. An example of a spatial covariate, say elevation, and a realization from an inhomogeneous Poisson point process with $\mu(\mathbf{s}, \boldsymbol{\beta}) = \exp(\beta_0 + \beta_1 \text{ELEV}(\mathbf{s}))$ where $\beta_0 = -6$ and $\beta_1 = 1$.

10047 new regions or create predicted density surface maps by plugging the parameter estimates
10048 into Eqs. 11.2.1 and 11.2.2.

10049 This example demonstrates that if we had the data we wish we had, i.e. if we knew
10050 the coordinates of the activity centers, we could easily estimate the parameters governing
10051 the underlying point process and make inferences about spatial variation in density and
10052 abundance. Unfortunately, in virtually all animal ecology studies, including SCR, the
10053 locations of the N animals, or the N activity centers, cannot be directly observed. Thus
10054 we need extra information to estimate the locations of these unobserved points. In SCR,
10055 this information comes from the locations where each animal is captured.

11.4 FITTING INHOMOGENEOUS POINT PROCESS SCR MODELS

10056 11.4.1 Continuous space

10057 In this example, we will use the same set of points simulated in the previous section to
10058 generate spatial capture-recapture data. Specifically, we overlay a grid of 49 traps on
10059 the map shown in Fig. 11.3 and simulate capture histories conditional upon the activity
10060 centers. Then, we will attempt to estimate the activity center locations as though we did
10061 not know where they were, as is the case in real applications. We will also estimate β_0
10062 and β_1 as before and see how the estimates compare when the points are not actually
10063 observed. The following R code simulates encounter histories under a Poisson observation
10064 model (see Chapt. 9), which could be appropriate in camera trapping studies or when
10065 using other methods in which animals could be detected multiple times at a trap during

```

10066 a single occasion.

10067 xsp <- seq(20, 80, by=10); len <- length(xsp)
10068 X <- cbind(rep(xsp, each=len), rep(xsp, times=len)) # traps
10069 ntraps <- nrow(X); nooccasions <- 5
10070 y <- array(NA, c(N, ntraps, nooccasions)) # capture data
10071 sigma <- 5 # scale parameter
10072 lam0 <- 1 # basal encounter rate
10073 lam <- matrix(NA, N, ntraps)
10074 set.seed(5588)
10075 for(i in 1:N) {
10076   for(j in 1:ntraps) {
10077     # The object "s" was simulated in previous section
10078     distSq <- (s[i,1]-X[j,1])^2 + (s[i,2] - X[j,2])^2
10079     lam[i,j] <- exp(-distSq/(2*sigma^2)) * lam0
10080     y[i,j,] <- rpois(noccasions, lam[i,j])
10081   }
10082 }
10083 # data augmentation
10084 nz <- 80; M <- nz+nrow(y)
10085 yz <- array(0, c(M, ntraps, K))
10086 yz[1:nrow(y),,] <- y # Fill data augmentation array

```

10087 Now that we have a simulated capture-recapture dataset y and we have augmented it to
 10088 create the new data object yz , we can estimate the parameters using MCMC. A commented
 10089 Gibbs sampler written in **R** is available in the accompanying **R** package **scrbook** (see
 10090 [?scrIPP](#)). This function is not meant to be an all purpose tool for fitting SCR models
 10091 using MCMC—instead, it is presented so that interested readers can better understand
 10092 the computational aspects of the problem and can modify it for their purposes. The
 10093 function can be used as so:

```

10094 set.seed(3434)
10095 fm1 <- scrIPP(yz, X, M, 10000, xlims=c(0,100), ylims=c(0,100),
10096   space.cov=elev.fn,
10097   tune=c(0.4, 0.2, 0.3, 0.3, 7))
10098 plot(mcmc(fm1$out))

```

10099 which requests 10000 posterior samples and estimates the effect of the spatial covariate,
 10100 elevation, on density. Currently, the function places uniform priors on the parameters σ ,
 10101 λ_0 , β_0 and β_1 , although this could easily be modified. Note that any spatial covariate
 10102 that returns a real value for any location in the state-space can be supplied using the
 10103 **space.cov** argument. The resulting trace plots of the Markov chains and the posterior
 10104 distributions for three parameters are shown in Fig. 11.3. The chains appear to converge
 10105 rapidly but may need to be run longer to reduce Monte Carlo error.

10106 Summaries of the posterior distributions are presented in Table 11.1. The posterior
 10107 means for β_0 and β_1 are quite similar to MLEs from the analysis in the previous section
 10108 in which we assumed no observation error. However, we see that the confidence intervals
 10109 are wider. With respect to the other parameters in the model, we see that all of the

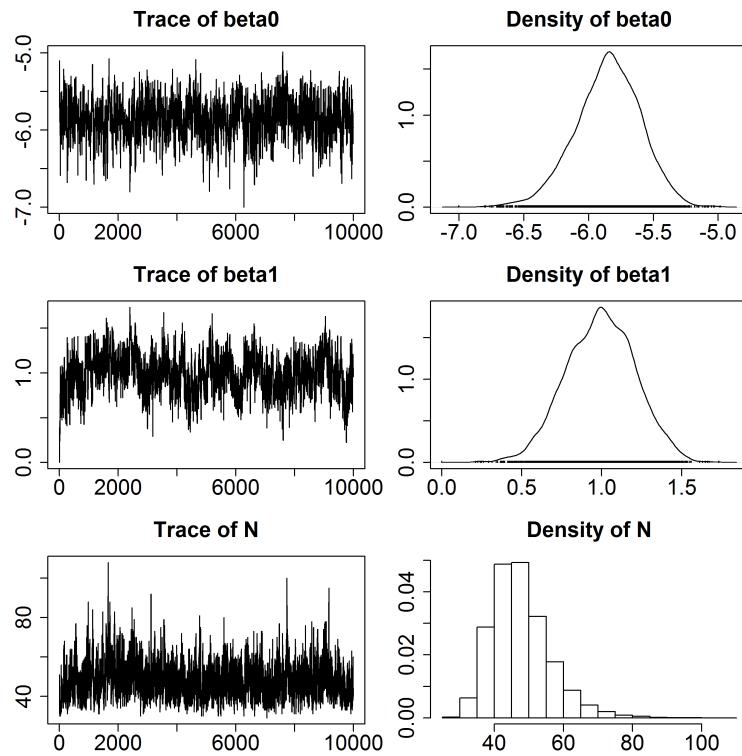


Figure 11.3. Trace plots and posterior distributions from MCMC analysis of SCR model with inhomogeneous point process. Analysis was conducted using the `scrIPP` function in the accompanying **R** package `scrbook`.

Table 11.1. Summary of posterior distributions from SCR model with inhomogeneous point process.

Parameter	Mean	SD	2.5%	97.5%
$\sigma = 5$	5.232	0.310	4.681	5.858
$\lambda_0 = 1$	0.802	0.119	0.595	1.049
$\beta_0 = -6$	-5.856	0.2542	-6.376	-5.393
$\beta_1 = 1$	0.985	0.209	0.575	1.378
$N = 41$	47.615	8.041	35.000	66.000
$\mathbb{E}[N] = 39.9$	47.551	10.992	29.837	71.332

10110 data generating parameter fall within the 95% credible intervals. One interesting thing to
 10111 note is that, although the point estimates for the expected and realized values of N are
 10112 quite similar, the estimate is more precise for the realized value. This is to be expected
 10113 because the uncertainty associated with the realized value of N is entirely determined by
 10114 the encounter rate parameters. That is, if we could perfectly detect all of the individuals
 10115 in \mathcal{S} , there would be no uncertainty about N . In contrast, the variance for expected value
 10116 of N is affected by the variance of all the parameters in the model, not just the encounter
 10117 rate parameters. See Efford and Fewster (2012) for additional discussion on the difference
 10118 between realized and expected values of abundance.

10119 Fitting continuous space inhomogenous point process models is somewhat difficult
 10120 in **BUGS** because the “IPP” prior $[s_i|\beta]$ —unlike the uniform prior—is not one of the
 10121 available distributions that comes with the software. It is possible to add new distributions
 10122 in **BUGS**, but it is somewhat cumbersome. **secr** allows users to fit continuous space
 10123 models using linear or polynomial functions of the x- and y- coordinates, but it does not
 10124 accept truly continuous covariates that are functions of space. However, these are not
 10125 really important limitations because discrete space versions of the model are straight-
 10126 forward, and virtually all spatial covariates are, or can be, defined as such.

10127 11.4.2 Discrete space

10128 To fit inhomogeneous point process models using covariates in discrete space, i.e. in raster
 10129 format, we follow the same steps as outlined in Chapter 9—we define s_i as pixel ID,
 10130 and we use the categorical distribution as a prior. This effectively changes the problem
 10131 from estimating the coordinates of an activity center, to estimating the pixel in which
 10132 an activity center is located. As pixel size approaches zero, these two become equivalent.
 10133 A good example is found in (Mollet et al., 2012). Here we present an analysis of the
 10134 simulated data shown in the Fig. 11.3. The spatial covariate, let’s call it forest canopy
 10135 height (CANHT), was simulated using using the code shown on the help page `ch11` in
 10136 **scrbook**. The points are the number of activity centers in each pixel, generated from
 10137 a single realization of the inhomogeneous point process model with intensity $\mu(s,\beta) =$
 10138 $\exp(\beta_0 + \beta_1 \text{CANHT}(s)) \times \text{pixelArea}$, where $\beta_0 = -6$ and $\beta_1 = 1$.

10139 The **BUGS** description of the model is shown in panel 11.1. The vector `probs[]` is
 10140 the prior probability defined by Eq. 11.2.5, which is the probability that an individual’s
 10141 activity center is located at pixel s . `Sgrid` is the matrix of coordinates for each pixel.

10142 This model can also be fit in **secr**, which refers to the raster data as a “habitat
 10143 mask”. **R** code to format the data and fit the models using **secr** and **JAGS** is available
 10144 in **scrbook**—see `help(ch9secrYjags)`. Results of the comparison are shown in Table 11.2
 10145 and are similar as expected. The differences that do exist can be explained by a variety of
 10146 reasons. For one, there exists some Monte Carlo error in the Bayesian posterior summaries.
 10147 There is also the fact that posterior summaries can be computed in numerous ways—
 10148 for example, we could have presented posterior modes or medians instead of means—
 10149 or, we could have shown highest posterior density credible intervals instead of simple
 10150 percentiles. The posteriors would also differ if we chose more informative priors than
 10151 the uniform distributions used here. We see no reason why these issues should be seen as
 10152 limitations of the Bayesian analysis, rather we would argue that the posterior distribution,
 10153 which describes the probability that the parameter equals any particular value, is a better

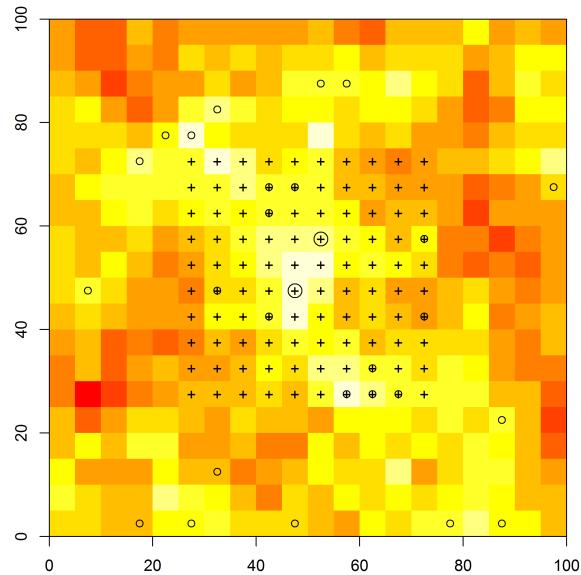


Figure 11.4. Simulated activity centers in discrete space. The spatial covariate, canopy height, is highest in the lighter areas and density increases with canopy height. A single activity center is shown as a small circle, and larger circles represent two activity centers in a pixel. Trap locations are shown as crosses.

descriptor uncertainty than any particular point estimator or confidence interval.

11.5 ECOLOGICAL DISTANCE AND DENSITY COVARIATES

Habitat characteristics that affect spatial variation in density can also affect home range size and movement behavior. For example, a species that occurs at high density in a forest may be reluctant to venture from a forest patch into an adjacent field. Thus, even if a trap placed in a field is located very close to an animal's activity center, the probability of capture may be very low. In this case, forest cover is a covariate of both density and encounter probability, and we could model it as such by combining the methods described in this chapter with those described in Chapter 12.

To demonstrate, we continue with our analysis of the data shown in Fig 11.4.2. Once again, we suppose that density increases with canopy height, but this time, we also make the assumption that home range size decreases as density increases. This commonly-observed phenomenon can be explained by numerous factors such as intra-specific competition (Sillett et al., 2004) or optimal foraging behavior (Tufto et al., 1996; Saïd and Servany, 2005). To model this effect, we introduce the parameter θ , which determines the "cost" of moving between pixels. If $\theta = 0$, then the animal perceives distance as Euclidean. If $\theta > 0$, then least-cost distance (LCD) is greater than Euclidean distance

```

model{
  sigma ~ dunif(0, 20)
  lam0 ~ dunif(0, 5)
  beta0 ~ dunif(-10, 10)
  beta1 ~ dunif(-10, 10)
  for(j in 1:nPix) {
    mu[j] <- exp(beta0 + beta1*CANHT[j])*pixArea
    probs[j] <- mu[j]/EN
  }
  EN <- sum(mu[]) # Expected value of N, E[N]
  psi <- EN/M
  for(i in 1:M) {
    w[i] ~ dbern(psi)
    s[i] ~ dcat(probs[])
    x0g[i] <- Sgrid[s[i],1]
    y0g[i] <- Sgrid[s[i],2]
    for(j in 1:ntraps) {
      dist[i,j] <- sqrt(pow(x0g[i]-traps[j,1],2) + pow(y0g[i]-traps[j,2],2))
      lambda[i,j] <- lam0*exp(-dist[i,j]*dist[i,j]/(2*sigma*sigma)) * w[i]
      y[i,j] ~ dpois(lambda[i,j])
    }
  }
  N <- sum(w[]) # Realized value of N
}

```

Panel 11.1: **BUGS** code for fitting inhomogeneous point process model in discrete space.

10170 (ED). In most cases, we would not expect, or should not even consider the possibility
 10171 of $\theta < 0$ because this implies that LCD<ED, which would mean that an animal could
 10172 view 1000km as 1m. In addition to the fact that this is not biologically justifiable, it also
 10173 suggests that the area of the state-space could be infinitely large. Thus, one may want to
 10174 enforce the constraint that θ is ≥ 0 . See Chapter 12 for more details.

10175 A question that arises is: Is possible to estimate β and θ using standard SCR data? In
 10176 other words, can we model spatial variation in density and connectivity at the same time,
 10177 using standard SCR data? Currently, it is not possible to model least-cost distance using
 10178 **JAGS** or **secr**, so we wrote our own function, **scrDED**, to fit the model using maximum
 10179 likelihood. An example analysis is provided on the help page for the function in our
 10180 **R** package **scrbook**. We briefly note here that the function requires the capture history
 10181 data, the trap locations, and the raster data formatted using the **raster** package (van
 10182 Etten, 2012). The linear model for the intensity parameter $\mu(\mathbf{s}, \beta)$ and the least-cost

Table 11.2. Comparison of **secr** and **JAGS** results. Point estimates from the Bayesian analysis are posterior means. Intervals are lower and upper 95% CIs.

Parameter	Truth	Software	Mean	SD	2.5%	97.5%
λ_0	1.00	JAGS	1.04	0.087	0.88	1.22
	1.00	secr	1.08	0.089	0.92	1.27
σ	10.00	JAGS	10.16	0.373	9.46	10.94
	10.00	secr	9.84	0.350	9.18	10.55
β_1	1.00	JAGS	1.20	0.350	0.50	1.88
	1.00	secr	1.09	0.316	0.47	1.71
N	30.00	JAGS	26.63	2.585	23.00	33.00
	30.00	secr	28.19	3.037	24.49	37.39
$\mathbb{E}[N]$	32.30	JAGS	26.39	5.048	17.25	36.96
	32.30	secr	28.19	6.117	18.52	42.93

10183 distance function $lcd(\theta)$ are specified using **R**'s formula interface. A simple function call
 10184 is

```
10185 fm <- scrDED(y, traplocs=X, den.formula=~elev, dist.formula=~elev,
10186 rasters=elev.raster)
```

10187 To assess the possibility of estimating both β and θ , we conducted a small simulation
 10188 study, generating 500 datasets from the model with both parameters set to 1, which
 10189 corresponds to the conditions described above. The results indicate that it is possible to
 10190 estimate both parameters (Fig 11.5).

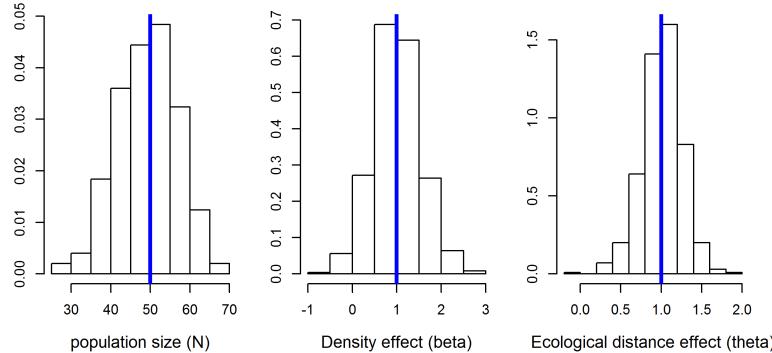


Figure 11.5. Histograms of parameter estimates from 500 simulations under the model in which both density and ecological distance are affected by the same covariate, canopy height. The vertical lines indicate the data-generating value.

11.6 THE JAGUAR DATA

10191 Estimating density of large felines has been a priority for many conservation organizations,
10192 but few robust methodologies existed before the advent of SCR. Distance sampling is not
10193 feasible for such rare and cryptic species, and traditional capture-recapture methods yield
10194 estimates that are highly sensitive to the subjective choice of the effective survey area.
10195 SCR models provide a powerful alternative because density can be estimated directly and
10196 data can be collected using non-invasive methods such as camera traps or hair snares.

10197 In this example, we demonstrate how readily density can be estimated for a globally
10198 imperiled species using SCR. Furthermore, we show how inhomogeneous point process
10199 models can be used to test important hypotheses regarding the factors affecting density.
10200 The data come from an 8-year camera-trapping study designed to assess the impacts
10201 of poaching on jaguar density in Argentina, near the borders of Brazil and Paraguay.
10202 Additional information about the study is presented in Paviolo et al. (2008) and Paviolo
10203 et al. (2009). Although jaguars themselves are occasionally killed by poachers, the larger
10204 concern is the influence of poaching on prey species. To protect jaguars and related
10205 species, protected areas have been established and three levels of protection are recognized
10206 as depicted in Fig. 11.6. The dark green area is the Iguazú National Park that is patrolled
10207 regularly by law enforcement officials. The light green areas are officially protected, but
10208 due to resource limitations, are not patrolled as often. The beige areas are not protected
10209 at all, and the gray areas are large soybean monocultures, which provide no habitat.

10210 To test for differences in density between the three regions, we modeled the point
10211 process intensity parameter as a function of protection status (PROTECT), which we
10212 treated as an ordinal variable: $\mu(\mathbf{s}, \boldsymbol{\beta}) = \exp(\beta_0 + \beta_1 \text{PROTECT}(\mathbf{s}))$. We hypothesized
10213 that $\beta_1 > 0$ indicating that jaguar density increases with protection status. In addition
10214 to modeling spatial variation in density, we also modeled the scale parameter of the half-
10215 normal or Gaussian encounter model as sex-specific, since male cats typically have larger
10216 home ranges than females (Sollmann et al., 2011). Since sex is an individual-specific
10217 covariate, and not observed for the individuals that were not captured, a prior is required
10218 for the sex of uncaptured individuals. We used an Bernoulli prior with probability 0.5 to
10219 describe our uncertainty about sex ratio.

10220 The geometry of the state-space differs greatly from the simple square regions that we
10221 have considered throughout this chapter, which raises a few questions. First, how would
10222 one integrate Eq. 11.2.4 over a complex spatial region? Earlier we used the function **cuhre**
10223 in **R** for the two-dimensional integration, but its **lower** and **upper** arguments essentially
10224 assume that the state-space is square. There are methods of transforming the state-space
10225 to make this work, but once again we find that it is most convenient to work in discrete
10226 space and sum over all the pixels defining \mathcal{S} . In this example, we restricted the state-
10227 space to exclude the large soybean monocultures surrounding the study area, and we only
10228 considered area south of the Iguazú River, which runs along the northern border of the
10229 park shown in dark green in Fig. 11.6. Rather than restricting the state-space, we could
10230 have modeled the permeability of the river using the methods described in the previous
10231 section and in Chapter 12; however, no sampling was conducted on the northern side of
10232 the river, and ancillary data indicates that jaguars rarely forge the waterway.

10233 We fit the model to data from a single year of data from 46 camera stations, each
10234 consisting of a pair of cameras placed along roads or small trails. Forty-five detections of
10235 16 jaguars (8 males and 8 females) were made over a 95-day sampling period. The mean

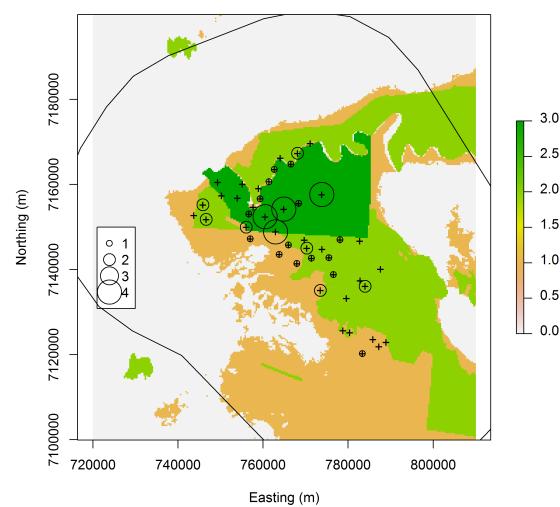


Figure 11.6. Jaguar detections at 46 camera trap stations. The three levels of protection status are no protection (beige), some protection (light green), and Iguazú National Park (dark green). Non-habitat (soybean monocultures) is shown in gray.

Table 11.3. Summaries of posterior distributions from the model of jaguar density. σ is the scale parameter of the half-normal detection function. λ_0 is base-line encounter rate. β_1 is the effect of protection status on jaguar density. ρ is the sex-ratio. N is population size. The last three parameters are the density estimates (jaguars/100km²) for the three levels of protection.

	Mean	SD	2.5%	97.5%
σ_{female}	5434.886	883.7433	4093.6069	7549.062
σ_{male}	6208.341	822.6217	4881.4060	8093.759
λ_0	0.013	0.0036	0.0068	0.021
β_0	-4.667	0.2866	-5.2527	-4.093
β_1	0.196	0.3672	-0.5179	0.961
ρ	0.541	0.0551	0.4286	0.644
N	36.428	9.6986	23.0000	61.000
D_{low}	0.921	0.3851	0.3789	1.894
D_{med}	0.775	0.3006	0.2653	1.503
D_{high}	1.444	0.3325	0.8791	2.110

number of sampling days at each camera station was 48.2. The raw capture data shown in Fig. 11.6 suggest that the highest number of captures was in the national park, but there were also several traps in the park with no captures. Furthermore, few cameras were placed far from the protected areas, making it somewhat difficult to detect differences in density. R code to fit the model is available in `scrbook` on the help page `jaguarDataCh9`. Parameter estimates are shown in Table 11.3.

The results indicate that efforts to protect jaguars by reducing poaching in protected areas are not working as well as hoped for. The posterior probability that $\beta_0 > 0$ was 0.705, and the posterior mean of realized density was only 58% higher in the national park than in the unprotected area. Fig. 11.6 shows the estimated density surfaces. The first map is the expected density (posterior mean) in each of the three values, which was computed by plugging in the posterior mean values of β_0 and β_1 into the log-linear intensity function. The second map is the realized density surface. Conditional on the N , this is the probability distribution for the number of activity centers in each pixel of the rasterized state-space—here shown as the posterior mean. The expected values would be used if we were interested in making inferences about other areas or time periods, whereas the realized map is the best description of the system during the study period.

We note that there is room for improvement in our analysis and our results should be considered preliminary. The political boundaries used to demarcate protected areas are not as concrete as we might like. In reality poaching pressure is likely higher near remote park boundaries than in well-guarded park interiors. One option for addressing this would be to use a continuous measure of poaching pressure such as distance from the nearest town, or some other accessibility metric. It would also be interesting to model density separately for each sex. Many of the detections outside of the park were of males, and thus it is possible that the sexes use habitat differently (Conde et al., 2010). Other extensions worth investigation include treating PROTECT as a categorical, rather than ordinal, variable; and, it would be interesting to assess the effects of roads and trails on jaguar movement using the methods described in Chapt. 12. Developing models for these extensions could be readily accomplished by modifying the fitting functions found in the

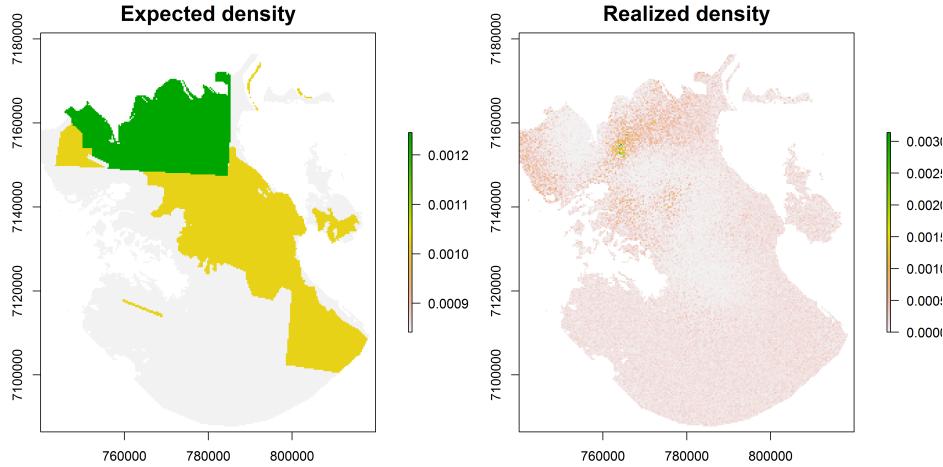


Figure 11.7. Estimated density surfaces from the analysis of the jaguar data.

10265 R package `scrbook`.

11.7 SUMMARY AND OUTLOOK

10266 One of the distinguishing features of spatial capture-recapture models is that they allow
 10267 for inference about spatial variation in density without relying on ad hoc approaches
 10268 for determining the amount of area surveyed. The approach described in this chapter
 10269 involves modeling the locations of activity centers as outcomes of an inhomogeneous point
 10270 process with intensity determined by covariates defined at all locations in the state-space.
 10271 Covariate effects can be evaluated in exactly the same way as is done in generalized linear
 10272 models, making it easy to interpret the results.

10273 All the examples in this section included a single state-space covariate, but this was
 10274 for simplicity only. Including multiple covariates poses no additional challenges. Similarly,
 10275 additional model structure such sex-specific encounter rate parameters or behavioral re-
 10276 sponses can be accommodated and fit using `secr`, **BUGS**, or by extending the functions
 10277 in `scrbook`. It is also possible to consider covariates that affect both density and ecolog-
 10278 ical distance. The ramifications of this are enormous for applied ecological research and
 10279 conservation efforts because researchers can use capture-recapture data to identify areas
 10280 where both density and landscape connectivity are high (Royle et al., 2012a). Address-
 10281 ing such questions is simply not possible using standard, non-spatial capture-recapture
 10282 methods.

10283 Although we focused on modeling the point process intensity as a function of covariates,
 10284 other options for fitting inhomogeneous model exist Illian et al. (2008). Cox processes are
 10285 models in which the intensity parameter is a function of spatial random effects. Such
 10286 methods are useful for accommodating overdispersion, but it seems unlikely that most

10287 SCR datasets could support such complexity. Gibbs processes are another important
10288 class of models that are distinguished by the interactions of points. Although little work
10289 has been done on such models in the context of SCR studies (Reich et al., 2012), we
10290 expect they will receive more attention because they can be used to model processes
10291 such territoriality (points repel one another) or aggregation (points attract one another).
10292 Neyman-Scott processes are another option for modeling aggregation or clustering, and
10293 could be useful for studying gregarious species.

10294
10295

12

10296

MODELING LANDSCAPE CONNECTIVITY

10297 XXXXx Kimmy: As you go through this chapter, we need to know which R
10298 packages are used in the chapter (and other chapters too) XXXXX

10299 Every spatial capture-recapture model that we have considered so far has expressed
10300 encounter probability as function of the Euclidean distance between individual activity
10301 centers s and trap locations x . As a practical matter, models based on Euclidean distance
10302 imply circular, symmetric, and stationary home ranges of individuals, and these are not
10303 often biologically realistic. While these simple encounter probability models will often be
10304 sufficient for practical purposes, especially in small data sets, sometimes developing more
10305 complex models of the detection process as it relates to space usage of individuals will
10306 be useful. Animals may not judge distance in terms of Euclidean distance but, rather,
10307 according to quality of local habitat, landscape connectivity, perceived mortality risk, and
10308 other considerations affecting movement behavior. Moreover, because encounter proba-
10309 bility and the distance metric upon which it is based represent outcomes of individual
10310 movements about their home range, ecologists might have explicit hypotheses about how
10311 environmental variables affect the distance metric, and it is therefore desirable to incor-
10312 porate these hypotheses directly into SCR models so that they may be formally evaluated
10313 statistically.

10314 Assessing the impacts of habitat fragmentation and habitat loss on population density
10315 and landscape connectivity are high priorities in applied ecological research. Landscape
10316 connectivity is defined as the degree to which landscape structure impedes or facilitates
10317 movement (Tischendorf and Fahrig, 2000) and is widely recognized to be an important
10318 component of population viability (With and Crist, 1995). Although much theory has been
10319 developed to predict the effects of decreasing connectivity, few empirical studies have been
10320 conducted to test these predictions due to the paucity of formal methods for estimating
10321 connectivity parameters (Cushman et al., 2010). Instead, ecologists often rely on expert
10322 opinion or *ad hoc* methods of specifying connectivity values, even in important applied
10323 settings (Adriaensen et al., 2003; Beier et al., 2008; Zeller et al., 2012). In addition, no
10324 methods are available for simultaneously estimating population density and connectivity
10325 parameters, in spite of theory predicting interacting effects of density and connectivity

on population viability (Tischendorf et al., 2005; Cushman et al., 2010). In this chapter, following Royle et al. (2012a), we provide a framework for modeling landscape connectivity using SCR models, by parameterizing models for encounter probability based on “ecological distance”. A natural candidate framework for modeling ecological distance is the least-cost path which is used widely in landscape ecology for modeling connectivity, movement and gene flow (Adriaensen et al., 2003; Manel et al., 2003; McRae et al., 2008). In practical applications, variables that influence landscape connectivity, or the effective cost of moving across the landscape, include things like highways (e.g., Epps et al., 2005), elevation (Cushman et al., 2006), ruggedness (Epps et al., 2007), snow cover (Schwartz et al., 2009), distance to escape terrain (Shirk et al., 2010), range limitations (McRae and Beier, 2007), or distance from urban areas, highways, human disturbance or other factors that animals might avoid. Together multiple environmental variables create a resistance surface, which forms the linchpin of all connectivity planning (Spear et al., 2010).

Recently Royle et al. (2012a) provided an SCR framework based on least-cost path for modeling landscape connectivity. They parameterized encounter probability *not* based on Euclidean distance but, rather, based on the least-cost path between an individual’s activity center and a trap location. This is parameterized in terms of one or more parameters that relate the *resistance* of the landscape to explicit covariates. In this way, SCR models can explicitly accommodate landscape structure and account for connectivity of the landscape. For these models based on least-cost path, it is convenient to use a likelihood-based inference framework which we follow here in this chapter. Using this methodological extension of SCR models, it is possible to make formal statistical inferences about movement and connectivity from capture-recapture studies that generate sparse individual encounter history data without subjective prescription of resistance or cost surfaces, which is commonly done in practice. While we believe there should be much ecological interest in developing SCR models that account for landscape connectivity, it is also important for obtaining more accurate estimates of density. Royle et al. (2012a) showed that, under simple models of landscape connectivity (governed by a single covariate), a misspecified model based on Euclidean distance can produce substantial bias in estimates of N and hence density.

12.1 SHORTCOMINGS OF EUCLIDEAN DISTANCE MODELS

In the standard SCR models encounter probability is modeled as a function of Euclidean distance. For example, using the binomial observation model (Chapt. 5), let y_{ij} be individual- and trap specific binomial counts with sample size K and probabilities p_{ij} . The Gaussian model is

$$p_{ij} = p_0 \exp(-d_{ij}^2/(2\sigma^2))$$

where $d_{ij} = \|\mathbf{x}_j - \mathbf{s}_i\|$ is Euclidean distance. As usual, we will sometimes adopt the log-scale parameterization based on $\log(p_{ij}) = \alpha_0 + \alpha_1 d_{ij}^2$ where $\alpha_0 = \log(p_0)$ and $\alpha_1 = -1/(2\sigma^2)$.

The main problem with the Euclidean distance metric in this encounter probability model is that it is unaffected by habitat or landscape structure, and it implies that the space used by individuals is stationary and symmetric which may be unreasonable assumptions for some species. By stationary here we mean in the formal sense of invariance to translation. That is, the properties of an individual home range centered at some point

10368 \mathbf{s} are exactly the same as any other point say \mathbf{s}' . As an example, if the common detection
 10369 model based on a bivariate normal probability distribution function is used, then the implied space usage by *all* individuals, no matter their location in space or local habitat
 10370 conditions, is symmetric with circular contours of usage intensity.

10371 In the framework of Royle et al. (2012a), SCR models explicitly incorporate information
 10372 about the landscape so that a unit of distance is variable depending on identified covariates, say $z(\mathbf{x})$. Thus, where an individual lives on the landscape, and the state of the surrounding landscape, will determine the character of its usage of space. In particular,
 10373 they suggest distance metrics, based on least-cost path, that imply irregular, asymmetric and non-stationary home ranges of individuals. As an example, Fig. 12.1 shows a typical symmetric home range (left panel), and a compressed home range (right panel) resulting from the effect of an environmental variable (center panel) on an animal's movement behavior. We might think of the environmental variable as representing an elevation gradient of a valley and so, for a species that avoids high elevation, space usage will be concentrated in flatter terrain at lower elevations and therefore producing the elliptical home range shape. We reproduce the application from Royle et al. (2012a) later in this chapter, in addition to providing an alternative applied context that involves computing distances within odd-shaped landscape patches (sec. 12.7).

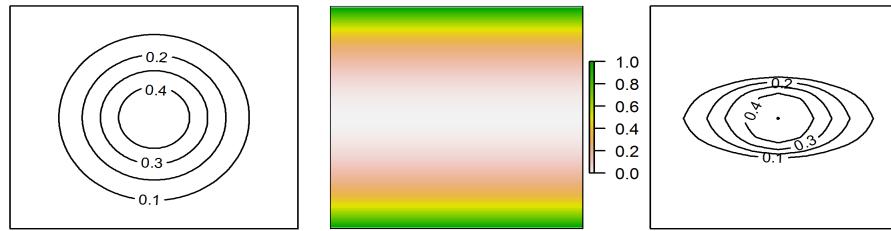


Figure 12.1. A symmetric home range (left), a habitat variable (center) such as representing an elevation gradient, and a non-symmetric home range (right) resulting from the cost imposed on movement by the habitat variable.

12.2 LEAST-COST PATH DISTANCE

10386 We adopt a cost-weighted distance metric here which defines the effective distance between
 10387 points by accumulating pixel-specific costs determined using a cost function defined by the user.
 10388 The idea of cost-weighted distance to characterize animal use of landscapes is widely
 10389 used in landscape ecology for modeling connectivity, movement and gene flow (Beier et al.,
 10390 2008). For reasons of computational tractability we consider a discrete landscape defined
 10391 by a raster of some prescribed resolution. The distance between any two points \mathbf{x} and
 10392 \mathbf{x}' can be represented by a sequence of line segments connecting neighboring pixels, say
 10393 $\mathbf{l}_1, \mathbf{l}_2, \dots, \mathbf{l}_m$. Then the cost-weighted distance between \mathbf{x} and \mathbf{x}' is

$$d(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{m-1} cost(\mathbf{l}_i, \mathbf{l}_{i+1}) \|\mathbf{l}_i - \mathbf{l}_{i+1}\| \quad (12.2.1)$$

10394 where $cost(\mathbf{l}_i, \mathbf{l}_{i+1})$ is the user-defined cost to move from pixel \mathbf{l}_i to neighboring pixel
 10395 \mathbf{l}_{i+1} in the sequence. Given the cost of each pixel, it is a simple matter to compute the
 10396 cost-weighted distance between any two pixels, along *any* path, simply by accumulating
 10397 the incremental costs weighted by distances. In the context of spatial capture-recapture
 10398 models (and, more generally, landscape connectivity) we are concerned with the *minimum*
 10399 cost-weighted distance, or the *least-cost path*, between any two points which we will denote
 10400 by d_{lcp} , which is the sequence $\mathcal{P} = (\mathbf{l}_1, \mathbf{l}_2, \dots, \mathbf{l}_m)$ that minimizes the objective function
 10401 defined by Eq. 12.2.1. That is,

$$d_{lcp}(\mathbf{x}, \mathbf{x}') = \min_{\mathcal{P}} \sum_{i=1}^{m-1} cost(\mathbf{l}_i, \mathbf{l}_{i+1}) \|\mathbf{l}_i - \mathbf{l}_{i+1}\| \quad (12.2.2)$$

10402 The least-cost path distance can be calculated in many geographic information systems
 10403 and other software packages, including the R package **gdistance** (van Etten, 2011) which
 10404 we use below.

10405 The key ecological aspect of least-cost path modeling is the development of models for
 10406 pixel-specific cost. In this paper we model cost as a function of one or more covariates
 10407 defined on every pixel of the according raster. For example, using a single covariate $z(\mathbf{x})$
 10408 we define the cost of moving from some pixel \mathbf{x} to neighboring pixel \mathbf{x}' as

$$\log(cost(\mathbf{x}, \mathbf{x}')) = \alpha_2 \left(\frac{z(\mathbf{x}) + z(\mathbf{x}')}{2} \right) \quad (12.2.3)$$

10409 Thus, if $\alpha_2 = 0$ then substituting $cost(\mathbf{x}, \mathbf{x}') = \exp(0) = 1$ into Eq. 12.2.2 will pro-
 10410 duce the ordinary Euclidean distance between points. Here we assume the covariate z is
 10411 positive-valued and constrain $\alpha_2 \geq 0$ so as to avoid negative costs. While not necessarily
 10412 problematic from a mathematical standpoint, negative costs are unrealistic biologically.

10413 The use of least-cost path models to model landscape connectivity has been around
 10414 for a long time. And, although α_2 is rarely known, conservation biologists design linkages
 10415 that require this resistance value as input (see Beier et al., 2008, and articles cited therein).
 10416 However, formal inference (e.g., estimation) of parameters is not often done. Instead, in
 10417 many existing applications of least-cost path analysis, the parameter α_2 is fixed by the
 10418 investigator, or based on expert opinion (Beier et al., 2008), although recently researchers
 10419 have begun to define costs based on resource selection functions, animal movement (Tracy,
 10420 2006; Fortin et al., 2005), or genetic distance data (e.g., Gerlach and Musolf (2000); Epps
 10421 et al. (2007); Schwartz et al. (2009)). We address the integration of resource selection
 10422 models based on telemetry data with SCR models in Chapt. 13.

10423 To formalize the use of cost-weighted distance in SCR models, we substitute Eq.
 10424 12.2.2 in the expression for encounter probability (Eq. ??) and maximize the resulting
 10425 likelihood which we address below. In doing so, we can directly estimate parameters of
 10426 the least-cost path model, evaluate how landscape covariate influence connectivity, and
 10427 test explicit hypotheses about these things using only individual level encounter history
 10428 data from capture-recapture studies.

10429 12.2.1 Example of Computing Cost-weighted distance

10430 XXXX Kimmy: Reconcile the raster example here with what is in the Ap-
 10431 pendix of the accepted Ecology paper. XXXXX

10432 As an example of the cost-weighted distance calculation consider the following land-
 10433 scape comprised of 16 pixels with unit spacing identified as follows, along with the pixel-
 10434 specific cost:

10435	pixel ID				Cost			
10436	1	5	9	13	100	1	1	1
10437	2	6	10	14	100	100	1	1
10438	3	7	11	15	100	100	100	1
10439	4	8	12	16	100	100	1	1

10440 This simple cost raster is shown in Fig. 12.2. We assume the scale is such that the
 10441 distance between neighboring pixels in any cardinal direction is 1 unit, and the distance
 10442 between neighbors on a diagonal is $\sqrt{2}$ units. We assigned low cost of 1 to “good habitat”
 10443 pixels (or pixels we think of as “highly connected” by virtue of being in good habitat)
 10444 and, conversely, we assign high cost (100) to “bad habitat”. So the shortest cost-weighted
 10445 distance between pixels 5 and 9 in this example is just 1 unit, the shortest cost-distance
 10446 between pixels 5 and 10 is $\sqrt{2}(1+1)/2 = 1.414214$ units, the shortest distance between
 10447 pixels 4 and 8 is 100 units, while the shortest cost-distance between 4 and 12 is 150.5. A
 10448 tough one is: what is the shortest distance between 7 and 16? An individual at pixel 7
 10449 can move diagonal (which has distance $\sqrt{2}$) and pay $\text{sqrt}(2) * (100 + 1) / 2 + 1 = 72.41778$.

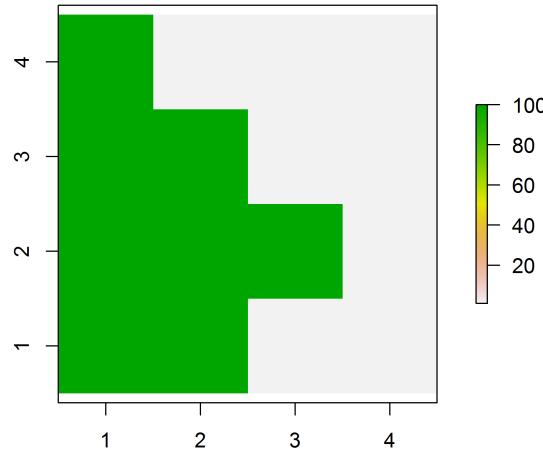


Figure 12.2. A 4×4 raster depicting a binary cost surface, with cost = 1 (white) or 100 (shaded) to represent ease of movement across a pixel.

10450 Once the cost raster is created, the least-cost path distances are computed with just a
 10451 couple **R** commands, and those can be inserted directly into the likelihood construction for
 10452 an ordinary spatial capture-recapture model. The **R** package **gdistance** calculates least-
 10453 cost path using Dijkstra's algorithm (Dijkstra, 1959) (from the **igraph** package (Csardi
 10454 and Nepusz, 2006)). Using **gdistance**, we define the incremental cost of moving from one
 10455 pixel to another as the distance-weighted *average* of the 2 pixel costs. We demonstrate
 10456 how to do this subsequently.

10457 **Kimmy: Make sure the structure of this example is copied directly from**
 10458 **the Ecology paper that was accepted. There are some things slightly different.**

10459 The **R** commands for computing the least-cost distance between all pairs of pixels are
 10460 as follows:

```
10461 r<-raster(nrows=4,ncols=4)
10462 projection(r)<- "+proj=utm +zone=12 +datum=WGS84"
10463 extent(r)<-c(.5,4.5,.5,4.5)
10464 costs1<- c(100,100,100,100,1,100,100,100,1,1,100,1,1,1,1,1)
10465 values(r)<-matrix(costs1,4,4,byrow=FALSE)
10466 par(mfrow=c(1,1))
10467 plot(r)
```

10468 Then we use the functions **transition**, **geoCorrection XXX Kimmy do we need**
 10469 **this function? XXXX** (which is only necessary if the data are not projected or if
 10470 cells are considered to have more than 4 neighbors) and **costDistance** to compute the
 10471 distance matrix. The transition function computes the cost of making a transition be-
 10472 tween any two pixels, and it operates on the inverse-scale ("conductance") and so the
 10473 **transitionFunction** argument is given as $1/\text{mean}(x)$. To compute the cost distance we
 10474 prescribe a set of points, or we can compute it between two sets of points (which is handy
 10475 when one of the sets is of trap locations, and the other is of individual activity centers).
 10476 To compute the distances for pixels in a raster, we use the center points of each raster.
 10477 The **R** commands altogether are as follows:

```
10478 tr1<-transition(r,transitionFunction=function(x) 1/mean(x),directions=8)
10479 tr1CorrC<-geoCorrection(tr1,type="c",multpl=FALSE,scl=FALSE)
10480 pts<-cbind( sort(rep(1:4,4)),rep(4:1,4))
10481 costs1<-costDistance(tr1CorrC,pts)
10482 outD<-as.matrix(costs1)
```

10483 Now we can look at the result and see if it makes sense to us. Here we produce the
 10484 first 5 columns of this distance matrix to illustrate a couple of examples of calculating
 10485 the minimum cost-weighted distance between points: **XXXX Kimmy: All of these**
 10486 **numbers are probably different in the paper's appendix and we need to change**
 10487 **them here XXXXX**

```
10488 > outD[1:5,1:5]
10489      1       2       3       4       5
10490 1  0.0000 100.00000 200.0000 205.2426 50.50000
10491 2 100.0000   0.00000 100.0000 200.0000 71.41778
10492 3 200.0000 100.00000   0.0000 100.0000 171.41778
```

10493 4 205.2426 200.00000 100.0000 0.0000 154.74264
 10494 5 50.5000 71.41778 171.4178 154.7426 0.00000

10495 An interesting case is that between point 1 and 4. Note that simply taking the shortest
 10496 Euclidean distance, weighted by cost, produces a cost-weighted distance of 100×1 to
 10497 move from pixel 1 to pixel 2, and similarly from 2 to 3 and 3 to 4, producing a total cost-
 10498 weighted distance of 300. However, the actual *least-cost path* has cost-weighted distance
 10499 205.2426 which has an individual moving from pixel 1 to 5, then 5 to 10, 10 to 15, 15 to
 10500 12, 12 to 8 and 8 to 4, adding up to a cost weighted distance of 205.2426.

12.3 SIMULATING SCR DATA USING ECOLOGICAL DISTANCE

10501 Royle et al. (2012a) simulated data based on two hypothetical landscapes typical of how
 10502 cost-weighted distance models might be used in real capture-recapture problems. They
 10503 defined a 20×20 pixel covariate raster with extent $= [0.5, 4.5] \times [0.5, 4.5]$ which we imagine
 10504 to be a coarse landscape covariate, with pixels having some arbitrary scaling. For example,
 10505 think of each pixel as representing, say, a 1×1 km grid cell in which case the raster defines
 10506 a landscape of 20×20 km. We suppose that 16 camera traps are established at the integer
 10507 coordinates $(1, 1), (1, 2), \dots, (4, 4)$. We could think of this as a landscape within which
 10508 we're studying a population of ocelots, lynx or some other cat.

10509 For our analyses here, we characterize cost by one covariate, and we consider two
 10510 specific cases. First is an increasing trend from the NW to the SE ("systematic covariate"),
 10511 where $z(\mathbf{x})$ is defined as $z(\mathbf{x}) = r(\mathbf{x}) + c(\mathbf{x})$ and $r(\mathbf{x})$ and $c(\mathbf{x})$ are just the row and column,
 10512 respectively, of the raster. This might mimic something related to distance from an urban
 10513 area or a gradient in habitat quality due to land use, or environmental conditions such
 10514 as temperature or precipitation gradients. In the second case we make up a covariate by
 10515 generating a field of spatially correlated noise to emulate a typical patchy habitat covariate
 10516 ("patchy covariate") such as tree or understory density. The two covariates are shown in
 10517 Fig. 12.3, along with a sample realization of $N = 100$ individuals (left panel only). For
 10518 both covariates we use a cost function in which transitions from pixel \mathbf{x} to \mathbf{x}' is given by:

$$\log(\text{cost}(\mathbf{x}, \mathbf{x}')) = \alpha_2 \frac{z(\mathbf{x}) + z(\mathbf{x}')}{2}$$

10519 where $\alpha_2 = 1$ for simulating the observed data. Remember that with $\alpha_2 = 0$ the model
 10520 reduces to one in which the cost of moving across each pixel is constant, and therefore
 10521 Euclidean distance is operative.

10522 When distance is defined by the cost-weighted distance metric given by Eq. 12.2.2
 10523 then individual space-usage varies spatially in response to the landscape covariate(s) used
 10524 in the distance metric. As a consequence, home ranges contours are no longer circular, as
 10525 in SCR models based on Euclidean distance. For example, using one of the covariates we
 10526 use in our simulation study below (Fig. 12.3, right panel) with a Gaussian pdf detection
 10527 function but having distance metric defined by Eq. 12.2.2, produces home ranges such as
 10528 those shown in Fig. 12.4.

10529 To simulate data, we have to load the `scrbook` package and call the function `make.EDcovariates`
 10530 to generate our raster covariates (see the help file for how that is done). We process the co-
 10531 variate into a least-cost path distance matrix, and then simulate observed encounter data
 10532 using standard methods which we have used many times previously in this book. The

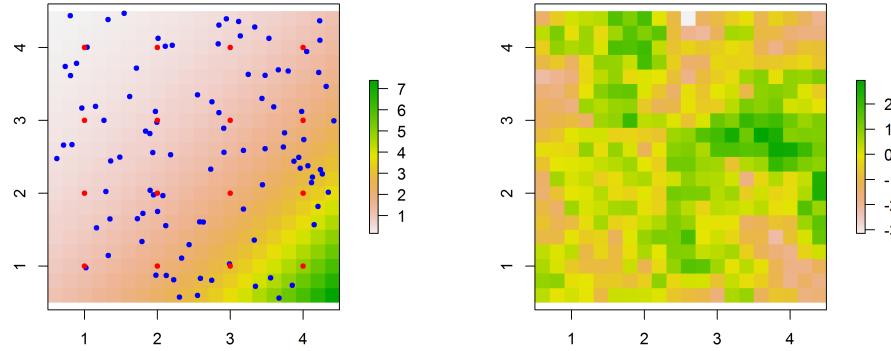


Figure 12.3. Two covariates (defined on a 20×20 grid) used in simulations. Left panel shows a covariate with systematic structure meant to mimic distance from some feature, and the right panel shows a “patchy” covariate. A hypothetical realization of $N = 100$ activity centers (blue dots) is superimposed on the left figure, along with 16 trap locations.

10533 complete set of R commands is: XXXX Kimmy: Test this code! XXXXX XXX
 10534 Kimmy: Do we need the geocorrection() statement in this code? XXXXX

```

10535 #### Grab a covariate
10536 library("scrbook")
10537 out<-make.EDcovariates()
10538 covariate<-out$covariate.patchy
10539 set.seed(2013)

10540 #### prescribe some settings
10541 N<-200
10542 alpha0<- -2
10543 sigma<- .5
10544 alpha1<- 1/(2*sigma*sigma)
10545 alpha2<-1
10546 K<- 5
10547 S<-cbind(runif(N,.5,4.5),runif(N,.5,4.5))

10549 # make up some trap locations
10550 xg<-seq(1,4,1); yg<-4:1
10551 traplocs<-cbind( sort(rep(xg,4)),rep(yg,4))
10552 points(traplocs,pch=20,col="red")
10553 ntraps<-nrow(traplocs)
10554

```

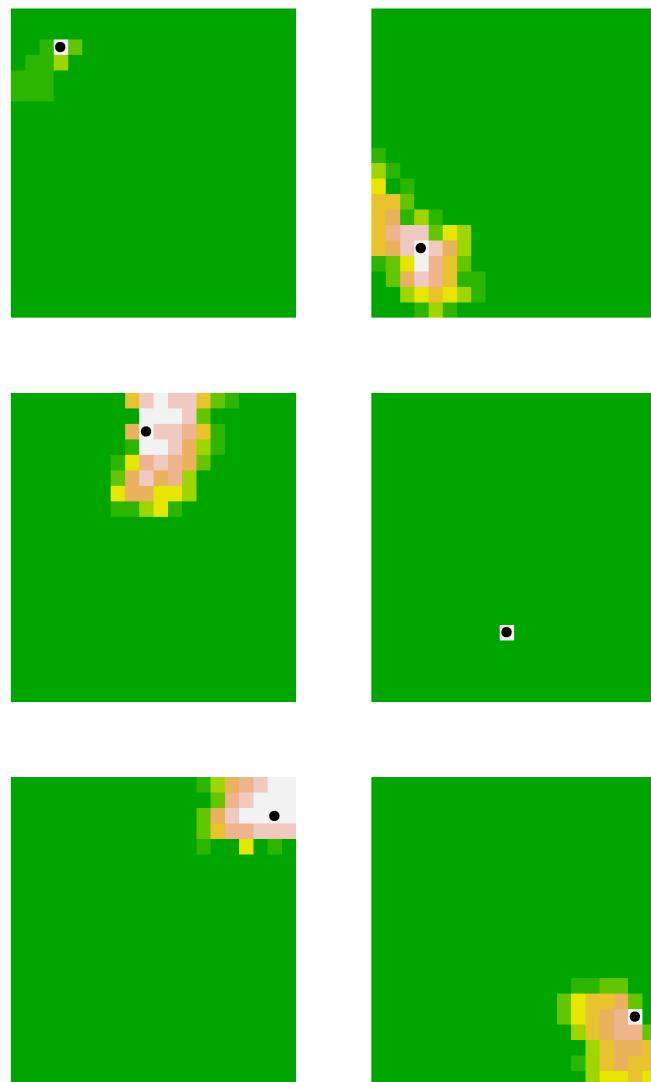


Figure 12.4. Typical home ranges for 6 individuals based on the cost surface shown in the right panel of Fig. 12.3 with $\alpha_2 = 1$. The black dot indicates the home range center and the pixels around each home range center are shaded according to the probability of encounter, if a trap were located in that pixel.

```

10555
10556 ### make a raster and fill it up with the "cost"
10557 r<-raster(nrows=20,ncols=20)
10558 projection(r)<- "+proj=utm +zone=12 +datum=WGS84"
10559 extent(r)<-c(.5,4.5,.5,4.5)
10560 cost<- exp(alpha2*covariate)
10561
10562 ### compute least-cost path distance
10563 tr1<-transition(cost,transitionFunction=function(x) 1/mean(x),directions=8)
10564 tr1CorrC<-geoCorrection(tr1,type="c",multpl=FALSE,scl=FALSE)
10565 D<-costDistance(tr1CorrC,S,traplocs)
10566 probcap<-plogis(alpha0)*exp(-alpha1*D*D)
10567
10568 # now generate the encounters of every individual in every trap
10569 # discard uncaptured individuals
10570 Y<-matrix(NA,nrow=N,ncol=ntraps)
10571 for(i in 1:nrow(Y)){
10572   Y[i,]<-rbinom(ntraps,K,probcap[i,])
10573 }
10574 Y<-Y[apply(Y,1,sum)>0,]

```

12.4 LIKELIHOOD ANALYSIS OF ECOLOGICAL DISTANCE MODELS

10575 Throughout much of this book we rely on Bayesian analysis by MCMC mostly using
 10576 **BUGS**, but sometimes (as in Chapt. 17) developing our own implementations. However,
 10577 occasionally we prefer to use likelihood estimation, such as when we can compare a set
 10578 of models directly by likelihood either to do a direct hypothesis test of a parameter, or
 10579 to tabulate a bunch of AIC values. For the class of models that use least-cost path, we
 10580 also prefer likelihood methods not because they have any conceptual or methodological
 10581 benefit, but simply because they are more computationally efficient to implement (Royle
 10582 et al., 2012a).

10583 There are no technical considerations in adapting our formulation of maximum likeli-
 10584 hood estimation (Borchers and Efford, 2008) from Chapt. 6 for the class of models based
 10585 on least-cost path (see the appendix in Royle et al. (2012a) for complete details). Likeli-
 10586 hood analysis is really just a straightforward adaptation in which we replace the Euclidean
 10587 distance with least-cost path. Consider the Bernoulli model in which the individual- and
 10588 trap-specific observations have a binomial distribution conditional on the latent variable
 10589 \mathbf{s}_i :

$$y_{ij} | \mathbf{s}_i \sim \text{Binomial}(K, p_{\boldsymbol{\alpha}}(d_{lcp}(\mathbf{x}_j, \mathbf{s}_i; \boldsymbol{\alpha}_2); \boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1)) \quad (12.4.1)$$

10590 where we have indicated the dependence of p on the parameters $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2)$, and
 10591 also d_{lcp} which itself depends on $\boldsymbol{\alpha}_2$, and the latent variable \mathbf{s} . We note that the only
 10592 difference between likelihood analysis of this model and the standard Bernoulli model, is
 10593 the use of d_{lcp} here.

10594 XXXXX Andy here XXXXXXXXXXXXXXXXXXXXXXX

10595 For the random effect we have $\mathbf{s}_i \sim \text{Uniform}(\mathcal{S})$, we can easily compute the integrated
 10596 likelihood.

10597 We have an R script in the `scrbook` package? adapted from Royle et al.
 10598 XXXXXXXX

10599 We provide an **R** function to evaluate the likelihood, and optimize it using the **R**
 10600 function `nls`. The likelihood is given in the `scrbook` package as the function `intlik3ed`.
 10601 The help file provides an example of its usage and for simulating data. To use this
 10602 function the cost covariate $z(\mathbf{x})$ has to be of class `RasterLayer` which requires packages
 10603 `sp` and `raster` to manipulate.

10604 12.4.1 Example of SCR with Least-Cost Path

10605 Now we use the **R** function `nls` along with our `intlik3ed` function to obtain the MLEs
 10606 of the model parameters for the data simulated in sec. 12.3. We'll do that for both the
 10607 standard Euclidean distance and then for the ecological distance based on the "patchy"
 10608 covariate using the following commands:

```
10609 frog1<-nls(intlik3ed,c(alpha0,alpha1,3)),hessian=TRUE,y=Y,K=K,X=traplocs,  

10610           distmet="euclid",covariate=covariate,alpha2=1)  

10611  

10612 frog2<-nls(intlik3ed,c(alpha0,alpha1,3,-.3),hessian=TRUE,y=Y,K=K,X=traplocs,  

10613           distmet="ecol",covariate=covariate,alpha2=NA)
```

10614 The abbreviated output for the two model fits is shown in Table XXX. XXXXXX
 10615 **KIMMY: Put this in a table like the one later in this chapter XXXXXX**

Distance	-LL	alpha0	alpha1	log(n0)	alpha2
true value		-2	2	[??]	1
euclidean	133.4951	-1.885005	1.247305	3.549064	--
LCP (truth)	70.11916	-1.78029983	2.47083431	4.45867628	0.04560194

10620 The model based on least-cost path (the data generating model) appears to be much
 10621 preferred in terms of negative log-likelihood. The output parameter order is $(\alpha_0, \alpha_1, \log(n_0), \text{and} \log(\alpha_2))$
 10622 (remember, we want to keep α_2 positive, so its logarithm is estimated). The data gener-
 10623 ating parameter values were $\alpha_0 = -2$, $\alpha_1 = 2$ and $\log(\alpha_2) = 0$. The simulated sampling
 10624 produced a sample of 96 individuals and so $n_0 = 104$, so $\log(n_0) = 4.64$. We see that the
 10625 MLEs of the least-cost path model are pretty close whereas they are not so close under
 10626 the misspecified model based on Euclidean distance.

12.5 BAYESIAN ANALYSIS

10627 While implementation of these ecological distance SCR models is reasonably straightfor-
 10628 ward, we do not believe the model can be fitted in the **BUGS** engines because least-cost
 10629 path distance cannot be computed. It would be possible to fit the models in **BUGS** if
 10630 the parameter α_2 was fixed. In that case, one could compute the distance matrix ahead of
 10631 time and reference the required elements for a given `s`. Alternatively, it would be possible
 10632 to write a custom MCMC routine using the methods we present in Chapt. 17, although
 10633 we have not yet developed our own MCMC implementation of SCR models with ecological
 10634 distance metrics.

12.6 SIMULATION EVALUATION OF THE MLE

10635 Royle et al. (2012a) carried-out a limited simulation study to evaluate the general statisti-
 10636 cal performance of the density estimator under this new model, the effect of mis-specifying
 10637 the model with a normal Euclidean distance metric and evaluate the general bias and pre-
 10638 cision properties of the MLE. We recapitulate their results here. For population sizes of
 10639 100 and 200, individuals with activity centers randomly distributed on the 20×20 land-
 10640 scape, they subjected individuals to encounter by 16 traps arranged in a 4×4 grid using
 10641 a Gaussian encounter model with least-cost path distance metric:

$$\log(p_{ij}) = \alpha_0 + \alpha_1 d_{lcp}(\mathbf{x}_j, \mathbf{s}_i; \alpha_2)^2$$

10642 where $\alpha_0 = -2$ and $\alpha_1 = 2$, the latter value corresponding to $\sigma = 0.5$ of a stationary
 10643 bivariate normal home range model. Different numbers of replicate samples were consid-
 10644 ered, $K = 3, 5, 10$ (e.g., nights in a camera trapping study), in order to produce varying
 10645 sample sizes. For each of the “systematic” and “patchy” landscapes defined previously,
 10646 200 data sets were simulated and, for each of those, two different models were fitted:
 10647 the misspecified Euclidean distance model; and (ii) the true data-generating model but
 10648 estimating the relative cost parameter by maximum likelihood.

10649 12.6.1 Simulation Results

10650 For both landscapes and all simulation conditions (levels of K and N) the average sample
 10651 sizes of individuals captured are given in Tab. 12.1. The simulation results for estimating

Table 12.1. Expected sample sizes of captured individuals under each configuration of N (population size for the prescribed state-space) and K (number of replicate samples).

	Systematic		Patchy	
	$N=100$	$N=200$	$N=100$	$N=200$
$K=3$	38.69	78.17	37.30	74.93
$K=5$	51.10	103.18	51.89	103.71
$K=10$	65.81	132.39	69.44	138.76

10651
 10652 N for the prescribed state-space are presented in Tab. 12.6.1. For the “patchy” landscape
 10653 we see extreme bias in estimates of N when the Euclidean distance is used. There is
 10654 moderate small sample bias of 3-5% in the MLE of N using the least-cost distance which
 10655 becomes negligible as K increases. For $N = 200$ the bias is on the order of 2% for the
 10656 lowest sample size case ($K = 3$) but negligible otherwise. Interestingly, for the landscape
 10657 exhibiting systematic structure, there is a persistent bias in the MLE of N of 1-3% even
 10658 for the highest level of K . As noted by Royle et al. (2012a), this is due to the fact that the
 10659 state-space is small relative to the extent of the trapping grid and sensitivity to a state-
 10660 space that is too small is expected because the support of the integrand is truncated.
 10661 In the particular case of the systematic landscape, we find that, in the NW corner of
 10662 the raster where cost of movement is low, individuals use large areas of space, and the
 10663 fitted model is under-stating the apparent heterogeneity in encounter probability for the
 10664 prescribed raster. Royle et al. (2012a) found that the issue is resolved when the traps are
 10665 moved away from the boundary (results shown in Tab. 12.6.1).

10666 The performance of estimating the cost parameter α_2 mirrors the results for estimating
 10667 N for the prescribed state space. In the patchy landscape where we don't expect a
 10668 systematic gradient in space usage around the edge of the state-space, we see (Table 12.3)
 10669 that α_2 is estimated with diminishing bias as the sample size increases, but with persistent
 10670 bias due to truncation of the likelihood under the systematic landscape which, as with the
 10671 MLE of N , is resolved by moving the traps away from the edge of the raster. Equivalently,
 10672 in practice, this could be resolved by expanding the raster away from the trap locations
 10673 so that all regions used by animals exposed to capture are included in the state-space.

Table 12.2. Simulation results for estimating population size N for a prescribed state-space with $N = 100$ or $N = 200$ and various levels of replication (K) chosen to affect the observed sample size of individuals (Tab. 12.1). For each simulated data set, the SCR model was fitted by maximum likelihood with standard Euclidean distance ("euclid"), or least-cost path ("lcp"), which was the true data-generating model. The summary statistics of the sampling distribution reported are the mean, standard deviation ("SD") and quantiles (0.025, 0.50, 0.975).

Systematic trend raster:										
	N=100					N=200				
	mean	SD	0.025	0.50	0.975	mean	SD	0.025	0.50	0.975
K=3										
euclid	63.65	12.62	44.77	61.17	90.98	126.68	17.05	98.93	124.49	168.26
lcp	101.93	21.68	67.95	101.56	156.21	201.58	28.14	154.96	200.15	263.20
K=5										
euclid	64.60	7.11	51.52	63.86	77.33	130.02	10.25	113.48	128.96	151.32
lcp	98.94	12.97	74.68	99.00	123.88	198.80	19.60	166.87	197.97	239.46
K=10										
euclid	69.24	4.83	59.37	69.47	79.18	139.83	7.62	125.65	139.65	154.82
lcp	97.53	8.18	82.02	97.62	113.16	195.19	13.28	171.63	194.58	217.96
Patchy "random" raster:										
	N=100					N=200				
	mean	SD	0.025	0.50	0.975	mean	SD	0.025	0.50	0.975
K=3										
euclid	78.68	18.12	49.40	76.34	125.47	154.34	33.74	107.00	146.34	221.43
lcp	110.96	28.65	69.55	106.98	181.84	208.77	49.29	141.68	197.89	325.77
K=5										
euclid	77.85	11.55	59.17	77.44	101.14	153.39	15.57	129.31	149.54	185.38
lcp	104.44	15.79	78.38	101.47	139.55	200.91	20.78	164.42	200.47	246.46
K=10										
euclid	78.01	5.26	68.00	77.96	87.81	156.27	8.51	142.17	156.05	174.55
lcp	100.42	7.56	86.72	100.34	115.47	198.45	11.44	180.06	198.04	219.52

Table 12.3. Mean of sampling distribution of the cost function parameter α_2 for the different simulation conditions.

	Patchy		Systematic	
	N=100	N=200	N=100	N=200
$K = 3$	1.05	1.03	1.17	1.14
$K = 5$	1.02	1.01	1.12	1.12
$K = 10$	1.01	1.00	1.10	1.08

Table 12.4. Simulation results for estimating population size N for a prescribed state-space with $N = 100$ or $N = 200$ and various levels of replication (K) chosen to affect the observed sample size of individuals. These results correspond to those of the systematic landscape in Table XXXXXX except with the traps moved 0.5 units in from the boundary of the landscape. Each grouping of 2 rows (for a given value of K) summarizes the performance of \hat{N} under models based on Euclidean distance ("euclid") and a model based on least-cost path, which was the true data-generating model. The summary statistics of the sampling distribution reported are the mean, standard deviation ("SD") and quantiles (0.025, 0.50, 0.975).

	N=100					N=200				
	mean	SD	0.025	0.50	0.975	mean	SD	0.025	0.50	0.975
K=3										
euclid	84.48	20.42	51.16	81.51	140.62	163.70	24.55	126.64	157.67	223.63
lcp	105.90	26.19	65.95	103.40	182.30	201.34	29.54	161.88	192.36	268.98
K=5										
euclid	81.21	11.33	61.35	79.20	98.86	163.27	13.06	140.21	162.97	185.94
lcp	100.84	13.15	79.96	99.51	119.08	200.25	16.53	168.88	199.29	227.39
K=10										
euclid	80.10	7.81	66.45	79.14	93.33	158.40	9.25	142.74	157.86	173.18
lcp	100.10	9.88	82.31	100.91	116.27	197.52	13.03	169.49	200.68	217.82

12.7 DISTANCE IN AN IRREGULAR PATCH

10674 We provide another illustration of how to employ ecological distance calculations in SCR
 10675 models. This example is meant to mimic a situation where we have something like a hard
 10676 habitat boundary such as a habitat corridor or park unit or some other block of relatively
 10677 homogeneous good-quality habitat for some species. This particular system (shown in
 10678 Fig. 12.5) could be habitat surrounded by a suburban wasteland of McDonalds and Wal-
 10679 Marts, much less hospitable habitat for most species. For our purposes, we suppose that
 10680 individuals live within the buffered “f-shaped” region, although we could also imagine the
 10681 negative of the situation in which individuals live outside of the region, so that the polygon
 10682 represents a barrier (a lake) or bad habitat (an urban area) or similar. We describe the
 10683 steps for creating this landscape shortly, so that you can use a similar process to generate
 10684 more relevant landscapes for your own problems.

10685 In this case we’re not going to estimate any parameters of the cost function (though
 10686 you could adapt the analyses of the previous sections to do that) but instead we’re going
 10687 to use ecological distance ideas only to constrain movement within (or to avoid) landscape
 10688 features. Note that, normally, distance “as the crow flies” would not be suitable for
 10689 irregular habitat patches such as that shown in Fig. 12.5.

10690 12.7.1 Basic Geographic Analysis in R

10691 In practical applications our landscape will contain one or more polygons which de-
 10692 lineate good or bad habitat or other important characteristics of the landscape. These
 10693 might exist as GIS shapefiles or merely as a text file with coordinates defining polygon
 10694 boundaries. To work with polygons in the context of SCR models we need to create a
 10695 raster, overlay the polygon and assign values to each pixel depending on whether pixels
 10696 are in the polygon or not, or how far they are from polygon boundaries. These opera-
 10697 tions are relatively easy to do within a GIS system but we need to be able to do them
 10698 in **R** in order to compute the least-cost paths needed in the likelihood evaluation. Some
 10699 additional geographic analyses have been discussed in secs. ?? and 17.5 where we talked
 10700 about reading in the shapefile and doing SCR calculations on that.

10701 Often we will have GIS shapefiles that define polygons but, here, we create a set of
 10702 polygons by buffering and joining some line segments. In the **R** library **scrbook**, we
 10703 provide a function **make.seg** which allows you to make such line segments given a specific
 10704 trap region. To involve **make.seg** we first create a plot region and then call **make.seg**
 10705 which has a single argument being the number of points used to define the line segment.
 10706 The user will click on the visual display until the required number of points has been
 10707 obtained by **make.sec**. In the following set of commands we generate two line segments,
 10708 11 consisting of 9 points and 12 consisting of 5 points, and these reside in a geographic
 10709 region enclosed by $[0, 10] \times [0, 10]$:

```
10710 library("scrbook")
10711 library("sp")
10712 plot(NULL,xlim=c(0,10),ylim=c(0,10))
10713 l1<-make.seg(9)
10714 plot(l1)
10715 l2<-make.seg(5)
```

```
10716 plot(l1)
10717 lines(l2)
```

10718 We used this function as above to create a habitat corridor composed of line segments
 10719 of class **SpatialLines** from the **R** package **sp**. The corridor can be loaded from **scrbook**
 10720 by typing the command `data("fakecorridor")`. This data list has 2 line files in it (l1
 10721 and l2) and a trap locations file (traps). We use some functions from the **R** packages
 10722 **sp** and **rgeos** to join and buffer (by 0.5 units) the two segments. The commands are as
 10723 follows and the result is shown in Fig. 12.5.

10724 **XXXX Kimmy: check all of this code! XXXXX**

```
10725 data("fakecorridor")
10726 library("sp")
10727 library("rgeos")
10728
10729 buffer<- 0.5
10730 par(mfrow=c(1,1))
10731 aa<-gUnion(l1,l2)
10732 plot(gBuffer(aa,width=buffer),xlim=c(0,10),ylim=c(0,10))
10733 pg<-gBuffer(aa,width=buffer)
10734 pg.coords<- pg@polygons[[1]]@Polygons[[1]]@coords
10735
10736 xg<-seq(0,10,,40)
10737 yg<-seq(10,0,,40)
10738
10739 delta<-mean(diff(xg))
10740 pts<- cbind(sort(rep(xg,40)),rep(yg,40))
10741 points(pts,pch=20,cex=.5)
10742
10743 in_pts<-point.in.polygon(pts[,1],pts[,2],pg.coords[,1],pg.coords[,2])
10744 points(pts[in_pts==1,],pch=20,col="red")
```

10745 In this example, we're not going to estimate parameters of the cost function. Instead, d
 10746 the point is compute ordinary Euclidean distance but restricted by the boundaries of the
 10747 corridor (or patch geometry in general) and thus not distance "as the crow flies." To do
 10748 this, we imagine that animals will tend to severely avoid leaving the buffered habitat zone.
 10749 Therefore, we assign `cost = 1` if a pixel is within the buffer, and `cost = 10000` if a pixel
 10750 is outside of a buffer. Therefore the cost to move to a neighboring pixel outside of the
 10751 buffered area is 5000.5 compared to the cost of 1 to move to a neighboring pixel inside the
 10752 buffer. With this cost specification, we can compute the least-cost path distance matrix
 10753 one time and modify our likelihood code to accept the distance matrix as input. We give
 10754 that likelihood in the library **scrbook** as the function `intlik3edv2`. We note also that this
 10755 function accepts a habitat mask in the form of a vector of 0's and 1's **XXXX Kimmy:**
 10756 **check that this is true XXXXX** that define any potential state-space restrictions.
 10757 i.e., 1 if the pixel is an element of the state-space and 0 if it is not, and so additional
 10758 modifications to the geometry of the region could be made. However, in the analysis of
 10759 this simulated data set, we define the state-space to be the buffered corridor system. Here

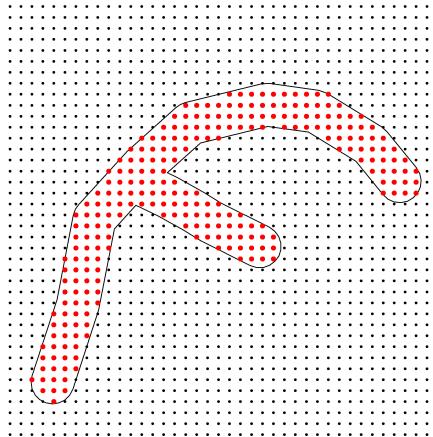


Figure 12.5. A made-up wildlife corridor or reserve. The boundary outlines a polygon of suitable habitat surrounded by suburban development.

we simulate a population of $N = 200$ individuals in the corridor system and so we restrict our state-space accordingly for purposes of fitting the model. However we encourage you to refit the model without the state-space restriction (for fitting the model only) and then compare the results. The code for doing all of this is in the help file for `intlik3edv2`, which contains the likelihood function and sample **R** script (`?intlik3edv2`).

```

10765  ### Define the cost structure
10766  cost<-rep(NA,nrow(pts))
10767  cost[in pts==1]<-1      # low cost to move among pixels but not 0
10768  cost[in pts!=1]<-10000  # high cost
10769
10770  ### Stuff costs into a raster
10771  library("raster")
10772  r<-raster(nrows=40,ncols=40)
10773  projection(r)<- "+proj=utm +zone=12 +datum=WGS84"
10774  extent(r)<-c(0-delta/2,10+delta/2,0-delta/2,10+delta/2)
10775  values(r)<-matrix(cost,40,40,byrow=FALSE)
10776
10777  # check what it looks like

```

```

10778 plot(r)
10779 points(pts,pch=20,cex=.4)
10780
10781 # compute ecological distances:
10782 library("gdistance")
10783 tr1<-transition(r,transitionFunction=function(x) 1/mean(x),directions=8)
10784 tr1CorrC<-geoCorrection(tr1,type="c",multpl=FALSE,scl=FALSE)
10785 costs1<-costDistance(tr1CorrC,pts)
10786 outD<-as.matrix(costs1)

10787 In the next block of code we simulate some data and then fit a model to the simulated
10788 data. KIMMY: I forget where “traps” came from. Is it in the “fakecorridor”
10789 object? Test this code XXXXXXXX XXX ANDY: change beta to alpha1 XXXX

10790 library('scrbook')
10791 traplocs<-traps$loc
10792 trap.id<-traps$locid
10793 ntraps<-nrow(traplocs)
10794
10795 set.seed(2013)
10796 N<-200
10797 S.possible<- (1:nrow(pts))[in pts==1]
10798 S.id<-sample(S.possible,N,replace=TRUE)
10799 S<- pts[S.id,]

10800 D<- outD[S.id,trap.id]
10801 eD<- e2dist(S,traplocs)
10802 Dtraps<-outD[trap.id,]

10803
10804 alpha0<- -1.5
10805 sigma<- 1.5
10806 beta<- 1/(2*sigma*sigma)
10807 K<-10
10808
10809 probcap<-plogis(alpha0)*exp(-beta*D*D)
10810 Y<-matrix(NA,nrow=N,ncol=ntraps)
10811 for(i in 1:nrow(Y)){
10812   Y[i,]<-rbinom(ntraps,K,probcap[i,])
10813 }
10814 Y<-Y[apply(Y,1,sum)>0,]

10815
10816 frog1<-nlm(intlik3edv2,c(-2.5,2,log(4)),hessian=TRUE,y=Y,K=K,X=traplocs,
10817   S=pts,D=Dtraps,inpoly=in.pts)
10818 frog2<-nlm(intlik3edv2,c(-2.5,2,log(4)),hessian=TRUE,y=Y,K=K,X=traplocs,
10819   S=pts,D=Deuclid,inpoly=in.pts)

```

10820 These two models fit, with the correctly specified ecological distance, constrained by
10821 the patch boundaries, and that with the ordinary (misspecified) Euclidean distance are

Table 12.5. Fitting results XXXXXXXXXXXXXXXX

Distance	neg. LL	alpha0	alpha1	log(n0)
constrained	-21.8921	-1.3380122	0.3321878	4.3530026
Euclidean	-21.1280	-1.3071132	0.3821317	4.2116319

summarized in Table 12.5. We find little difference between the two models. In particular, 150 individuals were captured and so truth is $\log(n_0) = 3.9$. The correct model produces only a slightly more accurate estimate, and it is favored by only .7 negative log-likelihood units. Therefore, for this single instance, the results are not too different. This is primarily because the distance between individuals, and traps that they are likely to be captured in, is well-approximated by the Euclidean distance.

12.8 SUMMARY AND OUTLOOK

Almost all published applications of SCR models to date have been based on models for the encounter probability that are functions of the Euclidean distance between individual activity centers and traps. The obvious limitations of such models are that Euclidean distance is unaffected by landscape or habitat structure and implies stationary, isotropic and symmetrical home ranges. These are standard criticisms of the basic SCR model which we have seen many times in referee reports, or heard in discussions with colleagues. However, this should not be seen as criticism that is inherent to the basic conceptual formulation of SCR models, because, we have shown here that one can modify the Euclidean distance metric to accommodate more realistic formulations of distance that allow for inference to be made about landscape connectivity, and model “distance” as a function of local habitat characteristics. As such, effective distance between individual home range centers and traps varies depending on the local landscape.

How animals use space and therefore how distance to a trap is perceived by individuals is not something that can ever be known. We can only ever conjure up models to describe this phenomenon and fit those models to limited data on a sample of individuals during a limited amount of time. Here we have shown that there is hope to estimate parameters, from capture-recapture data, that describe how animals use space and thereby allow for irregular home range geometry that is influenced by landscape structure.

The simulation study of Royle et al. (2012a) demonstrated (see Table XXXX) that the MLE of model parameters is approximately unbiased in moderate sample sizes. Moreover, the effect of ignoring ecological distance and using normal Euclidean distance in the model for encounter probability, has the logical effect of causing negative bias in estimates of N . This is expected because the effect is similar to failing to model heterogeneity, i.e., if we mis-specify “model M_h ” (Otis et al., 1978) with “model M_0 ” (Otis et al., 1978) then we will expect to under-estimate N . So the effect of mis-specifying the ecological distance metric with a standard homogeneous Euclidean distance has the same effect. As a practical matter, it stands to reason that many previous applications of SCR models based on homogeneous distance metrics have under-stated density of the focal population.

In our view, this bias is not really the most important reason to consider models of ecological distance. Rather, inference about the structure of ecological distance is fundamental to many problems in applied and theoretical ecology related to modeling

10860 landscape connectivity, corridor and reserve design, population viability analysis, gene
10861 flow, and other phenomena. Models based on least-cost path distance allow investigators
10862 to evaluate landscape factors that influence movement of individuals over the landscape
10863 from non-invasively collected capture-recapture data. Therefore SCR models based on
10864 ecological distance metrics might aid in understanding aspects of space usage and move-
10865 ment in animal populations and, ultimately, in addressing conservation-related problems
10866 such as corridor design.

10867
10868

13

10869
10870
10871

INTEGRATING RESOURCE SELECTION WITH SPATIAL CAPTURE-RECAPTURE MODELS

10872 Up to this point we have developed many variations of SCR models to describe the obser-
10873 vation process. These included models of the relationship between encounter probability
10874 and distance, and different types of covariates such as behavioral responses that can affect
10875 detection probability. Although these different observation models are immensely useful,
10876 they are rather basic in the sense that they imply simplistic models of how individuals use
10877 space (section 5.4) and how individuals are distributed in space. In the following several
10878 chapters we generalize some of the core SCR assumptions to accommodate more realistic
10879 notions of how animals use space.

10880 In Sec. 5.4 we briefly discussed the notion of how SCR encounter probability models
10881 relate to models of space usage. When we use symmetric and stationary encounter prob-
10882 ability models, SCR models imply that space usage is a decreasing function of distance
10883 from an individuals home range center. In this chapter, we extend SCR models to in-
10884 corporate models of resource selection (RS), such as when one or more explicit landscape
10885 covariates are available which the investigator believes might affect how individual ani-
10886 mals use space within their home range (this is what (Johnson, 1980) called *third-order*
10887 selection). Our treatment follows Royle et al. (2012b) who integrated a standard family
10888 of resource selection models based on auxilliary telemetry data into the capture-recapture
10889 model for encounter probability. The approach is consistent with the manner in which
10890 classical “resource selection function” (RSF) models (Manly et al., 2002) or utilization
10891 distributions (Worton, 1989; Fieberg and Kochanny, 2005; Fieberg, 2007) are estimated
10892 from animal telemetry data. Royle et al. (2012b) argued that SCR models and resource
10893 selection models estimated from telemetry are based on the same basic underlying model
10894 of space usage. The important distinction between SCR and RSF studies are that, in
10895 SCR studies, encounter of individuals is imperfect (i.e., “ $p < 1$ ”) whereas, with RSF data
10896 obtained by telemetry, encounter is perfect (or, rather, detection is not a *stochastic* out-

10897 come). We can think of the two as being exactly equivalent either if we have a dense
 10898 array of trapping devices, or if our telemetry apparatus is imperfect such as only samples
 10899 a small area of space (this would be consistent with telemetry stations for sampling fish
 10900 which only measure passage).

10901 Telemetry studies are extremely common in animal ecology for studying movement and
 10902 resource selection, and SCR studies frequently obtain such data on a subset of individuals.
 10903 Thus, formal integration of capture-recapture with telemetry data for the purposes of
 10904 modeling resource selection has a number of immediate benefits. For one, telemetry data
 10905 provide direct information about σ (Sollmann et al., 2012b, in revision). As a result, this
 10906 leads to improved estimates of model parameters, and also has design consequences (see
 10907 Sec. 10.4). In addition, active resource selection by animals induces a type of heterogeneity
 10908 in encounter probability, which is misspecified by standard SCR encounter probability
 10909 models. As a result, estimates of population size or density under models that do not
 10910 account for resource selection can be biased (Royle et al., 2012b). Finally, because the
 10911 resource selection model translates directly to a model for encounter probability for spatial
 10912 capture-recapture data, the implication of this is that it allows us to estimate resource
 10913 selection model parameters directly from SCR data, i.e., *absent* telemetry data. This
 10914 fact should broaden the practical relevance of spatial capture-recapture for studying or
 10915 estimating not just density, but also for directly studying movement and resource selection.

10916 Telemetry data has been widely used in conjunction with capture-recapture data.
 10917 For example, White and Shenk (2001) and Ivan (2012) suggested using telemetry data
 10918 to estimate the probability that an individual is exposed to sampling. However, their
 10919 estimator requires that individuals are sampled in proportion to this unknown quantity,
 10920 which seems impossible to achieve in many studies. In addition, they do not directly
 10921 integrate the telemetry data with the capture-recapture model so that common parameters
 10922 are jointly estimated. Sollmann et al. (in revision) and Sollmann et al. (2012b) used
 10923 telemetry data to directly inform the parameter σ from the bivariate normal SCR model
 10924 in order to improve estimates of density, although these models did not include an explicit
 10925 resource selection component.

13.1 A SIMPLE MODEL OF SPACE USAGE

10926 We assume here that our landscape is defined in terms of a discrete raster of one or more
 10927 covariates, having the same dimensions and extent. Let $\mathbf{x}_1, \dots, \mathbf{x}_{nG}$ identify the center
 10928 coordinates of nG pixels that define a landscape. We organize these coordinates into the
 10929 matrix $\mathbf{X}_{nG \times 2}$. Let $z(\mathbf{x})$ denote a covariate measured (or defined) for every pixel \mathbf{x} . For
 10930 clarity, and without loss of generality, we develop the basic ideas here in terms of a single
 10931 covariate. We suppose that a population of individuals wanders around space in some
 10932 manner related to the covariate $z(\mathbf{x})$, and their locations accumulate in pixels by some
 10933 omnipotent accounting mechanism. We will define “use of \mathbf{x} ” to be the event that an
 10934 individual animal appeared in some pixel \mathbf{x} during some interval of time.

10935 As a biological matter, use is the outcome of individuals moving around their home
 10936 range (Hooten et al., 2010), i.e., where an individual is at any point in time is the result
 10937 of some movement process. However, to understand space usage, it is not necessary to
 10938 entertain explicit models of movement, just to observe the outcomes, and so we don’t
 10939 elaborate further on what could be sensible or useful models of movement, but we imagine

existing methods of hierarchical or state-space models are suitable for this purpose (Jonsen et al., 2005; Forester et al., 2007; Patterson et al., 2008; Hooten et al., 2010; McClintock et al., 2012). XXX Also cite Ovasakinen papers XXXXXX We consider explicit movement models in the context of SCR models later chapters of this book (Chapt. XXX and XXX).

If an individual moves from a pixel \mathbf{x} to another pixel \mathbf{x}' this is defined as a decision to “use” pixel \mathbf{x}' . This also induces a definition of “truth” – that is, over any prescribed time interval, the animals makes some number, say R of use decisions, and they are, conceivably, observable by our omnipotent accounting mechanism (e.g., continuous telemetry). In this case, let r_{ij} be the *true* use frequency of pixel j by individual i – i.e., the number of times individual i used pixel j . We assume the vector of use frequencies $\mathbf{r}_i = (r_{i1}, \dots, r_{iG})$ has a multinomial distribution:

$$\mathbf{r}_i \sim \text{Multinom}(R, \boldsymbol{\pi}_i)$$

where

$$\pi_{ij} = \frac{\exp(\alpha_2 z(\mathbf{x}_j))}{\sum_x \exp(\alpha_2 z(\mathbf{x}))}$$

This is the standard RSF model (Manly et al., 2002) used to model telemetry data. One thing about Manly et al 2002 is that they offer numerous ways of modeling resource selection. They offer three “protocols” (pg 5) describing how used and unused resources are sampled. What we are discussing is their protocol A where all available resources (pixels) are censused, and used pixels are sampled randomly for each individual. They also describe 3 designs that vary in whether or not individual level data is collected. I think it is just worth being aware of this stuff because everybody that talks about RSFs thinks in these terms. The parameter α_2 is the effect of the landscape covariate $z(\mathbf{x})$ on the relative probability of use. Thus, if α_2 is positive, the relative probability of use increases as the covariate increases.

In practice, we don’t get to observe r_{ij} for all individuals but, instead, only for a small subset which we capture and install telemetry devices on. For the telemetered individuals, we assume their behavior is according to the same RSF model as the population as a whole.

We extend this model slightly to make it more realistic spatially. Let \mathbf{s} denote the centroid of an individuals home range and let $D_{ij} = \|\mathbf{x}_j - \mathbf{s}_i\|$ be the distance from the home range center of individual i , \mathbf{s}_i , to pixel j , \mathbf{x}_j . We modify the space usage model to accommodate that space use will be concentrated around an individuals home range centroid:

$$\pi_{ij} = \frac{\exp(-\alpha_1 D_{ij}^2 + \alpha_2 z(\mathbf{x}_j))}{\sum_x \exp(-\alpha_1 D_{ij}^2 + \alpha_2 z(\mathbf{x}_j))} \quad (13.1.1)$$

where $\alpha_1 = 1/(2\sigma^2)$ describes the rate at which capture probability declines as a function of distance. This has some context w.r.t. Johnson et al. and Forester in terms of modeling “availability” as a function of distance. But it is not necessary to distinguish between use vs. availability – really this model is cleanly interpreted as individuals using space as a function of how far away \mathbf{x} is from the individuals home range center. Don’t see a need to call that “availability”.

Note that Eq. 13.1.1 resembles standard encounter models used in spatial capture-recapture but with an additional covariate $z(\mathbf{x})$ (and see Chapt. 9). In particular, under this model for space usage or resource selection, if you have no covariates at all, or if $\alpha_2 = 0$, then the probabilities π_{ij} are directly proportional to the SCR model for encounter

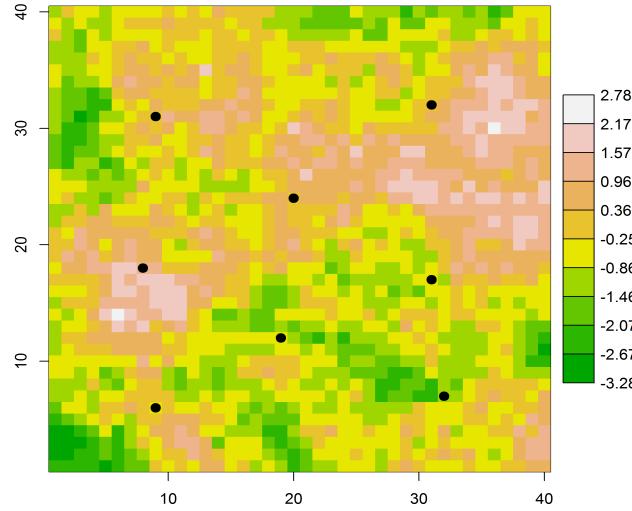


Figure 13.1. A typical habitat covariate reflecting habitat quality or hypothetical utility of the landscape to a species under study. Home range centers for 8 individuals are shown with black dots.

10981 probability. For example, setting $\alpha_2 = 0$, then this implies probability of use for pixel j
 10982 is:

$$p_{ij} \propto \exp(-\alpha_1 D_{ij}^2)$$

10983 so whatever function of distance we use in our RSF implies an equivalent model of space
 10984 usage (sec. 5.4) when used in SCR models. In particular, for whatever model we choose
 10985 for p_{ij} in an ordinary SCR model, we can modify the distance component in the RSF
 10986 function in Eq. 13.1.1 accordingly to be consistent with that model, by using whatever
 10987 function p_{ij} we choose according to

$$\pi_{ij} \propto \exp(\log(p_{ij}) + \alpha_2 z(\mathbf{x}_j))$$

10988 One difference between this observation model and those that we have considered in
 10989 previous chapters is that it includes the normalizing constant $\sum_x \exp(-\alpha_1 D_{ij}^2 + \alpha_2 z(\mathbf{x}_j))$,
 10990 which ensures that the use distribution is a proper probability density function. Thus we
 10991 are able to characterize the probability of encounter in terms of both distance and space
 10992 use.

10993 As an illustration of space usage patterns under this model, we simulated a covariate
 10994 that represents variation in habitat structure (Fig. 13.1) such as might correspond to
 10995 habitat quality. This was simulated by using a kriging interpolator with the following R
 10996 commands:

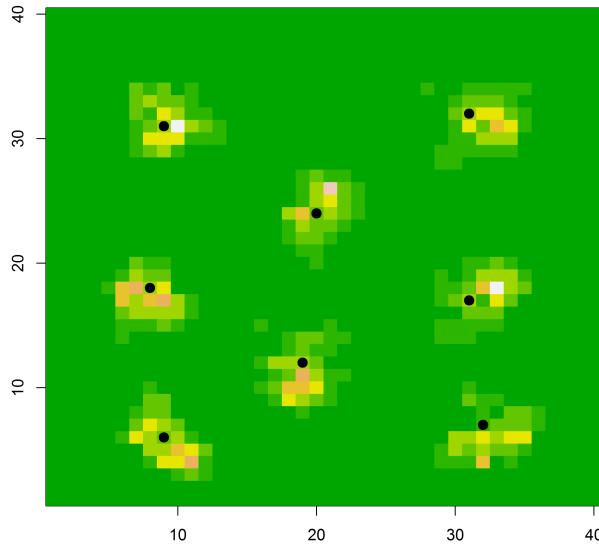


Figure 13.2. Space usage patterns of 8 individuals under a space usage model that contains a single covariate (shown in Fig. 13.1). Plotted value is the multinomial probability π_{ij} for pixel j under the model in Eq. 13.1.1.

```

10997 set.seed(1234)
10998 gr<-expand.grid(1:40,1:40)
10999 Dmat<-as.matrix(dist(gr))
11000 V<-exp(-Dmat/5)
11001 z<-t(chol(V))%*%rnorm(1600)

```

11002 Space usage patterns for 8 individuals are shown in Fig. 13.2, simulated with $\alpha_1 = 1/(2\sigma^2)$
 11003 with $\sigma = 2$ and the coefficient on $z(\mathbf{x})$ set to $\alpha_2 = 1$. These space usage densities – “home
 11004 ranges” – exhibit clear non-stationarity in response to the structure of the underlying
 11005 covariate, and they are distinctly asymmetrical. We note that if α_2 were set to 0, the 8
 11006 home ranges shown here would resemble bivariate normal kernels with $\sigma = 2$. Another
 11007 interesting thing to note is that the activity centers are not typically located in the pixel
 11008 of highest use or even the centroid of usage. That is, the observed “average” location is
 11009 not an unbiased estimator of \mathbf{s} under the model in Eq. 13.1.1.

13.1.1 Poisson use model

11011 A natural way to motivate the multinomial model of space usage is to assume that in-
 11012 dividuals make a sequence of resource selection decisions so that the outcomes r_{ij} are
 11013 marginally *independent*, having a Poisson distribution:

$$r_{ij} \sim \text{Poisson}(\lambda_{ij})$$

11014 where

$$\log(\lambda_{ij}) = a_0 - \alpha_1 D_{ij}^2 + \alpha_2 z(\mathbf{x})$$

11015 In this case, the number of visits to any particular cell is affected by the covariate $z(\mathbf{x})$
 11016 but has a baseline rate ($\exp(a_0)$) related to the amount of movement occurring over some
 11017 time interval. This is an equivalent model to the multinomial model given previously in
 11018 the sense that, if we condition on the total sample size $r_i = \sum_j r_{ij}$, then the vector \mathbf{r}_i
 11019 has a multinomial distribution with probabilities

$$\pi_{ij} = \frac{\lambda_{ij}}{\sum_j \lambda_{ij}}$$

11020 which is the same as Eq. 13.1.1 (see also Chapt. 9) because a_0 cancels from the numerator
 11021 and denominator of the multinomial cell probabilities and thus this parameter is not
 11022 relevant to understanding space usage.

11023 Also note that if use frequencies are summarized over individuals for each pixel, i.e.,
 11024 create the totals $r_j = \sum_i r_{ij}$, then a standard Poisson regression model for the resulting
 11025 “quadrat counts” is reasonable. This is “Design I” in Manly et al. (2002).

13.1.2 Thinning

11027 Suppose our sampling is imperfect so that we only observe a smaller number of telemetry
 11028 fixes than true use, r_{ij} . As developed in sec. 5.4, we assume that the observed number of
 11029 uses is

$$m_{ij} \sim \text{Bin}(r_{ij}, \phi_0).$$

11030 We can think of these counts as arising by thinning the underlying point process (here,
 11031 aggregated into pixels) where ϕ_0 is the thinning rate of the point process. In this case,
 11032 the marginal distribution of m_{ij} is also Poisson but with mean

$$\log(\lambda_{ij}) = \log(\phi_0) + a_0 - \alpha_1 D_{ij}^2 + \alpha_2 z(\mathbf{x}).$$

11033 Thus, the space-usage model (RSF) for the thinned counts m_{ij} is the same as the space-
 11034 usage model for the original variables r_{ij} . This is because if we remove r_{ij} from the
 11035 conditional model by summing over its possible values, then the vector of \mathbf{m}_i is *also*
 11036 multinomial with cell probabilities

$$\pi_{ij} = \frac{\phi_0 \lambda_{ij}}{\sum_j \phi_0 \lambda_{ij}}$$

11037 and so the nuisance parameter ϕ_0 cancels from the numerator and denominator. Thus,
 11038 the underlying RSF model applies to the true unobserved count frequencies \mathbf{r}_i and also
 11039 those produced from thinning or sampling, \mathbf{m}_i .

In summary, if we conduct a telemetry study we observe \mathbf{r}_i , the $nG \times 1$ vector of pixel-counts for each individual $i = 1, \dots, N_{tel}$. We declare these data to be “resource-selection data” which are typical of the type used to estimate resource-selection functions (RSFs) (Manly et al., 2002). Sometimes in RSF modeling activities we might have continuous covariates and so the denominator in Eq. 13.1.1 involves an integration over a distribution for the covariate which is the conditional intensity of observed point locations in a point process model. However, in a discrete landscape, entertaining pdfs for the covariates isn’t necessary (Royle et al., 2012b) when we recognize that the denominator should be the expectation over *space* and not the pdf of some covariate.

13.1.3 Capture-recapture Data

XXXXXXXXXXXXXXXXXXXX RC says: After reconciling SCR and RSF, cite that paper by Boyce and McDonald where they try to do accomplish the same objective using ad-hoc methods. That is the only effort to do something similar that I am aware of. XXXXXXXXXXXXXXXXXXXXXXX

The key to combining RSF data with SCR data is to work with this underlying resource utilization process and formulate SCR models in terms of that process. The idea in (Royle et al., 2012b) was to define the true use frequency for each pixel as the intermediate latent variable to which both telemetry data and SCR data are linked. Obviously we have to assume that both telemetered individuals and SCR individuals are using space according to the same resource selection model. The difference is that, for SCR data, we do not have sampling devices in all locations (pixels) in the landscape, and hence the data are only recorded at a subsample of them. XXXXX move the following XXXXX In other words, imagine that we have a sampling device, such as a camera trap, in *every* pixel. If the device operates continually then it is no different from a telemetry instrument. If it operates intermittently and does not expose the entire area of each pixel then a reasonable model for this imperfect observation is the “thinned” binomial model given above, where $\lambda_0 \equiv \exp(\phi_0)$ represents the sampling effectiveness of the device. So we imagine that the hypothetical perfect data from a camera trapping study are the thinned counts m_{ij} for every pixel j .

Introducing the latent use frequencies m_{ij} , and considering the Bernoulli SCR model where $y_{ij} = 1$ if the individual i visited the pixel containing a trap and was detected, then we imagine that y_{ij} is related to the latent variable m_{ij} being the event $m_{ij} > 0$, which occurs with probability

$$p_{ij} = 1 - \exp(-\lambda_{ij})$$

where

$$\log(\lambda_{ij}) = \log(\phi_0) + a_0 - \alpha_1 D_{ij}^2 + \alpha_2 z(\mathbf{x}).$$

We combine the constants so that $\alpha_0 = \log(\phi_0) + a_0$ is the baseline encounter rate which includes the constant intensity of use by the individual and also the baseline rate of detection, conditional on use. The Bernoulli observation model implies that the observed encounter frequencies for individual i and trap j , from sampling over K encounter periods is:

$$y_{ij} | \mathbf{s}_i \sim \text{Bin}(K; p_{ij})$$

11079 We imagine that any of the standard SCR observation model could be implemented here
 11080 with only minor modifications of the encounter probability model (following the develop-
 11081 ments of Chapt. 9.

13.2 THE JOINT RSF/SCR LIKELIHOOD

11082 To construct the likelihood for SCR data when we have auxiliary covariates on space usage
 11083 or direct information on space usage from telemetry data, we regard the two samples (SCR
 11084 and RSF) as independent of one another. In practice, this might not always be the case
 11085 but (1) often time the telemetry data come from a previous study; (2) the individuals are
 11086 not the same at all; (3) or even if they are some of the same individuals being captured,
 11087 we might not be able to match individuals captured by a sampling method such as hair-
 11088 snares with the individuals wearing radio-collars; (4) In cases where we *can* match some
 11089 individuals between the two samples, regarding them as independent should only entail a
 11090 minor loss of efficiency because we are disregarding more precise information on a small
 11091 number of activity centers. Moreover, we believe, it is unlikely in practice to expect the
 11092 two samples to be completely reconcilable and that the independence formulation is the
 11093 most generally realistic.

11094 Regarding the two data sets as being independent, our approach here is to form the
 11095 likelihood for each set of observations as a function of the same underlying parameters and
 11096 then combine them. In particular, let $\mathcal{L}_{scr}(\alpha_0, \alpha_1, \alpha_2, N; \mathbf{y}_{scr})$ be the likelihood for the
 11097 SCR data in terms of the basic encounter probability parameters and the total (unknown)
 11098 population size N , and let $\mathcal{L}_{rsf}(\alpha_1, \alpha_2; \mathbf{m}_{rsf})$ be the likelihood for the RSF data based
 11099 on telemetry which, because the sample size of such individuals is fixed, does not depend
 11100 on N . Assuming independence of the two datasets, the joint likelihood is the product of
 11101 these two pieces:

$$\mathcal{L}_{rsf+scr}(\alpha_0, \alpha_1, \alpha_2, N; \mathbf{y}_{scr}, \mathbf{m}_{rsf}) = \mathcal{L}_{scr} \times \mathcal{L}_{rsf}$$

11102 In what follows, we provide a formulation of each likelihood component, along with an **R**
 11103 function for obtaining the MLEs of model parameters using standard methods available
 11104 in **R**.

11105 Where the $L(scr)$ is the normal integrated likelihood (chapt XXX, equation XXXX)
 11106 and the rsf likelihood is the multinomial telemetry likelihood from eq. XXXX above.
 11107 That is, if we estimate s for the telemtered guys by the mean location then we could use
 11108 that plug-in otherwise we also have to integrate the multinomial likelihood For the
 11109 RSF data from the sample of individuals with telemetry devices we adopt the same basic
 11110 strategy of describing the conditional-on- s likelihood and then computing the marginal
 11111 likelihood by averaging over possible values of s . We have \mathbf{m}_i , the vector of pixel counts
 11112 for individual i , where these counts are derived from a telemetry study or similar. The
 11113 conditional-on- \mathbf{s}_i distribution of the telemetry data from individual i is:

$$[\mathbf{m}_i | \boldsymbol{\alpha}] = \prod_{g=1}^G \pi_{ij}(\mathbf{s}_i, z(\mathbf{x}_j))^{r_{ij}}$$

11114 where

$$\pi_{ij} = \frac{\exp(-\alpha_1 D_{ij}^2 + \alpha_2 z(\mathbf{x}_j))}{\sum_g \exp(-\alpha_1 D_{ij}^2 + \alpha_2 z(\mathbf{x}_j))}$$

11115 The marginal pmf is:

$$[\mathbf{m}_i | \boldsymbol{\alpha}] = \int_S [\mathbf{m}_i | \mathbf{s}_i, \boldsymbol{\alpha}] g(\mathbf{s}_i) d\mathbf{s}_i$$

11116 and therefore the likelihood for the RSF data is

$$\mathcal{L}_{rsf}(\boldsymbol{\alpha} | \mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_{Ntel}) = \prod_{i=1}^{Ntel} [\mathbf{m}_i | \boldsymbol{\alpha}].$$

11117 An R script is given in `scrbook` which is the function XXXX from manuscript ADD
 11118 TO REPO XXXXXX... will handle estimated s by sbar or whatever. Should be obvious
 11119 that estimating s with sbar is not the greatest thing because there is a covariate affecting
 11120 use, and so we expect the geographic centroid to be biased for s.

13.3 APPLICATION: NEW YORK BLACK BEAR STUDY

11121 Royle et al. (2012b) applied the integrated SCR+RSF model to data from a study of
 11122 black bears in a region of approximately 4,600 km² in southwestern New York (Sun,
 11123 in prep). We reproduce their results here. The data can be loaded from the `scrbook`
 11124 library with the command `data(nybears)`. **XXXX do this on the repo XXXXX**
 11125 A noninvasive, genetic, mark-recapture study was conducted to estimate density, and a
 11126 concurrent telemetry study was conducted in the same study area to understand patterns
 11127 of landscape connectivity and space usage. The study used DNA sampling obtained from
 11128 103 hair snares (Fig. ??) set from 6 June - 9 July, 2011. Hair snares were baited and
 11129 scented and checked weekly for hair. See Sun (in prep) for details of the genetic analysis.

11130 The study yielded captures of 33 individuals and a total of 14 recaptures (27 individuals
 11131 captured 1 time only; 3 individuals captured twice; 1 individual each three and four times).
 11132 Extra trap recaptures included 3 individuals captured in 2 traps, 1 individual in each of 3
 11133 and 4 traps. We used data from 3 radio-telemetered individual bears (2M, 1F) from the
 11134 same time period as the SCR data. Radio fixes were obtained approximately once per
 11135 hour and a total of 1,948 fixes on the 3 individuals were obtained. We thinned these hourly
 11136 fixes to once per 10 hours to approximate the data as independent movement outcomes,
 11137 producing 195 telemetry locations used in the RSF component of the model. We used the
 11138 covariate elevation in the model, derived from a one arc-second digital elevation model
 11139 (USGS National Elevation Dataset, accessed June 2012). This is shown in Fig. ?? (on a
 11140 standardized scale) which also shows the locations of each capture (multiple captures at
 11141 a trap location are dithered by adding random noise).

11142 We fitted a sequence of models based on the Gaussian hazard model (eq. 15.2.1)
 11143 including an ordinary SCR model with no covariates or telemetry data, the SCR model
 11144 with elevation affecting either λ_0 or density $D(\mathbf{x})$, and models that use telemetry data. We
 11145 have not discussed modeling covariate effects on density, but such models are described
 11146 by Borchers and Efford (2008) and we have not provided any novel treatment of that
 11147 modeling aspect here. The full list of models (with labels) is as follows:

11148 Model 1: SCR – ordinary SCR model

11149 Model 2: SCR+p(z) – ordinary SCR model with elevation as a covariate on baseline
 11150 encounter probability λ_0 .

11151 Model 3: SCR+D(z) – ordinary SCR model with elevation as a covariate on density
11152 only.
11153 Model 4: SCR+p(z)+D(z) – ordinary SCR model with elevation as a covariate on both
11154 baseline encounter probability and density.
11155 Model 5: SCR+p(z)+RSF – SCR model including data from 3 telemetered individuals.
11156 Model 6: SCR+p(z)+RSF+D(z) – SCR model including telemetered individuals and
11157 with elevation as a covariate on density.

11158 The first 4 models can be viewed together for purposes of model-selection by AIC since
11159 they are nested models. The last two models can be viewed together but cannot be
11160 compared to the first 4 because they include telemetry data. The results of fitting these
11161 6 models – the parameter estimates and standard errors are shown in Table 13.1. We
11162 provide a full R script for fitting all of these models to simulated data in Appendix 1.

11163 Among models 1-4, those models *without* the telemetry data, we see that the two mod-
11164 els with elevation affecting density are preferred – and, there is a large positive response
11165 to elevation. This is consistent with the visual pattern apparent in Fig. ?? where we see
11166 individual captures favoring high elevation sites. We also see a negative effect of elevation
11167 on *space usage* (the parameter α_2). It is interesting that the sign of the estimate of α_2
11168 changes from positive to negative when we add elevation as a covariate on density. Thus,
11169 the effect of elevation on density appears to have masked its effect on space usage. The
11170 estimate of N for the 4600 km² state-space is about 103 bears ($\exp(4.25) + 33$).

11171 In the two models that include the additional telemetry data, a couple points stand out:
11172 Clearly the elevation effect on density is important, reducing the negative log-likelihood
11173 by 5 units. The effect of elevation on density and space usage are roughly consistent with
11174 Model 4 which did not use telemetry data. Furthermore, the standard errors (SE) of
11175 those two parameter estimates are reduced considerably when the model uses telemetry
11176 data, as is the SE for estimating $\log(\sigma)$. The SE for estimating $\log(n_0)$ is only improved
11177 incrementally compared to the models without telemetry data. We used the best model,
11178 $\text{SCR}+\text{p}(z)+\text{RSF}+\text{D}(z)$, to produce a map of density (Fig. ??) which shows clearly the
11179 pattern induced by elevation. We also produced a map (Fig. ??) to illustrate the effect
11180 of elevation on space usage. This shows the relative probability of using a pixel x relative
11181 to one of mean elevation, and of the same distance from an individual's activity center.

11182 Resource selection can be described in hierarchical orders (Johnson, 1980), from selec-
11183 tion of a geographical area (first-order selection), selection of a home range within a study
11184 area (second-order), or selection of resources with that home range (third-order). Animals
11185 may select resources at different scales as a result of variability in the distribution of re-
11186 sources on the landscape (Mayor et al., 2009). Indeed, black bears make habitat selection
11187 decisions at multiple spatial scales, and decisions made at the second-order can differ from
11188 those at the third-order (Lyons et al., 2003; Sadeghpour and Ginnett, 2011). As a result
11189 of multi-scale resource selection, we can expect that the modeled covariates (elevation in
11190 our example) may affect density and space usage differently. We suggest that density is
11191 operating at the second-order and is largely related to the spacing of individuals and their
11192 associated home ranges across the landscape. On the other hand, our RSF was defined
11193 based on selection of resources within the home range (third-order). Because density and
11194 our third-order RSF were at different spatial scales, there is no expectation that the mod-
11195 eled covariate describing space usage (elevation) would influence each in a similar manner.
11196 Consistent with our positive relationship between elevation and density, the distribution

Table 13.1. Summary of model-fitting results for the black bear study. Parameter estimates are $\alpha_0 = \log(\lambda_0)$ and σ is the scale parameter of the half-normal hazard rate encounter model. The SCR data are based on $n = 33$ individuals, and the telemetry data are based on 3 individuals.

model	α_0	$\log(\sigma)$	α_2	$\log(n_0)$	β	-loglik
SCR+p(z)	-2.8600	-1.1170	0.1750	4.1400		122.7380
SE	0.3899	0.1390	0.2478	0.3657		
SCR	-2.7290	-1.1220	—	4.1100		122.9900
SE	0.3454	0.1404		0.3618		
SCR+D(z)	-2.7150	-1.1330	—	4.1140	1.2470	118.0070
SE	0.3526	0.1394		0.3575	0.4083	
SCR+p(z)+D(z)	-2.4840	-1.1570	-0.3840	4.2550	1.5710	117.0750
SE	0.3910	0.1421	0.2761	0.3768	0.4630	
SCR+RSF	-3.0680	-0.8140	-0.2810	3.8840		1271.7390
SE	0.2722	0.0364	0.1176	0.3626		
SCR+RSF+D(z)	-3.0700	-0.8100	-0.3710	4.0280	1.2730	1266.7000
SE	0.2720	0.0368	0.1239	0.3661	0.4110	

of a black bear population in the central Appalachian Mountains was positively associated with elevation (Frary et al., 2011). At the second-order, however, we observed a negative effect of elevation on space usage. Our study was conducted during summer, and seasonal shifts in elevation have been widely documented in black bears, often attributed to seasonal variation in food availability (Reynolds and Beecham, 1980; Gruber and White, 1983). The negative relationship between elevation and space usage during the summer could be attributable to either access to food resources at lower elevations, or access to river and stream corridors. Within their home ranges, black bears selected areas with high stream densities (Fecske et al., 2002), and in our study area, lower elevations were associated with river corridors which likely provided bears cooler conditions during the heat of summer.

13.4 SIMULATION STUDY

Royle et al. XXXXX carried-out a simulation study using the landscape shown in Fig. 13.1, and based on a population of $N = 100$ and $N = 200$ individuals with activity centers distributed uniformly over the landscape. This patchy covariate was simulated by generating a field of spatially correlated noise to emulate a typical patchy habitat covariate such as tree or understory density, or some other covariate relevant to habitat quality for a species. We subjected individuals to sampling over $K = 10$ sampling periods, using a 7×7 array of trapping devices located on the integer coordinates ($u * 5, v * 5$) for $u, v = 1, 2, 3, 4, 5, 6, 7$. The model parameters were

$$\text{cloglog}(p_{ij}) = -2 - \frac{1}{2\sigma^2} D_{ij}^2 + 1 \times z(\mathbf{x}_j)$$

for $\sigma = 2$. In the absence of the covariate z , this corresponds to an individual having a bivariate normal home range with standard deviation 2. These settings yielded an average of about $n = 61$ individuals captured for the $N = 100$ case and about $n = 123$ for the

11219 $N = 200$ case. The later case represents what we believe is an extremely large sample size
 11220 based on our own experience and thus it should serve to gauge the large sample bias of
 11221 the likelihood estimator (note: we expect little to no large sample bias).

In addition to simulating data from this capture-recapture study, we simulated 2, 4, 8, 12, 16 telemetered individuals to assess the improvement in precision as sample size increases. For all cases we observed 20 telemetry fixes *per* individual. The main things we're focused on with this simulation study were: (1) how much does the SE of estimated N improve as we add or increase the number of telemetered individuals? (2) How well does the SCR model do at estimating the parameter of the RSF with *no* telemetry data? (3) How much does the precision of the RSF parameter improve if we add SCR data to the telemetry data?

```

11230 N=100, 300 iters each, mean SCR only N: 99.418      N=200, 500 iters. Mean SCR only N = 199.712
11231 n=2          Nhat RMSE ahat RMSE sighat RMSE      Nhat RMSE ahat RMSE sighat RMSE
11232 SCR only:  99.73  9.97  0.99  0.14  2.00  0.124  198.85 14.24  0.99  0.10  2.00  0.091
11233 SCR/RSF:   99.94  9.54  0.99  0.12  2.00  0.097  199.37 12.80  0.99  0.09  2.00  0.078
11234 sbar       98.89  9.50  0.93  0.14  1.97  0.100  197.87 13.94  0.96  0.10  1.99  0.080
11235 RSF only   --    --    1.03  0.33  2.00  0.160  --    --    1.04  0.33  1.99  0.169
11236 n=4
11237 SCR only   99.10  9.83  0.99  0.13  2.00  0.127  200.06 15.34  1.00  0.09  2.00  0.092
11238 SCR/RSF    99.17  9.47  0.99  0.11  2.00  0.086  200.25 14.36  1.00  0.08  2.01  0.073
11239 sbar       97.43  9.68  0.89  0.16  1.97  0.090  198.14 14.31  0.94  0.10  1.98  0.075
11240 RSF only   --    --    0.98  0.22  2.00  0.119  --    --    1.02  0.21  2.01  0.122
11241 n=8
11242 SCR only   99.59 10.00  1.00  0.13  2.00  0.130  200.85 14.06  1.00  0.09  2.00  0.087
11243 SCR/RSF    98.90 10.02  0.99  0.10  2.00  0.071  200.29 13.98  1.00  0.08  2.00  0.061
11244 sbar       96.07 10.37  0.84  0.19  1.96  0.078  196.46 14.59  0.90  0.13  1.97  0.069
11245 RSF only   --    --    0.98  0.16  2.01  0.084  --    --    0.99  0.16  2.00  0.084
11246 n=12
11247 SCR only   99.44 10.73  0.98  0.13  2.02  0.128  198.76 14.47  0.99  0.10  2.00  0.091
11248 SCR/RSF    99.96 10.26  1.00  0.09  2.00  0.059  198.72 14.14  1.00  0.08  2.00  0.054
11249 sbar       96.30 10.49  0.82  0.20  1.96  0.071  193.83 15.14  0.87  0.15  1.97  0.063
11250 RSF only   --    --    1.01  0.12  2.00  0.069  --    --    1.01  0.13  2.00  0.069
11251 n=16
11252 SCR only   99.23 10.74  0.99  0.14  2.00  0.128  200.04 14.09  0.99  0.10  2.01  0.088
11253 SCR/RSF    99.20  9.79  1.00  0.09  1.99  0.057  200.25 13.40  1.00  0.07  2.00  0.047
11254 sbar       95.10 10.17  0.80  0.22  1.95  0.075  194.38 14.26  0.85  0.17  1.96  0.059
11255 RSF only   --    --    1.00  0.10  1.99  0.061  --    --    1.00  0.11  2.00  0.055
11256
11257 To check misspecification with isotropic h/r model I refitted the N =
11258 cases and fit the SCR only and SCR/RSF models IN ADDITION to the
11259 SCRO model with isotropic encounter model.
11260 n=2
11261 SCR only 199.11  14.28  0.99  0.09  2.00  0.090
11262 SCR/RSF   199.11  13.80  0.99  0.09  2.00  0.079
11263 SCRO     161.48  39.98  --    --    1.84  0.180
11264

```

11265	n=4						
11266	SCR only	199.67	13.87	1.00	0.09	2.00	0.090
11267	SCR/RSF	199.65	13.59	1.00	0.09	2.00	0.072
11268	SCRO	161.32	40.00	--	--	1.83	0.191
11269							
11270	n=8						
11271	SCR only	199.24	15.49	0.99	0.10	2.01	0.093
11272	SCR/RSF	199.55	14.17	0.99	0.08	2.00	0.063
11273	SCRO	161.46	40.06	--	--	1.84	0.184
11274							
11275	n=12						
11276	SCR only	200.41	15.16	0.99	0.10	2.00	0.086
11277	SCR/RSF	200.95	13.04	1.00	0.08	2.00	0.051
11278	SCRO	162.40	38.95	--	--	1.84	0.185
11279							
11280	n=16						
11281	SCR only	199.16	15.62	1.00	0.09	2.00	0.095
11282	SCR/RSF	199.63	13.38	1.00	0.07	2.00	0.052
11283	SCRO	160.93	40.44	--	--	1.84	0.190

The replicate runs of the SCR-only situation give us an idea of the inherent MC error in these simulations, which is roughly about 0.25 and 0.89 on the N scale for the $N = 100/N = 200$ cases. The mean N under “SCR only” across all 5 simulations for $N = 100$ was $\text{mean}(\hat{N}) = 99.418$, an empirical bias of 0.6%. For $N=200$, the estimated N across all 5 simulations (5 levels of ntel) was $\text{mean}(\hat{N}) = 199.712$, an empirical bias of about 0.15%, within the MC error of the true value of $N = 200$. The results suggests a very small bias of < 1% in the MLE of N in general when estimation is based on the full marginal likelihood. However, there is apparent bias of as much as 2-4% when s is estimated by the average observed location. The bias is slightly diminished as we double the expected sample size by doubling N from 100 to 200. In practice, we expect a small amount of bias in MLEs as likelihood theory only guarantees asymptotic unbiasedness. Moreover, the landscape resolution is fairly coarse relative to σ in our study, having a 1 km resolution whereas $\sigma = 2$, which we expect to introduce a small amount of negative bias because it is an explicit under-statement of the true heterogeneity in p due to the spatial context of the problem. The apparent bias that arises as a result of estimating s is expected because the average location of an individual would be unbiased for s only if the individual is moving according to a stationary isotropic kernel. Under the model of space usage with covariate $z(x)$, then the average location is biased to favor good values of $z(x)$ and so \bar{s} is really biased for s .

In terms of RMSE of the MLEs, generally there is about a 5% reduction in RMSE when we have at least 2 telemetered individuals, and, although there is a lot of MC error in the RMSE quantities, it might be as much as a 10% reduction (tops) as n increases under the higher $N = 200$ setting. This makes sense because we nail down the parameters and still don’t know where guys are, and get info about mean p , i.e. α_0 , only from the SCR data. Thus estimating N benefits only slightly from the addition of telemetry data.

Estimating the RSF parameter α_2 exhibits negligible or no bias except when s is estimated and, interestingly, it is well-estimated from SCR data alone and even better

than RSF data alone (in terms of RMSE) until we have more than 200 or so telemetry observations. The big improvement comes in estimating the home range parameter σ which is unbiased except when we estimate s in which case it exhibits only modest bias. However, there is huge improvement in RMSE of $\hat{\sigma}$, perhaps as much as 50-60% in some cases, but that really doesn't translate much into estimating N . Improvement due to adding RSF data from telemetry diminishes as the expected sample sizes increases, and so telemetry data does less to improve the precision of $\hat{\sigma}$ and $\hat{\alpha}_2$ for $N = 200$ than for $N = 100$.

We simulated a low p situation in which $\alpha_0 = -3$ producing $E[n] = 37$ under the $N = 100$ scenario. The effect is we have only incremental relative improvements in RMSE of N but relatively more improvement in RMSE for estimating σ . The MLE of N is positively biased. Interestingly this bias opposes slightly negative bias for the estimator based on estimating s and so that the wrong estimator actually does better. This is a complete chance occurrence and we should not get too excited by that.

N=100, low p, 500 iterations						
n=2	Nhat	RMSE	ahat	RMSE	sighat	RMSE
SCR only	103.85	22.88	1.00	0.19	2.02	0.261
SCR/RSF	102.90	20.98	1.00	0.17	2.00	0.136
sbar	101.55	20.91	0.90	0.19	1.96	0.136
RSF only	--	--	1.02	0.30	1.99	0.163
n=4						
SCR only	105.65	26.52	1.01	0.20	2.01	0.258
SCR/RSF	103.55	22.92	1.01	0.14	2.00	0.104
sbar	100.86	22.57	0.86	0.20	1.95	0.113
RSF only	--	--	1.01	0.21	1.99	0.114
n=8						
SCR only	107.41	45.05	0.99	0.19	2.01	0.254
SCR/RSF	104.28	22.13	1.00	0.12	2.00	0.076
sbar	99.82	21.55	0.80	0.23	1.95	0.091
RSF only	--	--	1.01	0.15	1.99	0.081
n=12						
SCR only	106.35	27.32	0.99	0.19	2.00	0.255
SCR/RSF	104.11	21.81	1.00	0.10	2.00	0.063
sbar	99.21	20.86	0.77	0.24	1.95	0.077
RSF only	--	--	1.01	0.12	2.00	0.065
n=16						
SCR only	104.05	31.41	0.99	0.19	2.02	0.252
SCR/RSF	101.98	20.78	1.00	0.09	2.00	0.055
sbar	96.78	20.25	0.76	0.26	1.95	0.070
RSF only	--	--	1.00	0.10	2.00	0.056

13.5 SUMMARY AND OUTLOOK

How animals use space is a fundamental interest to ecologists, and important in the conservation and management of many species. Normally this is done by telemetry and models referred to as resource selection functions (Manly et al., 2002). Conversely, spatial

capture-recapture models have grown in popularity over the last several years (Efford, 2004; Borchers and Efford, 2008; Royle, 2008; Efford et al., 2009b; Royle et al., 2009a; Gardner et al., 2010a,b; Kéry et al., 2010; Sollmann et al., 2011; Mollet et al., 2012; Gopalaswamy et al., 2012b). These, and indeed, most, development and applications of SCR models have focused on density estimation, not understanding space usage. However, it is intuitive that space usage should affect encounter probability and thus it should be highly relevant to density estimation in SCR applications. Despite this, a description of the relationship between encounter probability and space usage has not been developed explicitly in the literature on spatial capture-recapture models. Here we developed an SCR model in terms of a basic underlying model of space or resource use, that is consistent with existing views of resource selection functions (RSFs) (Manly et al., 2002).

Basically everyone does telemetry with SCR even though no one knows what to do with this stuff.

Our new class of integrated SCR/RSF models allows investigators to model how the landscape and habitat influence movement and space usage of individuals around their home range, using non-invasively collected capture-recapture data or capture-recapture data augmented with telemetry data. This should improve our ability to understand, and study, aspects of space usage and it might, ultimately, aid in addressing conservation-related problems such as reserve or corridor design. And, it should greatly expand the relevance and utility of spatial capture-recapture beyond simply its use for density estimation.

Integration of RSF data from telemetry with SCR models achieves a number of useful extensions of both ordinary SCR and RSF models: (1) Integration of the two distinct data sources (capture-recapture and telemetry) leads to an improvement in our ability to estimate density, and also an improvement in our ability to estimate parameters of the RSF function. As many animal population studies have auxiliary telemetry information, the ability to incorporate such information into SCR studies has broad applicability to many studies. It seems possible even to estimate density now, with no spatial recaptures, provided telemetry data are available. (2) The integrated model allows for the estimation of RSF model parameters directly from SCR data *alone*. This establishes clearly that SCR models *are* explicit models of space usage. In our view, this greatly broadens the utility and importance of capture-recapture studies beyond their primary historical use of estimating density or population size. (3) It is also now clear that one of the important parameters of SCR models, that controlling “home range radius”, is also directly estimable from telemetry data alone, and certainly its estimation is greatly improved with even moderate amounts of telemetry data. We pursue this topic from a design standpoint in Chapt. 10. (4) Resource selection can be viewed as inducing a type of heterogeneous encounter probability in capture-recapture studies. We say (Royle et al., 2012b) that misspecification of a simple resource selection model with a symmetric encounter probability model produces extremely biased estimates of N when the population of individuals does exhibit resource selection. As such, it is important to account for space usage when important covariates are known to influence space usage patterns.

In our formulation of the joint likelihood for RSF and SCR data, we assumed the data from a capture-recapture and telemetry studies were independent of one another. This implies that whether or not an individual enters into one of the data sets has no effect on whether it enters into the other data set. We cannot foresee situations in which violation of

11400 this assumption should be problematic or invalidate the estimator under the independence
11401 assumption. In some cases it might so happen that some individuals appear in *both* the
11402 RSF and SCR data sets. In this case, ignoring that information should entail only an
11403 incremental decrease in precision because a slight bit of information about an individual's
11404 activity center is disregarded. Heuristically, an SCR observation (encounter in a trap)
11405 is like one additional telemetry observation, and so the misspecification (independence)
11406 regards the two pieces of information as having separate activity centers. Our model
11407 pretends that we don't know anything about the telemetered individuals in terms of
11408 their encounter history in traps. In principle it shouldn't be difficult to admit a formal
11409 reconciliation of individuals between the two lists. In that case, we just combine the two
11410 conditional likelihoods before we integrate \mathbf{s} from the conditional likelihood. This would
11411 be almost trivial to do if *all* individuals were reconcilable (or none as in the case we have
11412 covered here) but, in general, we think you will always have an intermediate case – i.e.,
11413 either none will be or at most a subset of telemetered guys will be known. More likely
11414 you have variations of “well, that guy looks telemetered but we don't know which guy it
11415 is...hmmm” and that case, basically a type of marking uncertainty or misclassification, is
11416 clearly more difficult to deal with.

11417 In our formulation of the combined likelihood for RSF and SCR data, we assumed
11418 the data from capture-recapture and telemetry studies were independent of one another.
11419 This implies that whether or not an individual enters into one of the data sets has no
11420 effect on whether it enters into the other data set. We cannot foresee situations in which
11421 violation of this assumption should be problematic or invalidate the estimator under the
11422 independence assumption. In some cases it might so happen that some individuals appear
11423 in *both* the RSF and SCR data sets. In this case, ignoring that information should entail
11424 only an incremental decrease in precision because a slight bit of information about an
11425 individual's activity center is disregarded.

11426 Discussion point: Note that we could relax the uniformity assumption by specifying
11427 an inhomogeneous point process model (Borchers and Efford, 2008) as shown in Chapt.
11428 XXXXX. This allows for modeling second-order habitat selection as defined by Johnson
11429 (1980). Thus, SCR models provide insight into the hierarchical nature of habitat selection.
11430 Simultaneously we model all types of habitat selection in a single unified model based on
11431 capture-recapture data.

11432 Bayesian analysis might have an advantage in situations where the landscape is charac-
11433 terized by a very fine covariate raster, or even continuous covariates, because the individual
11434 activity centers can be updated in the MCMC algorithm by evaluating the likelihood con-
11435 ditional on a single candidate value of \mathbf{s} for each individual. Conversely, evaluation of
11436 the marginal likelihood becomes tedious and memory intensive as the size of the raster
11437 increases, and so some effort has to be made to efficiently calculate the likelihood in such
11438 cases (e.g., see Warton and Shepherd, 2010). Independent of its effect on integration,
11439 raster size is itself an important practical concern. Whenever we have explicit spatial
11440 covariates, it is possible that selection is occurring at a much finer resolution than is re-
11441 quired to effectively integrate the likelihood over the state-space of \mathbf{s} . In this case, too
11442 coarse of a raster will likely cause biased parameter estimates (having an effect analogous
11443 to measurement error in regression, we suspect). Too fine, however, creates concomitant
11444 effects on computing and memory requirements. Choice of raster size or spatial resolution
11445 is thus both a fundamental scientific question, but also very much a practical computing

11446 issue.

11447 We developed the model in a discrete landscape which regarded potential trap locations
11448 and the covariate $z(\mathbf{x})$ as being defined on the same set of points. In practice, trap locations
11449 may have been chosen independent of the definition of the raster and this does not pose
11450 any challenge or novelty to the model as we developed it. In that case, the covariate(s)
11451 need to be defined at each trap location. The model should be applicable also to covariates
11452 that are naturally continuous (e.g., distance-based covariates) although, in practice, it will
11453 usually be sufficient to work with a discrete representation of such covariates.

11454 We used an RSF model for telemetry data that is most suitable for independent obser-
11455 vations of space usage. This would be reasonable if telemetry fixes are made reasonably
11456 far apart in time, or if the telemetry data is thinned, as we did in our analysis of the black
11457 bear data. However, use of the independence model for non-independent data is probably
11458 only a minor problem for estimating density or other model parameters because we ex-
11459 pect that the pixel use frequencies should remain unbiased in this case¹. We imagine that
11460 precision should be over-stated for the parameters of the RSF model because the sample
11461 size is not reflecting the dependence of the observations. In general, however, it will be
11462 desirable to incorporate more general (or explicit) models of movement into the framework
11463 proposed here, so that SCR models can be used to improve inferences about animal move-
11464 ment, and because more explicit models may improve inferences about density obtained by
11465 capture-recapture studies. As we noted, our specific model of independence corresponds
11466 to a limiting case of the Gaussian process movement model (Johnson et al., 2008), but
11467 including the general RSF movement model for correlated data from Johnson et al. (2008)
11468 should not pose any difficulty in terms of constructing the combined SCR+RSF likelihood
11469 (but contain one additional parameter).

¹As a technical matter, we think regular movement models should exhibit an ergodic property analogous to standard MCMC algorithms, time-series models and related dynamical systems.

11470
11471

14

11472
11473

STRATIFIED POPULATIONS: MULTI-SESSION AND MULTI-SITE DATA

11474 In this chapter, we describe SCR models for situations when we have multiple, distinct
11475 groups, strata or “sessions” (multi-session models using the `secr` terminology). The mod-
11476 els are extremely general and provide a flexible hierarchical modeling framework for mod-
11477 eling abundance (Converse and Royle, 2012; Royle et al., 2012c). We believe that such
11478 “stratified” populations are extremely commonplace, yet most SCR applications have been
11479 based on models that are distinctly single-population models. This is done either by an-
11480 alyzing separate data sets one-at-a-time or by pooling data from multiple study areas. A
11481 standard example that arises frequently is that in which multiple distinct patches (often
11482 refuges, parks or reserves) are sampled independently with the goal of estimating the pop-
11483 ulation size in each reserve. It makes sense to combine the data together into a single
11484 model that permits the sharing of information about some parameters, but provides in-
11485 dividual estimates of abundance for each land unit. A similar situation is that in which
11486 a number of replicate trap arrays are located within a landscape, sometimes for purposes
11487 of evaluating management actions or landscape structure. This is extremely common in
11488 studies of small mammals (Converse et al., 2006a,b; Converse and Royle, 2012), or in mist-
11489 netting of birds (DeSante et al., 1995) (BETTER REF HERE WOULD BE NICE), but
11490 there are examples of large-scale monitoring of carnivores and other species, e.g., tigers
11491 (Jhala et al., 2011).

11492 Stratified or multi-session SCR models are also directly relevant when the grouping is
11493 based on distinct time samples, either periods within a biological season, or even across
11494 years. Unlike in the case of having spatial strata, with temporally defined samples, we
11495 imagine a fully dynamic, or demographically open, model that involves survival and re-
11496 cruitment might be suitable. We deal with those models specifically in Chapt. 16. How-
11497 ever, the stratified (multi-session) models we deal with in this chapter can be thought of
11498 as a primitive type of model for open systems, but in which the populations are assumed
11499 to be *independent*. Whereas the underlying model may be one of Markovian dynamics
11500 (survival, recruitment), we could *ignore* that dependence for convenience or perhaps the
11501 dynamics are not distinctly estimable because individual recapture rate is low.

11502 We focus mostly on Bayesian analysis of stratified SCR models using data augmentation (Royle et al., 2012c; Royle and Converse, in review). The technical modification
 11503 of data augmentation to deal with such models is that it is based on a model for the joint distribution of the stratum-specific population sizes, N_g , *conditioned* on their total.
 11504 This results in a multinomial distribution which we can analyze in some generality using data augmentation.
 11505 As a practical matter, specification of this multinomial distribution for the N_g parameters *induces* a distribution for an individual covariate, say g_i , which
 11506 is “group membership”. This is extremely handy to analyze by MCMC in the various
 11507 **BUGS** engines that you are familiar with by now.

11511 The **R** package **secr** fits a class of multi-session models which we have already seen
 11512 (sec. 6.5.4) and we used this class of models to analyze the ovenbird data in **secr** (sec.
 11513 9.2.4). Later in this chapter we will provide a Bayesian analysis of the ovenbird data in
 11514 **BUGS** using an analogous class of models.

14.1 DATA STRUCTURE

11515 We suppose that $g = 1, 2, \dots, G$ populations, having sizes N_g , state-spaces S_g are sampled
 11516 using some capture-recapture method producing sample sizes of n_g unique individuals and
 11517 encounters y_{ijk} for individual $i = 1, 2, \dots, \sum_{g=1}^G n_g$. Right now we won’t be concerned
 11518 with the details of every type of capture-recapture observation model so, for context,
 11519 we consider the Bernoulli model in which individual and trap-specific encounter frequen-
 11520 cies are binomial counts: $y_{ij} \sim \text{Binomial}(K, p_{ij})$. Let g_i be a covariate (integer-valued,
 11521 $1, \dots, G$) indicating the population membership of individual i . This covariate is *observed*
 11522 for the sample of captured individuals but not for individuals that are not captured.

11523 A key idea that we develop shortly is that the assumption of certain models for the
 11524 collection of abundance variables N_g *implies* a specific model for the population mem-
 11525 bership variable g_i . Then, the data from all populations can essentially be pooled, and
 11526 analyzed as data from a single population with the appropriate model on g_i , without
 11527 having to model the N_g parameters *directly*. In this way, we can easily build hierarchical
 11528 models for stratified populations, using an *individual* level parameterization of the model.
 11529 Obviously this is important for SCR models as they all possess at least onee random effect
 11530 in the form of the activity center **s**. Moreover, in the context of stratified or multi-session
 11531 type models, the “population membership” variable g_i is a *categorical* type of individual
 11532 covariate (Huggins 1989; Alho 1990; Royle 2009).

11533 To illustrate the prototypical data structure for stratified SCR data, we suppose that
 11534 a population comprised of 4 sub-populations is sampled $K = 5$ times. Then a plausible
 11535 data set has the following structure:

```
11536      individual (i) : 1 2 3 4 5 6 7 8 9 10
11537      total   encounters (y) : 1 1 3 1 1 2 2 4 1 1
11538      group (g)       : 1 1 1 2 3 3 3 3 4 4
```

11539 This data set indicates three individuals were captured in subpopulation 1 (captured 1,
 11540 1, and 3 times), a single individual was captured in population 2, four individuals were
 11541 captured in population 3, and two individuals were captured in subpopulation 4.

14.2 MULTINOMIAL ABUNDANCE MODELS

11542 The Poisson GLM is commonly used throughout ecology to model variation in abundance.
 11543 Consider sampling $g = 1, 2, \dots, G$ populations having unknown sizes N_g :

$$N_g \sim \text{Poisson}(\lambda_g) \quad (14.2.1)$$

11544 with

$$\log(\lambda_g) = \beta_0 + \beta_1 x_g \quad (14.2.2)$$

11545 where x_g is some measured attribute for population g . Under this Poisson model, by
 11546 conditioning on the total population size over all G populations, the N_g variables have a
 11547 multinomial distribution:

$$\mathbf{N} = (N_1, \dots, N_G) | \{N_T = \sum_g N_g\} \sim \text{Multinom}(\boldsymbol{\pi} | N_T). \quad (14.2.3)$$

11548 with multinomial probabilities $\pi_g = \lambda_g / \sum_g \lambda_g$. This relationship between Poisson and
 11549 multinomial random variables is a standard distribution theory result.

11550 **XXXX below here needs edited and reworked XXXXX XXXX don't need**
 11551 **any of this ... just the z/g model cite back to R/C papers XXXX**

11552 To devise a data augmentation scheme for this model of population size, we embed
 11553 the multinomial for $\{N_s\}$ into a multinomial of the same dimension but with larger, fixed
 11554 sample size. Specifically, we introduce a latent super-population variable M_s which we
 11555 assume has the desired Poisson distribution but with scaled mean: $M_s \sim \text{Poisson}(A\lambda_s)$
 11556 where $A \gg 1$ where A exists (can be chosen) to ensure that M_s is arbitrarily larger
 11557 than N_s . Conditional on the total super-population size $M_T = \sum_s M_s$, then \mathbf{M} has a
 11558 multinomial distribution:

$$\mathbf{M} | M_T \sim \text{Multinom}(M_T; \boldsymbol{\pi}) \quad (14.2.4)$$

11559 where $\pi_s = \lambda_s / \sum_s \lambda_s$ which are the same probabilities as for the target multinomial
 11560 for \mathbf{N} . This multinomial model for the super-population sizes M_s is equivalent to the
 11561 following:

$$g_i \sim \text{Categorical}(\boldsymbol{\pi})$$

11562 for $g_i; i = 1, 2, \dots, M_T$. Given \mathbf{M} or, equivalently, g_i , we specify a model for $\{N_g\}$ that
 11563 differentiates between "real" and "pseudo-" individuals by a Bernoulli sampling model:

$$N_s \sim \text{Binom}(M_s, \psi)$$

11564 where $\psi \sim \text{Uniform}(0, 1)$. Bernoulli sampling preserves the marginal Poisson assumption
 11565 (Takemura 1999). That is, N_s is Poisson, unconditional on M_s and, also, conditional
 11566 on $N_T = \sum_s N_g$, \mathbf{N} has a multinomial with probabilities $\boldsymbol{\pi}$ and index N_T . Note also
 11567 that $N_T \sim \text{Binom}(M_T, \phi)$ which is consistent with data augmentation applied to total
 11568 population size N_T . This binomial sampling model can be represented, equivalently, by
 11569 the set of Bernoulli variables:

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

11570 for $i = 1, 2, \dots, M_T$.

11571 The multinomial construction makes it clear that ψ is confounded with $\exp(\beta_0)$. By
 11572 constructing the model conditional on the total, we lose information about the intercept

11573 β_0 , but this is recovered in the data augmentation parameter ψ . One of these parameters
 11574 has to be fixed. We can set $\beta_0 = 0$ or else we can fix ψ . The constraint can be specified
 11575 by noting that, under the binomial data augmentation model $\mathbb{E}(N_T) = \psi M_T$ and, under
 11576 the Poisson model, $\mathbb{E}(N_T) = \sum_g \exp(\beta_0 + \beta_1 x_g)$ and so we can set

$$\psi = \frac{1}{M_T} \sum_g \exp(\beta_0 + \beta_1 x_g).$$

11577 The equivalence of ψ and β_0 can be thought of in terms of pooling data from the different
 11578 sub-populations. In a model with *no* covariates, we could pool all of the data and estimate
 11579 a single parameter ψ or β_0 but not both. In this sense, pooling data from multiple spatial
 11580 samples is justifiable (in terms of sufficiency arguments) under a Poisson assumption on
 11581 local abundance (which was noted by Royle 2004b; Royle and Dorazio 2008, sec. 5.5.1).

11582 By introducing the latent M_g structure, and the Bernoulli sampling of N_s , the model
 11583 is equivalently represented by the latent variable pair (g_i, z_i) where g_i is categorical with
 11584 prior probabilities π_s and $z_i \sim \text{Bern}(\psi)$. In particular, the multinomial assumption for
 11585 the latent variables G_s is formulated in terms of “group membership” for each individual
 11586 in the super-population of size M according to:

$$g_i \sim \text{Categorical}(\boldsymbol{\pi})$$

11587 with $\boldsymbol{\pi} = (\pi_1, \dots, \pi_S)$ and $\pi_s = \lambda_s / (\sum_s \lambda_s)$. Note that aggregating these M categorical
 11588 variables yields a set of multinomial variables consistent with Eq. 14.2.4. That is, define
 11589 $G_1 = \sum_{i=1}^M I(g_i = 1)$, $G_2 = \sum_{i=1}^M I(g_i = 2)$, etc., where $I()$ is the indicator function. The
 11590 binomial sampling from the super-population, $N_T \sim \text{Binom}(M, \psi)$ can be described at
 11591 the level of the individual also, by introducing the binary variables z_1, \dots, z_M such that

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

11592 where ψ is constrained as noted in the previous section. We implement this individual-level
 11593 formulation of the model in BUGS in Panel 14.1.

11594 A second implementation of the model is suggested by working from Eq. (14.2.3) –
 11595 we can marginalize N_T over the prior $N_T \sim \text{Binom}(M, \phi)$ to see that the $(S+1) \times 1$
 11596 vector $(N_1, \dots, N_S, N_{S+1})$ has, conditional on M , a multinomial distribution with cell
 11597 probabilities $\pi_s^+ = \pi_s \psi$ for $s = 1, 2, \dots, S$ and $\pi_{S+1}^+ = (1 - \psi)$ for the last cell which
 11598 corresponds to individuals of the super-population that are not members of any of the S
 11599 populations that were subject to sampling. Thus,

$$N|M \sim \text{Multinom}(\boldsymbol{\pi}^+).$$

11600 where the superscript + here indicates that $\boldsymbol{\pi}^+$ is a larger version of $\boldsymbol{\pi}$ from 14.2.4. In this
 11601 case,

$$g_i \sim \text{Categorical}(\boldsymbol{\pi}^+) \text{ for } i = 1, \dots, M_T \quad (14.2.5)$$

11602 The two distinct implementations are shown in Panel 14.1 for an ordinary closed popula-
 11603 tion model (model M_0).

11604 14.2.1 Observation Models

11605 Any observation model is cool here, no worries. We show a multinomial model below.

11606 Bernoulli model.... replace binom command in WB code with a double-loop, y[i,j]
 11607 etc...

```

model {
# This will show that psi and b0
#   are confounded.
  p ~ dunif(0,1)
  b0 ~ dnorm(0,.1)
  b1 ~ dnorm(0,.1)
  psi ~ dunif(0,1)
  for(s in 1:S){
    log(lam[s]) <- b0 + b1*x[s]
    gprobs[s] <- lam[s]/sum(lam[1:S])
  }
  for(i in 1:M){
    g[i] ~ dcat(gprobs[])
    z[i] ~ dbern(psi)
    y[i] ~ dbin(mu[i], J)
    mu[i] <- z[i]*p
  }
  N <- sum(z[1:M])
}

```

```

model {
# This version constrains psi with
#   the intercept parameter
  p ~ dunif(0,1)
  b0 ~ dnorm(0,.1)
  b1 ~ dnorm(0,.1)
  psi <- sum(lam[])/M
  for(j in 1:K){
    log(lam[j]) <- b0 + b1*x[j]
    gprobs[j] <- lam[j]/sum(lam[1:K])
  }
  for(i in 1:M){
    g[i] ~ dcat(gprobs[])
    z[i] ~ dbern(psi)
    y[i] ~ dbin(mu[i], J)
    mu[i] <- z[i]*p
  }
  N <- sum(z[1:M])
}

```

Panel 14.1: BUGS model specification for a capture-recapture model with constant encounter probability and Poisson subpopulation sizes, N_k , with mean depending on a single covariate $x[j]$. Two versions of the model: The first one describes the model in terms of the intercept β_0 and DA parameter ψ , which are confounded. The required constraint is indicated in the specification on the RHS.

14.2.2 Simulating group structured capture-recapture data

11609 It is helpful, as always, to simulate some data in order to understand the model. Suppose
 11610 we carry-out a trapping study, say using hair-snares within baited wooden boxes for a
 11611 species of *Mustelid*, and we establish arrays of 25 hair snares organized in an opportunistic
 11612 along stream networks within 20 watersheds. Actually, we didn't know anything about
 11613 SCR when we did this study but we set up hairsnares to be at least 1 stream km apart
 11614 from each other, based on a systematic sampling of all stream and wetland boundaries.
 11615 The main objective is to study the effect of development on mink density, measured by
 11616 building structures per km^2 , but each watershed also differs in the amount of available
 11617 habitat which we characterize the km of stream plus km of lake, pond and major wetland
 11618 shoreline. We imagine that

$$\log(\lambda_g) = \log(area_g)\beta_0 + \beta_1 habitat + \beta_2 development$$

11619 simulate..... R script

14.2.3 Fitting in BUGS

11621 Bernoulli observation model.....
 11622 Fit the model in WinBUGS with about 10 lines of code.....

14.2.4 Approach B modeling ψ

11624 Another idea here is to model the DA parameter ψ as being variable for each subject as
 11625 a function of group-specific variables (Tenan ref....?). That is, if x_g is the value of some
 11626 stratum covariate then we could have $z_i \sim Bern(\psi_i)$ with

$$\text{logit}(\psi_i) = \beta_0 + \beta_1 x_{g_i}$$

11627 This implies a binomial model for the stratum population sizes:

$$N(g) \sim Bin(psi(g), M)$$

11628 and also a multinomial for the vector $N_1, \dots, N_G, M - \sum N$ with probabilities ψ_g and, for
 11629 the last cell, $1 - \sum_g \psi_g$. This is almost the same multinomial as produced by the other
 11630 approach. Also, if M is sufficiently large then the $N(g)$ are approximately independent
 11631 Poisson random variables with means $\psi_g M$.

14.3 SPATIAL CAPTURE-RECAPTURE

11632 We describe a model for the encounter histories conditional on knowing to which pop-
 11633 ulation each observed individual belongs. Let $\mathbf{y}_{ik} = (y_{i1k}, y_{i2k}, \dots, y_{ijk})$ be the spatial
 11634 encounter history for individual i , a sequence of 0's and 1's for individual i during sample
 11635 k .

11636 A standard type of model which applies to detection devices such as hair snares
 11637 (Borchers and Efford 2008; Gardner et al. 2010) is that in which the y_{ijk} are independent
 11638 Bernoulli trials so that an individual can be captured in any number of the J traps during

11639 a sample occasion. In that case, the probability of encounter in trap j is modeled by some
 11640 function of the distance between trap j (a 2-dimensional vector \mathbf{x}_j), and the individual
 11641 activity center \mathbf{s}_i which is regarded as a latent variable (i.e., “random effect”) in the model
 11642 (Borchers and Efford 2008; Royle and Young 2008). For example, a common model is the
 11643 “half-normal” model:

$$\Pr(y_{ijk} = 1) = p_{ijk} = p_0 \exp\left(-\frac{1}{2\sigma^2} \text{dist}(\mathbf{x}_j, \mathbf{s}_i)^2\right)$$

11644 Or, equivalently, we can express this as a linear function on a suitable scale: $\log(p_{ijk}) =$
 11645 $\alpha_0 + \alpha_1 \text{dist}(\mathbf{x}_j, \mathbf{s}_i)^2$ where $\alpha_0 = \log(p_0)$ and $\alpha_1 = 1/(2\sigma^2)$.

11646 For standard live traps – also called “single catch” traps (Efford 2004), an individual
 11647 can be captured in at most one trap. Then, the vector $(y_{i1k}, y_{i2k}, \dots, y_{iJk}, y_{i,J+1,k})$,
 11648 where the last element $y_{i,J+1,k}$ corresponds to “not captured”, contains a single 1 and the
 11649 remaining values are 0. This $(J + 1) \times 1$ vector \mathbf{y}_{ik} is a multinomial trial:

$$\mathbf{y}_{ik} \sim \text{Multinomial}(n = 1; \boldsymbol{\pi}_{ik})$$

11650 where $\boldsymbol{\pi}_{ik}$ is a $(J + 1) \times 1$ vector where each element represents the probability of being
 11651 encountered in a trap (for elements 1, ..., J) or not captured at all (element $J + 1$).

11652 For the multinomial case, we also model the encounter probability vector as a function
 11653 of distance between trap locations and individual activity centers, but for this case we use
 11654 the multinomial logit transform. The equivalent half-normal model is:

$$\text{mlogit}(\pi_{ij}) = \eta_{ij} = \alpha_0 + \alpha_1 * \text{dist}(\mathbf{x}_j, \mathbf{s}_i)^2 \quad (14.3.1)$$

11655 where $\alpha_1 = 1/(2\sigma^2)$ and σ is the scale parameter of the half-normal detection model.
 11656 Then

$$\boldsymbol{\pi}_{ij} = \exp(\eta_{ij}) / [1 + \sum_j \exp(\eta_{ij})]$$

11657 for each $j = 1, 2, \dots, J$, and the last cell corresponding to the event “not captured” is:

$$\pi_{i,J+1} = 1 - \sum_{j=1}^J \pi_{ij}$$

11658 It is easy to build additional covariates into this model including those that vary by
 11659 sample occasion. For example, to model a behavioral effect (which we do in the example
 11660 below), let C_{ik} be a covariate of previous encounter (i.e., $C_{ik} = 0$ before the occasion of
 11661 first capture, and $C_{ik} = 1$ thereafter), then

$$\text{mlogit}(\pi_{ijk}) = \eta_{ijk} = \alpha_0 + \alpha_1 * \text{dist}(\mathbf{x}_j, \mathbf{s}_i)^2 + \alpha_2 * C_{ik}$$

11662 We note, in this case, the multinomial probabilities depend not only on individual and
 11663 trap, but also on sample occasion.

14.4 APPLICATION

11664 Single-catch traps
 11665 XXX move this stuff XXXXX

11666 For the single-catch system, the independent multinomial model we employed is a
 11667 misspecification of the true observation model. This is because competition for single-
 11668 catch traps renders the independence assumption invalid. As Efford et al. (2009) noted,
 11669 we expect “bias to be small when trap saturation (the proportion of traps occupied) is low.
 11670 Trap saturation will be higher when population density is high...”, relative to trap density,
 11671 or when net encounter probability is high. Efford et al. (2009) did a limited simulation
 11672 study and found essentially no effective bias and concluded that estimators of density
 11673 from the misspecified independent multinomial model are robust to the mild dependence
 11674 induced when trap saturation is low. Conversely, properly specifying the likelihood for the
 11675 correct single-catch system is challenging and, so far, has eluded formal characterization
 11676 by researchers.

11677 XXXXXX

11678 We applied this model to data described by Converse et al. (2006). The data were
 11679 collected as part of an effort to understand the impacts of fuel reduction treatments on
 11680 small mammal populations at 2 replicate study sites in northern New Mexico (Fig. 14.1;
 11681 the Jemez Mountains Study area of the National Fire and Fire Surrogate Study; McIver
 11682 et al. 2008), with trapping over 3 years (2001-2003) in each of 4 replicate experimental
 11683 units per study site (i.e., number of groups $G = 24$, 8 units by 3 years, in this exam-
 11684 ple). The experimental design included plans for thinning, burning, and thinning/burning
 11685 combination treatments, as well as a control, at each study site. However, during the
 11686 period when these data were collected, the thinning only treatment was completed on a
 11687 single experimental unit at the JM-B study area (see Converse et al. 2006:1713), and at
 11688 the JM-C study area, all 4 study experimental units were burned in a wildfire. Both the
 11689 thinning treatment and the wildfire took place between the 2002 and 2003 study seasons.

11690 Trapping was conducted over 10 occasions (2 per day) at each experimental unit, with
 11691 half the units at each site randomly selected in the first year for trapping in trapping
 11692 session 1, which lasted 5 days, and half in session 2, an additional 5 days. The assignment
 11693 to session then alternated over years. In 2001, the traps in each experimental unit were
 11694 configured in a 6 by 6 grid, with 50 m between each trap. After a pilot project to assess
 11695 the effects of trap spacing (Converse et al. 2004) the trap density was increased such that
 11696 there was 25 m between traps, and so the grid was an 11 by 11 grid with 121 total trap
 11697 stations. Multiple species were captured in the grids, but we base our analyses on the
 11698 species with the largest number of captures, the deer mouse (*Peromyscus maniculatus*).

11699 The detection model is related to covariates through the multinomial logit transform in
 11700 which the trap-specific encounter probabilities are given by Eq. 14.3.1. In the application
 11701 we have

$$\eta_{ijk} = \alpha_{0,g_i} + \alpha_1 * C_{ik} + \alpha_{2,g_i} * d_{ij}^2$$

11702 where $d_{ij} \equiv dist(\mathbf{s}_i, \mathbf{x}_j)$, $\alpha_{0,1}, \dots, \alpha_{0,G}$ are group-specific intercepts, α_1 is the behavioral
 11703 response parameter, C_{ik} is a covariate of previous encounter (i.e., $C_{ik} = 0$ before the
 11704 occasion of first capture, and $C_{ik} = 1$ thereafter), and α_{2,g_i} is a group-specific coefficient on
 11705 distance (related to σ_{g_i} by: $\alpha_{2,g_i} = 1/(2\sigma_{g_i})$), allowing for the possibility that treatments
 11706 influence home range size.

11707 To accommodate differences in trap array configuration (e.g., 6×6 vs. 11×11 grids),
 11708 we introduce a trap-operation matrix, \mathbf{A} where $A_{j,k}^g = 1$ if, for group g , trap j is oper-
 11709ational during period k and $A_{j,k}^g = 0$ otherwise. A similar approach could be used if, in
 11710 practice, certain traps were not operational during certain occasions. This could occur, for

example, if traps were sprung or damaged by animals. Then we include trap availability as multiplying $\exp(\eta_{ijk})$ so that, in the multinomial logit transform, the cell probability is zeroed out for an inoperative trap.

For the abundance model, we assume that N_g is Poisson with mean

$$\lambda_g = \exp(\beta_{0,g} + \mathbf{x}'_g \boldsymbol{\beta})$$

where $\beta_{0,g}$ is a group-specific random effect (see below), \mathbf{x}'_g is a vector of population-specific covariates, and including an intercept. In our analysis here, $\mathbf{x}_g = (\text{year1}_g, \text{year2}_g, \text{thin}_g, \text{fire}_g)$ where **year1** and **year2** are dummy variables indicating years 2001 and 2002) i.e., **year1**_{*g*} = 1 if group *g* occurred in 2001, **season2**_{*g*} = 1 if group *g* occurred in 2002; **thin** and **fire** are binary treatment effects being **thin**_{*g*} = 1 if group *g* was a thinned experimental unit, and **fire**_{*g*} = 1 if group *g* was a burned experimental unit.

We used proper uniform prior distributions for each of the regression coefficients: $\beta_m \sim \text{Unif}(-10, 10)$ for $m = 1, 2, 3, 4$, $\alpha_1 \sim \text{Unif}(-10, 10)$, and $\alpha_2 \sim \text{Unif}(-10, 10)$. For the group-specific intercept parameters $\beta_{0,g}$ we assumed:

$$\beta_{0,g} \sim \text{Normal}(0, \tau_\lambda)$$

with $\sigma_\lambda = (1/\sqrt{\tau_\lambda}) \sim \text{Unif}(0, 10)$. The mean of the normal distribution for $\beta_{0,g}$ is 0 because the intercept of the abundance model is confounded with the data augmentation parameter ψ . That is, ψ is providing the information on the total abundance which is equivalent information to the intercept in the abundance model (Royle et al. 2012). The effect of this group-specific random effect is to induce extra-Poisson variation in the group-specific abundance parameters N_g . It is convenient to use the normal distribution on the $\log(\lambda)$ scale here but a gamma noise term multiplying λ is equivalent to a negative binomial abundance model (Royle et al. 2012). For the group-specific intercept parameter α_0 we assumed then to be independent with normal prior

$$\alpha_{0,g} \sim \text{Normal}(\mu_p, \tau_p)$$

and flat priors on the hyperparameters μ_p and standard-deviation: $\mu_p \sim \text{Unif}(-10, 10)$, $\sigma_p = 1/\sqrt{\tau_p} \sim \text{Unif}(0, 10)$. We assumed a normal prior for $\alpha_{2,g}$ also, having parameters μ_{α_2} and standard deviation σ_{α_2} .

14.4.1 Results

There was a positive response of deer mouse population density to both thinning (β_2) and wildfire (β_3) (Table 1, Figure 1). There were also reasonably strong annual effects on density. Overall density of the species, across all groups, was estimated to be 0.00025 per m^2 , or 2.5 per ha. The conclusion that both thinning and fire had a positive effect on density of deer mice was consistent with the conclusion reached by Converse et al. (2006). We also found strong trap-happy responses (i.e., animals that had been trapped previously had a higher capture probability, see α_1 , in Table 1).

Table 14.1. Point estimates (posterior mode) and 95% credible intervals for parameters in the observation process portion of the model as well as the ecological process portion of the model, for the joint estimation and modeling of density of *Peromyscus* spp. on experimental units at the Jemez Mountains Study Area, New Mexico. See text for explanation of parameters.
Footnotes: (a) Only 2 fixed season effects are separately estimable. The third effect = $-1 * (\beta_1[\text{seas}1] + \beta_1[\text{seas}2])$. (b) Overall abundance is summed across all 24 groups, each with an implied area = 12.25 ha. (c) Overall density is reported as individuals/m².

Parameter	Estimate	95% Lower	95% Upper
Observation Process			
μ_p	-1.85	-2.19	-1.57
σ_p	0.55	0.37	0.88
α_1	0.22	0.05	0.41
μ_{α_2}	-1.28	-1.49	-1.07
σ_{α_2}	0.46	0.34	0.68
Ecological Process			
σ_λ	0.17	0.05	0.46
β [seas 1]	-0.60	-0.80	-0.43
β [seas 2]	-0.17	-0.33	-0.01
β [seas 3](a)	0.79	0.59	0.97
β [fire]	0.60	0.14	1.03
β [thin]	0.38	0.12	0.77
N (b)	747	708	797
Density (c)	2.54x10-4	2.41x10-4	2.71x10-4

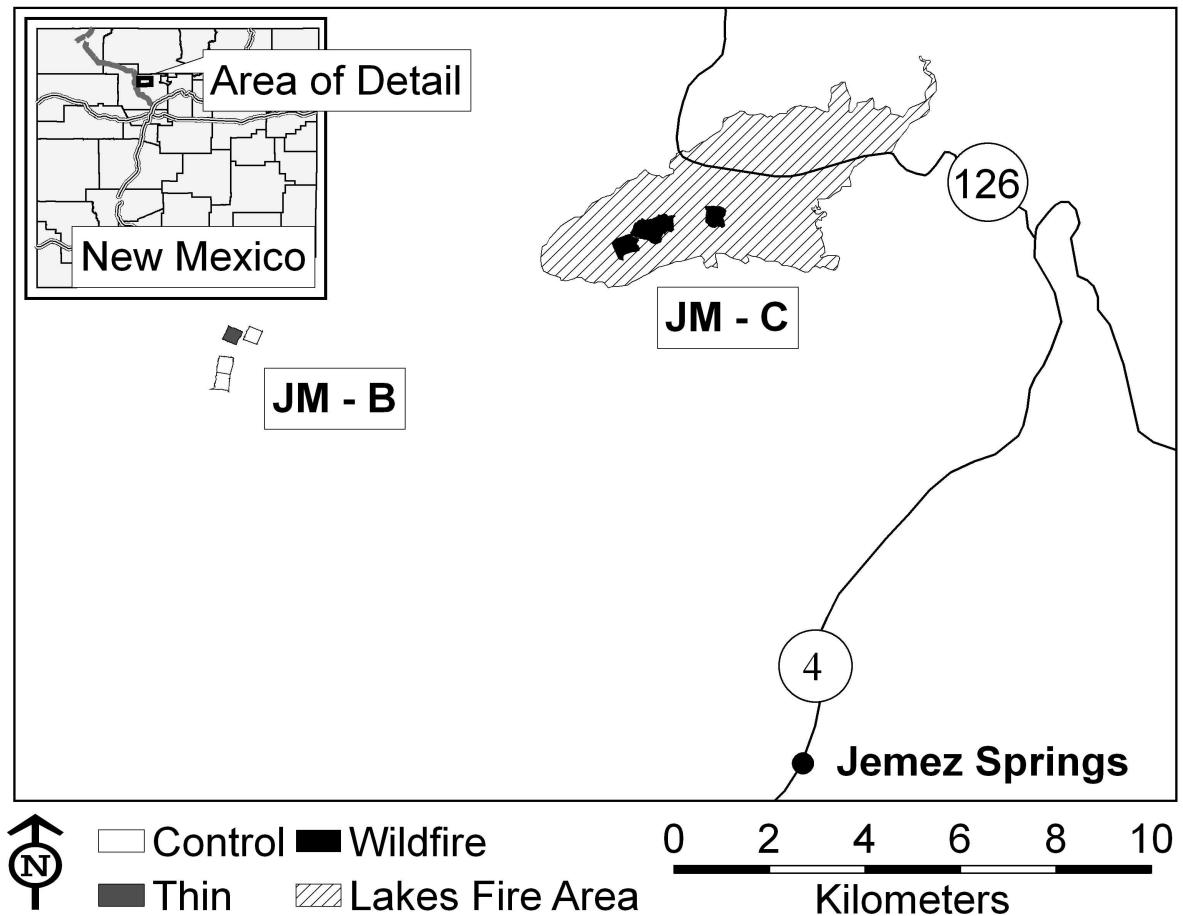


Figure 14.1. Central New Mexico Study Area from Converse et al. (2006).

14.5 TOPICS IN MULTI-SESSION MODELS

11744 The thing is this induces a slight bit of *dependence* among the counts but, for even a
 11745 smallish number of populations , and of moderate size, the dependence is imperceptible
 11746 and we think basically irrelevant from an inference standpoint.
 11747 Over-dispersion. Everyone loves it. Could have normal random effect or Gamma
 11748 distribution.

14.5.1 Temporal models

11750 The case here is we have $g = 1, \dots, G$ samples over time but individuals are coming
 11751 and going. We might capture some individuals over time but we ignore the individual
 11752 recaptures across primary periods. (See chapter 16). So instead of modeling the dynamics
 11753 at the individual level we just model net change in N_g .

14.5.2 Dependence – is it a problem?

11754 **XXX This should probably go in the body of the text or something XXXX**
 11755 In time – ignoring the dependence of N_g probably entails a little *loss* of efficiency
 11756 but should have no effect on anything. In space, there might be some individuals shared
 11757 by multiple groups and we don't think that should cause any bias or anything, even in
 11758 statement of uncertainty. So we view these models as pretty generally useful and relevant.
 11759 A few points worth discussing: If you have grids that are in relatively close proximity you
 11760 might want to build a model in which the total state-space is used in the model. i.e., form
 11761 the union of the state-spaces and model that. That will be more computationally tedious
 11762 but on the other hand it preserves the real landscape and any interactions that might be
 11763 affecting grids simultaneously.

11764 Conceptually we can apply models like this which assume N_g are independent even if
 11765 they're not... as long as we don't care about the underlying dynamics explicitly and also
 11766 possibly with some loss of efficiency.

11767 GROUPS, STRATA, POPULATIONS, ETC...????

14.6 MULTI-SESSION MODELS IN SECR

11769 We talked about this back in sec. xxx and also in sec. xxxxx....
 11770 The R package **secr** (Efford 2011) implements an estimator for “multiple sessions”
 11771 that could be applied to data from multiple trap arrays or other meaningfully grouped
 11772 data. The multi-session model in **secr** arises by an explicit Poisson assumption on N ,
 11773 but uses a classical likelihood analysis in which the parameters N_g are removed from the
 11774 likelihood by marginalizing the conditional-on- N likelihood over a Poisson prior. One
 11775 advantage of our Bayesian formulation based on data augmentation which enables direct
 11776 implementation in widely available software (WinBUGS, JAGS, OpenBUGS) is that it
 11777 is more versatile in terms of the model specification. For example, here we allowed for
 11778 multiple group-specific random effects in the detection model, which is not accommodated
 11779 in the **secr** package. As another (potential) example, we believe the model could be
 11780 extended to open populations (Gardner et al. 2010) without much difficulty.

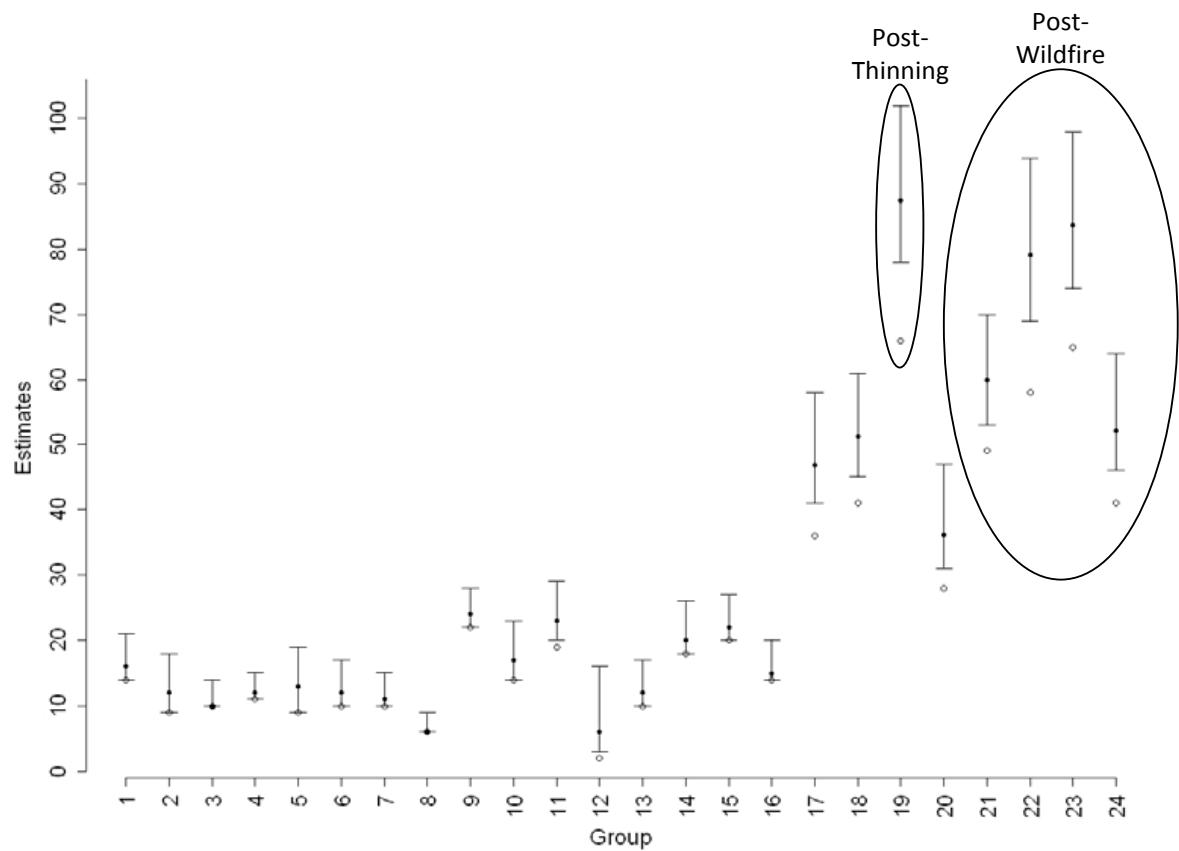


Figure 14.2. Abundance estimates for *Peromyscus* spp. per experimental unit (with area = 5.0625 ha) for each of 24 groups composed of 8 experimental units in year 1 (groups 1:8), and the same 8 experimental units in year 2 (groups 9:16) and year 3 (groups 17:24) at the Jemez Mountains Study Area, New Mexico. Point estimates (filled circles) are posterior modes, and error bars reflect 95% credible intervals. Also shown are the number of individuals captured per group (open circles).

11781 14.6.1 Ovenbird data in WinBUGS?

11782 Multi-catch observation model.... it is worth fitting this for sure.

11783 14.6.2 Converse data in secr?

11784 probably leave this out too much stuff..... but a simplified model would run a lot faster
11785 in secr I bet.

14.7 SUMMARY AND OUTLOOK

11786 The other context is temporally indexed data – multi-session or group-structured models
11787 are a simplified type of open model, one without explicit Markovian dynamics. The models
11788 are not incorrect per se, but just simpler, reduced versions of the more general Markovian
11789 models. We do cover the Markovian models in Chapt. 16

11790 SCR data are not always collected as single isolated studies but, rather, usually a
11791 number of replicate trap arrays are used. Often this is motivated by specific objectives,
11792 e.g., the trap arrays represent experimental replicates, and oftentimes just to obtain more
11793 valid estimates of density by obtaining a representative sample of space within some region.
11794 Thus there is a need to combine data from multiple arrays or sites in a single unified model
11795 that accommodates explicit sources of variation in density among sites. This is naturally
11796 accomplished by developing an explicit model for variation in N , e.g., a Poisson GLM or
11797 similar.

11798 In this paper we extended SCR models to allow for modeling variation in N with
11799 explicit assumptions on N . We adopt the data augmentation strategy from Converse and
11800 Royle (2012), and Royle et al. (2012) and extended this to the spatial capture-recapture
11801 observation model, and applied that model to data from a study of forest disturbance ef-
11802 fects on small mammal populations. The framework for combining multiple sites is general
11803 and will work for any kind of SCR observation model. We demonstrated a multi-catch
11804 (ovenbird), single-catch (microtus) and a bernoulli (simulated data) model..... Imple-
11805 mentation in a Bayesian framework allows for modeling of individual effects (and hence
11806 makes SCR possible) but also facilitates efficient modeling of nuisance variation via hier-
11807 archical structures (for example, on detection parameters, block effects, or time effects).
11808 However, certain types of models can be fitted in secr easily, and

11809 Previously people always did ad hoc shit when confronted with multi-session types of
11810 capture-recapture data. They would get Nhats and do regression on this. For example,
11811 our small-mammal trapping case study comes from Converse et al (2006), who used a
11812 3 step process to complete the analysis of these data: first a closed capture-recapture
11813 analysis to estimate abundance, second an analysis of mean maximum distance moved
11814 (Wilson and Anderson 1985) to allow conversion of abundance to density, and third a
11815 weighted regression analysis of the resulting density estimates. The weighted analysis was
11816 necessary to accomodate the non-zero sampling covariances resulting from the first 2 steps.
11817 The analysis shown herein is both more streamlined and also integrates the improvements
11818 that spatial capture-recapture methods bring to the estimation of capture-recapture data.
11819 In addition, the Bayesian analysis we present makes the use of hierarchical structures
11820 simple, such as the random effects for modeling variation in components of detection.
11821 Converse and Royle (2012) showed that the use of random effects for modeling variation

11822 in components of detection provides a good compromise between model complexity and
11823 parsimony, and can result in the lowest root mean square error in analyses of replicated
11824 capture-recapture data.

11825
11826

11827
11828

15

MODELS FOR SEARCH-ENCOUNTER DATA

11829 In this chapter we discuss models for search-encounter data. These are models that arise
11830 where you get actual location data of individuals not biased by trap locations but rather by
11831 searching space in some fashion. In most cases both detection probability and movement
11832 parameters are resolvable. i.e., models that preserve the these movement outcomes – the
11833 u_{it} variables. The models we differentiate here depend on a number of things related to
11834 data structure or protocol – basically whether or not we record the exact location and how
11835 we record it. All models have an underlying movement model which may be completely
11836 latent or not.

11837 How exactly are these different from models for data from fixed arrays? (1) sample
11838 units are either continuous space polygons or lines, not points; (2) we have location information
11839 that is not biased by trap locations (but is biased by the observation device
11840 somehow); (3) because we have direct observations of location that exist independent of
11841 traps we can often build an explicit model of space usage or an explicit movement model.

11842 A few variations of the models exist – a long sample path through a sample region
11843 where we note the locations of individuals seen along the way, *and their identity* (this is
11844 different from distance sampling int hat sense). Or we could search a region systematically
11845 and so forth. The canonical situation is Royle and Young (2008) which involved a plot
11846 search for lizards. They assumed the plot was uniformly searched which justified the
11847 assumption of constant p within the plot boundaries. The data set was ≥ 1 location
11848 observations for each of a sample of n individuals. The recent paper by Efford XXXX
11849 discussed likelihood analysis of similar models. In the jargon of `secr` such models are
11850 referred to as models for *polygon detectors*. An extension of this model was described by
11851 (Royle et al., 2011a).

11852 Search-encounter models also provide something of a bridge between the standard
11853 models for fixed trap arrays (e.g., Chapt. 5), and the models of (Chandler and Royle,
11854 In press) where no individual identity is present. The latter are search-encounter models
11855 where the movement process (and outcomes) are completely latent. Another type of model
11856 is SCR/DS – this is a SCR model with

15.1 SEARCH-ENCOUNTER SAMPLING DESIGNS

11857 For our purposes here we recognize 4 basic sampling designs, each of which might have
11858 variations due to modification of the basic sampling protocol. In later sections of this
11859 chapter we will do some examples but not of all of them.

11860 **Design 1: Fixed Search Path.** The ideal situation is where we have a continuous
11861 search-path or lines, or multiple such lines, in some region (Fig. XXXX 1 XXXXX). This
11862 is the type of problem described by Royle et al. (2011 MEE). We assume the path or
11863 lines are laid out a priori in some manner that is done independent of the activity centers
11864 of individuals and the collection of data does not affect the lines. Sometimes the lines
11865 are within well-defined polygons but the polygon boundaries are not meaningful in terms
11866 of the observation process. A number of variations of the data collection protocol are
11867 possible:

11868 Protocol (1a) has us just record the locations of individuals
11869 Protocol (1b) has us record location of individuals AND location on the transect where
11870 we observed the individual
11871 Protocol (1c) has us record neither of those things, instead we record the closest perpen-
11872 dicular distance. This is a typical distance sampling situation which produces exactly
11873 a DS type of a model (or a CR-DS model). We don't recommend recording closest
11874 perpendicular distance and we don't discuss these models too much here
11875 Protocol (1d) . In this case, observations are restricted to the line itself. We imagine
11876 that the line is evolving in response to search activity. It is not quite like the other
11877 ones so let's call it "ad hoc". In this case we use small bins as traps and the length of
11878 the line in each grid as a covariate. Unstructured survey data. Thompson et al. and
11879 Russell et al.

11880 **Design 2: Uniform search intensity.** In this case we have one or more well-defined
11881 sample areas (polygons), such as a quadrat or a transect, and we imagine that the area is
11882 uniformly searched so that $p = p_0$ is constant within the search area. Sampling produces
11883 locations of individuals within the well-defined boundaries of the sample area. The polygon
11884 boundaries defining the sample unit are important because it tells us that $p = 0$ by design
11885 outside of the boundary.

11886 Using the example from the Figure above, we could imagine that each quadrat was
11887 uniformly searched. The individual quadrat boundaries are irrelevant and we only need
11888 to be concerned about the "total" boundary of the composite polygon (the intersection of
11889 all little ones). That said for analysis in BUGS it is easier to work with square polygons.
11890 We show a simulation example here and we analyze it either using a bivariate normal
11891 movement model or else a 2-d random walk type of model. But we don't provide a real
11892 example as Royle and Dorazio 2008 did a reanalysis of the lizard data and see also Efford
11893 (XXXX).

11894 **Design 3: bad implementation of 1 or 2** We set up search polygons (e.g., the grid
11895 cells of above figure) and record locations of encountered individuals but we do not do a
11896 uniform search of quadrats and we forgot to record the GPS path. Analysis of this design:
11897 We imagine that we can assume a uniform search intensity here and maybe it won't be so

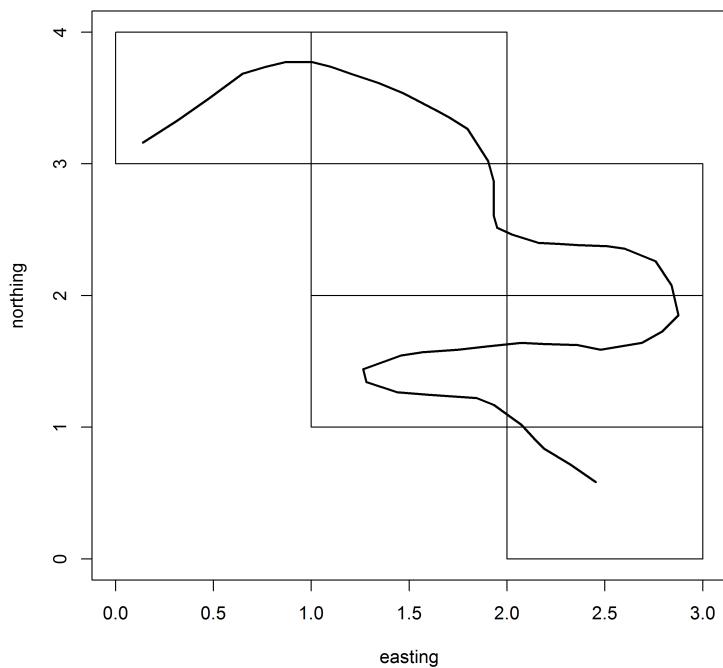


Figure 15.1. snake line.... showing design 1. more here.

11898 bad. We should do a simulation study of this somehow. I am working on methods to lay
 11899 down some standard sets of lines for simulating data, and then ignoring the lines in doing
 11900 an analysis. Alternative 2 for analysis: We could map each location to the CENTER of
 11901 the grid cell and pretend this is trap array (traps at center of each grid). This was the
 11902 idea of Kery et al. and some other papers.

11903 **Design 4: Really bad implementation of 1 or 2** In this case we screw up even
 11904 further and forget to record the locations of individuals within a bunch of quadrats. I
 11905 believe Richard has been thinking along these lines - using the underlying movement
 11906 model as a latent model. There are two variations of Design 4:

11907 Protocol (4a) - We imagine that you could have counts BY individual identity within
 11908 each quadrat. Not sure what analysis model this would be.

11909 Protocol (4b) - We don't have individual identities but just total counts. This is Chan-
 11910 dler and Royle (2012/13).

11911 The capricailie example: search of polygons – could be search encounter with uniform
 11912 search intensity but we ignored the polygon boundaries and just mapped each observation
 11913 to the center point. The fisher data: we had a GPS line but it was not really fixed , it
 11914 evolved as dogs searched around. Therefore as a practical matter the locations of samples
 11915 were all *on* the line. We therefore mapped to a center point of a grid. We make a grid of the
 11916 sampled area and we assume within each grid if a species is present then it is independent
 11917 actually if the grid is placed INDEPENDENT of the lines then its probably safe to
 11918 make some kind of independence assumption. Russell et al. – similar situation, they have
 11919 a search parth but not really independent.

11920 For the rest of this chapter, we will provide some model formulations for some cases,
 11921 provide code for simulating and analyzing the data, and some real examples but not for
 11922 every situation. A number of published examples have been given. The Royle et al. 2011
 11923 paper on the MHB. The Royle and Young 2008 (see also Marques et al. and Efford 2011).
 11924 We also have the Thompson et al. XXX and Russell et al. XXXX and Capricaillie paper
 11925 XXXXX.

11926 Possible examples to provide:

11927 Example 1: Analysis of the Swiss MHB survey using Design 1

11928 Example 1b: Lizard data. No need to analyze this as it was done in RD book. Mention
 11929 polygon detectors in secr.

11930 Example 2: Fisher data possibly - lion data or – or Capricaillie data?

15.2 A MODEL FOR SEARCH-ENCOUNTER DATA

11931 We cover the basic Design 1 here which also is relevant to Design 2 as a special case....
 11932 this comes from Royle et al. 2011.

11933 XXX t below has to be k XXXX

11934 Our approach is to parameterize a model for the encounter histories y_{ik} in terms of
 11935 \mathbf{u}_{ik} , the two-dimensional location of capture at the instant of sample, k . In contrast to
 11936 most of the models describe in this book, we develop models for encounter probability that
 11937 depend explicitly on the instantaneous location \mathbf{u}_{ik} , i.e., $p_{ik} \equiv p(\mathbf{u}_{ik}) = \Pr(y_{ik} = 1 | \mathbf{u}_{ik})$.
 11938 Note that \mathbf{u} is unobserved for the $y = 0$ observations and thus we cannot analyze the

11939 conditional-on- \mathbf{u} likelihood directly. Instead, we regard \mathbf{u} as random effects and assume
 11940 a distribution for them, which allows us to handle the problem of missing \mathbf{u}_{ik} values.

11941 To develop encounter probability models for this problem we cannot just use the
 11942 previous models because the “trap” is actually a line or collection of line segments (e.g.,
 11943 Fig XXXX). Intuitively, $\Pr(y_{ik} = 1 | \mathbf{u}_{ik})$ should increase as \mathbf{u}_{ik} comes “close” to the line
 11944 segments \mathbf{X} . It seems reasonable to express closeness by some distance metric $\|\mathbf{u}_{ik} - \mathbf{X}\| =$
 11945 $dist(\mathbf{u}_{ik}, \mathbf{X})$ and then assume

$$\text{logit}(p_{ik}) = \alpha_0 + \alpha_1 \|\mathbf{u}_{ik} - \mathbf{X}\|.$$

11946 For the case where \mathbf{X} describes a wandering line, some kind of average distance from \mathbf{u} to
 11947 the line might be reasonable; possible alternatives include the absolute minimum distance
 11948 or the mean over specific segments of the line (within some distance), etc. Because the
 11949 line \mathbf{X} is not a single point (like a camera trap) we have to somehow describe the total
 11950 encounter probability to the line. We adopt a similar idea to the hazard modeling idea
 11951 in survival analysis (also adopted in distance sampling by Hayes & Buckland (1983) and
 11952 Skaug & Schweder (1999) and, in the context of arrays of fixed traps by Borchers & Efford
 11953 (2008)). The individual is detected (analogous to mortality) if encountered at any point
 11954 along \mathbf{X} . Naturally, covariates are modeled as affecting the hazard rate and we think of
 11955 distance to the line as a covariate acting on the hazard. Let $h(\mathbf{u}_{ik}, \mathbf{x})$ be the hazard of
 11956 individual i being encountered by sampling at a point \mathbf{x} on occasion t . For example, one
 11957 possible model assumes, for all points $\mathbf{x} \in \mathbf{X}$,

$$\log(h(\mathbf{u}_{ik}, \mathbf{x})) = \alpha_0 + \alpha_1 * dist(\mathbf{u}_{ik}, \mathbf{x}). \quad (15.2.1)$$

11958 The total hazard to encounter anywhere along the survey path, for an individual located at
 11959 \mathbf{u}_{ik} , say $H(\mathbf{u}_{ik})$, is obtained by integrating over the surveyed line, which we will evaluate
 11960 numerically by a discrete sum where the hazard is evaluated at the set of points \mathbf{x}_j along
 11961 the surveyed path:

$$H(\mathbf{u}_{ik}) = \exp(\alpha_0) \left\{ \sum_{x_j} \exp(\alpha_1 * dist(\mathbf{u}_{ik}, \mathbf{x}_j)) \right\} \quad (15.2.2)$$

11962 where \mathbf{x}_j is the j^{th} row of \mathbf{X} defining the survey path as a collection of line segments
 11963 which can be arbitrarily dense, but should be regularly spaced. Then the probability of
 11964 encounter is

$$p_{it} \equiv p(\mathbf{u}_{it}) = 1 - \exp(-H(\mathbf{u}_{it})). \quad (15.2.3)$$

11965 This is a reasonably intuitive type of encounter probability model in that the probability
 11966 of encounter is large when an individual’s location \mathbf{u}_{it} is close to the line in the average
 11967 sense defined by Eq. (15.2.2), and vice versa. Note that p_{it} also depends on the sample
 11968 path \mathbf{X} , i.e., $p(\mathbf{u}_{it}, \mathbf{X})$ which we suppress in our notation because \mathbf{X} is fixed for any specific
 11969 analysis. We note that we don’t require all line segments are surveyed during each sample
 11970 period, as this simply affects the construction of the encounter probability p for each
 11971 sample. Thus, different line segments may be surveyed at different times, which results in
 11972 considerable flexibility in the design of a survey. Additional covariates could be included
 11973 in the hazard function. For example, in some situations observers might record weather
 11974 conditions along the route, time-of-day, effort or other covariates (Kéry *et al.* 2005).

11975 This formulation of total hazard and encounter probability assumes that encounter
 11976 at each point along the line, \mathbf{x}_j , is independent of each other point. Then, the event
 11977 that an individual is encountered *at all* is the complement of the event that it is not
 11978 encountered *anywhere* along the line (see also Hayes and Buckland 1983). In terms of
 11979 the survival/hazard analogy, the survival function is $S(\mathbf{u}_{ik}, \mathbf{x}_j) = \exp(-h(\mathbf{u}_{ik}, \mathbf{x}_j))$ and so
 11980 the probability that an individual “survives” all J points is $\prod_j \exp(-h(\mathbf{u}_{ik}, \mathbf{x}_j))$ and the
 11981 encounter probability is therefore the complement of this, which is precisely the expression
 11982 given by Eq. (15.2.3).

11983 Consider the case of a single survey point, i.e., $\mathbf{X} \equiv \mathbf{x}$, which we might think of as a
 11984 camera trap location. In this case note that Eq. (15.2.3) is equivalent to

$$\log(-\log(1 - p_{ik})) = \alpha_0 + \alpha_1 * dist(\mathbf{u}_{ik}, \mathbf{x})$$

11985 which is to say that distance is a covariate on detection that is linear on the complementary
 11986 log-log scale, which is similar to the “trap-specific” encounter probability of our Bernoulli
 11987 encounter probability model (see Chapt. 5). The difference is that, here, the relevant
 11988 distance is between the “trap” (i.e. the survey lines) and the individual’s present location,
 11989 \mathbf{u}_{ik} , which is observable. On the other hand, in the context of camera traps, the distance
 11990 is that between the trap and a latent variable, \mathbf{s}_i , representing an individual’s home range
 11991 or activity center which is not observed.

11992 15.2.1 Ecological process model

11993 We have so far described the model for the encounter data in a manner that is conditional
 11994 on the locations \mathbf{u}_{ik} , some of which are unobserved. That consideration alone justifies
 11995 the need for a 2nd level model – a “random effects” distribution – for the \mathbf{u}_{ik} variables.
 11996 In addition, biologically we expect that these variables should be correlated because they
 11997 correspond to repeated measures on the same individual. To develop such a model, we
 11998 adopt what is now customary in spatial capture-recapture problems – we assume that
 11999 individuals are characterized by a latent variable, \mathbf{s}_i , which represents a center of activity
 12000 or territory or simply “home range”. This leads to a natural model for the variables \mathbf{u}_{ik} .
 12001 In particular, we can now think of \mathbf{u}_{ik} as the outcomes of a *movement process*, conditional
 12002 on \mathbf{s}_i . Here we make use of the bivariate normal model:

$$\mathbf{u}_{ik} | \mathbf{s}_i \sim \text{Normal}(\mathbf{s}_i, \sigma^2 \mathbf{I}),$$

12003 where \mathbf{I} is the 2×2 identity matrix. This is a primitive model of individual movements
 12004 about their home range but, in most capture-recapture studies, we will only have one to
 12005 several observations on each individual and thus very limited ability to estimate complex
 12006 home range models. Therefore, we believe that the bivariate normal model will be sufficient
 12007 for most real-life spatial capture-recapture problems.

12008 We adopt our now customary assumption for the activity centers \mathbf{s} :

$$\mathbf{s}_i \sim \text{Unif}(\mathcal{S}); \quad i = 1, 2, \dots, N.$$

12009 The usual considerations apply in specifying the state-space \mathcal{S} – either choose a large
 12010 rectangle, or prescribe a habitat mask to restrict the potential locations of \mathbf{s} .

12011 15.2.2 Other stuff

12012 We have specified the model “conditional on N ”, where N is the total population of
 12013 individuals residing in the state-space \mathcal{S} . We need to account for the fact that N is
 12014 unknown which we do using our standard approach of data augmentation. As usual, under
 12015 data augmentation, the observations $y_{it} = 0$ correspond to an excess zero when $z_i = 0$ and
 12016 to a sampling zero when $z_i = 1$ – in the latter case an individual is indeed a member of the
 12017 population of size N . The known- N observation model is modified from $y_{it} \sim \text{Bern}(p_{it})$
 12018 (as above) to $y_{it} \sim \text{Bern}(w_i p_{it})$ and the latent variables w_i for $i = n + 1, \dots, M$ are
 12019 updated with the remaining model parameters in the MCMC algorithm (see below).

12020 Any model for encounter probability can be converted to a hazard model so that
 12021 encounter probability based on total hazard can be derived. Royle et al. 2011 considered
 12022 a bunch of other hazard models including that described previously

$$\log(h(\mathbf{u}_{it}, \mathbf{x})) = \alpha_0 + \alpha_1 * \text{dist}(\mathbf{u}_{it}, \mathbf{x}).$$

12023 which is usually called the Gompertz hazard function in survival analysis, and it is most
 12024 often written $h(t) = a \exp(b*t)$ in which case $\log(h(t)) = \log(a) + b*t$. Model 2 (squared-
 12025 distance) is a quadratic function of distance

$$\log(h(\mathbf{u}_{it}, \mathbf{x})) = \alpha_0 + \alpha_1 * \text{dist}(\mathbf{u}_{it}, \mathbf{x})^2.$$

12026 We’ve used this model quite a bit in the book, and it implies a bivariate normal hazard
 12027 rate model. Model 3 is from Borchers & Efford (2008):

$$h(\mathbf{u}_{it}, \mathbf{x}) = -\log(1 - \text{expit}(\alpha_0) \exp(\alpha_1 * \text{dist}(\mathbf{u}_{it}, \mathbf{x})^2))$$

12028 which produces a normal kernel model for *probability of detection* at the point level. i.e.,
 12029 $\Pr(y = 1) = 1 - \exp(-h) = h_0 \exp(\alpha_1 * \text{dist}(\mathbf{u}_{it}, \mathbf{x})^2)$ where $h_0 = \text{expit}(\alpha_0)$. Model 4 is

$$\log(h(\mathbf{u}_{it}, \mathbf{x})) = \alpha_0 + \alpha_1 * \log(\text{dist}(\mathbf{u}_{it}, \mathbf{x}))$$

12030 which is a Weibull hazard function.

15.3 EXAMPLES

12031 We just simulate data and fit it in JAGS – in the repo.

12032 Example: Simulator and WinBUGS code for this example [in repo]

```
> wbout
Inference for Bugs model at "model0.txt", fit using jags,
  3 chains, each with 5000 iterations (first 1000 discarded)
n.sims = 12000 iterations saved
      mu.vect sd.vect    2.5%     25%     50%     75%   97.5% Rhat n.eff
12038 N         94.448   5.237  81.000  92.000  96.000  98.000 100.000 1.006   520
12039 beta0     -0.539   0.743 -1.714 -1.072 -0.637 -0.102   1.258 1.042    54
12040 beta1    -11.943   2.196 -17.229 -13.236 -11.665 -10.375 -8.378 1.035    62
12041 psi        0.936   0.056   0.792   0.908   0.951   0.979   0.998 1.005   650
12042 sigma      0.340   0.043   0.268   0.309   0.337   0.366   0.436 1.004   820
```

12043 deviance 206.987 25.474 160.405 189.069 205.779 224.080 259.491 1.017 130
 12044
 12045 For each parameter, n.eff is a crude measure of effective sample size,
 12046 and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
 12047
 12048 DIC info (using the rule, pD = var(deviance)/2)
 12049 pD = 319.4 and DIC = 526.3
 12050 DIC is an estimate of expected predictive error (lower deviance is better).

12051 **15.3.1 Hard plot boundaries**

12052 1.3. Hard quadrat boundaries: Quadrat boundaries might be relevant or might not be.
 12053 If they are then, for Bayesian analysis, a value of u outside the boundary has p forced to
 12054 0, its as simple as that. In general, we define $p[I,t] = p[I,t]*I(u \text{ in } X)$. We see how this
 12055 relates to the uniform search intensity model. $P[I,t] = p_0$ then defines precisely the model
 12056 of Royle and Young (2008).

12057 **Hard plot boundaries** – The previous development assumed that encounters can be
 12058 made anywhere in space but that the encounter probability decreases with distance from
 12059 the survey path. In practice, as in the MHB, we might delineate a plot which restricts
 12060 where individuals might be observed (as in the situation considered by Royle & Young
 12061 (2008)). For such cases we truncate the encounter probability function such as

$$p(\mathbf{u}_{it}) = (1 - \exp(-H(\mathbf{u}_{it})))I(\mathbf{u}_{it} \in \mathcal{X})$$

12062 where \mathcal{X} is the surveyed polygon and the indicator function $I(\mathbf{u}_{it} \in \mathcal{X}) = 1$ if $\mathbf{u}_{it} \in \mathcal{X}$ and
 12063 0 otherwise. That is, the probability of encounter is identically 0 if an individual is located
 12064 *outside* the plot at sample period t . Given this modified encounter probability function,
 12065 it is clear that the model is a modified form of Royle & Young (2008) where their model
 12066 – “uniform search intensity” – replaces the above expression with

$$p(\mathbf{u}_{it}) = p_0 I(\mathbf{u}_{it} \in \mathcal{X})$$

12067 Analysis of lizard data from Royle and Young 2008

12068 **Multiple survey plots** – It is common in wildlife surveys to have multiple spatial
 12069 sample units which need to be integrated into a single model. It is convenient if the
 12070 population sizes for each plot are independent. In the case of the MHB data, the closest
 12071 two plots were 10 km apart and, for this species, it is reasonable to assume independence.
 12072 Moreover, the MHB plots represent (approximately) a random sample and thus indepen-
 12073 dence is probably justified from a design-based perspective. With multiple plots, it is
 12074 convenient computationally to organize the plots in some modified coordinate system that
 12075 keeps them far enough apart so that individual movement outcomes cannot be located in
 12076 multiple plots. This enables an implementation by data augmentation based on a single
 12077 augmented data set. To construct the point process state-space, the 7 plots were embed-
 12078 ded into a 30.8 km rectangular state-space having a minimum of 0.6 km buffer, which
 12079 we judged to be sufficient given the estimate of σ (see below) so that individuals cannot
 12080 appear in > 1 plot during the MCMC simulation (i.e., 0.6 is large relative to the estimate
 12081 of σ).

15.3.2 Analysis of other protocols

Analysis of 1b is a distance-sampling like model but with an additional hierarchical structure that describes the individual location scatter about the home range center. This is precisely a type of DS with measurement error. Analysis of 1c is a similar idea except it represents an explicit model misspecification since one is approximating the observation process by the nearest perpendicular to the line. Analysis of 1d is the “unstructured survey data” like from Thompson et al. or Russell et al. Note also that the capcrap paper is a version of this - grids or polygons were sampled but no information on the search path is available. This could be a Design 3 problem but that is excess computation I think.

Protocol (1b) has us record location of individuals AND location on the transect where we observed the individual. This is an easier problem I think, but you have to account for “not seen” prior to x_0 so maybe some kind of a cumulative hazard model or something.

Protocol (1c) has us record neither of those things, instead we record the closest perpendicular distance. This is a typical distance sampling situation which produces exactly a DS type of a model (or a CR-DS model). We don’t recommend recording closest perpendicular distance and we don’t discuss these models too much here

Protocol (1d). In this case, observations are restricted to the line itself. We imagine that the line is evolving in response to search activity. It is not quite like the other ones so let’s call it “ad hoc”. In this case we use small bins as traps and the length of the line in each grid as a covariate. Thompson et al. and Russell et al. Simulation results

15.4 DESIGN 3: AD HOC IMPLEMENTATION OF DESIGN 1.

We don’t do anything new in terms of modeling here but we look at how bad do we do if we don’t have the search path and use the USI model? We consider 4 cases: Case 1 and 2: regular searching of low and high intensity. E.g., for a 1 unit block, then we can have a sinusoid track through each block of length 1.5 or 2 and then 4 or 5 km. For case 3 and 4 we use heterogeneity in search intensity.

15.5 CAPRICAILLIE CRAP**15.6 DESIGN 4 – NO LOCATION INFO**

We imagine a series of models for situations where we forget altogether to record location information within the sample unit. We further assume the design was such that the sample units represent contiguous quadrats or at least close enough together so that individuals may be counted in multiple units. The idea here is that by being exposed to multiple units, there is a spatial dependence induced and this spatial dependence provides a little bit more of information about model parameters.

We have two specific cases here: Imagine we have a bunch of quadrats or segments that are contiguous and we do the surveys like above and record counts PER individual but no other sampling information. Not sure what to do about this. The other case is that we don’t record individual ID at all – instead we just have total count frequencies in each plot. This model is precisely the one considered by (Chandler and Royle, In press) and this is the focus of Chapt. 18.

Comments on inference for the first situation?

15.7 SUMMARY AND OUTLOOK

₁₂₁₂₁ Searching space for scat is , we imagine, the future of all animal sampling.
₁₂₁₂₂ SCR/DS?

12123
12124

12125

16

OPEN POPULATION MODELS

16.1 INTRODUCTION

12126 All of the previous chapters focused on closed population models for estimating density
12127 and for inference about spatial variation in density. However, a thorough understanding
12128 of population dynamics requires information about both spatial and temporal variation
12129 in population density and demographic parameters. In this chapter, we develop a frame-
12130 work for inference about the processes governing spatial and temporal dynamics, namely
12131 survival, recruitment, and movement (migration, dispersal, etc...). The ability to estimate
12132 these parameters is critical to both basic and applied ecological research. For example,
12133 testing hypotheses about life history trade-offs requires accurate estimates of both sur-
12134 vival and fecundity ([citations](#)). Inference about density-dependent population regulation,
12135 which has fascinated theoretical ecologists for well over a century, is likewise best accom-
12136 plished by studying the factors affecting survival and fecundity, rather than the more
12137 common approach of modeling time series data (Nichols et al., 2000). Modeling vital
12138 rates is just as important for applied ecologists and conservation biologists, because a
12139 mechanistic understanding of population decline requires it. Furthermore, if we know how
12140 environmental variables affect demographic parameters, we can make predictions about
12141 population changes under different future scenarios. We can also assess the sensitivity of
12142 parameters such as population growth rate to variation in survival or fecundity. Although
12143 matrix population models are often used for these purposes (Caswell, 1989; Sæther and
12144 Bakke, 2000), the same objectives can be accomplished by computing posterior predictive
12145 distributions as part of the MCMC algorithm.

12146 For the first time, we can fully integrate the movement of individuals onto and off of the
12147 trap array with their encounter histories to simultaneously estimate density, survival, and
12148 recruitment in a spatial model. For many species, such as those that are rare or not often
12149 observed by researchers, this allows us to make inference about survival and recruitment
12150 without having to physically capture individuals. Additionally, another reason extending
12151 our SCR models to open populations arises purely from a sampling perspective. We often
12152 need longer time periods to sample rare or elusive species to ensure that enough captures
12153 and recaptures are produced. This extended time frame can quickly lead to violations

in the assumption of population closure. For example, the European wildcat study that was presented in chapter XXX insert ref XXX was conducted over a year long period. While the researchers in that study used a closed population, they did model variation in detection as a function of time. Another approach would have been to use an open population model (the spatial capture recapture open models had not been developed at the time of the wildcat study, so we'll forgive the authors for not having used this more appropriate model).

The modeling framework we will develop in this chapter is based on a formulation of Cormack-Jolly-Seber (CJS) and Jolly-Seber (JS) type models (Cormack, 1964; Jolly, 1965; Seber, 1965) that are amenable to modeling individual effects, including individual covariates. There is a long history of use of these models in fisheries, wildlife, and ecology studies (Pollock et al., 1990; Lebreton et al., 1992; Pradel, 1996; Williams et al., 2002; Schwarz and Arnason, 2005; Gimenez et al., 2007). Additionally, there have been many modifications and developments of the CJS and JS models including dealing with transients, multi-state, and spatially implicit models.

16.1.1 Overview of Population Dynamics

The most basic formulation of models for population growth stem from an idea originally used in accounting, the balance sheet. In this case, we can think of population size as a function of credits (i.e., births and immigrants) and debits (i.e., deaths and emigrants). We can then set up the population at time $t + 1$ as a function of these four components:

$$N(t + 1) = N(t) + B(t) + I(t) - D(t) - E(t)$$

where $N(t)$ is the population size at time t , $B(t)$ and $I(t)$ are the credits (additions) from births and immigrants at time t , and $D(t)$ and $E(t)$ are the debits (losses) due to deaths and emigration. This balance equation model is known as the “BIDE model”. We can easily derive a simple population growth model under density independence, by assuming no immigration or emigration.

$$N(t + 1) = N(t) + N(t)r(t)$$

where $r(t) = b(t) - d(t)$. Here, $b(t)$ and $d(t)$ are the per-capita birth and death rates and thus $r(t)$ is the per-capita growth rate. Density-dependent, age structured, stochastic effects on growth, spatially structured, and competition models (e.g., Lotka-Volterra) all are basic derivations of the BIDE model.

In closed population models, we focus on estimating $N(t)$, but in open population models we are interested in the dynamics that arise between years or seasons and thus we focus not only on $N(t)$ but on these so-called “credits” and “debits” that drive the population changes. If we take the basic parameters in the BIDE model and reconceptualize them, we can relate these to the commonly used parameters in JS and CJS models, described in more detail below. For example, survival ($\phi(t)$) is defined as the probability of an individual surviving from time t to $t + 1$, and often we call this ‘apparent’ survival because deaths and emigration cannot be separated. Mortality, the probability of dying from time t to $t + 1$, is $1 - \phi(t)$. Recruitment (γ) is the probability of a new individual entering the population between t to $t + 1$, which includes those both those born into the population and immigrants.

12194 16.1.2 Animal movement related to population demography

12195 Density may influence demographic parameters such as survival rates, population growth,
12196 etc., it is also likely that movement of individuals can influence these parameters. For
12197 example, we know that movement of transients will affect our estimates of survival, causing
12198 us to typically refer to estimates as “apparent survival”. This is because an animal that
12199 appears in the population for a short period of time and then leaves is going to appear
12200 as though it has died. Due to this problem, there has been a significant amount of
12201 work developing models to deal with transients in both closed and open capture-recapture
12202 models **NONE OF THESE ARE IN BIB FILE: kendalletal:1997, pradeletal:1997, hinesetal:2003, claveletal:2008**. Because we estimate movement within the SCR framework, we
12203 can better understand the impact of animals moving onto and off of the trap array and
12204 hence we can improve our estimates of survival by combining the traditional CJS and JS
12205 models with the SCR model.

12206 But what if movement and space usage of individuals directly influences the survival
12207 rates or recruitment? It is generally accepted that population structure (i.e., age, stage, or
12208 size distribution) can affect both population size and growth over time. We also know that
12209 how animals associate themselves in space can directly influence the age or stage structure
12210 of a population – this can be behavioral, habitat related, or some combination of factors.
12211 For example, if habitat is limited, some younger members of the population might have
12212 trouble finding and/or defending a territory. Ultimately, this may lower survival for a
12213 certain age class in the population directly impacting the population structure. Dispersal
12214 can also affect population structure. In many animal populations, dispersal is linked with
12215 reproduction and population regulation. Thus, movement including spatial arrangement of
12216 activity centers and dispersal are key components to population dynamics. We start here
12217 by showing how to extend the SCR models to open populations, but this chapter opens
12218 the door for how we would go about incorporating space usage into models of demographic
12219 dynamics. For example, we could incorporate space and movement into age-dependent
12220 multistate capture-recapture models to address the impact of dispersal on recruitment or
12221 survival.

12223 16.1.3 Basic assumptions of JS and CJS models

12224 Before extending the classic open models to our SCR framework, let's first look at the
12225 basic assumptions of both models. No tag (or mark) loss is assumed in both models.
12226 If a marked animal loses its tag or mark, then that animal cannot be recaptured and
12227 this could appear as though the animal has died. Hence, to maintain unbiased estimates
12228 of survival, no tag or mark loss is important. Additionally, capture and release should
12229 be instantaneous (or as close as possible), otherwise the time interval between capture
12230 occasions could differ for individuals and that would result in individual heterogeneity of
12231 survival. Individuals must also be recorded accurately.

12232 In the standard CJS models, it is also assumed that all emigration from the study
12233 area is permanent and that capture and survival probabilities are constant within each
12234 sample occasion and group. A group can be created based on sex, age, area, etc. In
12235 the CJS model, we condition on the captured individuals, and therefore we estimate
12236 only the probability of recapture and the survival rates. Here, survival is considered the
12237 “apparent” survival because emigration and mortality are confounded within the model,

thus apparent survival is always estimated lower than true survival when emigration is not zero. In the JS version of the model, we do not condition on marked individuals. Thus we can estimate survival like we do in the CJS, but now we can also model recruitment (new individuals coming into the population) and the total abundance/density of the population. Estimating more parameters does require a few more assumptions including that all individuals in the population have the same probability of capture. Under a “robust design” (Pollock, 1982), which we will demonstrate in this chapter, we can estimate heterogeneity in capture probabilities.

16.2 TRADITIONAL JOLLY-SEBER MODELS

There are a number of ways that researchers have formulated the JS model and while all are slightly different, the resulting estimates of abundance and the driving parameters such as survival and some form of recruitment should be the same. The most commonly used formulations are the Link-Barker, Pradel-recruitment, Burham JS, and the Pradel-l models. In all of these models, we are interested in recruitment, or how new individuals arrive into the population. Therefore one of the main differences between the various models is how new entrants into the population are parameterized.

Pollock (1982) created the robust design in order to allow for heterogeneity in capture probability under the JS model. The basic idea is that there are primary occasions (e.g., years, seasons) and we allow the population to be “open” between the primary occasions. This means that individuals can enter and leave the population (i.e., births, deaths, immigration, emigration can occur) between the primary occasions. However, within a primary occasion, the population is assumed to be closed to these processes. The standard JS model does not allow for variation in detection probability between individuals or within a primary occasion because only one sample is collected per primary period. However, when multiple samples are taken within a primary occasion (we call these “secondary occasions”), then variation in detection probability can be modeled and thus our estimates of N can be improved. To that extent, we can envision the data as arising from repeated sampling over seasons or years (or *primary* periods) within which one or more samples (e.g., nights) might be taken (*secondary* periods). Fig. 16.1 demonstrates the sampling process graphically. Comparing this with all of our previous work, the sample occasions (e.g., trap nights, weeks, etc...) described in the closed population chapters are called *secondary* sampling occasions.

Based on the robust design, we can easily create a non-spatial JS model. We define y_{ikt} as the encounter history for individual i at secondary occasion k during primary occasion t . If we have a Bernoulli encounter process then we can describe the observation model, specified conditional on $z(i, t)$, as:

$$y_{ikt}|z(i, t) \sim \text{Bernoulli}(pz(i, t)).$$

Thus, if individual i is alive at time t ($z(i, t) = 1$), then the observations are Bernoulli with detection probability p as before. Conversely, if the individual is not alive ($z(i, t) = 0$), then the observations must be fixed zeros with probability 1.

Survival and recruitment in the open population are manifest in a model for the latent state variables $z(i, t)$ describing individual mortality and recruitment events. An important aspect of the hierarchical formulation of the model that we adopt here is that the

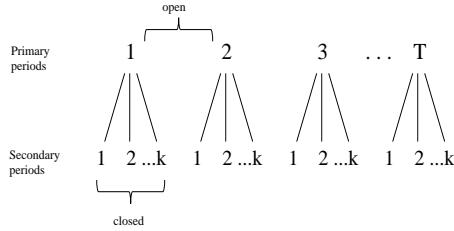


Figure 16.1. Schematic of the robust design with T primary sampling periods and K secondary periods. The populations are considered open between primary periods and closed within the secondary

model for the state variables is described conditional on the total number of individuals ever alive during the study (a parameter which we label N) based on T periods, as in Schwarz and Arnason (1996). Data augmentation induces a special interpretation on the latent state variables $z(i, t)$. In particular, “not alive” includes individuals that have died, or individuals that have not yet been recruited. Using this formulation simplifies the state model and also allows it to be implemented directly in the **JAGS** software Royle and Dorazio (2008). For example, considering the case $T = 2$, the state model is composed of the following two components: First the initial state is described by:

$$z(i, 1) \sim \text{Bern}(\psi)$$

and then a model describing the transition of individual states from $t = 1$ to $t = 2$:

$$z(i, 2) \sim \text{Bern}(\phi z(i, 1) + \gamma(1 - z(i, 1))).$$

If $z(i, 1) = 1$, then the individual may survive to time $t = 2$ with probability ϕ whereas, if $z(i, 1) = 0$, then the “pseudo-individual” may be recruited with probability γ .

We can then generalize this model for $T > 2$ time periods and allow survival and recruitment to be time dependent. Initialize the model for time $T = 1$ as we have done above and then the model describing the transition of individual states from t to $t + 1$ is:

$$z(i, t + 1) \sim \text{Bern}(\phi_t z(i, t) + \gamma_t(1 - z(i, t))).$$

This parameterization then results in $T - 1$ survival and recruitment parameters. The main difference here from the CJS model, described below, is that we include recruitment and are interested in estimating N for each t . Since this state model described above is conditional-on- N , we must deal with the fact that N is unknown, which is done through data augmentation similar to how we used it in the closed population models.

12298 16.2.1 Data Augmentation for the Jolly-Seber Model

12299 The fundamental challenge in carrying out a Bayesian analysis of this model is that the
 12300 parameter N (the total number of individuals alive during the study) is not known. We
 12301 have discussed and demonstrated data augmentation in many previous chapters; however,
 12302 with the open population model, we have to take care that two issues are addressed:
 12303 (1) the data augmentation is large enough to accommodate all potential individuals alive
 12304 in the population during the entire study and (2) that individuals cannot die and then
 12305 re-enter the population. To begin, let's consider the role of γ in the model.

12306 Data augmentation formally reparameterizes the model, replacing N , the number of
 12307 individuals ever alive with the parameter ψ which is interpretable as the population size
 12308 expressed as a fraction of M . That is, the expected value of N under the model is equal
 12309 to ψM . As a result of this reparameterization, the recruitment parameters γ_t are also
 12310 relative to the number of “available recruits” on the data augmented list of size M , and
 12311 not directly related to the population size. This is easily resolved by deriving N_t , and
 12312 R_t , the population size and number of recruits in year t , as a function of the latent state
 12313 variables $z(i, t)$. In particular, the total number of individuals alive at time t is

$$N_t = \sum_{i=1}^M z(i, t)$$

12314 and the number of recruits is

$$R_t = \sum_{i=1}^M (1 - z(i, t-1)) z(i, t)$$

12315 which is the number of individuals *not* alive at time $t-1$ but alive at time t .

12316 In the case of just two primary periods, this process is straightforward. When the
 12317 number of primary sample occasions is greater than 2, we must formulate the model for
 12318 recruitment by introducing another latent variable. We do this in order to ensure that
 12319 an individual can only be recruited once into the population. Here, this formulation of
 12320 the model uses a set of latent indicator variables $r(i, t)$ which describe the time interval
 12321 $(t-1, t)$ at which individual i is recruited into the population. Let $r(i, t) = 1$ if individual
 12322 i is recruited in time interval $(t-1, t)$ otherwise $r(i, t) = 0$. To construct the recruitment
 12323 process we make use of the standard conditional binomial construction of a removal process
 12324 (Royle and Dorazio 2008). The initial state is given by:

$$r(i, 1) \sim \text{Bin}(1, \gamma_1)$$

12325 for $i = 1, 2, \dots, N$. Then, for $t > 1$

$$r(i, t) | r(i, t-1) \dots r(i, 1) \sim \text{Bin}\left((1 - \sum_{\tau=1}^{t-1} r(i, \tau)) \times \gamma_t, 1\right)$$

12326 Each recruitment variable is conditional on whether the individual was previously
 12327 recruited and this construction forces the recruitment variable after initial recruitment to
 12328 be degenerate (have a sample size of 0). Then, we can describe the state variables $z(i, t)$
 12329 by a 1st order Markov process. For $t = 1$, the initial states are fixed:

$$z(i, 1) \equiv r(i, 1)$$

12330 and, for subsequent states, we have

$$z(i, t) | z(i, t - 1), r(i, t) \sim \text{Bern}(\phi_t z(i, t - 1)) + r(i, t).$$

12331 Thus, if an individual is in the population at time t (i.e., $z(i, t) = 1$), then that individual's
 12332 status at time $t+1$ is the outcome of a Bernoulli random variable with parameter (survival
 12333 probability) ϕ_t . If the individual, however, is not in the population at time t (i.e., $z(i, t) =$
 12334 0), then the outcome is a Bernoulli random variable with probability γ_t , a parameter that
 12335 is related to *per capita* recruitment. We carry out this process in **JAGS** by using the **sum()**
 12336 and **step()** functions together to ascertain if a particular individual i was ever previously
 12337 alive. Individuals that were ever previously alive are no longer eligible to be "recruited"
 12338 into the population. The implementation of this model in **JAGS** is shown in panel 16.1.

```
model{

psi ~ dunif(0,1)
phi ~ dunif(0,1)
p.mean ~ dunif(0,1)

for(t in 1:5){
  N[t] <- sum(z[1:M,t])
  gamma[t] ~ dunif(0,1)
}

for(i in 1:M){
  z[i,1] ~ dbern(psi)
  cp[i,1] <- z[i,1]*p.mean
  Y[i,1] ~ dbinom(cp[i,1], K)
  a[i,1] <- (1-z[i,1])

  for(t in 2:5){
    a1[i,t] <- sum(z[i, 1:t])
    a[i,t] <- 1-step(a1[i,t] - 1)

    mu[i,t]<- (phi*z[i,t-1]) + (gamma[t]*a[i,t-1])
    z[i,t] ~ dbern(mu[i,t])
    cp[i,t] <- z[i,t]*p.mean
    Y[i,t] ~ dbinom(cp[i,t], K)
  }
}
}
```

Panel 16.1: **JAGS** model specification for the non-spatial JS model.

12339 16.2.2 Mist-netting example

12340 We now return to the ovenbird data collected during a mist-netting study, and initially
 12341 presented in Chapt. 9. These data are available in the `secr` package (see, Efford et al.
 12342 (2004); Borchers and Efford (2008)). To refresh your memory: 44 mist nets spaced 30 m
 12343 apart on the perimeter of a 600-m x 100-m rectangle (see Fig. XXXX) were operated on
 12344 9 or 10 non-consecutive days in late May and June for 5 years from 2005-2009.

12345 In Chapt. 9, we dealt with this dataset as a type of “multi-season” model where
 12346 abundance in each year, N_t , was estimated separately. This is the simplest approach for
 12347 modeling data collected over multiple years, but it does not allow for inference about
 12348 demographic processes, as does the JS model.

12349 The first issue at hand is that each line in our 3-D encounter history array of data
 12350 must correspond to a single individual. Previously, we were not interested in individual
 12351 identity across years so this was not of concern; however, we need to maintain the order of
 12352 individuals across years in order to estimate the survival and recruitment of the individual
 12353 into the population. We organize the data set so that each row in our array represents
 12354 just one individual across all primary periods. For the ovenbird dataset, we can organize
 12355 the data by creating a master list of all individuals captured during the entire study.
 12356 From this list, we can assign each individual a unique row in our dataset (in the following
 12357 **R** commands, we do this by using the `unique()` function on the row names for each year of
 12358 our 3-D array and use `pmatch()` to associate the data to the correct column). Additionally,
 12359 in Chapt. 9 we carried out data augmentation for each year separately; however, we must
 12360 consider for example that individuals captured in year t could have been alive in year
 12361 $t - 1$. Our data augmentation must be large enough to include individuals alive during
 12362 any of the time periods and to account for that, we set $M=200$. For this example, we
 12363 hold survival constant but allow recruitment to be time dependent (since γ is essentially
 12364 a function of the data augmentation process as described above, it does not make sense
 12365 to hold recruitment constant and we therefore make it time specific).

```
12366 library("secr")
12367 library(scrbook)
12368 data(ovenbird)
12369
12370 X<-traps<-traps(ovenCH)
12371 xlim<-c(min(X[[1]][,1])-150,max(X[[1]][,1])+150)
12372 ylim<-c(min(X[[1]][,2])-150,max(X[[1]][,2])+150)
12373 ntraps<- nrow(traps[[1]])
12374 Y<-ovenCH
12375 K<-10
12376 M<-200 # do constant data augmentation to all years
12377 Sst<-cbind(runif(M,xlim[1],xlim[2]),runif(M,ylim[1],ylim[2]))
12378 Sst<-array(Sst,dim=c(M,2,5))
12379
12380
12381 hold<- unique(c(unlist(dimnames(Y[[1]])[1]), unlist(dimnames(Y[[2]])[1]),
12382      unlist(dimnames(Y[[3]])[1]), unlist(dimnames(Y[[4]])[1]),
12383      unlist(dimnames(Y[[5]])[1])))
12384
```

```

12384
12385 Yarr<-array(ntraps+1,dim=c(M,K,5))
12386 for(i in 1:5){
12387   tmp<-Y[[i]]
12388   tmp[tmp<0]<-tmp[tmp<0]*(-1) ## one guy died, we ignore that here
12389   tmp[tmp==0]<-ntraps+1
12390   nind<-nrow(tmp)
12391   nrep<-ncol(tmp)
12392   tmp2<-matrix(ntraps+1,nrow=M,ncol=10) # pad last col with NA for year 1
12393   tmp2[pmatch(unlist(dimnames(Y[[i]])) [1]), , hold], 1:nrep]<-tmp
12394   Stmp<-Sst[, , i]
12395   Stmp[pmatch(unlist(dimnames(Y[[i]])) [1]), , hold], 1:2]<-
12396     spiderplot(tmp2[pmatch(unlist(dimnames(Y[[i]])) [1]), , hold], 1:nrep),
12397     as.matrix(X[[i]]))$avg.s ##$
12398   Sst[, , i]<-Stmp
12399   Yarr[, , i]<-tmp2
12400 }

12401 Yarr[Yarr < 45] <- 1
12402 Yarr[Yarr == 45] <- 0
12403 Ybin=matrix(NA, M, 5)
12404 for(t in 1:5){
12405   Ybin[, t] <- rowSums(Yarr[, , t])
12406 }
12407
12408 zst<-c(rep(1,M/2),rep(0,M/2))
12409 zst<-cbind(zst,zst,zst,zst,zst)
12410
12411
12412 inits <- function(){list (z=zst,sigma=rnorm(1,25,100), gamma=rnorm(5,0,1)) }
12413 parameters <- c("psi","N","phi", "p.mean", "gamma")
12414 data <- list (K=10,Y=Ybin,M=M)
12415
12416 library("rjags")
12417 out1 <- jags.model("modelNSJS.txt", data, inits, n.chains=3, n.adapt=500)
12418 out2NSJS <- coda.samples(out1,parameters,n.iter=20000)

```

12419 We find in this non-spatial JS model that N is estimated to be between about 22 and
12420 33 for each of the 5 years (see Table 16.1 for results). The posterior mean for detection
12421 ($p.\text{mean}$ in the model) was 0.14, it is not included in the table because the spatial models
12422 do not have a parameter that directly corresponds to this one.

12423 16.2.3 Shortcomings of the traditional JS models

12424 As we have previously discussed, one of the biggest shortcomings of the non-spatial JS
12425 model is that we estimate N but have no explicit spatial reference area for that value. As
12426 you see in Table 16.1, the density estimate from the non-spatial JS model is listed as NA.
12427 This is because, again, the effective sampling area is unknown leaving us to determine

12428 that area in an ad hoc manner. Not making use of the spatial information in the data
 12429 makes the estimation of density a non-formal process. As we saw in the closed models,
 12430 the explicit incorporation of spatial information will allow us to provide a robust estimate
 12431 of density. This improvement should also carry through in our estimation of other demo-
 12432 graphic parameters such as survival and recruitment. Also, while we can potentially model
 12433 the relationship between density and the demographic parameters we are interested in by
 12434 using standard JS models, we can make no inference regarding the spatial arrangement of
 12435 individuals in the landscape nor the direct impact of movement.

16.3 SPATIAL JOLLY-SEBER MODELS

12436 To parameterize the spatial JS models, we essentially follow all of the same steps as the
 12437 non-spatial model but we also include the trap location information into our detection
 12438 function. Essentially, we are using the closed population SCR model to estimate the
 12439 detection parameters and initial population size, and the open component is carried out
 12440 in the process of how we model the transition of $z(i, t)$ to $z(i, t + 1)$ which is the same
 12441 as in the non-spatial JS model. To do so, we describe the Bernoulli observation model,
 12442 specified conditional on $z(i, t)$, as we have done throughout the book:

$$y_{ijk} | z(i, t) \sim \text{Bernoulli}(p_{ijk} z(i, t)).$$

12443 with

$$p_{ijk} = p_0 * \exp(-\alpha_1 d_{ij}^2) \quad (16.3.1)$$

12444 where $d_{ij} = \|s_i - x_j\|$, the distance between s_i and x_j .

12445 If individual i is alive at time t ($z(i, t) = 1$), then the observations are Bernoulli as
 12446 before. Conversely, if the individual is not alive ($z(i, t) = 0$), then the observations must
 12447 be fixed zeros with probability 1. We can of course consider other encounter models such
 12448 as the Poisson or multinomial models described in Chapt. 9.

12449 We initialize the model for time $T = 1$ and then model the transition of individual
 12450 states from t to $t + 1$ as:

$$z(i, t + 1) \sim \text{Bern}(\phi_t z(i, t) + \gamma_t(1 - z(i, t))).$$

12451 Previously, we described how this formulation of the model uses a set of latent indicator
 12452 variables $r(i, t)$ which describes if individual is recruited into the population during time
 12453 $(t - 1, t)$. Therefore, $r(i, t) = 1$ if individual i is recruited in time interval $(t - 1, t)$ otherwise
 12454 $r(i, t) = 0$. Determining the number of recruits into the population, can be done using
 12455 two steps. For example, to estimate the number of recruits from time period 1 to 2, we
 12456 count those individuals not in the population at time 1 ($z_{i,1} = 0$) but alive at time 2
 12457 ($z_{i,2} = 1$). We can determine if individual i has entered the population at time $t = 2$
 12458 by using the formula: $R_{i,2} = (1 - z_{i,1})z_{i,2}$ and then sum $R_{i,2}$ over M to get the total
 12459 number of recruits. We can do this for all the primary periods in our study, as shown in
 12460 the **JAGS** code below.

12461 16.3.1 Mist-netting example

12462 In the previous analysis of the ovenbird data, we did not make use of the spatial location
 12463 for each net the ovenbirds were captured in. However, there were 44 mist nets operational

12464 during each of the sampling occasions. We already organized the data above so that
 12465 our 3-D encounter histories are set up. The data set is then $M = 200$ individuals by
 12466 $K = 10$ secondary occasions by $T = 5$ primary occasions. In the non-spatial version, we
 12467 reduced the data to captured or not-captured; however, the encounter history array Yarr)
 12468 contains the number of the net that each individual was captured in and contains a 45 if
 12469 the individual was not captured. The encounter history array, Yarr), was created above
 12470 in the code, so we do not reproduce the code here.

```

12471 cat("
12472 model {
12473   psi ~ dunif(0,1)
12474   phi ~ dunif(0,1)
12475   alpha0 ~ dnorm(0,10)
12476   sigma ~ dunif(0,200)
12477   alpha1<- 1/(2*sigma*sigma)
12478
12479   A <- ((xlim[2]-xlim[1]))*((ylim[2]-ylim[1]))
12480
12481   for(t in 1:5){
12482     N[t] <- sum(z[1:M,t])
12483     D[t] <- N[t]/A
12484     gamma[t] ~ dunif(0,1)
12485   }
12486
12487   for(i in 1:M){
12488     z[i,1] ~ dbern(psi)
12489
12490     #to estimate the number of recruits, we need a few derivations
12491     R[i,1]<- z[i,1]
12492     R[i,2]<-(1-z[i,1])*z[i,2]
12493     R[i,3]<- (1-z[i,1])*(1-z[i,2])*z[i,3]
12494     R[i,4] <-(1-z[i,1])*(1-z[i,2])*(1-z[i,3])*(1-z[i,4])*z[i,5]
12495     R[i,5] <-(1-z[i,1])*(1-z[i,2])*(1-z[i,3])*(1-z[i,4])*z[i,5]
12496
12497
12498   for(t in 1:5){
12499     S[i,1,t] ~ dunif(xlim[1],xlim[2]) # XXXX This needs to be justified XXXX
12500     S[i,2,t] ~ dunif(ylim[1],ylim[2])
12501
12502     for(j in 1:ntraps){
12503       d[i,j,t] <- pow(pow(S[i,1,t]-X[j,1],2) + pow(S[i,2,t]-X[j,2],2),1)
12504     }
12505
12506     for(k in 1:K){
12507       for(j in 1:ntraps){
12508         lp[i,k,j,t] <- exp(alpha0 - alpha1*d[i,j,t])*z[i,t]
12509

```

```

12510     cp[i,k,j,t] <- lp[i,k,j,t]/(1+sum(lp[i,,t]))
12511   }
12512   cp[i,k,ntraps+1,t] <- 1-sum(cp[i,k,1:ntraps,t]) #last cell = not captured
12513   Ycat[i,k,t] ~ dcat(cp[i,k,,t])
12514 }
12515 }
12516
12517 a[i,1]<-(1-z[i,1])
12518
12519 for(t in 2:5){
12520   a1[i,t] <- sum(z[i, 1:t])
12521   a[i,t] <- 1-step(a1[i,t] - 1)
12522
12523   mu[i,t]<- (phi*z[i,t-1]) + (gamma[t]*a[i,t-1])
12524   z[i,t]~dbern(mu[i,t])
12525   }
12526 }
12527
12528 R1<-sum(R[1:M,1])
12529 R2<-sum(R[1:M,2])
12530 R3<-sum(R[1:M,3])
12531 R4<-sum(R[1:M,4])
12532 R5<-sum(R[1:M,5])
12533 }
12534
12535 ",file="modelJS.txt")
12536 ####
12537
12538
12539 zst<-c(rep(1,M/2),rep(0,M/2))
12540 zst<-cbind(zst,zst,zst,zst,zst)
12541
12542 inits <- function(){list (z=zst,sigma=runif(1,25,100), gamma=runif(5,0,1) ,S=Sst,alpha0=runif(1,-2,-1) ) }
12543 parameters <- c("psi","alpha0","alpha1","sigma","N","D", "phi", "gamma", "R2", "R3", "R4", "R5")
12544 data <- list (X=as.matrix(X[[1]]),K=10,Ycat=Yarr,M=M,ntraps=ntraps,ylim=ylim,xlim=xlim)
12545
12546 library("rjags")
12547 out1 <- jags.model("modelJS.txt", data, inits, n.chains=3, n.adapt=500)
12548 out2JS <- coda.samples(out1,parameters,n.iter=10000)
12549

```

12550 Our results for density, alpha0, and alpha1 are rather similar to those found in the
12551 multi-season analysis from Chapt. 9. Since all of our parameters including alpha0 and
12552 alpha1 are shared between seasons, we would expect these results to be similar between the
12553 multi-season model and the JS model (see Table 16.1). There are some slight differences in
12554 the parameter estimates, for example, the density is smaller in year 4 in the multi-season
12555 model than in the JS model. This maybe be due to a smaller sample size in that year

Table 16.1. Posterior mean of model parameters for the non-spatial JS model (NS-JS), the spatial JS model (S-JS), and the spatial multi-season model (S-MS) fitted to the ovenbird data set.

	NS-JS	S-JS	S-MS
D[1]	NA	9.6e-05	9.3e-05
D[2]	NA	1.0e-04	1.0e-04
D[3]	NA	1.1e-04	1.2e-04
D[4]	NA	1.1e-04	8.9e-05
D[5]	NA	7.9e-05	7.6e-05
N[1]	26.5	33	32.4
N[2]	30.2	36	35.8
N[3]	33.1	39	42.1
N[4]	29.5	37	30.8
N[5]	21.7	28	26.2
alpha0	NA	-2.9	-2.88
alpha1	NA	1.2e-04	1.22e-04
sigma	NA	6.4	6.44
gamma[1]	0.50	0.50	NA
gamma[2]	0.09	0.09	NA
gamma[3]	0.11	0.13	NA
gamma[4]	0.13	0.16	NA
gamma[5]	0.07	0.08	NA
phi	0.48	0.53	NA
psi	0.14	0.17	NA
R2	NA	1.5e+01	NA
R3	NA	1.9e+01	NA
R4	NA	8.3e+00	NA
R5	NA	8.3e+00	NA

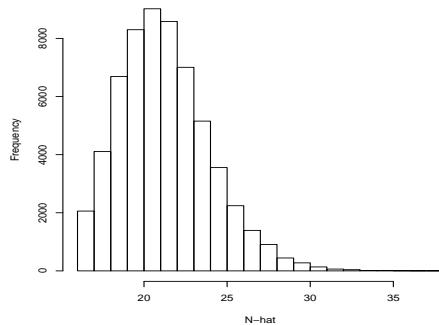


Figure 16.2. Posterior distribution of N_5 from the spatial JS model for the ovenbird dataset. This graph suggests that there is no truncation of the posterior of N_5 .

and the JS model is able to make use of the data a little more efficiently across the years. Because we have defined the same state space for the spatial JS model and multi-season, our estimates of N are directly comparable. However, the estimates of N under the non-spatial JS model are not directly comparable as we do not have a well-defined effective trapping area. We see from Table 16.1 that N is smallest for the non-spatial JS model across all years. This suggests that the actual effective trapping area is smaller than our state space, but we cannot know how much relative to the state-space to make useful comparisons between the N s.

In the JS formulation of the model, we also estimate the recruitment for each year, and we can look at our derived values for recruitment (R_2 , R_3 , R_4 , and R_5). R_2 is the number of new recruits from primary period 1 to 2; R_3 is the number of new recruits from primary period 2 to 3; and so forth. R_2 and R_3 are almost double that of R_4 and R_5 , suggesting that less animals were recruited into the population in the latter years of the study. The density in the last year of the study was lower than previous years. It is good to check your results when you see a pattern like this – the number of recruits declining each year – because this could be an indication that the data augmentation was not large enough. In this example, we checked to make sure that $M=200$ was sufficiently large by examining the recruitment parameter, γ . If γ is close to 1 during any of the time periods, then there are not enough augmented individuals in the overall dataset. In this case, the 97.5% quantile of γ_5 , the recruitment probability in the final year of the study, was 0.14 and none of the other γ s were close to 1 either. You can also look at the posterior distributions of N to make sure they are not truncated, Fig. 16.2 shows that the posterior distribution of N_5 is not truncated. The posterior mean for survival, ϕ , was 0.53. Although we did not do it here, it should be easy to see that we could allow survival to vary by time, as we did with recruitment. Our estimates of survival seem reasonable when compared with the literature. Some studies have found annual male ovenbird survival to be around 0.62 (Porneluzzi and Faaborg, 1999; Bayne and Hobson, 2002); however, female ovenbird survival was much lower (0.21, Bayne and Hobson (2002)). With more individuals, we

12584 could run this model with survival estimated for each sex separately. However, we should
 12585 be careful not to over-parameterize our model based on the amount of data available.

16.4 TRADITIONAL CJS MODELS

12586 The Cormack-Jolly-Seber models are used extensively in the literature to estimate survival
 12587 probabilities. There are essentially two ways to fit these models, using either a multinomial
 12588 approach (Lebreton et al., 1992) or a state-space likelihood approach (Gimenez et al., 2007;
 12589 Royle, 2008).

12590 We can adopt a simple hierarchical parameterization of the basic single state, non-
 12591 spatial CJS model in which the observation model is described conditional on the latent
 12592 state variables $z(i, t)$ – the alive state which describe whether individual i is alive ($z(i, t) =$
 12593 1) or not ($z(i, t) = 0$) during each of $t = 1, 2, \dots, T$ primary periods. Let y_{it} indicate
 12594 the observed encounter data of individual i in primary period t . The model, specified
 12595 conditional on $z(i, t)$, is:

$$y_{it}|z_{it} \sim \text{Bernoulli}(p_t z_{it}).$$

12596 Analogous to the JS model, if individual i is alive at time t ($z_{it} = 1$), then the observations
 12597 are Bernoulli with probability of detection p_t . If the individual is not alive ($z_{it} = 0$), then
 12598 the observations must be fixed zeros with probability 1. In the CJS formulation, as
 12599 opposed to the JS, we condition on first capture which means that z_{it} will be 1 when t is
 12600 the first primary period of capture. We can denote this z_{if_i} where f_i indicates the primary
 12601 occasion in which individual i is first captured, which can vary from $1 \dots T$. This ensures
 12602 that each individual is alive upon entering the model.

12603 We can describe the "alive state" at time t for each individual as a function of the
 12604 state at the previous time step $t - 1$. Because we condition on the first capture, the initial
 12605 state is set to one:

$$z_{if_i} = 1$$

12606 and to model the transition of individual states from t to $t + 1$ for all $t > f_i$ we have

$$z_{it} \sim \text{Bernoulli}(\phi z_{i,t-1}).$$

12607 Because we start with $z_{if_i} = 1$, the individual survives with probability ϕ to time $f_i + 1$
 12608 and so forth. Once an individual leaves the population (i.e., $z_{it} = 0$), there is no mech-
 12609 anism for the individual to return. This means that under this specification individuals
 12610 cannot temporarily emigrate. In the CJS model we are not estimating N , so we do not
 12611 incorporate any data augmentation here. This version of the model is easy to construct in
 12612 the **BUGS** (or **JAGS**) language which is shown in Panel 16.2. Variations on this basic
 12613 model and associated code for fitting the model in **BUGS** are described in detail in Kéry
 12614 and Schaub (2012, Chaps. 7-9).

12615 16.4.1 Migratory fish example

12616 The motivation for this example stems from an interest in better understanding survival
 12617 and movement of migratory fishes. For this example, we will use data collected on Amer-
 12618 ican shad *Alosa sapidissima* in the New River in North Carolina, U.S.A. (see photo in
 12619 Fig. 16.3). The data were collected and analyzed in Raabe (2012). Using a resistance

```
model {
  phi ~ dunif(0,1)    #Survival (constant over time)

  for(t in 1:T){
    p[t] ~ dunif(0, 1)      #detection (varies with time)
  }

  for (i in 1:M){
    z[i,first[i]] ~ dbern(1)
    for (t in (first[i]+1):T) {
      tmp[i,t] <- z[i,t]*p[t]
      y[i,t] ~ dbern(tmp[i,t])
      phiUP[i,t] <- z[i,t-1]*phi
      z[i,t] ~ dbern(phiUP[i,t])
    }
  }
}
```

Panel 16.2: **JAGS** model specification for the non-spatial basic CJS model.



Figure 16.3. American shad caught in North Carolina, U.S.A. Credit: Joshua Raabe, North Carolina State University

Table 16.2. Results of the basic non-spatial CJS model for the American shad dataset.

	Mean	SD	2.5 %	50 %	97.5 %
p[1]	0.499	0.289	0.026	0.499	0.975
p[2]	0.627	0.058	0.511	0.628	0.738
p[3]	0.762	0.036	0.689	0.763	0.829
p[4]	0.880	0.025	0.828	0.882	0.925
p[5]	0.548	0.043	0.465	0.548	0.633
p[6]	0.259	0.038	0.190	0.258	0.337
p[7]	0.126	0.031	0.072	0.124	0.194
p[8]	0.236	0.045	0.155	0.234	0.332
p[9]	0.237	0.049	0.148	0.234	0.341
p[10]	0.589	0.072	0.447	0.590	0.728
p[11]	0.834	0.063	0.700	0.839	0.942
p[12]	0.468	0.072	0.330	0.466	0.614
ϕ	0.824	0.011	0.802	0.825	0.846

12620 board weir near the river mouth, 315 fish were tagged with passive integrated transponders
 12621 (PIT) in the spring of 2010. An array of 7 upstream PIT antennas passively recaptured
 12622 individuals during upstream and downstream migrations. Each time a fish passed over
 12623 the antenna, it was recorded and summarized weekly for 12 weeks. To apply the basic
 12624 CJS model, we create the encounter history for each individual for the 12 weeks and we
 12625 also create a vector to indicate the period of first capture.

12626 Table 16.2 shows the estimated detection probability for each of the 12 primary periods
 12627 in the study. The posterior mean for detection probability ranges from 0.126 to 0.880,
 12628 which could potentially be due to variation in water flow, stream depth, storms, etc...
 12629 The weekly survival probability, ϕ had a posterior mean estimate of 0.824. This estimate
 12630 could be considered low for a weekly probability, but is likely due to the fact that the
 12631 migration upstream can be quite energetically taxing. Additionally, the CJS model is
 12632 only estimating apparent survival and some fish may have left the stream temporally or
 12633 permanently. We demonstrate in panel 16.2 how to allow p to vary by time, but we could
 12634 also allow survival, ϕ to vary by time by implementing it exactly as we do p . As we move
 12635 into the multi-state model, we can test for movement and survival by state, allow us to
 12636 address more specific biological questions.

16.5 MULTI-STATE CJS MODELS

12637 The basic version of the CJS model only allows for estimation of survival and detection.
 12638 However, researchers are often interested in addressing other ecological questions such as
 12639 age-dependent survival rates, habitat based movements, etc. Multi-state models allow
 12640 researchers to directly address such questions by incorporating more than one state that
 12641 an individual may potentially be in Arnason (1972, 1973); Brownie et al. (1993). These
 12642 possible states can be geographic location, age class, or reproductive status among many
 12643 others. Instead of just having an encounter history for an individual, we will also have
 12644 auxiliary information on the state of that individual at capture (e.g., breeder or non-
 12645 breeder). Since our interest in movement of individuals, here we will consider states that
 12646 represent spatial units or geographic locations. Generally speaking, we might think that

the transition rates between locations could be due to habitat features (or quality) and we can use multi-state models to help us address such a question. In addressing movement through a multi-state modeling approach, the movement is often parameterized as random or Markovian between patches (Arnason (1972, 1973); Schwarz et al. (1993)).

In the simplest version of the multi-state model we have just two states. Thus, individuals can be marked and recaptured in one of two states (we'll call them A and B here). We will assume that the two "states" are different geographic sites. In our single-state model above, an individual i was either alive ($z_{it} = 1$) at time t or dead ($z_{it} = 1$). Now, we must consider that the individual could be alive in a given state or dead and that individuals can transition between states. An easy way to think about this is to look at the state transition matrix in Table 16.3. Here, ϕ_A is the probability of surviving in State A from time t to $t + 1$ and ϕ_B is analogous for State B. The movement parameters are ψ_{AB} and ψ_{BA} , where ψ_{AB} is the probability that an individual, which survived from t to $t + 1$ in Site A, moves to State B just before $t + 1$ and vice versa for ψ_{BA}

Table 16.3. Transition matrix for a multi-state model with just two states.

	State A	State B	Dead
State A	$\phi_A(1 - \psi_{AB})$	$\phi_A\psi_{AB}$	$1 - \phi_A$
State B	$\phi_B\psi_{BA}$	$\phi_B(1 - \psi_{BA})$	$1 - \phi_B$
Dead	0	0	1

We do not necessarily observe individuals in their given state though, so we must estimate detection separately for each of the states. Hence we also have p_A and p_B , the probability of detecting an individual in state A and state B respectively.

To relate this back to the description of multi-state models in Chapt. 9, we can define s as the index of which state an individual is in and u_{it} as the state in which individual i was observed during sample t . In this two state example, u_{it} can only take on values for being observed in A or B (i.e., 1 or 2).

We can define a simplistic model such that

$$u_{it} \sim \text{dcat}(\psi)$$

where ψ is a constant vector. We observe that individual with probability p_0 , that is:

$$\Pr(y_{it} = 1 | u_{it}) = p_0$$

The state-transition probabilities are constant.

To extend this model, we can define s as the index of which state an individual is in and then condition the observed locations, u_{it} as a function of the state an individual is in, s . This means that whether an individual moves or not, or where it moves to, is a function of where it is located.

This commonly used model has successive movement outcomes that are *iid*

$$u_{it} \sim \text{dcat}(\psi(s_i))$$

Conditional on the state in which individual i is located, we observe that individual with probability p_0 . That is:

$$\Pr(y_{it} = 1 | u_{it}) = p_0$$

Table 16.4. Results of the multi-state CJS model for the migratory fish example. p_A is the detection probability in the first state (A), which in this case is the down stream area. ϕ_A is the weekly survival probability in state A and ψ_{AB} is the probability that an individual, which survived from t to $t + 1$ in Site A, moves to State B just before $t + 1$.

	Mean	SD	2.5 %	50 %	97.5 %
p_A	0.777	0.045	0.689	0.777	0.866
p_B	0.434	0.027	0.382	0.434	0.489
ϕ_A	0.850	0.022	0.807	0.851	0.893
ϕ_B	0.782	0.019	0.743	0.782	0.820
ψ_{AB}	0.421	0.034	0.356	0.421	0.489
ψ_{BA}	0.927	0.014	0.897	0.937	0.952

12678 The state-transition probabilities are still constant, conditional on s . Other models for
 12679 these transition probabilities are possible and we will discuss those later.

12680 A slight modification of this model would define s as a “home area” for each individual.
 12681 Then the region the animal goes to is a function, not of where he was last time, but
 12682 which region is his home area. This model is only subtlety different from the Markovian
 12683 model and as was shown in Chapt. 9 for closed populations models is how we make the
 12684 technical transition from multi-state models to SCR models. Essentially increasing to a
 12685 large number of strata, this formulation of the multi-state model becomes an SCR model
 12686 where the “home area” s becomes the “activity center” for each individual.

12687 To program this model in **JAGS**, we use a slightly different formulation which es-
 12688 sentially combines u_{it} and y_{it} as defined above into one observation matrix such that
 12689 $y_{it} = 1, 2$, or 3 where 3 indicates “not observed”. Additionally, we use z_{it} to indicate the
 12690 true state of individual i such that $z_{it} = 1, 2$, or 3 where 1 indicates alive and in state 1, 2
 12691 indicates alive and in state 2, and 3 indicates “not alive”. Using this delineation, we just
 12692 need to set up the transition matrix based on Table 16.3 and define each item within the
 12693 model specification, shown in Panel 16.3. Note that this can become quite cumbersome
 12694 when dealing with models that have many states.

12695 16.5.1 Migratory fish example

12696 Previously, we analyzed the American shad data using a basic CJS model. However,
 12697 the researchers were interested in movement of fish during migration and so we classified
 12698 the stream into 2 states (regions) – “downstream” and “upstream”. Each antenna was
 12699 assigned to a state based on the location, those below 20 river kilometers were considered
 12700 in the downstream state. Each fish has an encounter history including whether or not the
 12701 fish was detected during each week of the 12 week study, but also the “state” of capture
 12702 (“downstream” or “upstream”). Again, a vector to indicate the period of first capture
 12703 was also created. Fish captured in more than one state during the week were assigned the
 12704 state in which they were captured most during that week.

12705 Survival between the two areas is quite different (see Table 16.4). This might suggest
 12706 that fish moving further upstream are expending more energy and are more likely to die.
 12707 While survival in the two states was different, it is intuitive that the average of the survival
 12708 probabilities for A and B is essentially the same as that from the basic non-spatial CJS

```

model {
  for(r in 1:2){
    phi[r] ~ dunif(0,1)
    psi[r] ~ dunif(0,1)
    p[r] ~ dunif(0,1)
  }

  for (i in 1:M){
    z[i,first[i]] <- y[i, first[i]]
    for (t in (first[i]+1):T){
      z[i,t] ~ dcat(ps[z[i,t-1], i, ])
      y[i,t] ~ dcat(po[z[i,t], i, ])
    }
    ps[1, i, 1] <- phi[1] * (1-psi[1])
    ps[1, i, 2] <- phi[1] * psi[1]
    ps[1, i, 3] <- 1-phi[1]
    ps[2, i, 1] <- phi[2] * (1-psi[2])
    ps[2, i, 2] <- phi[2] * psi[2]
    ps[2, i, 3] <- 1-phi[2]
    ps[3, i, 1] <- 0
    ps[3, i, 2] <- 0
    ps[3, i, 3] <- 1

    po[1, i, 1] <- p[1]
    po[1, i, 2] <- 0
    po[1, i, 3] <- 1-p[1]
    po[2, i, 1] <- 0
    po[2, i, 2] <- p[2]
    po[2, i, 3] <- 1-p[2]
    po[3, i, 1] <- 0
    po[3, i, 2] <- 0
    po[3, i, 3] <- 1
  }
}

```

Panel 16.3: **JAGS** model specification for a two state version of the multi-state CJS model. Code adjusted from (Kéry and Schaub, 2012, Chapt. 9).

($\phi = 0.82$, see Table 16.2). Also, it should be noted that $\psi_B A$ is very high, indicating that fish in this study are returning downstream after spawning in the upstream area. These results highlight the utility in using a multi-state model to understand movement between states; here, we used spatial states, but age, class, breeding status, etc. are all possibilities. We did have to reduce the dataset however to fit this model and information on spatial location was lost in creating just two states, downstream and upstream.

16.6 SPATIAL CJS MODELS

In Chapt. 9, we described how SCR models are essentially a type of multi-state model with spatially structured transition probabilities. As we noted, individuals can appear in > 1 states, simultaneously, which is not directly analogous to a standard multi-state model. However, building on the state-space and multi-state CJS models, we can explicitly incorporate individual movement as an individual covariate model (Royle, 2009a). To move from the basic and multi-state CJS models to the SCR version, we need only make a few changes to the model. Essentially, we will not have discrete states and thus the biggest difference is that individuals do not “transition” between a finite set of states, but instead are allowed to move in continuous space.

We may consider the same basic encounter models as described previously (i.e., Poisson, Bernoulli, or multinomial). In particular, let y_{ijkt} indicate the observed encounter data of individual i in trap j , during interval (secondary period or sub-sample) $k = 1, 2, \dots, K$ and primary period t . We note that in some cases we may have intervals ($K = 1$) which correspond to the design underlying a standard CJS or JS models whereas the case $K > 1$ corresponds to the “robust design” (Pollock 1982). The Poisson observation model, specified conditional on $z(i, t)$, is:

$$y_{ijkt}|z(i, t) \sim \text{Poisson}(\lambda_0 g_{ij} z(i, t)).$$

Conversely, if the individual is not alive ($z(i, t) = 0$), then the observations must be fixed zeros with probability 1. In the CJS formulation, we will condition on first capture which means that $z(i, t)$ will be 1 when t is the first primary period of capture. We can denote this $z(i, f_i)$ where f_i indicates the primary occasion in which individual i is first captured. This ensures that each individual is alive upon entering the model.

Modeling time-effects either within or across primary periods is straightforward. For that, we define $\lambda_0 \equiv \lambda_0(k, t)$ and then develop models for $\lambda_0(k, t)$ as in our closed SCR models (we note that trap-specific effects could be modeled analogously).

We follow the same model for survival as described in the non-spatial version of the CJS. The model is initialized by setting the alive state at first capture to one:

$$z(i, f_i) = 1$$

and for the transition of an individual’s alive state from t to $t + 1$, for all $t > f_i$, we have

$$z_{it} \sim \text{Bernoulli}(\phi z_{i,t-1}).$$

An individual survives with probability ϕ from one time step to the next. It is easy to see that we can let survival be time specific by allowing ϕ to vary with each time step:

$$z_{it} \sim \text{Bernoulli}(\phi_t z_{i,t-1}).$$

12744 In either case, once an individual leaves the population (i.e., $z_{it} = 0$), there is no
 12745 recruitment so individuals cannot return. Again, we are not estimating N in this model,
 12746 hence we do not need any data augmentation. This conveniently makes the model run
 12747 faster too!

12748 **16.6.1 Migratory fish example**

12749 Going back to our American shad example, we can consider that this is exactly a spatial
 12750 capture recapture problem. In stream networks, the placement of PIT antennas along the
 12751 stream mimics the type of spatial data collected in terrestrial passive detector arrays such
 12752 as camera traps, hair snares, acoustic recording devices, etc. The difference is that for
 12753 fish and aquatic species, the stream constrains the movement of individuals to a linear
 12754 network. Using the data from the array of 7 PIT antennas and the number of times each
 12755 fish passed over the antenna, we can apply the SCR CJS model to evaluate movement
 12756 up and downstream of these fishes. When we look at the individuals encountered at each
 12757 antenna for each of the primary periods, the dimensions of the data are 315 individuals by
 12758 7 antennas by 12 sample occasions. Individuals can encounter any antenna any number
 12759 of times during the week, which means we just sum the encounters over the week and
 12760 eliminate any need for explicit secondary occasions in the model. The result is a 3-D
 12761 array instead of a 4-D array. Given the structure of the encounters, we use a Poisson
 12762 encounter model in this example.

```
12763 library(reshape)
12764
12765 # Constants:
12766 M <- 315      # Number of individuals
12767 T <- 12       # Number of periods (weeks)
12768 nantenna <- 7 # weir, 6 antennas
12769 antenna.loc <- c(3,7,12,44,56,72,77) # antenna locations
12770
12771 # Input and format data matrix:
12772 AS10 <- read.table("AS10.txt" ,header=T)
12773 melted.rkm <- melt(AS10, id=c("TagID","RKM"))
12774 y <- cast(melted.rkm, TagID ~ RKM ~ value, fill=0, length)
12775 first=read.csv("firstcap.csv")
12776
12777 sink("ModelCJS.txt")
12778 cat("
12779
12800 model {
12801 # Priors
12802 sigma ~ dunif(0,80)
12803 sigma2 <- sigma*sigma
12804 lam0 ~ dgamma(0.1, 0.1)
12805 phi ~ dunif(0, 1) # Survival (constant across time)
12806 tauv~dunif(0, 30)
12807 tau<-1/(tauv*tauv)
```

```

12788
12789 for (i in 1:M){
12790   z[i,first[i]] <- 1
12791   S[i,first[i]] ~ dunif(0,50)
12792
12793 for(j in 1:nantenna) {
12794   D2[i,j,first[i]] <- pow(S[i,first[i]]-antenna.loc[j], 2)
12795   lam[i,j,first[i]]<- lam0*exp(- D2[i,j,first[i]]/(2*sigma2))
12796   tmp[i,j,first[i]] <- lam[i,j,first[i]]
12797   y[i,j,first[i]] ~ dpois(tmp[i,j,first[i]])
12798 }
12799
12800 for (t in first[i]+1:T) {
12801   S[i,t] ~ dunif(xl, xu) # XXXX above you have dunif(0,50)?
12802   for(j in 1:nantenna) {
12803     D2[i,j,t] <- pow(S[i,t]-antenna.loc[j], 2)
12804     lam[i,j,t] <- lam0 * exp(-D2[i,j,t]/(2*sigma2))
12805     tmp[i,j,t] <- z[i,t]*lam[i,j,t]
12806     y[i,j,t] ~ dpois(tmp[i,j,t])
12807   }
12808   phiUP[i,t] <- z[i,t-1]*phi
12809   z[i,t] ~ dbern(phiUP[i,t])
12810 }
12811 }
12812 }
12813
12814 ",fill = TRUE)
12815 sink()
12816
12817 data1<-list(y=y, first=first, M=M, T=T, xl=0, xu=80, nantenna=nantenna, antenna.loc=antenna.loc)
12818
12819 z=matrix(NA, M, T)
12820 for(i in 1:M){
12821   for(t in first[i]:12){
12822     z[i,t] <-1
12823   }
12824 }
12825
12826 inits = function() {list(z=z,phi=runif(1,0,1), lam0=runif(1,0,2),
12827                         tauv=runif(1,10, 20), sigma=runif(1,0,10)) }
12828
12829 parameters <- c("sigma", "phi", "lam0")
12830
12831 library("rjags")
12832 out1 <- jags.model("modelCJS.txt", data1, inits, n.chains=3, n.adapt=500)
12833 out2CJS <- coda.samples(out1,parameters,n.iter=20000)

```

Table 16.5. Results of the spatial CJS model fitted to the American shad data set.

	Mean	SD	2.5 %	50 %	97.5 %
lam0[1]	5.555	0.224	5.125	5.553	6.003
lam0[2]	4.442	0.155	4.143	4.437	4.752
lam0[3]	1.892	0.068	1.763	1.891	2.031
lam0[4]	1.126	0.055	1.021	1.125	1.238
lam0[5]	0.949	0.058	0.838	0.948	1.067
lam0[6]	0.359	0.040	0.284	0.357	0.443
lam0[7]	0.188	0.031	0.133	0.186	0.254
lam0[8]	0.309	0.044	0.230	0.307	0.402
lam0[9]	0.363	0.052	0.269	0.361	0.471
lam0[10]	0.627	0.072	0.493	0.625	0.777
lam0[11]	1.611	0.109	1.408	1.607	1.835
lam0[12]	0.939	0.139	0.697	0.929	1.241
ϕ	0.784	0.012	0.760	0.785	0.807
σ	13.954	0.197	13.573	13.950	14.350

12834 The baseline encounter rate, λ_0 , was allowed to vary by week and ranged from 0.188 to
 12835 5.555. We use the Poisson encounter model in this spatial CJS example rendering λ_0 not
 12836 directly comparable to p_0 from the non-spatial and multi-state versions which arises as the
 12837 detection probability based under the Binomial encounter model. The posterior mean for
 12838 ϕ was 0.784 (see Table 16.5), again showing that the survival probability is generally low,
 12839 just as we saw in the two previous example analysis of these data. Here, we are modeling
 12840 survival probability as constant, but there is reason to believe that it might vary by time
 12841 (similar to detection) and we might consider this additional parameterization in a more
 12842 complete analysis of the data set. The other parameter of interest is σ , the movement
 12843 parameter, which had a posterior mean of 13.954. Our system here is linear, so we do not
 12844 think of fish as having a home range radius in this system. However, σ can still inform
 12845 us about the linear distance fish are moving. One final note about this example, we have
 12846 simplified the dataset for analysis here and some parameter estimates are different than
 12847 found in Raabe (2012).

16.7 MOVING ACTIVITY CENTERS

12848 We extend the model of individual encounter histories by specifying an additional model
 12849 component that describes the spatial distribution of individual activity centers. A plau-
 12850 sible “null model” for the distribution of individual activity centers is to assume they
 12851 are static over time and do not change across primary periods, i.e., $s_i \sim \text{Unif}(\mathcal{S})$. It
 12852 might seem more likely that activity centers change over time but are independent from
 12853 year to year for a given individual such $s_i \sim \text{Unif}(\mathcal{S})$. This is how the spatial version
 12854 of the JS and CJS models were formulated above. Another option would be to assume
 12855 that $s(i, t) \sim \text{Normal}(s(i, t - 1), \tau^2 \mathbf{I})$ for $t > 1$ so that individual home range centers are
 12856 perturbed randomly from their previous value.

12857 We could use this specification to model changes in home range centers with regards
 12858 to habitat. For example, if our primary period is a season, we may expect that individuals
 12859 move as the available food sources change. Using telemetry data and/or capture recapture

models a number of developments have been made to understand animal movement patterns relative to habitat or dynamic systems(e.g., Jonsen et al. (2005); Hooten and Wikle (2010)). Similarly, if we have an indicator of habitat that varies by season, then in SCR models we can model the location of activity centers as a function of the change in habitat. There are a number of options for modeling variation in activity centers or animal locations as a function of covariates such as habitat, season, or behavior. Other approaches to analyzing movement in a mark-recapture framework include but are not limited to diffusion and auto-regressive models(Ovaskainen (2004); Ovaskainen et al. (2008)), agent-based (Grimm et al. (2005); Hooten et al. (2010)) and dispersal kernels (Fujiwara et al. (2006)). For example, we define u_{ikt} as the individual's observed location at secondary period k in primary period t . Then $u_{ikt} \sim \text{Normal}(\mathbf{s}(i, t), \Sigma_t)$ where Σ_t is the variance-covariance matrix at time t . This is the model we have assumed quite frequently throughout the book, i.e., that individual observed locations are assumed to follow a bivariate normal distribution about the activity center, \mathbf{s} . This is similar to the Guassian and Laplace dispersal kernels. We could then allow the observed locations to follow an auto-regressive model such that $u_{ikt} \sim \text{Normal}(\rho\mathbf{s}(i, t - 1), \Sigma_t)$ XXX figure out the variance XXX. This is just one simple example, as more information becomes available and data are collected over longer time periods, the ability to use different movement models will continue to be employed in open SCR models.

Rathbun and Cressie (1994) articulate model for marked point processes where they separate out the spatial birth, growth, and survival processes for longleaf pine trees. Because of the application, these demographic parameters are slightly different than how they are often considered in wildlife, but are still analogous. Allowing birth, growth, and survival as well as density to arise from different spatially varying processes is the next stage in development of the open SCR models.

16.7.1 Migratory Fish Example Notes

In our American shad example above, we had reason to believe that individual movement is directly related to stream flow. When the stream flow is low, we might expect that the fish move very little, and when the stream flow is high, they might move upstream to spawn. In this case, we could model the effect of stream flow in two ways. First, we might allow σ to be a function of flow and to vary for each primary occasion.

$$\log(\sigma_t) = \mu_S + \alpha_2 \text{Flow}_t$$

But if we think that the change in activity centers between primary periods might be related to the overall movement of fish, then we could allow the variation in locations to be a function of flow. This means that we assume the activity centers are correlated so we have

$$\mathbf{s}(i, t) \sim \text{Normal}(\mathbf{s}(i, t - 1), \tau^2 \mathbf{I})$$

where

$$\log(\tau) = \mu_T + \alpha_2 \text{Flow}_t$$

These are just a few thoughts on simple ways to model movement as a function of habitat variables. As we discussed in the previous section, there are many other movement models that could be used.

16.8 SUMMARY AND OUTLOOK

12899 In this chapter we have described a framework for making inference not only about spatial
12900 and temporal variation in population density, but also demographic parameters including
12901 survival, recruitment, and movement. The ability to model population vital rates is es-
12902 sential for ecology, management, and conservation; and the models described here allow
12903 researchers to examine the spatial and temporal dynamics governing those population
12904 parameters.

12905 As open models are further developed, mechanisms for dealing directly with dispersal
12906 and transients will provide improved inference frameworks for understanding movement
12907 as well as the potential to estimate *true* survival instead of only *apparent* survival. This is
12908 a function of explicitly modeling movement, which means we can separate movement from
12909 mortality providing a huge advantage over traditional models. Also, models of individual
12910 dispersal can be used to examine dynamics of population dynamics relative to habitat,
12911 density-dependence, or climatic events.

12912 Birth and death processes, as well as movement, all have the potential to be related
12913 to the space usage of animals in the landscape. Understanding the impact of spatially
12914 varying density on survival and recruitment will provide insights into the basic ecology
12915 of species. With the advent of non-invasive techniques, like camera trapping and genetic
12916 analysis of tissue, we can start to understand the population dynamics of species that are
12917 rarely observed in the wild. As more and more data are collected, we can use the models
12918 to explore the spatio-temporal patterns of survival, recruitment, density, and movement of
12919 species, providing incredibly useful biological and ecological information as we face broad
12920 changes in climate, land-use, habitat fragmentation, etc..

Part IV

12921

12922

12923

Super-Advanced SCR Models

12924
12925

17

12926
12927

DEVELOPING MARKOV CHAIN MONTE CARLO SAMPLERS

12928 In this chapter we will dive a little deeper into Markov chain Monte Carlo (MCMC)
12929 sampling. We will construct custom MCMC samplers in **R**, starting with easy-to-code
12930 GLMs and GLMMs and moving on to simple CR and SCR models. Finally, we will
12931 illustrate some alternative ready-to-use software packages for MCMC sampling. We will
12932 NOT provide exhaustive background information on the theory and justification of MCMC
12933 sampling – there are entire books dedicated to that subject and we refer you to Robert and
12934 Casella (2004) and Robert and Casella (2010). Rather we aim to provide you with enough
12935 background and technical know-how to start building your own MCMC samplers for SCR
12936 models in **R**. You will find that quite a few topics that come up in this chapter have already
12937 been covered in previous chapters, particularly the introduction into Bayesian analysis in
12938 Chapt. 3. To keep you from having to leaf back and forth we will in some places briefly
12939 review aspects of Bayesian analysis, but we try to focus on the more technical issues of
12940 building MCMC samplers relevant to SCR models.

12941 17.0.1 Why build your own MCMC algorithm?

12942 The standard programs we have used so far to do MCMC analyses are **WinBUGS** (Gilks
12943 et al., 1994) and **JAGS** (Plummer, 2003). The wonderful thing about these **BUGS**
12944 engines is that they automatically use appropriate and, most of the time, reasonably
12945 efficient forms of MCMC sampling for the model specified by the user.

12946 The fact that we have such a Swiss Army knife type of MCMC machine begs the
12947 question: Why would anyone want to build their own MCMC algorithm? For one, there
12948 are a limited number of distributions and functions implemented in **BUGS**. While **Open-**
12949 **BUGS** provides more options, some more complex models may be impossible to build
12950 within these programs. A very simple example from spatial capture-recapture that can
12951 give you a headache in **WinBUGS** is when your state-space is an irregular-shaped poly-
12952 gon, rather than an ideal rectangle that can be characterized by four pairs of coordinates.

12953 It is easy to restrict activity centers to any arbitrary polygon in **R** using an ESRI shapefile
 12954 (and we will show you an example in a little bit), but you cannot use a shape file in a
 12955 **BUGS** model. Similarly, models of space usage that take into account ecological distance
 12956 (Chapt. 12) cannot be implemented in the **BUGS** engines. Moreover, there are classes of
 12957 SCR models that we have not been able to implement effectively using likelihood methods,
 12958 and are inefficient to run in the **BUGS** engines. Examples of those models are covered in
 12959 Chaps. 18 and 19.

12960 Sometimes implementing an MCMC algorithm in **R** may be faster than in **WinBUGS**
 12961 - especially if you want to run simulation studies where you have hundreds or more sim-
 12962 ulated data sets, several years' worth of data or other large models, this can be a big
 12963 advantage.

12964 Finally, building your own MCMC algorithm is a great exercise to understand how
 12965 MCMC sampling works. So while using the **BUGS** language requires you to understand
 12966 the structure of your model, building an MCMC algorithm requires you to think about
 12967 the relationship between your data, priors and posteriors, and how these can be efficiently
 12968 analyzed and characterized. Not to mention that, if you are an **R** junkie, it can actually
 12969 be fun. However, if you don't think you will ever sit down and write your own MCMC
 12970 sampler, consider skipping this chapter - apart from coding it will not cover anything
 12971 SCR-related that is not covered by other, more model-oriented chapters as well.

17.1 MCMC AND POSTERIOR DISTRIBUTIONS

12972 MCMC is a class of simulation methods for drawing (correlated) random numbers from
 12973 a target distribution, which in Bayesian inference is the posterior distribution. As a re-
 12974 minder, the posterior distribution is a probability distribution for an unknown parameter,
 12975 say θ , given observed data and its prior probability distribution (the probability distribu-
 12976 tion we assign to a parameter before we observe data). The great benefit of having the
 12977 posterior distribution of θ is that it can be used to make probability statements about
 12978 θ , such as the probability that θ is equal to some value, or the probability that θ falls
 12979 within some range of values. The posterior distribution summarizes all we know about a
 12980 parameter and thus, is the central object of interest in Bayesian analysis. Unfortunately,
 12981 in many if not most practical applications, it is nearly impossible to directly compute the
 12982 posterior. Recall Bayes' theorem:

$$[\theta|y] = \frac{[y|\theta][\theta]}{[y]}, \quad (17.1.1)$$

12983 where θ is the parameter of interest, y is the observed data, $[\theta|y]$ is the posterior, $[y|\theta]$ the
 12984 likelihood of the data conditional on θ , $[\theta]$ the prior probability of θ , and, finally, $[y]$ is the
 12985 marginal probability of the data, defined as

$$[y] = \int [y|\theta][\theta]d\theta$$

12986 This marginal probability is a normalizing constant that ensures that the posterior
 12987 integrates to 1. Often, the integral is difficult or impossible to evaluate, unless you are
 12988 dealing with a really simple model. For example, consider a Normal model, with a set of

12989 n observations, $y_i; i = 1, 2, \dots, n$:

$$y_i \sim \text{Normal}(\mu, \sigma),$$

12990 where σ is known and our objective is to obtain an estimate of μ . To fully specify the
 12991 model in a Bayesian framework, we first have to define a prior distribution for μ . Recall
 12992 from Chapt. 3 that for certain data models, certain priors lead to conjugacy, i.e. if you
 12993 choose a certain prior for your parameter, the posterior distribution will be of a known
 12994 parametric form. The conjugate prior for the mean of a Normal model is also a Normal
 12995 distribution:

$$\mu \sim \text{Normal}(\mu_0, \sigma_0^2)$$

12996 If μ_0 and σ_0^2 are fixed, the posterior for μ has the following form (for some of the algebra
 12997 behind this, see Chapt. 2 in Gelman et al. (2004)):

$$\mu|y \sim \text{Normal}(\mu_n, \sigma_n^2) \quad (17.1.2)$$

12998 where

$$\mu_n = \left(\frac{\sigma^2}{\sigma^2 + n\sigma_0^2} \right) \times \left(\mu_0 + \frac{n\sigma_0^2}{\sigma^2 + n\sigma_0^2} \right) \times \bar{y}$$

12999 And

$$\sigma_n^2 = \frac{\sigma^2 \sigma_0^2}{\sigma^2 + n\sigma_0^2}$$

13000 We can directly obtain estimates of interest from this Normal posterior distribution, such
 13001 as the mean $\hat{\mu}$ and its variance; we do not need to apply MCMC, since we can recognize
 13002 the posterior as a parametric distribution, including the normalizing constant $[y]$. But
 13003 generally we will be interested in more complex models with several, say m , parameters.
 13004 In this case, computing $[y]$ from Eq. 17.1.1 requires m -dimensional integration, which can
 13005 be difficult or impossible. Thus, the posterior distribution is generally only known up to
 13006 a constant of proportionality:

$$[\theta|y] \propto [y|\theta][\theta]$$

13007 The power of MCMC is that it allows us to approximate the posterior using simulation
 13008 without evaluating the high dimensional integrals and to directly sample from the pos-
 13009 terior, even when the posterior distribution is unknown! The price is that MCMC is
 13010 computationally expensive. Although MCMC first appeared in the scientific literature in
 13011 1949 (Metropolis and Ulam, 1949), widespread use did not occur until the 1980s when
 13012 computational power and speed increased (Gelfand and Smith, 1990). It is safe to say that
 13013 the advent of practical MCMC methods is the primary reason why Bayesian inference has
 13014 become so popular during the past three decades.

13015 In a nutshell, MCMC lets us generate sequential draws of θ (the parameter(s) of in-
 13016 terest) from distributions approximating the unknown posterior over T iterations. The
 13017 distribution of the draw at t depends on the value drawn at $t-1$; hence, the draws from a
 13018 Markov chain¹. As T goes to infinity, the Markov chain converges to the desired distri-
 13019 bution, in our case the posterior distribution for $\theta|y$. Thus, once the Markov chain has
 13020 reached its stationary distribution, the generated samples can be used to characterize the
 13021 posterior distribution, $[\theta|y]$, and point estimates of θ , its standard error and confidence
 13022 bounds, can be obtained directly from this approximation of the posterior.

¹Remember that for T random samples $\theta^{(1)}, \dots, \theta^{(T)}$ from a Markov chain the distribution of $\theta^{(t)}$ depends only on the immediately preceding value, $\theta^{(t-1)}$.

17.2 TYPES OF MCMC SAMPLING

13023 There are several general MCMC algorithms in widespread use, the most popular being
 13024 Gibbs sampling and Metropolis-Hastings sampling, both of which were briefly introduced
 13025 in Chapt. 3. We will be dealing with these two classes in more detail and use them to
 13026 construct MCMC algorithms for SCR models. Also, we will briefly review alternative
 13027 techniques that are applicable in some situations.

13028 **17.2.1 Gibbs sampling**

13029 Gibbs sampling was named after the physicist J.W. Gibbs by Geman and Geman (1984),
 13030 who applied the algorithm to a Gibbs distribution². The roots of Gibbs sampling can
 13031 be traced back to work of Metropolis et al. (1953), and it is actually closely related to
 13032 Metropolis sampling (see Chapt. 11.5 in Gelman et al. (2004), for the link between the
 13033 two samplers). We will focus on the technical aspects of this algorithm, but if you find
 13034 yourself hungry for more background, Casella and George (1992) provide a more in-depth
 13035 introduction to the Gibbs sampler.

13036 Let's go back to our simple example from above to understand the motivation and
 13037 functioning of Gibbs sampling. Recall that for a Normal model with known variance and
 13038 a Normal prior for μ , the posterior distribution of $\mu|y$ is also Normal. Conversely, with a
 13039 fixed (known) μ , but unknown variance, the conjugate prior for σ^2 is an Inverse-Gamma
 13040 distribution with shape and scale parameters a and b :

$$\sigma^2 \sim \text{Inverse-Gamma}(a, b),$$

13041 With fixed a and b , the posterior $[\sigma^2|\mu, y]$ is also an Inverse-Gamma distribution, namely:

$$\sigma^2|\mu, y \sim \text{Inverse-Gamma}(a_n, b_n), \quad (17.2.1)$$

13043 where $a_n = n/2 + a$ and $b_n = (1/2) \sum_{i=1}^n (y_i - \mu)^2 + b$. However, what if we know neither

13044 μ nor σ^2 , which is probably the more common case? The joint posterior distribution of μ
 13045 and σ^2 now has the general structure

$$[\mu, \sigma^2|y] = \frac{[y|\mu, \sigma^2][\mu][\sigma^2]}{\int [y|\mu][\mu][\sigma^2]d\mu d\sigma^2}$$

13046 or

$$[\mu, \sigma^2|y] \propto [y|\mu, \sigma^2][\mu][\sigma^2]$$

13047 This cannot easily be reduced to a distribution we recognize. However, we can con-
 13048 dition μ on σ^2 (i.e., we treat σ^2 as fixed) and remove all terms from the joint posterior
 13049 distribution that do not involve μ to construct the full conditional distribution,

$$[\mu|\sigma^2, y] \propto [y|\mu][\mu]$$

13050 The full conditional of μ again takes the form of the Normal distribution shown in
 13051 Eq. 17.1.2; similarly, $[\sigma^2|\mu, y]$ takes the form of the Inverse-Gamma distribution shown in

²a distribution from physics we are not going to worry about, since it has no immediate connection with Gibbs sampling other than giving its name

Eq. 17.2.1, both distribution we can easily sample from. And this is precisely what we do when using Gibbs sampling: we break down high-dimensional problems into convenient one-dimensional problems by constructing the full conditional distributions for each model parameter separately; and we sample from these full conditionals, which, if we choose conjugate priors, are known parametric distributions. Let's put the concept of Gibbs sampling into the MCMC framework of generating successive samples, using our simple Normal model with unknown μ and σ^2 and conjugate priors as an example. These are the steps you need in order to build a Gibbs sampler:

Step 0: Begin with some initial values for θ , say $\theta^{(0)}$. In our example, we have to specify initial values for μ and σ , for example by drawing a random number from some Uniform distribution, or by setting them close to what we think they might be. (Note: This step is required in any MCMC sampling; chains have to start from somewhere. We will get back to these technical details a little later.)

Step 1: Draw $\theta_1^{(1)}$ from the conditional distribution $[\theta_1^{(1)} | \theta_2^{(0)}, \dots, \theta_d^{(0)}]$. Here, θ_1 is μ , which we draw from the Normal distribution in Eq. 17.1.2 using $\sigma^{(0)}$ as value for σ .

Step 2: Draw $\theta_2^{(1)}$ from the conditional distribution $[\theta_2^{(1)} | \theta_1^{(1)}, \theta_3^{(0)}, \dots, \theta_d^{(0)}]$. Here, θ_2 is σ , which we draw from the Inverse-Gamma distribution of Eq. 17.2.1, using $\mu^{(1)}$ as value for μ .

Step 3, ..., d: Draw $\theta_3^{(1)}, \theta_4^{(1)}, \dots, \theta_d^{(1)}$ from their conditional distribution $[\theta_3^{(1)} | \theta_1^{(1)}, \theta_2^{(1)}, \theta_4^{(0)}, \dots, \theta_d^{(0)}], \dots, [\theta_d^{(1)} | \theta_1^{(1)}, \dots, \theta_{d-1}^{(1)}]$. In our example we have no additional parameters, so we only need step 0 through to 2.

Repeat Steps 1 to d for $T =$ a large number of samples.

In terms of **R** coding, this means we have to write Gibbs updaters for μ and σ^2 and embed them into a loop over T iterations. The final code in the form of an **R** function is shown in Panel 17.1.

This is it! You can go ahead and simulate some data, $y \sim \text{Normal}(5, 0.5)$ and then use the function **NormGibbs()** in the **R** package **scrbook** to run your first Gibbs sampler (note that the **R** function **rnorm** requires you to supply the standard deviation σ and we have written **NormGibbs** so that it returns σ instead of σ^2 so you can easily compare your input value and parameter estimate).

```
13082 set.seed(13)
13083
13084 #true mean and sd are 5 and 0.5
13085 y<-rnorm(1000, 5,0.5) #data
13086
13087 mu_0<-0 #prior mean
13088 sigma2_0<-100 #prior variance
13089
13090 #Inverse-Gamma hyperparameters
13091 a<-0.1
13092 b<-0.1
13093
13094 mod=Norm.Gibbs(y, mu_0, sigma2_0, a,b,niter=10000)
```

13095 Your output, `mod`, will be a table with two columns, one per parameter, and T rows,
 13096 one per iteration. For this 2-parameter example you can visualize the joint posterior by
 13097 plotting samples of μ against samples of σ (Fig. 17.1):

13098 `plot(out[,1], out[,2])`

13099 The marginal distribution of each parameter is approximated by examining the samples
 13100 of this particular parameter. You can visualize it by plotting a histogram of the samples
 13101 (Fig. 17.2 upper left and right):

13102 `par(mfrow=c(1,2))`
 13103 `hist(out[,1]); hist(out[,2])`

13104 Finally, recall an important characteristic of Markov chains, namely, that the chain
 13105 has to have converged (reached its stationary distribution) in order to regard samples as
 13106 coming from the posterior distribution. In practice, that means you have to throw out
 13107 some of the initial samples called the burn-in. We will talk about this in more detail
 13108 when we talk about convergence diagnostics. For now, you can use the `plot(out[,1])` or
 13109 `plot(out[,2])` command to make a time series plot of the samples of each parameter and
 13110 visually assess how many of the initial samples you should discard. Fig. 17.2 bottom left
 13111 and right shows plots for the estimates of μ and σ from our simulated data set; you see
 13112 that in this simple example the Markov chain apparently reaches its stationary distribution
 13113 very quickly – the chains look ‘grassy’ seemingly from the start. It is hard to discern a
 13114 burn-in phase visually (but we will see examples further on where the burn-in is clearer)
 13115 and you may just discard the first 500 draws to be sure you only use samples from the
 13116 posterior distribution. The mean of the remaining samples are your estimates of μ and σ :

13117 `summary(mod[501:10000,])`
 13118 mu sig
 13119 Min. :4.935 Min. :0.4652
 13120 1st Qu.:4.988 1st Qu.:0.4930
 13121 Median :4.998 Median :0.5006
 13122 Mean :4.998 Mean :0.5008
 13123 3rd Qu.:5.009 3rd Qu.:0.5084
 13124 Max. :5.062 Max. :0.5486

13125 17.2.2 Metropolis-Hastings sampling

13126 Although it is applicable to a wide range of problems, the limitations of Gibbs sampling are
 13127 immediately obvious: what if we do not want to use conjugate priors or what if we cannot
 13128 recognize the full conditional distribution as a parametric distribution, or simply do not
 13129 want to worry about these issues? The most general solution is to use the Metropolis-
 13130 Hastings (MH) algorithm, which also goes back to the work by Metropolis et al. (1953).
 13131 You saw the basics of this algorithm in Chapt. 3. In a nutshell, because we do not
 13132 recognize the posterior $[\theta|y]$ as a parametric distribution, the MH algorithm generates
 13133 samples from a known proposal distribution, say $h(\theta)$, that depends on the value of θ at
 13134 the previous time step, θ^{t-1} . The candidate value θ^* is accepted with probability.

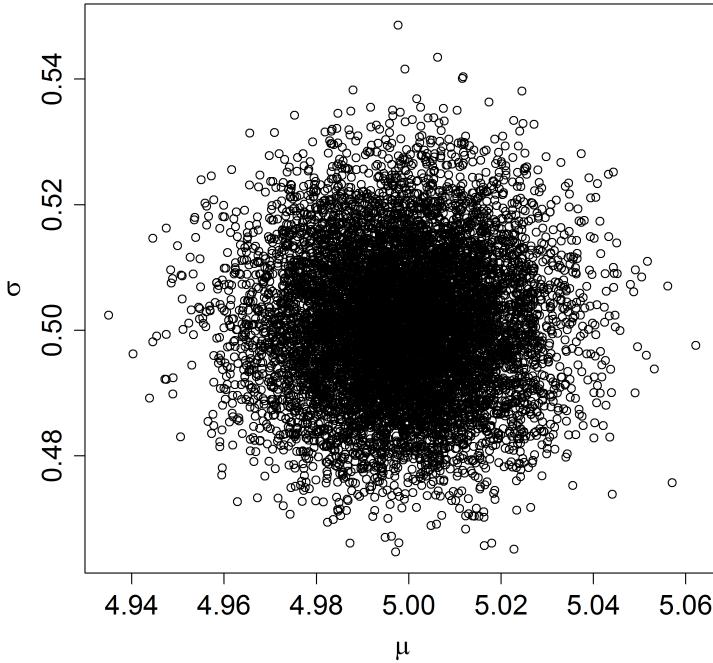


Figure 17.1. Joint posterior distribution of μ and σ from a Normal Model

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

Proposal distributions can be absolutely anything! You can generate candidate values from a $\text{Normal}(0,1)$ distribution, from a $\text{Uniform}(-3455,3455)$ distribution, or anything of proper support. Note, however, that good choices of $h()$ are those that approximate the posterior distribution. Obviously if $h() = [\theta|y]$ (i.e., the posterior) then you always accept the draw, and it stands to reason that proposals that are more similar to $[\theta|y]$ will lead to higher acceptance probabilities.

The original Metropolis algorithm required $h(\theta)$ to be symmetric so that

$$h(\theta^*|\theta^{t-1}) = h(\theta^{t-1}|\theta^*)$$

In that case these two terms just cancel out from the MH acceptance probability and r is then just the ratio of the target density evaluated at the candidate value to that evaluated at the current value. A later development of the algorithm by Hastings (1970) lifted this

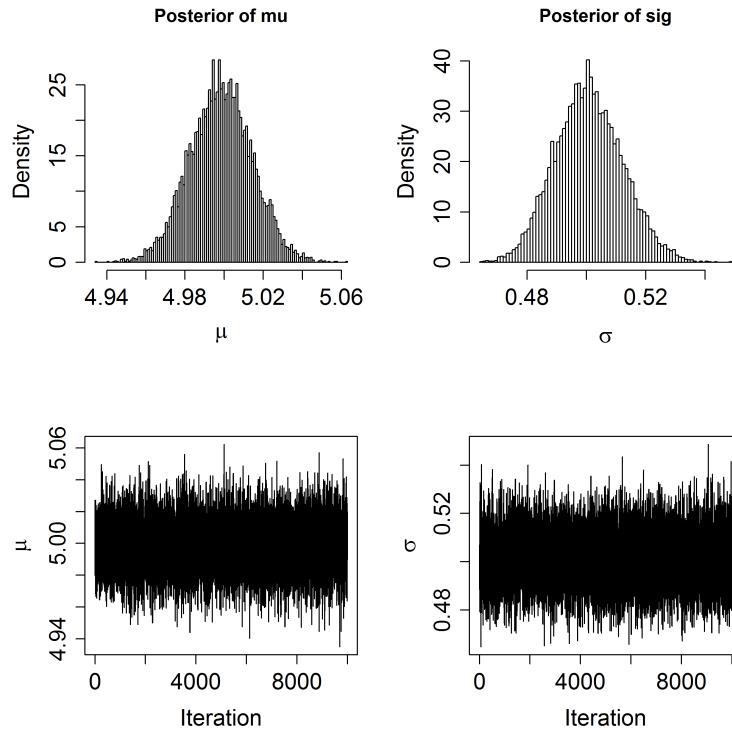


Figure 17.2. Plots of the posterior distributions of μ (upper left) and σ (upper right) from a Normal model and time series plots of μ (lower left) and σ (lower right).

13145 condition. Since using a symmetric proposal distribution makes life a little easier, we are
 13146 going to focus on this specific case. A type of symmetric proposal useful in many situations
 13147 is the so-called *random-walk* proposal distribution where candidate values are drawn from
 13148 a normal distribution with mean equal to the current value and some standard deviation,
 13149 say δ , which is prescribed by the user (see below for further explanation).

13150 **Parameters with bounded support:** Many models contain parameters that have
 13151 bounded support. E.g., variance parameters live on $[0, \infty]$, parameters that represent
 13152 probabilities live on $[0, 1]$, etc.. For such cases, it is sometimes convenient to use a random
 13153 walk proposal distribution that can generate any real number (e.g., a normal random walk
 13154 proposal). Under these circumstances you should not constrain the proposal distribution
 13155 itself, but you can just reject parameters that are outside of the parameter space (sec. 6.4.1
 13156 in Robert and Casella, 2010). You will see plenty of examples of updating parameters with
 13157 bounded support in this chapter.

13158 It is worth knowing that there are alternatives to the random walk MH algorithm.

13159 For example, in the independent MH, θ^* does not depend on θ^{t-1} , while the Langevin
 13160 algorithm (Roberts and Rosenthal, 1998) aims at avoiding the random walk by favoring
 13161 moves towards regions of higher posterior probability density. The interested reader should
 13162 look up these algorithms in Robert and Casella (2004) or Robert and Casella (2010).

13163 Building a MH sampler can be broken down into several steps. We are going to
 13164 demonstrate these steps using a different but still simple and common model: the logit-
 13165 normal or logistic regression model. For simplicity, assume that

$$y \sim \text{Bernoulli} \left(\frac{\exp(\theta)}{1 + \exp(\theta)} \right)$$

13166 and

$$\theta \sim \text{Normal}(\mu, \sigma)$$

13167 The following steps are required to set up a random walk MH algorithm:

13168 **Step 0:** Choose initial values, $\theta^{(0)}$.

13169 **Step 1:** Generate a proposed value of θ from $h(\theta^* | \theta^{t-1})$. We often use a Normal proposal
 13170 distribution, so we draw $\theta^{(1)}$ from $\text{Normal}(\theta^{(0)}, \delta)$, where δ is the variance of the Normal
 13171 proposal distribution, the tuning parameter that we have to set.

13172 **Step 2:** Calculate the ratio of posterior densities for the proposed and the original value
 13173 for θ :

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

13174 In our example,

$$r = \frac{\text{Bernoulli}(y | \theta^*) \times \text{Normal}(\theta^* | \mu, \sigma)}{\text{Bernoulli}(y | \theta^{t-1}) \times \text{Normal}(\theta^{t-1} | \mu, \sigma)}$$

13175 **Step 3:** Set

$$\begin{aligned} \theta^t &= \theta^* \text{ with probability } \min(r, 1) \\ &= \theta^{t-1} \text{ otherwise} \end{aligned}$$

13176 We can do this last step by drawing a random number u from a $\text{Uniform}(0, 1)$ and
 13177 accept θ^* if $u < r$. Repeat for $t = 1, 2, \dots$ a large number of samples. The **R** code for this
 13178 MH sampler is provided in Panel 17.2.

13179 The reason we sum the logs of the likelihood and the prior, rather than multiplying
 13180 the original values, is simply computational. The product of small probabilities can be
 13181 numbers very close to 0, which computers do not handle well. Thus we add the logarithms,
 13182 sum, and exponentiate to achieve the desired result. Similarly, in case you have forgotten,
 13183 $x/y = \exp(\log(x) - \log(y))$, with the latter being favored for computational reasons.

13184 Comparing MH sampling to Gibbs sampling, where all draws from the conditional
 13185 distribution are used, in the MH algorithm we discard a portion of the candidate values,
 13186 which inherently makes it less efficient than Gibbs sampling – the price you pay for its
 13187 increased generality. In Step 1 of the MH sampler we had to choose a variance, δ , for
 13188 the Normal proposal distribution. Choice of the parameters that define our candidate
 13189 distribution is also referred to as ‘tuning’, and it is important since adequate tuning will

make your algorithm more efficient. δ should be chosen (a) large enough so that each step of drawing a new proposal value for θ can cover a reasonable distance in the parameter space, as otherwise, mixing of the Markov chain is inefficient and chains will tend to have strong autocorrelation; and (b) small enough so that proposal values are not rejected too often, as otherwise the random walk will 'get stuck' at specific values for too long. As a rule of thumb, your candidate value should be accepted in about 40% of all cases. Acceptance rates of 20 – 80% are probably ok, but anything below or above may well render your algorithm inefficient (this does not mean that it will give you wrong results, only that you will need more iterations to converge to the posterior distribution). In practice, tuning will require some 'trial-and-error', some common sense and, with enough experience, some intuition. Or, one can use an adaptive phase, where the tuning parameter is automatically adjusted until it reaches a user-defined acceptance rate, at which point the adaptive phase ends and the actual Markov chain begins. This is computationally a little more advanced. Link and Barker (2010) discuss this in more detail. It is important the samples drawn during the adaptive phase are discarded. To illustrate the effects of tuning, we ran the Metropolis-within-Gibbs algorithm in Panel 17.2 with $\delta = 0.01$, $\delta = 0.2$ and $\delta = 1$. The first 150 iterations for θ are shown in Fig. 17.3. We see that for a very small δ (the dashed line) the burn-in is extremely slow - after 150 iterations the chain isn't even half way there, while for the other two values of δ (solid and dotted) the burn-in phase seems to be over after only about 10 iterations. While $\delta = 0.2$ leads to reasonably good mixing, the chain clearly gets stuck on certain values with $\delta = 1$.

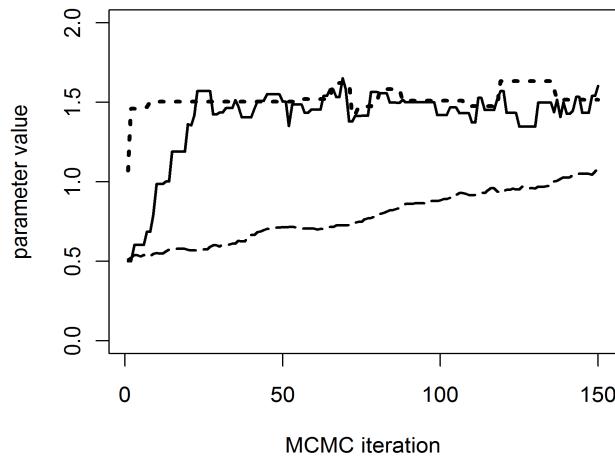


Figure 17.3. Time series plots of θ from a MH algorithm with tuning parameter $\delta = 0.01$ (dashed line), 0.2 (solid line) and 1 (dotted line).

13210 Other than graphically, you can easily check acceptance rates for the parameters you
 13211 monitor (that are part of your output) using the `rejectionRate()` function of the package
 13212 `coda` (we will talk more about this package a little later on). Do not let the term 'rejection
 13213 rate' confuse you; it is simply 1 – acceptance rate. There may be parameters – for example,
 13214 individual values of a random effect or latent variables – that you do not want to save,
 13215 though, and in our next example we will show you a way to monitor their acceptance rates
 13216 with a few extra lines of code.

13217 17.2.3 Metropolis-within-Gibbs

13218 One weakness of the MH sampler is that formulating the joint posterior when evaluating
 13219 whether to accept or reject the candidate values for θ becomes increasingly complex or
 13220 inefficient as the number of parameters in a model increases. As you already saw in
 13221 Chapt. 3, in these cases you can simply combine MH sampling and Gibbs sampling.
 13222 You can use Gibbs sampling to break down your high-dimensional parameter space into
 13223 easy-to-handle one-dimensional conditional distributions and use MH sampling for these
 13224 conditional distributions. Better yet, if you have some conjugacy in your model, you can
 13225 use the more efficient Gibbs sampling for these parameters and one-dimensional MH for all
 13226 the others. You have already seen the basics of how to build both types of algorithms, so
 13227 we can jump straight into an example here and build a Metropolis-within-Gibbs algorithm.

13228 **GLMMs: Poisson regression with a random effect** Let's assume a model that gets
 13229 us closer to the problem we ultimately want to deal with - a GLMM. Here, we assume
 13230 we have Poisson counts, y_{ij} , from $j = 1, 2, \dots, n$ plots in i different study sites, and we
 13231 believe that the counts are influenced by some plot-specific covariate, \mathbf{x} , but that there is
 13232 also a random site effect. So our model is:

$$y_{ij} \sim \text{Poisson}(\lambda_{ij})$$

$$\lambda_{ij} = \exp(\alpha_i + \beta x_{ij})$$

13234 Let's use Normal priors on α and β ,

$$\alpha_i \sim \text{Normal}(\mu_\alpha, \sigma_\alpha)$$

13235 and

$$\beta \sim \text{Normal}(\mu_\beta, \sigma_\beta)$$

13236 In this model, we do not specify μ_α and σ_α , but instead, estimate them as well, so we
 13237 have to specify hyperpriors for these parameters:

$$\mu_\alpha \sim \text{Normal}(\mu_0, \sigma_0)$$

$$\sigma_\alpha^2 \sim \text{Inverse-Gamma}(a_0, b_0)$$

13238 Note that for simplicity we assume that β is constant across the i study sites, and for
 13239 analysis we would set μ_β and σ_β . With the model completely specified, we can compile
 13240 the full conditionals, breaking the multi-dimensional parameter space into one-dimensional
 13241 components:

$$\begin{aligned} [\alpha_1 | \alpha_2, \alpha_3, \dots, \alpha_i, \beta, \mathbf{y}_1] &\propto [\mathbf{y}_1 | \alpha_1, \beta][\alpha_1] \\ &\propto \text{Poisson}(\mathbf{y}_1 | \exp(\alpha_1 + \beta \mathbf{x}_1)) \times \text{Normal}(\alpha_1 | \mu_\alpha, \sigma_\alpha) \end{aligned}$$

13242 where $\mathbf{y}_1 = (y_{11}, y_{12}, \dots, y_{1n})$ is the vector of observed counts for site $i = 1$ and, in general,
 13243 \mathbf{y}_i is the vector of all counts for site i ; analogous, \mathbf{x}_i is the vector of all observations of
 13244 the covariate for site i . The other full conditionals for each α_i are constructed similarly:

$$\begin{aligned} [\alpha_2 | \alpha_1, \alpha_3, \dots, \alpha_i, \beta, \mathbf{y}_2] &\propto [\mathbf{y}_2 | \alpha_2, \beta][\alpha_2] \\ &\propto \text{Poisson}(\mathbf{y}_2 | \exp(\alpha_2 + \beta \mathbf{x}_2)) \times \text{Norm}(\alpha_2 | \mu_\alpha, \sigma_\alpha) \end{aligned}$$

13245 and so on for all elements of α . The full-conditional for β is:

$$\begin{aligned} [\beta | \alpha, \mathbf{y}] &\propto [\mathbf{y} | \alpha, \beta][\beta] \\ &\propto \text{Poisson}(\mathbf{y} | \exp(\alpha + \beta \mathbf{x})) \times \text{Normal}(\beta | \mu_\beta, \sigma_\beta) \end{aligned}$$

13246 Finally, we need to update the hyperparameters for the random effects vector α :

$$\begin{aligned} [\mu_\alpha | \alpha] &\propto [\alpha | \mu_\alpha, \sigma_\alpha][\mu_\alpha] \\ [\sigma_\alpha | \alpha] &\propto [\alpha | \mu_\alpha, \sigma_\alpha][\sigma_\alpha] \end{aligned}$$

13248 Since we assumed α to come from a Normal distribution, the choice of priors for μ_α
 13249 (Normal) and σ_α^2 (Inverse-Gamma) leads to the same conjugacy we observed in our initial
 13250 Normal model, so that both hyperparameters can be updated using Gibbs sampling.

13251 Now let's build the updating steps for these full conditionals. Again, for the MH steps
 13252 that update α and β we use Normal proposal distributions with standard deviations δ_α
 13253 and δ_β .

13254 First, we set the initial values $\alpha^{(0)}$ and $\beta^{(0)}$. Then, starting with α_1 , we draw $\alpha_1^{(1)}$ from
 13255 $\text{Norm}(\alpha_1^{(0)}, \delta_\alpha)$, calculate the conditional posterior density of $\alpha_1^{(0)}$ and $\alpha_1^{(1)}$ and compare
 13256 their ratios,

$$r = \frac{\text{Poisson}(\mathbf{y}_1 | \exp(\alpha_1^{(1)} + \beta \mathbf{x}_1)) \times \text{Normal}(\alpha_1^{(1)} | \mu_\alpha, \sigma_\alpha)}{\text{Poisson}(\mathbf{y}_1 | \exp(\alpha_1^{(0)} + \beta \mathbf{x}_1)) \times \text{Normal}(\alpha_1^{(0)} | \mu_\alpha, \sigma_\alpha)}$$

13257 and accept $\alpha_1^{(1)}$ with probability $\min(r, 1)$. We repeat this for all α .

13258 For β , we draw $\beta^{(1)}$ from $\text{Norm}(\beta^{(0)}, \delta_\beta)$, compare the posterior densities of $\beta^{(0)}$ and
 13259 $\beta^{(1)}$,

$$r = \frac{\text{Poisson}(\mathbf{y} | \exp(\alpha + \beta^{(1)} \mathbf{x})) \times \text{Normal}(\beta^{(1)} | \mu_\beta, \sigma_\beta)}{\text{Poisson}(\mathbf{y} | \exp(\alpha + \beta^{(0)} \mathbf{x})) \times \text{Normal}(\beta^{(0)} | \mu_\beta, \sigma_\beta)},$$

13260 and accept $\beta^{(1)}$ with probability $\min(r, 1)$.

13261 For μ_α and σ_α^2 , we sample directly from the full conditional distributions (Eq. 17.1.2
 13262 and Eq. 17.2.1):

$$\mu_\alpha^{(1)} \sim \text{Norm}(\mu_n, \sigma_n^2)$$

13263 where

$$\mu_n = \frac{\sigma_\alpha^{2(0)}}{\sigma_\alpha^{2(0)} + n_\alpha \sigma_0^2} \times \mu_0 + \frac{n_\alpha \sigma_0^2}{\sigma_\alpha^{2(0)} + n_\alpha \sigma_0^2} \times \bar{\alpha}^{(1)}$$

13264 and

$$\sigma_n^2 = \frac{\sigma_\alpha^{2(0)}\sigma_0}{\sigma_\alpha^{2(0)} + n\sigma_0^2}$$

13265 Here, $\bar{\alpha}$ is the current mean of the vector α , which we updated before, and n_α is the
 13266 length of α . For σ_α^2 we use $\sigma_\alpha^{2(1)} \sim \text{Inverse-Gamma}(a_n, b_n)$, where $a_n = n_\alpha/2 + a_0$, and
 13267 $b_n = 0.5 \sum_{i=1}^{n_\alpha} (\alpha_i^{(1)} - \mu_\alpha^{(1)})^2 + b_0$.

13268 We repeat these steps over T iterations of the MCMC algorithm. Call the function
 13269 `PoisGLMM()` in `scrbook` to check out what this algorithm looks like in **R**.

13270 In this example we may not want to save each individual α , but are only interested in
 13271 their mean and standard deviation. Since these two parameters will change as soon as the
 13272 value for one element in α changes, their acceptance rates will always be close to 1 and
 13273 are not representative of how well your algorithm performs. To monitor the acceptance
 13274 rates of parameters you do not want to save, you simply need to add a few lines of code
 13275 into your updater to see how often the individual parameters are accepted. The code for
 13276 updating α from our Poisson GLMM below shows one way how to monitor acceptance of
 13277 individual α 's.

```

13278 #initiate counter for acceptance rate of alpha
13279 alphaUps<-0
13280
13281 #loop over sites, update intercepts alpha one at a time;
13282 #only data at site i contributes information
13283 #lev is the number of sites i
13284 for (i in 1:lev) {
13285   alpha.cand<-rnorm(1, alpha[i], delta_alpha)
13286   loglike<- sum(dpois (y[site==i], exp(alpha[i] + beta*x[site==i]),
13287     log=TRUE))
13288   logprior<- dnorm(alpha[i], mu_alpha,sig_alpha, log=TRUE)
13289   loglike.cand<- sum(dpois (y[site==i], exp(alpha.cand + beta *x[site==i])),
13290     log=TRUE))
13291   logprior.cand<- dnorm(alpha.cand, mu_alpha,sig_alpha, log=TRUE)
13292   if (runif(1)< exp((loglike.cand+logprior.cand) -(loglike+logprior))) {
13293     alpha[i]<-alpha.cand
13294     alphaUps<-alphaUps+1
13295   }
13296 }
13297
13298 #lets you check the acceptance rate of alpha at every 100th iteration
13299 if(iter %% 100 == 0) {
13300   cat("    Acceptance rates\n")
13301   cat("      alpha =", alphaUps/lev, "\n")
13302 }
```

13303 17.2.4 Rejection sampling and slice sampling

13304 While MH and Gibbs sampling are probably the most widely applied algorithms for posterior approximation, there are other options that work under certain circumstances and
 13305 may be more efficient when applicable. **WinBUGS** applies these algorithms and we want
 13306 you to be aware that there is more out there to approximate posterior distributions than
 13307 Gibbs and MH. One alternative algorithm is rejection sampling. Rejection sampling is
 13308 not an MCMC method, since each draw is independent of the others. The method can
 13309 be used when the posterior $[\theta|y]$ is not a known parametric distribution but can be ex-
 13310 pressed in closed form. Then, we can use a so-called envelope function, say, $g(\theta)$, that
 13311 we can easily sample from, with the restriction that $[\theta|y] < M \times g(\theta)$. We then sample a
 13312 candidate value for θ from $g(\theta)$, calculate $r = [\theta|y]/M \times g(\theta)$ and keep the sample with
 13313 the probability r . M is a constant that has to be picked so that r lies between 0 and 1, for
 13314 example by evaluating both $[\theta|y]$ and $g(\theta)$ at n points and looking at their ratios. Rejec-
 13315 tion sampling only works well if $g(\theta)$ is similar to $[\theta|y]$, and packages like **WinBUGS** use
 13316 adaptive rejection sampling (Gilks and Wild, 1992), where a complex algorithm is used to
 13317 fit an adequate and efficient $g(\theta)$ based on the first few draws. Though efficient in some
 13318 situations, rejection sampling does not work well with high-dimensional problems, since
 13319 it becomes increasingly hard to define a reasonable envelope function. For an example
 13320 of rejection sampling in the context of SCR models, see Chapt. 11, where we use it to
 13321 simulation non-stationary point processes.

13322 Another alternative is slice sampling (Neal, 2003). In slice sampling, we sample uni-
 13323 formly from the area under the plot of $[\theta|y]$. Considering a single univariate θ . Let's define
 13324 an auxiliary variable, $U \sim \text{Unif}(0, [\theta|y])$. Then, θ can be sampled from the vertical slice
 13325 of $[\theta|y]$ at U (Fig. 17.4):

$$\theta|U \sim \text{Unif}(B),$$

13326 where $B = \{\theta : [\theta|y] \geq U\}$

13327 Slice sampling can be applied in many situations; however, implementing an efficient
 13328 slice sampling procedure can be complicated. We refer the interested reader to Robert and
 13329 Casella (2010, Chapt. 7) for a simple example. Both rejection sampling and slice sampling
 13330 can be applied on one-dimensional conditional distributions within a Gibbs sampling setup.

17.3 MCMC FOR CLOSED CAPTURE-RECAPTURE MODEL MH

13331 By now you have seen MCMC samplers for some simple GL(M)M's. Now, to ease you
 13332 into more complex models, we construct our own MCMC algorithm using a Metropolis-
 13333 within-Gibbs sampler for the non-spatial model with individual heterogeneity in capture
 13334 probability, model M_h , developed in Chapt. 4.

13335 To recapitulate: Under the non-spatial model, each of the n observed individuals is
 13336 either detected (1) or not (0) during each of K sampling occasions. We estimate N using
 13337 data augmentation and have a Bernoulli model for the data augmentation variables z_i .

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

13338 The binomial observation model is expressed conditional on the latent variables z_i .

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

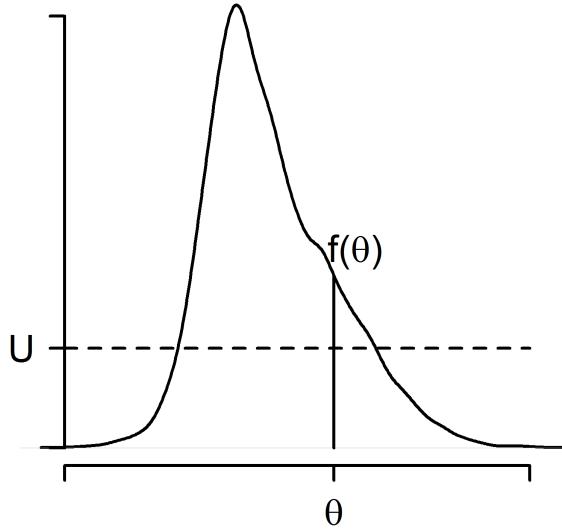


Figure 17.4. Slice sampling. For $U \sim \text{Unif}(0, [\theta|y])$, we can sample θ from the vertical slice of $[\theta|y]$ at U ; $\theta|U \sim \text{Unif}(B)$, where $B = \{\theta : [\theta|y] \geq U\}$.

13340 Further, we prescribe a distribution for the capture probability p_i . Here we assume

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

13341 As usual, we have to go through two general steps before we write the MCMC algo-
13342 rithm:

- 13343 (1) Identify the model with all its components (including priors)
13344 (2) Recognize and express the full conditional distributions for all parameters

13345 Our model components are as follows: $[y_i|p_i, z_i]$, $[p_i|\mu_p, \sigma_p]$, and $[z_i|\psi]$ for each $i =$
13346 $1, 2, \dots, M$ and then prior distributions $[\mu_p]$, $[\sigma_p]$ and $[\psi]$. The joint posterior distri-
13347 bution of all unknown quantities in the model is proportional to the joint distribution of
13348 all elements y_i, p_i, z_i and also the prior distributions of the prior parameters:

$$\left\{ \prod_{i=1}^M [y_i|p_i, z_i] [p_i|\mu_p, \sigma_p] [z_i|\psi] \right\} [\mu_p, \sigma_p, \psi]$$

13349 For prior distributions, we assume that μ_p, σ_p, ψ are mutually independent and for μ_p and
13350 σ_p we use improper uniform priors, and $\psi \sim \text{Unif}(0, 1)$. This is equivalent to Beta(1, 1),
13351 which will come in handy, as we will see in a moment. Note that the likelihood contribution
13352 for each individual, when conditioned on p_i and z_i , does not depend on ψ , μ_p , or σ_p . As

such, the full-conditional for the structural parameter ψ only depend on the collection of data augmentation variables z_i , and that for μ_p and σ_p will only depend on the collection of latent variables $p_i; i = 1, 2, \dots, M$. The full conditionals for all the unknowns are as follows:

(1) For p_i :

$$\begin{aligned} [p_i|y_i, \mu_p, \sigma_p, z_i] &\propto [y_i|p_i][p_i|\mu_p, \sigma_p^2] \text{ if } z_i = 1 \\ &\quad [p_i|\mu_p, \sigma_p] \text{ if } z_i = 0 \end{aligned}$$

(2) for z_i :

$$[z_i|y_i, p_i, \psi] \propto [y_i|z_i \times p_i]\text{Bernoulli}(z_i|\psi)$$

(3) For μ_p :

$$[\mu_p|p_i, \sigma_p] \sim \left\{ \prod_i [p_i|\mu_p, \sigma_p] \right\} \times \text{const}$$

(4) For σ_p :

$$[\sigma_p|p_i, \mu_p] \sim \left\{ \prod_i [p_i|\mu_p, \sigma_p] \right\} \times \text{const}$$

(5) For ψ :

$$[\psi|z_i] \propto \left\{ \prod_i [z_i|\psi] \right\} \text{Beta}(1, 1)$$

Remember that Beta(1,1) is equivalent to Uniform(0,1). The beta distribution is the conjugate prior to the binomial and Bernoulli distributions and the general form of a full conditional of a beta-binomial model with $x_i \sim \text{Bernoulli}(p)$ and $p \sim \text{Beta}(a, b)$ is

$$[p|\mathbf{x}] \propto \text{Beta}(a + \sum_i x_i, b + n - \sum_i x_i)$$

In our case that means

$$[\psi|z_i] \propto \text{Beta}(1 + \sum z_i, 1 + M - \sum z_i)$$

What we've done here is identify each of the full conditional distributions in sufficient detail to toss them into our Metropolis-Hastings algorithm (the constant term in the full conditionals for μ_p and σ_p reflects the improper prior we chose for both parameters). Below, you see the updating step for the detection parameter \mathbf{p} . Note that (1) we draw candidate values on the logit scale and (2) instead of looping through $1 - M$ individuals to update all p_i , we update all elements of the vector of \mathbf{p} in parallel.

```
13372  ### update the logit(p) parameters
13373  lp.cand<- rnorm(M,lp,1) # 1 is a tuning parameter
13374  p.cand<-plogis(lp.cand)
13375  ll<-dbinom(ytot,K,z*p, log=T)
13376  prior<-dnorm(lp,mu,sigma, log=T)
13377  llcand<-dbinom(ytot,K,z*p.cand, log=T)
13378  prior.cand<-dnorm(lp.cand,mu,sigma, log=T)
```

```

13379
13380 kp<- runif(M) < exp((llcand+prior.cand)-(ll+prior))
13381 p[kp]<-p.cand[kp]
13382 lp[kp]<-lp.cand[kp]

```

13383 The parameters μ_p and σ_p are also updated using MH steps (see the code for μ_p
13384 below). In truth, we could also sample μ_p and σ_p^2 directly with certain choices of prior
13385 distributions. For example, if $\mu_p \sim \text{Normal}(0, 1000)$ then the full conditional for μ_p is also
13386 Normal (see sec. 17.2.1), etc..

```

13387 p0.cand<- rnorm(1,p0,.05)
13388 if(p0.cand>0 & p0.cand<1){
13389   mu.cand<-log(p0.cand/(1-p0.cand))
13390   ll<-sum(dnorm(lp,mu,sigma,log=TRUE))
13391   llcand<-sum(dnorm(lp,mu.cand,sigma,log=TRUE))
13392   if(runif(1)<exp(llcand-ll)) {
13393     mu<-mu.cand
13394     p0<-p0.cand
13395   }
13396 }

```

13397 For ψ we can easily sample directly from the beta distribution:

```
13398 psi<-rbeta(1, sum(z) + 1, M-sum(z) + 1)
```

13399 To update the z_i we have opted for a MH updater (although they could be updated
13400 directly from their full-conditional). Since z_i can only take the values of 0 or 1, we generate
13401 candidate values using `z.cand<-ifelse(z==1,0,1)`. You can check out the full code by
13402 invoking `modelMh()` from the R package `scrbook`.

17.4 MCMC ALGORITHM FOR MODEL SCR0

13403 Conceptually, but also in terms of MCMC coding, it is only a small step from the non-
13404 spatial model M_h to a fully spatial capture-recapture model. Next, we'll walk you through
13405 the steps of building your own MCMC sampler for the basic SCR model (i.e. without any
13406 individual, site or time specific covariates) with both a Poisson and a Binomial encounter
13407 process. As usual, we will have to go through two general steps before we write the MCMC
13408 algorithm:

- 13409 (1) Identify the model with all its components (including priors)
- 13410 (2) Recognize and express the full conditional distributions for all parameters

13411 It is worthwhile to go through all of step 1 for an SCR model, but you have probably
13412 seen enough of step 2 in our previous examples to get the essence of how to express a full
13413 conditional distribution. Therefore, we will exemplify step 2 for some parameters and tie
13414 these examples directly to the respective R code.

13415 **Step 1 – Identify your model**

13416 Recall the components of the basic SCR model with a Poisson encounter process from
 13417 Chapt. 9: We assume that individuals i , or rather, their activity centers \mathbf{s}_i , are uniformly
 13418 distributed across the state space \mathcal{S} ,

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

13419 and that the number of times individual i encounters trap j , y_{ij} , is a Poisson variable
 13420 with mean λ_{ij} ,

$$y_{ij} \sim \text{Poisson}(\lambda_{ij})$$

13421 The link between individual location, movement and trap encounter rates is made by the
 13422 assumption that λ_{ij} , is a decreasing function of the distance between \mathbf{s}_i and the location
 13423 of j , \mathbf{x}_j , say $d_{ij} = \|\mathbf{s}_i - \mathbf{x}_j\|$, of the half-normal form

$$\lambda_{ij} = \lambda_0 \exp(-d_{ij}^2/2\sigma^2),$$

13424 where λ_0 is the baseline trap encounter rate at $d_{ij} = 0$ and σ controls the shape of the
 13425 half-normal function.

13426 In order to estimate the number of \mathbf{s}_i in \mathcal{S} (or any subset of \mathcal{S}), N , we use data
 13427 augmentation (sec. 4.2) and create $M - n$ all-zero encounter histories, where n is the
 13428 number of individuals we observed and M is a somewhat arbitrary number that is larger
 13429 than N . We estimate N by summing over the auxiliary data augmentation variables, z_i ,
 13430 which is 1 if the individual is part of the population and 0 if not, and assume that z_i is a
 13431 Bernoulli random variable,

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

13432 To link the two model components, we modify our trap encounter model to

$$\lambda_{ij} = \lambda_0 \times \exp(-d_{ij}^2/2\sigma^2) \times z_i.$$

13433 The model has the following structural parameters, for which we need to specify priors:

13434 ψ : the $\text{Uniform}(0, 1)$ is required as part of the data augmentation procedure and in general
 13435 is a natural choice of an uninformative prior for a probability. It will also lead to
 13436 conjugacy as we saw in the example of model M_h , so that we can update ψ directly
 13437 from its full conditional distribution using Gibbs sampling.

13438 \mathbf{s}_i : since \mathbf{s}_i is a pair of coordinates it is two-dimensional and we use a uniform prior
 13439 limited by the extent of our state-space over both dimensions.

13440 σ : we can conceive several priors for σ but let's assume an improper prior, one that is
 13441 Uniform over $(-\infty, \infty)$. We will see why this is convenient when we construct the full
 13442 conditionals for σ .

13443 λ_0 : analogous, we will use a $\text{Uniform}(-\infty, \infty)$ improper prior for λ_0 .

13444 The parameter that is the objective of our modeling, N , is a derived parameter that we
 13445 can obtain by summing all z_i :

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

13446 **Step 2 – Construct the full conditionals:** Having completed step 1, let's look at
 13447 the full conditional distributions for some of these parameters. We find that with improper

13448 priors, full conditionals are proportional only to the likelihood of the observations; for
 13449 example, consider σ :

$$[\sigma|\mathbf{s}, \lambda_0, \mathbf{z}, \mathbf{y}] \propto \left\{ \prod_i [y_i|\mathbf{s}_i, \lambda_0, z_i, \sigma] \right\} [\sigma]$$

13450 Since the improper prior implies that $[\sigma] \propto 1$, we can reduce this further to

$$[\sigma|\mathbf{s}, \lambda_0, \mathbf{z}, \mathbf{y}] \propto \left\{ \prod_i [y_i|\mathbf{s}_i, \lambda_0, z_i, \sigma] \right\}$$

13451 The R code to update σ is shown below. Notice that we automatically reject negative
 13452 candidate values, since σ cannot be < 0 .

```
13453 sig.cand <- rnorm(1, sigma, 0.1) #draw candidate value
13454 if(sig.cand>0){ #automatically reject sig.cand that are <0
13455   lam.cand <- lam0*exp(-(d*d)/(2*sig.cand*sig.cand))
13456   ll<- sum(dpois(y, lam*z, log=TRUE))
13457   llcand <- sum(dpois(y, lam.cand*z, log=TRUE))
13458   if(runif(1) < exp( llcand - ll ) ){
13459     ll<-llcand
13460     lam<-lam.cand
13461     sigma<-sig.cand
13462   }
13463 }
```

13464 These steps are analogous for λ_0 and \mathbf{s}_i and we will use MH steps for all of these
 13465 parameters. Similar to the random intercepts in our Poisson GLMM, we update each
 13466 \mathbf{s}_i individually. Note that to be fully correct, the full conditional for \mathbf{s}_i contains both
 13467 the likelihood and prior component, since we did not specify an improper, but a proper
 13468 Uniform prior on \mathbf{s}_i . However, with a Uniform distribution the probability density of
 13469 any value is $1/(\text{upper limit} - \text{lower limit}) = \text{constant}$. Thus, the prior components are
 13470 identical for both the current and the candidate value and can be ignored (formally, when
 13471 you calculate the ratio of posterior densities, r , the identical prior component appears
 13472 both in the numerator and denominator, so that they cancel each other out).

13473 We still have to update z_i . The full conditional for z_i is

$$[z_i|y_i, \sigma, \lambda_0, \mathbf{s}_i] \propto [y_i|z_i, \sigma, \lambda_0, \mathbf{s}_i][z_i]$$

13474 and since $z_i \sim \text{Bern}(\psi)$, the term has to be taken into account when updating z_i :

```
13475 zUps <- 0 #set counter to monitor acceptance rate
13476 for(i in 1:M) {
13477   #no need to update seen individuals, since their z =1
13478   if(seen[i])
13479     next
13480   zcand <- ifelse(z[i]==0, 1, 0)
13481   llz <- sum(dpois(y[i,,],lam[i,]*z[i], log=TRUE))
```

```

13482     llcand <- sum(dpois(y[i,], lam[i,]*zcand, log=TRUE))
13483
13484     prior <- dbinom(z[i], 1, psi, log=TRUE)
13485     prior.cand <- dbinom(zcand, 1, psi, log=TRUE)
13486     if(runif(1) < exp((llcand+prior.cand)-(llz+prior))){ 
13487         z[i] <- zcand
13488         zUps <- zUps+1
13489     }
13490 }
```

13491 The parameter ψ is a hyperparameter of the model, with an uninformative prior distribution of Uniform(0, 1) or Beta(1, 1), so that

$$[\psi|\mathbf{z}] \propto \text{Beta}\left(1 + \sum_i z_i, 1 + M - \sum_i z_i\right)$$

13493 These are all the building blocks you need to write the MCMC algorithm for the spatial
13494 null model with a Poisson encounter process. You can find the full **R** code by calling the
13495 function (**SCR0pois**) in the **R** package **scrbook**.

13496 17.4.1 SCR model with binomial encounter process

13497 The equivalent SCR model with a binomial encounter process is very similar. Here, each
13498 individual i can only be detected once at any given trap j during a sampling occasion k .
13499 Thus

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

13500 Where p_{ij} is some function of distance between \mathbf{s}_i and trap location \mathbf{x}_j . Here we use:

$$p_{ij} = 1 - \exp(-\lambda_{ij})$$

13501 Recall from Chapt. 3 that this is the complementary log-log (cloglog) link function,
13502 which constrains p_{ij} to fall between 0 and 1. For our MCMC algorithm that means that,
13503 instead of using a Poisson likelihood, $\text{Poisson}(y|\sigma, \lambda_0, \mathbf{s}, \mathbf{z})$, we use a Binomial likelihood,
13504 $\text{Binomial}(y|\sigma, \lambda_0, \mathbf{s}, \mathbf{z}; K)$, in all the conditional distributions. An exemplary updating
13505 step for λ_0 under a Binomial encounter model is shown below. The full MCMC code for
13506 the Binomial SCR with a clog-log link (**SCR0binom.cl**) can be found in the **R** package
13507 **scrbook**.

```

13508     lam0.cand <- rnorm(1, lam0, 0.1)
13509     #automatically reject lam0.cand that are <0
13510     if(lam0.cand >0){
13511         lam.cand <- lam0.cand*exp(-(d*d)/(2*sigma*sigma))
13512         p.cand <- 1-exp(-lam.cand)
13513         ll<- sum(dbinom(y, K, pmat *z, log=TRUE))
13514         llcand <- sum(dbinom(y, K, p.cand *z, log=TRUE))
13515         if(runif(1) < exp( llcand - ll) ){
13516             ll<-llcand
```

```

13517           pmat<-p.cand
13518           lam0<- lam0.cand
13519       }
13520   }

```

13521 Another possibility is to model variation in the individual and site specific detection
 13522 probability, p_{ij} , directly, without any transformation, such that

$$p_{ij} = p_0 \times \exp(-d_{ij}^2/(2\sigma^2))$$

13523 and $p_0 \in [0, 1]$. This formulation is analogous to how detection probability is modeled in
 13524 distance sampling under a half-normal detection function; however, in distance sampling
 13525 p_0 – detection of an individual on the transect line – is assumed to be 1 (Buckland et al.,
 13526 2001). Under this formulation the updater for p_0 becomes:

```

13527   p0.cand <- rnorm(1, p0, 0.1)
13528   if(p0.cand > 0 & p0.cand < 1 ){
13529     #automatically rejects lam0.cand that are not {0,1}
13530     p.cand <- p0.cand*exp(-(d*d)/(2*sigma*sigma))
13531     ll<- sum(dbinom(y, K, pmat *z, log=TRUE))
13532     llcand <- sum(dbinom(y, K, p.cand *z, log=TRUE))
13533     if(runif(1) < exp( llcand - ll ) ){
13534       ll<-llcand
13535       pmat<-p.cand
13536       p0<- p0.cand
13537     }
13538   }

```

13539 17.4.2 Looking at model output

13540 Now that you have an MCMC algorithm to analyze spatial capture-recapture data with,
 13541 let's run an actual analysis so we can look at the output. As an example, we will use the
 13542 Fort Drum bear data set we first introduced in Chapt. 1 and already analyzed in Chapt.
 13543 4 with traditional non-spatial models (and that you will see again in Chapt. 7). You can
 13544 load the Fort Drum data (`data(beardata)`), extract the trap locations (`trapmat`) and
 13545 detection data (`bearArray`) and build the augmented $M \times J$ array of individual encounter
 13546 histories:

```

13547 M=700
13548 trapmat<-beardata$trapmat
13549 #summarizes captures across occasions
13550 bearmat<-apply(beardata$bearArray, 1:2, sum)
13551 Xaug<-matrix(0, nrow=M, ncol=dim(trapmat)[1])
13552 Xaug[1:dim(bearmat)[1],]<-bearmat #create augmented data set

```

13553 In addition to these data, we need to specify the outermost coordinates of the state-
 13554 space. Since bears are wide ranging animals we add a 20-km buffer to the maximum and
 13555 minimum coordinates of the trap array:

```

13556 xl<- min(trapmat[,1])- 20
13557 yl<- min(trapmat[,2])- 20
13558 xu<- max(trapmat[,1])+ 20
13559 yu<- max(trapmat[,2])+ 20

```

13560 Finally, use the MCMC code for the binomial encounter model with the clog-log link
 13561 (`SCR0binom.cl`) and run 5000 iterations. This should take approximately 25 minutes (in
 13562 real life we would of course run the algorithm a lot longer but for demonstration purposes
 13563 let's stick with a number of iterations that can be run in a manageable amount of time).

```

13564 set.seed(13)
13565 mod0<-SCR0binom.cl(y=Xaug, X=trapmat, M=M, xl=xl, xu=xu, yl=yl,
13566 yu=yu, K=8, delta=c(0.1, 0.05, 2), niter=5000)

```

13567 Before, we used simple **R** commands to look at model results. However, there is a
 13568 specific **R** package to summarize MCMC simulation output and perform some convergence
 13569 diagnostics – package `coda` (Plummer et al., 2006). Download and install `coda`, then
 13570 convert your model output to an `mcmc` object

```

13571 chain<-mcmc(mod0)

```

13572 which can be used by `coda` to produce MCMC specific output.

13573 **Markov chain time series plots**

13574 Start by looking at time series plots of your Markov chains using `plot(chain)`. This com-
 13575 mand produces a time series plot and marginal posterior density plots for each monitored
 13576 parameter, similar to what we did before using the `hist()` and `plot()` commands. Fig.
 13577 shows an example of these plots for σ and λ_0 . Time series plots will tell you several
 13578 things: First, recall from sec. 17.2.2 that the way the chains move through the parameter
 13579 space gives you an idea of whether your MH steps are well tuned. If chains were constant
 13580 over many iterations you would need to decrease the tuning parameter of the (Normal)
 13581 proposal distribution. If a chain moves along some gradient to a stationary state very
 13582 slowly, you may want to increase the tuning parameter so that the parameter space is
 13583 explored more efficiently.

13584 Second, you will be able to see if your chains converged and how many initial sim-
 13585 ulations you have to discard as burn-in. In the case of the chains shown in Fig. 17.5,
 13586 we would probably consider the first 750 – 1000 iterations as burn-in, as afterwards the
 13587 chains seem to be fairly stationary.

13588 **17.4.3 Posterior density plots**

13589 The `plot()` command also produces posterior density plots and it is worthwhile to look
 13590 at those carefully. For parameters with priors that have bounds (e.g. Uniform over some
 13591 interval), you will be able to see if your choice of the prior is truncating the posterior
 13592 distribution. In the context of SCR models, this will mostly involve our choice of M , the
 13593 size of the augmented data set. If the posterior of N has a lot of mass concentrated close to
 13594 M (or equivalently the posterior of ψ has a lot of mass concentrated close to 1), as in the
 13595 example in Fig. 17.6, we have to re-run the analysis with a larger M . A diffuse posterior

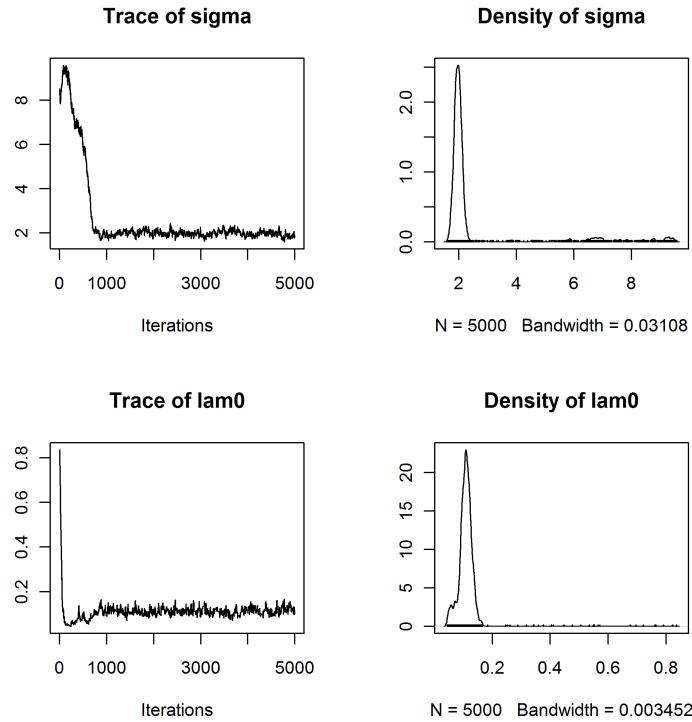


Figure 17.5. Time series and posterior density plots for σ and λ_0 for the Fort Drum black bear data.

plot suggests that the parameter may not be well-identified. There may not be enough information in your data to estimate model parameters and you may have to consider a simpler model. Finally, posterior density plots will show you if the posterior distribution is symmetrical or skewed – if the distribution has a heavy tail, using the mean as a point estimate of your parameter of interest may be biased and you may want to opt for the median or mode instead.

17.4.4 Serial autocorrelation and effective sample size

Checking the degree of autocorrelation in your Markov chains and estimating the effective sample size your chain has generated should be part of evaluating your model output. If you use **WinBUGS** through the **R2WinBUGS** package, the **print()** command will automatically return the effective sample size for all monitored parameters. In the **coda** package there are several functions you can use to do so. The function **effectiveSize()**

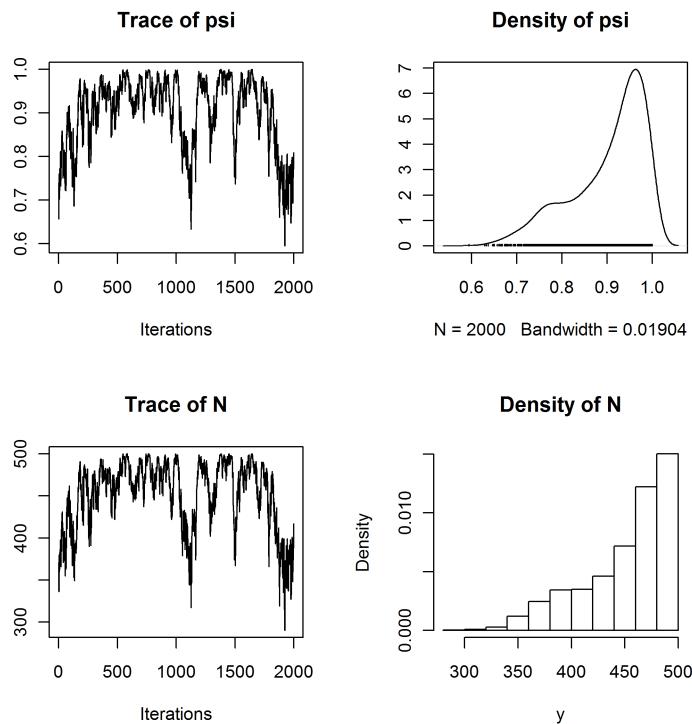


Figure 17.6. Time series and posterior density plots of ψ and N for the Fort Drum black bear data truncated by the upper limit of M (500).

13608 will directly give you an estimate of the effective sample size for the parameters:

```
13609 effectiveSize(window(chain, start=1001))
13610      sigma      lam0      psi      N
13611  93.89807 163.72311  51.96443  46.45394
```

13612 Alternatively, you can use the `autocorr.diag()` function, which will show you the
 13613 degree of autocorrelation for different lag values (which you can specify within the function
 13614 call, we use the defaults below):

```
13615 autocorr.diag(window(chain, start=1001))
13616      sigma      lam0      psi      N
13617 Lag 0  1.0000000 1.0000000 1.0000000 1.0000000
13618 Lag 1  0.9316928 0.91464875 0.9745833 0.9663320
13619 Lag 5  0.7603332 0.67445407 0.8525272 0.8500215
```

```

13620 Lag 10 0.6065374 0.48724122 0.7514657 0.7530124
13621 Lag 50 0.1122331 0.06564406 0.3811939 0.3823236

```

13622 In the present case we see that autocorrelation is especially high for the parameter ψ and
 13623 our effective sample size for this parameter is only 52! This means we would have to run
 13624 the model for much longer to obtain a reasonable effective sample size. Unfortunately,
 13625 with many SCR data sets we observe high degrees of serial autocorrelation. For now, let's
 13626 continue using this small number of samples to look at the output.

13627 17.4.5 Summary results

13628 Now that we checked that our chains apparently have converged and pretending that
 13629 we have generated enough samples from the posterior distribution, we can look at the
 13630 actual parameter estimates. The `summary()` function will return two sets of results: the
 13631 mean parameter estimates, with their standard deviation, the naïve standard error – i.e.
 13632 your regular standard error calculated for T (= number of iterations) samples without
 13633 accounting for serial autocorrelation – and the Time-series SE (in **WinBUGS** and earlier
 13634 in this book referred to as MC error), which accounts for autocorrelation. Remember our
 13635 rule of thumb that this error decreases with increasing chain length and should be 1% or
 13636 less of the parameter estimate. In **WinBUGS** the MC error is only given in the log output
 13637 within **BUGS** itself. You should adjust the `summary()` call by removing the burn-in from
 13638 calculating parameter summary statistics. To do so, use the `window()` command, which
 13639 lets you specify at which iteration to start 'counting'. In contrast to **WinBUGS**, which
 13640 requires you to set the burn-in length before you run the model, this command gives us
 13641 full flexibility to make decisions about the burn-in after we have seen the trajectories of
 13642 our Markov chains. For our example, `summary(window(chain, start=1001))` returns the
 13643 following output:

```

13644 Iterations = 1001:5000
13645 Thinning interval = 1
13646 Number of chains = 1
13647 Sample size per chain = 4000
13648
13649 1. Empirical mean and standard deviation for each variable,
13650      plus standard error of the mean:
13651
13652      Mean        SD  Naive SE Time-series SE
13653 sigma     1.9697  0.12534  0.0019818      0.012792
13654 lam0      0.1124  0.01521  0.0002405      0.001311
13655 psi       0.7295  0.11794  0.0018648      0.015278
13656 N        510.9190 81.99868 1.2965130     10.580567
13657
13658 2. Quantiles for each variable:
13659
13660      2.5%       25%       50%       75%     97.5%
13661 sigma    1.7288   1.8831   1.9666   2.0517   2.2240
13662 lam0     0.0863   0.1008   0.1112   0.1217   0.1449

```

```
13663 psi      0.5100  0.6423  0.7261  0.8170  0.9549
13664 N       359.0000 451.0000 508.0000 572.0000 668.0000
```

13665 Looking at the MC errors (column labeled **Time-series SE**), we see that in spite of the
 13666 high autocorrelation, the MC error for σ is below the 1% threshold, whereas for all other
 13667 parameters, MC errors are still above, another indication that for a thorough analysis we
 13668 should run a longer chain.

13669 Our algorithm gives us a posterior distribution of N , but we are usually interested
 13670 in the density, D . Density itself is not a parameter of our model, but we can derive a
 13671 posterior distribution for D by dividing each value of N (N at each iteration) by the area
 13672 of the state-space (here 3032.719 km^2) and we can use summary statistics of the resulting
 13673 distribution to characterize D :

```
13674 summary(window(chain[,4]/ 3032.719, start=1001))
13675
13676 Iterations = 1001:5000
13677 Thinning interval = 1
13678 Number of chains = 1
13679 Sample size per chain = 4000
13680
13681 1. Empirical mean and standard deviation for each variable,
13682 plus standard error of the mean:
13683
13684          Mean           SD        Naive SE Time-series SE
13685 0.1684690  0.0270380  0.0004275  0.0034888
13686
13687 2. Quantiles for each variable:
13688
13689    2.5%    25%    50%    75%   97.5%
13690 0.1184 0.1487 0.1675 0.1886 0.2203
13691 We see that our mean density of  $0.17/\text{km}^2$  is very similar to the estimate of  $0.18/\text{km}^2$ 
13692 obtained under the non-spatial model  $M_0$  in Chapt. 4.
```

13693 17.4.6 Other useful commands

13694 While inspecting the time series plot gives you a first idea of how well you tuned your
 13695 MH algorithm, use **rejectionRate()** to obtain the rejection rates (1 – acceptance rates)
 13696 of the parameters that are written to your output:

```
13697 rejectionRate(chain)
13698     sigma      lam0      psi        N
13699 0.42988598 0.78775755 0.00000000 0.03160632
```

13700 Recall (sec. 17.2.2) that rejection rates should lie between 0.2 and 0.8, so our tuning
 13701 seems to have been appropriate here. Draws of the parameter ψ are never rejected since
 13702 we update it with Gibbs sampling, where all candidate values are kept. And since N is
 13703 the sum of all z_i , all it takes for N to change from one iteration to the next are small

13704 changes in the z-vector, so the rejection rate of N is always low. If you have run several
 13705 parallel chains, you can combine them into a single mcmc object using the `mcmc.list()`
 13706 command on the individual chains (note that each chain has to be converted to an mcmc
 13707 object before combining them with `mcmc.list()`). You can then easily obtain the Gelman-
 13708 Rubin diagnostic (Gelman et al., 2004), in **WinBUGS** called Rhat, using `gelman.diag()`,
 13709 which will indicate if all chains have converged to the same stationary distribution. For
 13710 details on these and other functions, see the `coda` manual, which can be found (together
 13711 with the package) on the CRAN mirror.

17.5 MANIPULATING THE STATE-SPACE

13712 So far, we have constrained the location of the activity centers to fall within the outermost
 13713 coordinates of our rectangular state space by posing upper and lower bounds for x and y .
 13714 But what if \mathcal{S} has an irregular shape – maybe there is a large water body we would like
 13715 to remove from \mathcal{S} , because we know our terrestrial study species does not occur there. Or
 13716 the study takes place in a clearly defined area such as an island.

13717 As mentioned before, this situation is difficult to handle in **WinBUGS**. In some
 13718 simple cases we can adjust the state space by setting one of the coordinates of s_i to be
 13719 some function of the other and reject candidate s_i that do not fall within this modified
 13720 state space. In this manner, we can cut off corners of the rectangle to approximate the
 13721 actual state space³. To visualize this approach, plot the following rectangle, representing
 13722 your state space polygon, and line, representing, for example, the approximation of a shore
 13723 line:

```
13724 xlim<-c(-5,5)
13725 ylim<-c(-7,7)
13726 plot(xlim, ylim, type='n')
13727 abline(a=4, b=0.4)
```

13728 The Y coordinates limiting your state space to the habitat that is suitable to the species
 13729 you study can now be expressed as a linear function of the X coordinates, in this case,
 13730 $Y = 4 + 0.4 \times X$. To include this new limit in our **WinBUGS** model, we need to change
 13731 the following:

```
13732 #draw SX and SY as before
13733 SX[i] ~ dunif(xlim[1],xlim[2])
13734 SY[i] ~ dunif(ylim[1],ylim[2])
13735 #calculate upper limit for Y given X
13736 ymax[i]<-4+0.4*SX[i]
13737 # use step function to see if location [SX, SY]
13738 # is below the Y limit (Pin = 1) or not (Pin = 0)
13739 Pin[i] <- step(ymax[i] - SY[i])
13740 In[i] ~ dbern(Pin[i])
```

13741 In is a vector of M 1's, passed as data to the model. If $Pin = 0$, the likelihood will be 0
 13742 and the candidate [SX, SY] pair will be rejected. If $Pin = 1$, this bit of the likelihood is

³This idea was pitched to us by Mike Meredith, from WCS Malaysia

13743 equal to 1, and whether or not the candidate pair of coordinates is accepted depends
 13744 only on capture history of i . This approach can be very useful in some situations but is
 13745 clearly restricted by the functional form of the relationship between SX and SY that it
 13746 requires.

13747 In **R**, we are much more flexible, as we can use the actual state-space polygon to
 13748 constrain s_i . To illustrate that, let's look at a camera trapping study of raccoons (*Procyon*
 13749 *lotor*) conducted on South Core Banks, a barrier island within Cape Lookout National
 13750 Seashore, North Carolina (details of the study can be found in Sollmann et al. (2012b)
 13751 and in Chapt. 19 where we present the analysis of this data set with spatial mark-resight
 13752 models). Since camera-traps were spread across the entire length of the island, we set
 13753 the state space to be delineated by the shore line of the island (Fig. 17.7), which clearly
 13754 cannot easily be approximated as a rectangle. Instead, within **R** we can use an actual
 13755 shapefile of the island.

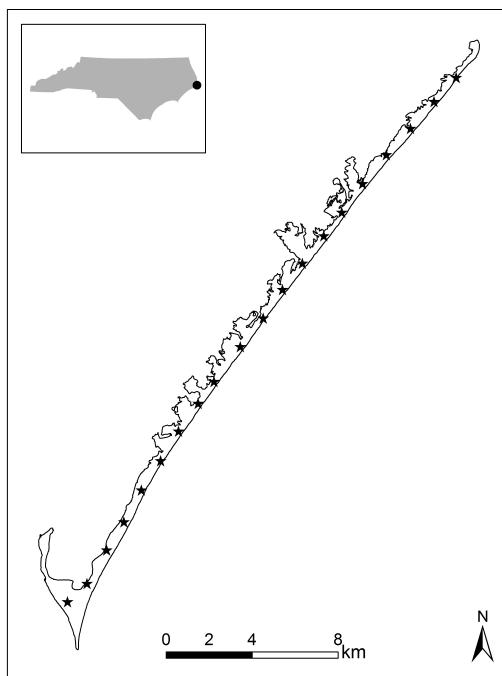


Figure 17.7. Camera traps (stars) set up on South Core Banks, a barrier island within Cape Lookout National Seashore, North Carolina (inset map) to estimate the raccoon population (see Chapt. 19 for details).

13756 In other circumstances you may still want to create the state space as before, by adding
 13757 some buffer to your trapping grid, but you may find that the resulting rectangle includes
 13758 water bodies, paved parking lots or any other kind of habitat you know is never used by the

species you study. In order to precisely describe the state-space, these features need to be removed. You can create a precise state-space polygon in **ArcGIS** and read it into **R**, or create the polygon directly within **R**, by intersecting two shape files – one of the rectangle defining the outer limits of your state-space state and one of the landscape feature you want to remove. While you will most likely have to obtain the shapefile describing the landscape of and around your trapping grid (coastlines, water bodies etc.) from some external source, the polygon shapefile buffering your outermost trapping grid coordinates can easily be written in **R**.

If **xmin**, **xmax**, **ymin** and **ymax** mark the most extreme *x* and *y* coordinates of your trapping grid and *b* is the distance you want to buffer with, load the package **shapefiles** (Stabler, 2006) and issue the following **R** commands:

```
13770 xl= xmin-b
13771 xu= xmax+b
13772 yl= ymin-b
13773 yu= ymax+b
13774
13775         #create data frame with coordinate pairs
13776 dd <- data.frame(Id=c(1,1,1,1,1),X=c(xl,xu,xu,xl,xl),
13777 Y=c(yl,yl,yu,yu,yl))
13778 ddTable <- data.frame(Id=c(1),Name=c("Item1"))
13779         #convert to shapefile, type polygon
13780 ddShapefile <- convert.to.shapefile(dd, ddTable, "Id", 5)
13781         # name and save to location of choice
13782 write.shapefile(ddShapefile, 'c:/Test', arcgis=T)
```

You can read shapefiles into **R** loading the package **maptools** (Lewin-Koh et al., 2011) and using the function **readShapeSpatial()**. Make sure you read in shapefiles in UTM format, so that units of the trap array, the movement parameter σ and the state-space are all identical. Intersection of polygons can be done in **R** also, using the package **rgeos** (Bivand and Rundel, 2011) and the function **gIntersect()**. The area of your (single) polygon can be extracted directly from the state-space object **SSp**:

```
13783 area <- SSp@polygons[[1]]@Polygons[[1]]@area /1000000
```

Note that dividing by 1000000 will return the area in km^2 if your coordinates describing the polygon are in UTM. If your state-space consists of several disjunct polygons, you will have to sum the areas of all polygons to obtain the size of the state-space. To include this polygon into our MCMC sampler we need one last spatial **R** package, **sp** (Pebesma and Bivand, 2011), which has a function, **over()**, which allows us to check if a pair of coordinates falls within a polygon or not.⁴ All we have to do is embed this new check into the updating steps for the s_i :

⁴Remember from the previous chapter (6.4.2) that the **over** function takes as its second argument (among others) an object of the class “**SpatialPolygons**” or “**SpatialPolygonsDataFrame**”. The former produces a vector while the latter produces a data frame (e.g., in the example above), which is important for how you index the output.

```

13797     #draw candidate value
13798 Scand <- as.matrix(cbind(rnorm(M, S[,1], 2), rnorm(M, S[,2], 2)))
13799     #convert to spatial points on UTM (m) scale
13800 Scoord<-SpatialPoints(Scand*1000)
13801     # check if scand is within the polygon
13802 SinPoly<-over(Scoord,SSp)
13803
13804 for(i in 1:M) {
13805     #if scand falls within polygon, continue update
13806     if(is.na(SinPoly[i])==FALSE) {
13807 ... [rest of the updating step remains the same]

```

13808 Note that it is much more time-efficient to draw all M candidate values for s and check
 13809 once if they fall within the state-space, rather than running the `over()` command for
 13810 every individual pair of coordinates. To make sure that our initial values for s also fall
 13811 within the polygon of \mathcal{S} , we use the function `runifpoint()` from the package `spatstat`
 13812 (Baddeley and Turner, 2005), which generates random uniform points within a specified
 13813 polygon. You'll find this modified MCMC algorithm (`SCR0poisSSp`) in the **R** package
 13814 `scrbook`.

13815 Finally, observe that we are converting candidate coordinates of \mathcal{S} back to meters to
 13816 match the UTM polygon. In all previous examples, for both the trap locations and the
 13817 activity centers we have used UTM coordinates divided by 1000 to estimate σ on a km
 13818 scale. This is adequate for wide ranging species like bears. In other cases you may center
 13819 all coordinates on 0. No matter what kind of transformation you use on your coordinates,
 13820 make sure to always convert candidate values for \mathcal{S} back to the original scale (UTM)
 13821 before running the `over()` command.

17.6 INCREASING COMPUTATIONAL SPEED

13822 Using custom written MCMC algorithms in **R** is not only more flexible but can also be
 13823 faster than using programs such as **JAGS** and especially **BUGS**. Also, **R** tends to use
 13824 much less memory than **JAGS**, which can be crucial if you are running a large model
 13825 but only have limited memory available. For example, you will see in Chapt. 7 that even
 13826 with a reasonable sized data set certain parameterizations of SCR models can max out
 13827 the memory of a 16 GB computer when using **JAGS**. These are mostly the models that
 13828 require us to look at individual sampling occasions instead of joining observations for a
 13829 given sampling location across the entire study, which requires us to introduce another
 13830 for-loop into the **JAGS** model. **BUGS** is limited in the amount of memory it can access
 13831 and thus will likely not max out your memory, but as a trade-off, it will take a long time
 13832 to run such models. In this chapter we have provided you with the guidelines to write
 13833 your own MCMC sampler. But beyond the material that we have covered there are a
 13834 number of ways you can make your sampler more efficient, through parallel computing
 13835 or by accessing an alternative computer language such as **c++**. Exploring these options
 13836 exhaustively is beyond the scope of this book; instead, in this section we will give you
 13837 some pointers to get started with these more advanced computational issues.

13838 17.6.1 Parallel computing

13839 If you are using a computer with several cores, you can make use of parallel computing to
 13840 speed up overall computation. In parallel computing we execute commands simultaneously
 13841 on different cores of the computer, instead of running them serially on one single core.
 13842 For example, imagine you have 4 cores available and you want to implement a for-loop in
 13843 **R**; instead of going through the loop iteration by iteration, you can prompt **R** to execute
 13844 iterations 1 to 4 at the same time on the 4 different cores. The core that finishes first will
 13845 then continue with iteration 5, and so on. There are several packages in **R** that allow you
 13846 to induce parallel computing, such as **snow** (Tierney et al., 2011) and **snowfall** (Knaus,
 13847 2010), and the more current versions of **R** (from 2.14.0 upwards) come with a pre-installed
 13848 set of functions grouped under the name **parallel**.

13849 The MCMC algorithms developed here and in other parts of this book come with plenty
 13850 of opportunities to parallelize computation. In various instances within the algorithm, we
 13851 have for-loops across our augmented data set of size M , or we may have for-loops across
 13852 sampling occasions. We also have for-loops across iterations of the algorithm, but since
 13853 one iteration of the Markov chain depends on the preceding iteration these should always
 13854 be run serially, not in parallel. There is another dimension we can think of, and that is
 13855 running multiple chains of an algorithm to assess convergence. This is a comparatively
 13856 easy implementation of parallel computing and thus provides a good starting point to
 13857 understand how it works in **R**.

13858 Let's go back to the Ft. Drum black bear data we analyzed above with the cloglog
 13859 version of the binomial SCR model (sec. 17.4.2) and run 3 parallel chains using **snowfall**.
 13860 All we need to do is wrap our function **SCR0binom.cl** within another function that can
 13861 then be executed in parallel, returning a list with one output matrix for each chain (install
 13862 **snowfall** before executing the code below; we assume the data objects are already in your
 13863 workspace from the previous analysis):

```
13864 library(snowfall)
13865 ## create wrapper function
13866 wrapper<-function(a){
13867   out<-SCR0binom.cl(y=Xaug, X=trapmat, M=M, xl=xl, xu=xu, yl=yl,
13868                         yu=yu, K=8, delta=c(0.1, 0.05, 2), niter=5000)
13869   return(out)
13870 }
```

13871 After creating the wrapper function we need to initialize the cluster of cores, defining
 13872 that we want computation to be implemented in parallel and how many cores we want it
 13873 to be run on. Here, we assume we have (at least) 3 cores, but if your computer only has 2,
 13874 make sure to adjust the code accordingly (i.e., set **cpus=2**). In that case, 2 of the 3 chains
 13875 will be run in parallel and whichever core finishes first will then pick up the third chain.
 13876 Further, we have to export all **R** libraries and data to all the cores, and set up a random
 13877 number generator, so that we do not get identical results from the different cores:

```
13878 sfInit( parallel=TRUE, cpus=3 ) #initialize cluster
13879 sfLibrary(scrbook) #export library scrbook
13880 sfExportAll() #export all data in current workspace
13881 sfClusterSetupRNG() #set up random number generator
```

```

13882 outL=sfLapply(1:3,wrapper) # execute 'wrapper' 3 times

13883 The object outL is a list of length 3, with one out matrix from the function SCRObinom.cl
13884 for each chain. After computation is complete, terminate the cluster using the command
13885 sfStop(). Note that the intermediate output of current values and acceptance rates in the
13886 R console is suppressed when using parallel computing. We can now look at the output
13887 as described previously using the package coda, by first defining outL to be a list of mcmc
13888 objects.

13889 library(coda)
13890 #turn output into MCMC list
13891 res<-mcmc.list(as.mcmc(outL[[1]]),as.mcmc(outL[[2]]),as.mcmc(outL[[3]]))
13892 summary(window(res, start=1001)) #remove first 1000 iterations as burn-in
13893
13894 [... some output removed ...]
13895
13896      Mean       SD  Naive SE Time-series SE
13897 sigma   1.9723  0.13093 0.0011952      0.0087055
13898 lam0    0.1115  0.01535 0.0001401      0.0009003
13899 psi     0.7130  0.10787 0.0009847      0.0077910
13900 N      499.6166 74.74934 0.6823650      5.4232653
13901
13902 2. Quantiles for each variable:
13903
13904      2.5%     25%     50%     75%   97.5%
13905 sigma  1.74339  1.8811  1.9637  2.0530  2.2618
13906 lam0   0.08443  0.1007  0.1105  0.1211  0.1438
13907 psi    0.52046  0.6350  0.7093  0.7814  0.9627
13908 N     366.00000 446.00000 497.00000 547.00000 674.0000

13909 Now that we have parallel chains we can also use the function gelman.diag to evaluate
13910 if chains have converged:
13911 gelman.diag(window(res, start=1001)) #assess chain convergence
13912
13913 Potential scale reduction factors:
13914
13915      Point est. Upper C.I.
13916 sigma      1.01      1.04
13917 lam0      1.01      1.02
13918 psi       1.07      1.21
13919 N        1.07      1.21
13920
13921 Multivariate psrf
13922
13923 1.05

13924 We can see that estimates are similar to what we observed when running a single
13925 chain (see sec. 17.4.2) and that all 3 chains appear to have converged, based on their

```

13926 point estimates of the \hat{R} statistic, but, as already noted before, for a real analysis we
 13927 might want to run this model for quite a bit longer, to bring down the upper confidence
 13928 interval limits on \hat{R} for ψ and N . If you have 3 cores then running these 3 parallel chains
 13929 should not have taken longer than running a single chain. Yet if you look at the effective
 13930 sample size now using `effectiveSize`, you can see that it has roughly tripled, as we would
 13931 expect:

```
13932 effectiveSize(window(res, start=1001))  

13933  

13934   sigma      lam0      psi        N  

13935 272.6935 411.8384 167.4192 168.3355
```

13936 17.6.2 Using C++

13937 Parallel computing is a great tool to speed up computations, but its usefulness is limited
 13938 by how many cores you have available. Even with a decent number of cores, large models
 13939 may still take a long time to run. A major reason for this is that for-loops in **R** are
 13940 time consuming, whereas they are handled much more time efficiently in other computer
 13941 languages such as **C++**. As we saw above, MCMC algorithms consist of for-loops within
 13942 for-loops, so that it stands to reason that implementing them in a language like **C++**
 13943 should make those algorithms run much faster. Being avid **R** users, we cannot claim to
 13944 be fluent in **C++** or to be aware of all the opportunities this language brings for faster
 13945 computing. It is also beyond the scope of this book to go into the nuts and bolts of
 13946 how **C++** works or provide a tutorial, and we refer you to the vast amounts of online
 13947 and print material designed to give the interested user an introduction to **C++**. Just
 13948 google “introduction C++” and you are sure to come across sites such as <http://www.cplusplus.com> that provide step by step instructions to get you started. Here, we only
 13949 want to point out one approach to linking **R** with **C++**: the packages `inline` (Sklyar
 13950 et al., 2010) and `RcppArmadillo` (Fran ois et al., 2011). These two packages provide a
 13951 very convenient interface between the two languages, but there are other other ways of
 13952 calling **C++** functions from within **R**, such as the `.Call` command. If you are interested,
 13953 we suggest you refer to the package manuals and vignettes, as well as the online document
 13954 “Writing R extensions” (at <http://cran.r-project.org/doc/manuals/R-exts.html>) for
 13955 a much more thorough treatment of this topic.

13956 In order to use **C++** you need a compiler such as `g++` that (together with other com-
 13957 pilers, for example for **C** and **FORTRAN**) comes with **Rtools**, which you can easily
 13958 download from the web (at <http://cran.r-project.org/bin/windows/Rtools/>). All of
 13959 these compilers are part of the GNU compiler collection (<http://gcc.gnu.org/>). Make
 13960 sure the version of **Rtools** matches your version of **R** or you may run into compilation
 13961 errors later on. To give you a taste of **C++** we will show you how to write a function
 13962 that calculates the squared distances of individual activity centers to all traps, as is im-
 13963 plemented in the `scrbook` package in the function `e2dist` (to be exact, `e2dist` calculates
 13964 the distance, not the squared distance), and compare performance between **R** and **C++**.
 13965 We will refer to these functions as “distance functions”. First, let us set up dummy data
 13966 – a matrix holding the coordinates of the trap array, outer limits of the state space and
 13967 uniformly distributed activity centers for $M = 700$ individuals:

```

13969  gx<-seq(1,10,1)
13970  gy<-seq(1,10,1)
13971  X<-as.matrix(expand.grid(gx, gy))
13972  M<-700
13973  J<-dim(X)[1]
13974  b<-3
13975  xl<-min(gx)-b
13976  xu<-max(gx)+b
13977  yl<-min(gy)-b
13978  yu<-max(gy)+b
13979  S<-cbind(runif(M, xl, xu), runif(M, yl,yu))

```

13980 Next, we can write a “pedestrian” version of `e2dist` and check how long it takes to
13981 calculate the squared distance matrix:

```

13982 Dfun<-function(M, J, S, X){
13983   D2<-matrix(0, nrow=M, ncol=J)
13984   for (i in 1:M){
13985     for(j in 1:J){
13986       D2[i,j]<-(S[i,1]-X[j,1])^2 + (S[i,2]-X[j,2])^2
13987     }
13988   return(D2)
13989 }
13990
13991 system.time(
13992 (D2R<-Dfun(M, J, S, X))
13993 )
13994
13995 user  system elapsed
13996 0.81    0.01    0.82

```

13997 The code to implement the same function in **C++** using the `inline` and `RcppArmadillo`
13998 packages is shown in panel 17.3. These packages allow you to use a range of data formats
13999 such as lists and matrices, and they take care of compiling the code in **C++** and loading
14000 the resulting function into **R**. This is also referred to compiling **C++** code “on the fly”.
14001 You will see that the way the code is set up is reasonably similar to **R**. One difference that
14002 is worthy to point out is that in **C++** indexes for vectors range from 0 to $n - 1$, NOT
14003 from 1 to n , as in **R**. Note that with `inline` we only need to write the core of the code and
14004 define the type of the variables we want to pass to the function, while the `cxxfunction`
14005 call takes care of the rest. Once your function is compiled and loaded you should check
14006 out the full **C++** code by calling `DfunArma@code`.

14007 Executing this code shows that it is also faster than the **R** version of the distance
14008 function or `e2dist`; in fact it is too fast for the time resolution of the `system.time()`
14009 function to even give us a time estimate:

```

14010 system.time(
14011 (out<-DfunArma(M,J,S,X)))
14012

```

```
14013 user    system elapsed
14014      0        0       0
```

14015 While speed differences of less than 1 second may seem negligible, remember that
 14016 each command has to be executed at each iteration of the Markov chain. Especially with
 14017 time-consuming models such as those for open populations (Chapt. 16) or multi-session
 14018 models (Chapt. 14) we believe that C++ holds large potential to make implementation
 14019 of such models more feasible.

17.7 SUMMARY AND OUTLOOK

14020 In a nutshell, programs like **WinBUGS** do everything that we went through in this chapter
 14021 (and quite a bit more). Looking through your model, they determine which parameters
 14022 they can use standard Gibbs sampling for (i.e. for conjugate full conditional distributions).
 14023 Then, they determine whether to use adaptive rejection sampling, slice sampling or – in
 14024 the ‘worst’ case – Metropolis-Hastings sampling for the other full conditionals (how the
 14025 sampler is chosen differs among softwares). For MH sampling, they will automatically
 14026 tune the updater so that it works efficiently.

14027 Although these programs are flexible and extremely useful to perform MCMC simulations,
 14028 it sometimes is more efficient to develop your own MCMC algorithm. Building an
 14029 MCMC code follows three basic steps: Identify your model including priors and express
 14030 full conditional distributions for each model parameter. If full conditionals are parametric
 14031 distributions, use Gibbs sampling to draw candidate parameter values from those dis-
 14032 tributions; otherwise use Metropolis-Hastings sampling to draw candidate values from
 14033 a proposal distribution and accept or reject them based on their posterior probability
 14034 densities.

14035 These custom-made MCMC algorithms give you more modeling flexibility than existing
 14036 software packages, especially when it comes to handling the state-space: In **BUGS**
 14037 (and **JAGS** for that matter) we define a continuous rectangular state-space using the
 14038 corner coordinates to constrain the Uniform priors on the activity centers **s**. But what if a
 14039 continuous rectangle isn’t an adequate description of the state-space? In this chapter we
 14040 saw that in **R** it only takes a few lines of code to use any arbitrary polygon shapefile as the
 14041 state-space, which is especially useful when you are dealing with coastlines or large bodies
 14042 of water that need removing from the state-space. Another example is the SCR **R** package
 14043 **SPACECAP** (Gopalaraswamy et al., 2012a) that was developed because implementation of an
 14044 SCR model with a discrete state-space was inefficient in **WinBUGS**.

14045 Another situations in which using **BUGS/JAGS** becomes increasingly complicated
 14046 or inefficient is when using point processes other than the Binomial point process (“uni-
 14047 formity”) which underlies the basic SCR model (see sec. 5.10 in Chapt. 5). In Chapt.
 14048 11 and XXXX BETHS PP CHAPTER XXXX you will see examples of different point
 14049 processes, implemented using custom-made MCMC algorithms.

14050 Finally, the Chapt. 18 and 19 deal with unmarked or partially marked populations
 14051 using hand-made MCMC algorithms to handle the (partially) latent individual encounter
 14052 histories. While some of these models can be written in **BUGS/JAGS**, they are painstakingly
 14053 slow; others cannot be implemented in **BUGS/JAGS** at all (e.g., the classes of
 14054 models considered in Chaps. 12 and 11). In conclusion, while you can certainly get by

- 14055 using **BUGS/JAGS** for standard SCR models, knowing how to write your own MCMC
14056 sampler allows you to tailor these models to your specific needs.

```
Norm.Gibbs<-function(y=y,mu_0=mu_0,sigma2_0=sigma2_0,a=a,b=b,niter=niter){

ybar<-mean(y)
n<-length(y)
mu<-1           #mean initial value
sigma2<-1        #sigma2 initial value
an<-n/2 + a      #shape parameter of InvGamma of sigma2
out<-matrix(nrow=niter, ncol=2)
colnames(out)<-c('mu', 'sig')

for (i in 1:niter) {

#update mu according to Eq. 7.2
mu_n<-((sigma2/(sigma2+n*sigma2_0))*mu_0
+ (n*sigma2_0/(sigma2 + n*sigma2_0))*ybar)
sigma2_n <- (sigma2*sigma2_0)/ (sigma2 + n*sigma2_0)
mu<-rnorm(1,mu_n, sqrt(sigma2_n))

#update sigma2 according to Eq. 7.3
bn<- 0.5 * (sum((y-mu)^2)) + b
sigma2<-1/rgamma(1,shape=an, rate=bn)
out[i,]<-c(mu,sqrt(sigma2))
}
return(out)
}
```

Panel 17.1: R-code for a Gibbs sampler for a Normal model with unknown μ and σ and conjugate priors (Normal and Inverse-Gamma, respectively) for both parameters.

```
Logreg.MH<-function(y=y, mu0=mu0, sig0=sig0, delta=delta, niter=niter) {  
  out<-c()  
  theta<-runif(1, -3,3) #initial value  
  for (iter in 1:niter){  
    theta.cand<-rnorm(1, theta, delta)  
    loglike<-sum(dbinom(y, 1, exp(theta)/(1+exp(theta)), log=TRUE))  
    logprior <- dnorm(theta,mu0 ,sig0, log=TRUE)  
    loglike.cand<-sum(dbinom(y, 1, exp(theta.cand)/(1+exp(theta.cand)),  
    log=TRUE))  
    logprior.cand <- dnorm(theta.cand, mu0, sig0, log=TRUE)  
    if (runif(1)<exp((loglike.cand+logprior.cand)-(loglike+logprior))){  
      theta<-theta.cand  
    }  
    out[iter]<-theta  
  }  
  return(out)  
}
```

Panel 17.2: **R** code to run a Metropolis sampler on a simple Logit-Normal model.

```
### calculate squared distances using RcppArmadillo
library(inline)
library(RcppArmadillo)

#write core of function code
code<-'
/*define input, assign correct class (matrix, vector etc)*/
arma::mat Sn=Rcpp::as<arma::mat>(S);
arma::mat Xn=Rcpp::as<arma::mat>(X);
int Ntot=Rcpp::as<int>(M);
int ntraps=Rcpp::as<int>(J);
/*create matrix to hold squared distances*/
arma::mat D2(Ntot, ntraps);

/*loop over M and J to calculate distances*/
for (int i=0; i<Ntot; i++){
  for(int j=0; j<ntraps; j++){
    D2(i,j)= pow(Sn(i,0)-Xn(j,0), 2) + pow(Sn(i,1)-Xn(j,1), 2);
  }
}
/*return D2 in R format*/
return Rcpp::wrap(D2);
'

# compile and load
DfunArma<-cxxfunction(signature(M="integer", J="integer", S="numeric",
X="numeric"), plugin="RcppArmadillo", body=code)
```

Panel 17.3: Code to compute squared distance between individual activity centers and traps in **C++** from within **R** using **inline** and **RcppArmadillo**

14057
14058

18

14059
14060

SPATIAL CAPTURE-RECAPTURE FOR UNMARKED POPULATIONS

14061
14062
14063
14064
14065
14066
14067
14068

todo: Royle-Nichols observaiton model

Traditional capture-recapture models share the fundamental assumption that each individual in a population can be uniquely identified when captured. Often, this can be accomplished by marking individuals with color bands, ear tags, or some other artificial mark that can be subsequently read in the field. For other species, such as tigers or marbled salamanders, individuals can be easily identified using only their natural markings, yet many species do not possess adequate natural markings and are difficult to capture, making it impractical to use standard capture-recapture techniques.

14069
14070
14071
14072
14073
14074
14075
14076
14077
14078
14079
14080

Estimating density when individuals are unmarked can be accomplished using a variety of alternatives to capture-recapture, but many of these methods have important limitations that warrant the exploration of alternative approaches. In this chapter we highlight the work of Chandler and Royle (In press) who demonstrated that the “individual recognition” assumption of capture-recapture models is not a requirement of spatial capture-recapture models. They showed that, under certain conditions described below, spatially-correlated count data are sufficient for making inference about animal distribution and density even when no individuals are marked. The Chandler and Royle (In press) “spatial count model” (hereafter the SC model) is virtually identical to other SCR models except that the encounter histories $\{z_{ijk}\}$ are not directly observed. Instead, the observed data are the counts realized by summing up the detections for each individual at a survey location during a sampling occasion $n_{jk} = \sum_i z_{ijk}$.

14081
14082
14083
14084
14085
14086
14087

The ability to fit SCR models to data from unmarked populations has important implications. For one, it means that SCR models can be applied to data collected using methods like points counts in which observers record simple counts of animals at an array of survey points. Camera trapping data on unmarked animals such as deer or coyotes could also also be suitable. In addition, this development has important implications for traditional SCR studies because many SCR datasets include some individuals that cannnot be identified due to poor photo quality or indistiguishable natural markings.

14088

It is also interesting to note that by disregarding individual identity, we wind up with

14089 a model that closely resembles another large class of spatial models, known as convolution
14090 models (Wolpert and Ickstadt, 1998; Higdon, 1998). These models have been used for a
14091 variety of purposes such as describing oceanic surface temperatures and correlation in tree
14092 locations within managed forests. The SC model offers an improvement in some respects
14093 over existing convolution models because it does not require arbitrary decisions about the
14094 location and number of “support points”. We will clarify this later in the chapter, and
14095 briefly mention how this model can be used outside of SCR contexts for general purpose
14096 spatial modeling of correlated count data.

18.1 EXISTING MODELS FOR INFERENCE ABOUT DENSITY IN UNMARKED POPULATIONS

14097 When capture-recapture methods are not a viable option, researchers often collect sim-
14098 ple count data or even detection/non-detection data to estimate population parameters.
14099 These data are often analyzed using Poisson regression or logistic regression, perhaps with
14100 random effects. When detection is imperfect, as it almost always is, these methods cannot
14101 be used to obtain unbiased estimates of population size or occurrence probability. Even
14102 when these data are used an index of abundance or occurrence, standard models may yield
14103 unreliable results when covariates affect both the state variable and detection probability.
14104 A classic example is the finding by Bibby and Buckland (1987) who reported that the de-
14105 tention probability of songbirds in restocked conifer plantations was negatively associated
14106 with vegetation height, yet population density was positively related to vegetation height.
14107 This intuitive and common phenomenon has led to the development of a vast number
14108 of models to estimate population size and detection probability when individuals are un-
14109 marked. A review of these models is beyond the scope of this chapter, but we mention a
14110 few deficiencies of existing methods that warrant the exploration of alternatives for robust
14111 inference when standard capture-recapture methods do not apply.

14112 Distance sampling (Buckland et al., 2001), which we briefly introduced in Chapter 1,
14113 is perhaps the most widely used method for estimating population density when individ-
14114 uals are unmarked and detection probability is less than one. This class of methods is
14115 known to work impeccably when estimating the number of stakes in a field or the number
14116 of duck nests in a wetland. Distance sampling can also work very well in more interesting
14117 situations, and is an extremely powerful method when the assumptions can be met. How-
14118 ever, the assumptions that distance data can be recorded without error and that animals
14119 are distributed randomly with respect to the transect can be easily violated by common
14120 processes such as animal movement and measurement error. Although numerous methods
14121 have been proposed to relax some of these assumptions Royle et al. (2004); Borchers et al.
14122 (1998); Johnson (2010); Chandler et al. (2011), another issue is that distance sampling is
14123 simply not practical in many settings. For example, many species are so rare and elusive
14124 that they can only be reliably surveyed using methods such as camera traps.

14125 Other common sampling methods used to estimate density when individuals are un-
14126 marked include double-observer sampling, removal sampling, and repeated counts, for
14127 which custom models have been developed (Nichols et al., 2000; Farnsworth et al., 2002;
14128 Royle, 2004b,a; Fiske and Chandler, 2011). To obtain reliable density estimates using
14129 these methods, the area surveyed must be well defined and closed with respect to move-
14130 ment and demographic processes. Given a short enough sampling interval, such as a 5-min

14131 point-count, the closure assumption may be reasonable. However, short sampling intervals
14132 limit the number of detections, so observers generally visit each survey location multiple
14133 times during a season. But then animal movement may invalidate the closure assumption,
14134 and a model of temporary emigration is required (Kendall et al., 1997; Chandler et al.,
14135 2011). Furthermore, distance-related heterogeneity in detection probability can introduce
14136 bias in these models, although this bias is negligible when the ratio of plot size to the scale
14137 parameter of the detection function is low (Efford and Dawson, 2009).

14138 We mention these issues not to suggest that existing models do not have value—indeed
14139 we believe that they can be used to obtain reliable density estimates in many situations—
14140 rather our aim is to highlight the need for alternative methods when the assumptions of
14141 existing methods cannot be met. Additionally, the spatial count model we discuss in this
14142 chapter serves as the foundation for a broad class of SCR models in which all or some of
14143 the individuals cannot be uniquely identified, which is the focus of the next chapter.

18.2 SPATIAL CORRELATION AS INFORMATION

14144 All of the previous methods require some sort of auxiliary information to separately model
14145 abundance and detection. That is, we need multiple observers or distance data or repeated
14146 visits to ensure that model parameters are identifiable¹. The same is true for SC model
14147 (Chandler and Royle, In press), but the auxiliary information comes in the form of spatial
14148 correlation, which requires no extra effort to collect.

14149 It is natural to be suspicious of the claim that spatial correlation is a good thing.
14150 Indeed, elaborate methods have been devised to deal with spatial correlation as a nuisance
14151 parameter (F Dormann et al., 2007), and ecologists have been admonished for failing to
14152 obtain “real” replicates uncontaminated by spatial correlation (Hurlbert, 1984). The
14153 following heuristic may be helpful.

14154 Imagine a 10×10 grid of camera traps and a single unmarked individual exposed to
14155 capture whose home range center lies in the center of the trapping grid. If the individual
14156 has a small home range size relative to the extent of the trapping grid, we can imagine
14157 what the spatial correlation structure of the encounters might look like. If the animal’s
14158 movement is symmetric around the activity center then the number of times the individual
14159 is detected at each trap (the trap counts) is a function of the distance between the home
14160 range center and the trap, *i.e.* traps with the same distance from the activity center
14161 will yield counts that are more highly correlated with one another than traps located
14162 at different distances from the activity center. Thus, the correlation in counts tells us
14163 something about the location of the activity center. It is relatively intuitive that spatial
14164 correlation carries information about distribution, but what about density?

14165 Imagine now that there are two activity centers located in our trapping grid. Using
14166 trap counts alone, can our model tell us both where the activity centers are and how
14167 many exist in the population exposed to capture? The answer is yes, at least under
14168 certain circumstances. Figure 18.1 illustrates the process. The map on the left shows 500
14169 simulated movement outcomes of the two individuals. The right panel shows the total
14170 counts made at each trap after 10 survey occasions. Assuming that animals have bivariate
14171 normal home ranges, the fact that there are two areas in the map with high counts that

¹Or we can make very strong model assumptions and get away without any auxiliary data (Lele et al., 2012; Sólymos et al., 2012)

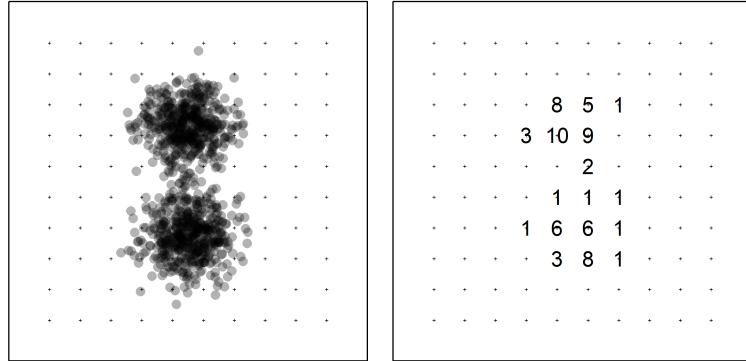


Figure 18.1. Movement outcomes (left) of two individuals with slightly overlapping home ranges. Crosses represent trap (or point count) locations. The right panel shows counts at each point. It is possible to estimate density using the count data alone.

dissipate in both dimensions suggests that the most likely number of individuals given these data is 2. Furthermore, the degree to which the counts dissipate from the two areas of highest intensity is information about the home range size parameter σ . These two pieces of information are enough to estimate density—again, given that a bivariate normal home range is a valid assumption. Departures from this assumption are discussed subsequently.

18.3 DATA

One of the important benefits of the SC model is that it can be applied to data collected using an enormous variety of survey methods. Whereas traditional SCR models require spatially-referenced encounter histories, this model requires simple count data. Once again, suppose that we have J “traps” operated on K time periods during which no births or deaths occur. We use the term trap very loosely in this context. A trap is simply some sampling device capable of recording the number of individuals detected, n_{jk} , so traps could be camera traps, hair snares, or even human observers standing at some location \mathbf{x}_j . Regardless of the sampling method, the requisite data are the counts n_{jk} and the coordinates of the traps \mathbf{x}_j . In some instances, we might have additional data such as trap-specific covariates, state-space covariates, information on the identities of a subset of individuals, or perhaps even distance data. Some of these extensions are covered in Chapters 19 and ??, but for the sake of simplicity we focus on the basic data structure in this chapter.

18.4 MODEL

The state model that we consider here is the same as in the basic spatial-capture setting, in which we assume a homogeneous point process $\mathbf{s}_i \sim Unif(\mathcal{S})$ where \mathbf{s}_i is the activity

center of individual $i = 1, \dots, N$, and \mathcal{S} is the state-space which is typically a polygon defining the region where the organism occur. This state model describes the number and locations of animals. The observation model is once again conditional on the state model and describes the encounter rate as a function of the distance between activity centers and traps.

As with all SCR models, the encounter process is specific to the sampling method, and here we consider the standard camera trapping situation in which an individual can be encountered at multiple traps during a single time period, say one night during a camera-trapping study, and it can be detected multiple times at a single trap during an occasion. This is the Poisson encounter model described in Chapt. 9. The model for the capture histories can be described by

$$z_{ijk} \sim \text{Poisson}(\lambda_{ij}). \quad (18.4.1)$$

where λ_{ij} is the encounter rate for individual i at trap j . A common form of this parameter is

$$\lambda_{ij} = \lambda_0 \exp(\|\mathbf{x}_j - \mathbf{s}_i\|/2\sigma^2)$$

where λ_0 is the baseline encounter rate and σ is the scale parameter describing the distance-related decay in encounter rate.

When individuals cannot be uniquely identified, the z_{ijk} cannot be directly observed, which seems like a massively insurmountable problem. The solution is the same one we routinely apply when we cannot directly observe the process of interest—we regard the encounter histories as latent variables. The data are now just a reduced-information summary of the latent encounter histories. That is, they are the sample- and trap-specific totals, aggregated over all individuals:

$$n_{jk} = \sum_{i=1}^N z_{ijk}.$$

This data structure, a matrix of counts made at a collection of sampling locations on one or more occasions is extremely common in ecology. Note also that we can get by with a single occasion of data ($J \equiv 1$) because under the Poisson model,

$$n_{jk} \sim \text{Poisson}(\Lambda_j) \quad (18.4.2)$$

where

$$\Lambda_j = \lambda_0 \sum_i k_{ij},$$

and because Λ_j does not depend on t , we can aggregate the replicated counts, defining $n_{j\cdot} = \sum_k n_{jk}$ and then

$$n_{j\cdot} \sim \text{Poisson}(K\Lambda_j)$$

As such, K and λ_0 serve equivalent roles as affecting baseline encounter rate as has been noted elsewhere (Efford et al., 2009b).

This formulation of the model in terms of the aggregate count simplifies computations as the latent variables z_{irt} do not need to be updated in the MCMC estimation scheme (see below). However, retaining z_{irt} in the formulation of the model is important if some individuals are uniquely marked, in which case modifying the MCMC algorithm to include both types of data is trivial is straight-forward. This is because uniquely identifiable individuals produce observations of some of the z_{irt} variables, which we elaborate on in the subsequent chapter.

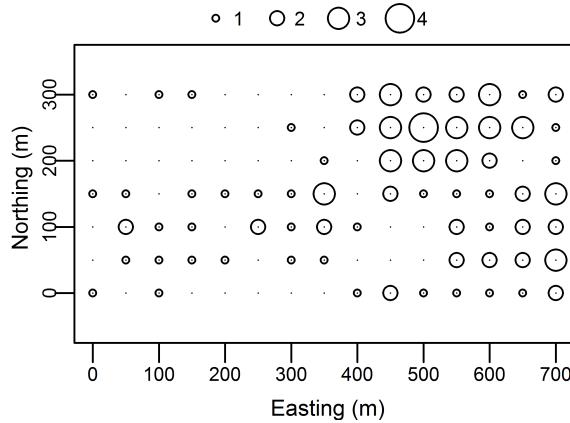


Figure 18.2. Spatially-correlated counts of northern parula on a 50-m grid. The size of the circle represents the total number of detections at each point.

18.5 NORTHERN PARULA EXAMPLE

14229 Here we re-analyze the Northern Parula (*Parula americana*) data described in Chandler
 14230 and Royle (In press). The data were collected at 105 points located on a 50-m grid at
 14231 the Patuxent Wildlife Research Center. Each point was surveyed 3 times during June
 14232 2006, and Fig. 18.2 depicts the resulting spatially-correlated counts ($n_{r,i}$). A total of 226
 14233 detections were made with a maximum count of 4 during a single survey. At 38 points,
 14234 no warblers were detected. All but one of the detections were of singing males, and this
 14235 one observation was not included in the analysis.

14236 In our analysis of the parula data, we defined the point process state-space by buffering
 14237 the grid of point count locations by 250 m and used $M = 300$.

14238 At this point in time there is no canned software to fit this model, and it is actually
 14239 not straight-forward to use **BUGS** because of the constraints in the model². However,
 14240 **JAGS** has a neat distribution called the `dsum` distribution, which was designed for this
 14241 type of situation where the observed data are a sum of random variables. Remember,
 14242 if we have 3 detections at a point, we assume that these results as $\sum_i z_{ijk}$. Thus, we
 14243 are summing up random variables. **JAGS** actually works rather well for this situation
 14244 although it is quite slow. Another limitation of using **JAGS** is that we can't mix data from
 14245 marked and unmarked individuals because `dsum` requires that we sum over unobserved
 14246 quantities, not a mix of observed and unobserved nodes. Thus, we can't use **JAGS** for the
 14247 situations considered in the next chapter, and thus we wrote our own MCMC algorithm
 14248 which overcomes these limitations, and it is somewhat faster. Nonetheless, here is the
 14249 **JAGS** code to analyze the NOPA data.

```
14250 model{
```

²Although it can be done using the so-called “ones-trick”

```

14251 sigma ~ dunif(0, 5)
14252 lam0 ~ dunif(0, 5)
14253 psi ~ dunif(0, 1)
14254 for(i in 1:M) {
14255     # Indicator of occurrence
14256     w[i] ~ dbern(psi)
14257     # Animal activity centers
14258     sx[i] ~ dunif(0, xSide)
14259     sy[i] ~ dunif(0, ySide)
14260     for(r in 1:nTraps) {
14261         # distance from plot center
14262         d[i, r] <- sqrt(pow(sx[i] - X[r, 1], 2) + pow(sy[i] - X[r, 2], 2))
14263         # encounter rate
14264         lam[i, r] <- lam0 * exp(-1*pow(d[i, r],2) / (2*pow(sigma,2))) * w[i]
14265         for(t in 1:nReps) {
14266             z[i, r, t] ~ dpois(lam[i, r])
14267         }
14268     }
14269 }
14270 for(r in 1:nTraps) {
14271     for(t in 1:nReps) {
14272         y[r, t] ~ dsum(z[1,r,t],z[2,r,t], ... ,z[100,r,t]) # code abbreviated
14273     }
14274 }
14275 N <- sum(w[])
14276 }
```

14277 Note that this code will not run as shown because we abbreviated the arguments to
 14278 `dsum`. In practice, you need to provide all 100 of them, if $M = 100$! This is kind of a drag,
 14279 but you can easily create the text using `paste` in R. Maybe Martyn Plummer will throw
 14280 us a bone and allow for a vector as an argument. Anyhow, the entire analysis is shown
 14281 on the `???XX` help page in `scrbook`.

14282 We simulated posterior distributions using three Markov chains, each consisting of
 14283 300000 iterations after discarding the initial 10000 draws. Convergence was satisfactory,
 14284 as indicated by an \hat{R} statistic of < 1.02 (Gelman and Rubin, 1992).

14285 The posterior distribution for N was highly skewed with a long right tail resulting in a
 14286 wide 95% credible interval (Table 18.1). Nonetheless, the interval for density, D , includes
 14287 estimates reported from more intensive field studies (Moldenhauer and Regelski, 1996).
 14288 As with any SCR model, we can produce a density surface map, as shown in Fig. 18.4

18.6 IMPROVING PRECISION WITH PRIOR INFORMATION

14289 We are asking a lot of a little data. Because both the activity centers and the encounter
 14290 histories are latent variables, there is inherently high uncertainty in the data, even if it is
 14291 “perfect” data simulated from the true model. This explains the low posterior precision
 14292 in the parula data.

Table 18.1. Posterior summary statistics for spatial Poisson-count model applied to the northern parula data. Two sets of priors were considered. $M = 300$ was used in both cases. Parulas/ha, D , is a derived parameter.

Par	Prior	Mean	SD	Mode	q0.025	q0.50	q0.975
σ	$U(0, \infty)$	2.154	1.222	1.230	0.896	1.665	5.170
λ_0	$U(0, \infty)$	0.284	0.149	0.212	0.084	0.256	0.665
N	$U(0, M)$	40.953	38.072	4.000	3.000	31.000	143.000
D	—	0.427	0.397	0.0417	0.0313	0.323	1.490
σ	$G(13, 10)$	1.301	0.258	1.230	0.889	1.266	1.908
λ_0	$U(0, \infty)$	0.298	0.132	0.240	0.098	0.279	0.603
N	$U(0, M)$	59.321	36.489	36.000	18.000	50.000	157.000
D	—	0.618	0.380	0.375	0.188	0.521	1.635

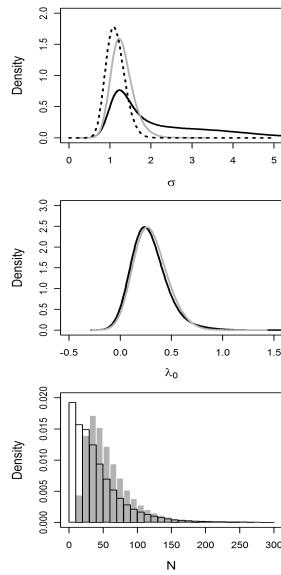


Figure 18.3. Effects of $\sigma \sim \text{Gamma}(13, 10)$ prior on the posterior distributions from the northern parula model. Posteriors from model with uniform priors are shown in black, and posteriors from the informative prior model are shown in gray. The prior itself is shown as dotted line in the upper panel.

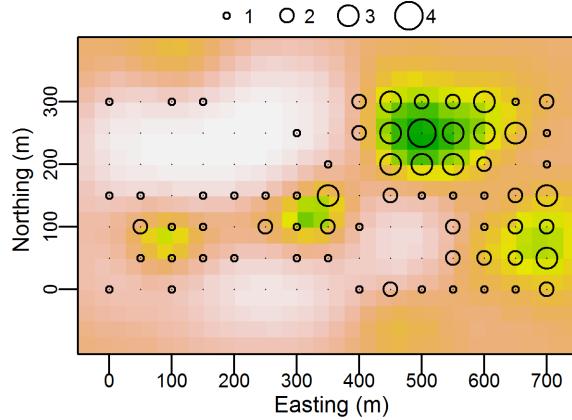


Figure 18.4. Estimated density surface of northern parula activity centers. The grid of point count locations with count totals is superimposed. See Fig. 1 for additional details.

14293 So why not just collect distance data or something? If you can, great—we are not
 14294 arguing against the use of other methods. But in many cases, other models are not
 14295 applicable. For instance, our model could be applied to camera trapping data collected
 14296 on species without natural marks, such as pumas or coyotes. In addition, this model
 14297 provides an important foundation for modeling data where other methods do not apply,
 14298 and the underlying state model is so damn cool because it corresponds to what we think
 14299 is happening in the field. Furthermore, the potential generalizations are numerous as we
 14300 will see later in this chapter and in the next chapter. In sum, the model can be applied
 14301 where no other models can, and it provides the foundation for important extensions, but
 14302 how can we improve precision?

14303 Indeed, extensive information on home range size has been compiled for many species in
 14304 diverse habitats (*e.g.*, DeGraaf and Yamasaki, 2001). It is easy to embody this information
 14305 in a prior distribution as we demonstrated for the parula data.

14306 One benefit of a Bayesian analysis is that it can accommodate prior information on
 14307 the home range size and encounter rate parameters, which are readily available for many
 14308 species. To illustrate, we analyzed the parula data using a new set of priors. Whereas in the
 14309 first analysis, all priors were improper, customary non-informative priors (see Table 18.1),
 14310 in the second set we used an informative prior for the scale parameter $\sigma \sim \text{Gamma}(13, 10)$.
 14311 We arrived at this prior using the methods described by Royle et al. (2011a) and published
 14312 information on the warbler's home range size and detection probability (Moldenhauer and
 14313 Regelski, 1996; Simons et al., 2009). More details on this derivation are found in ??????.
 14314 We briefly note here that this prior includes the biologically-plausible range of values from
 14315 σ suggested by the published literature.

14316 This was true when considering both sets of priors, although posterior precision was
 14317 higher under the informative set of priors. Specifically, the use of prior information reduced
 14318 posterior density at high, biologically implausible, values of σ , and hence decreased the

14319 posterior mass for low values of N (Fig. 18.3).

18.7 DESIGN ISSUES

14320 18.7.1 How Much Correlation Is Enough?

14321 σ shouldn't be too small or too large relative to trap spacing.

14322 Can we test for correlation using K-functions or something?

14323 18.7.2 Linear Designs

14324 Survey points are not always located on a grid with even spacing—in fact, it is rare to see
14325 a perfect 10×10 grid of points in any study because of habitat patchiness or rugged terrain
14326 or what have you. Instead, points are often distributed haphazardly or using some form
14327 of probability sampling. Such designs can still produce data amenable to the models we
14328 consider in this chapter if individuals can be encountered at multiple points, and none of
14329 the considerations discussed above need to be modified. But what about linear designs?

14330 In bird studies, point counts are often placed on linear transects. For example, the
14331 Breeding Bird Survey involves surveying 50 points spaced by 0.5 miles. The mountain-top
14332 bird survey in the White Mountain National Forest involves surveying 42 transects, each
14333 with 20? points spaced by 250-m (King et al., 2008). For many species, the 0.5 mile spacing
14334 of the BBS will ensure that individuals are not detected at multiple points. However, in
14335 the moutain-top survey, it's easy to imagine that a Bicknell's Thrush (*Catharus bicknelli*)
14336 could easily be heard from adjacent points. So can we apply our model to obtain density
14337 estimates with such simple counts?

14338 18.7.3 Quadrat counts

18.8 ALTERNATIVE OBSERVATION MODELS

14339 Chandler and Royle (In press) focused exclusively on the Poisson observation model, but
14340 noted that alternative models such as the Bernoulli model or the multinomial model
14341 (Chapt. 9) should be easily accomodated. Unfortunately, our experimentation with these
14342 models indicates that the base-line encounter probability parameter p_0 is not identifiable.
14343 At this point in time, it is not clear why this would be so. However, this situation is
14344 similar to that of traditional mark-resight models where the unmarked individuals provide
14345 no information about the parameters of the capture process. Under these models, capture
14346 or re-sight probability can only be estimated by marking a subset of the population. In
14347 the next chapter we demonstrate how data from marked and unmarked individuals can
14348 be combined to improve precision and allow for the estimation of parameters under the
14349 alternative observation models.

14350 18.8.1 Spatial point process models

14351 Our model has some direct linkages to existing point process models. We note that
14352 the observation intensity function (i.e., corresponding to the observation locations) is a

14353 compound Gaussian kernel similar to that of the Thomas process (Thomas, 1949; Møller
 14354 and Waagepetersen, 2004, pp. 61-62). Also, the Poisson-Gamma Convolution models
 14355 (Wolpert and Ickstadt, 1998) are structurally similar (see also Higdon (1998) and Best
 14356 et al. (2000)). In particular, our model is such a model but with a *constant* basal encounter
 14357 rate λ_0 and *unknown* number and location of “support points”, which in our case are the
 14358 animal activity centers, s_i . We can thus regard our model as a model for *estimating*
 14359 the location and local density of support points in such models, which we believe could
 14360 be useful in the application of convolution models. Best et al. (2000) devise an MCMC
 14361 algorithm for the Poisson-Gamma model based on data augmentation, which is similar
 14362 to the component of our algorithm for updating the z variables in the conditional-on- z
 14363 formulation of the model. We emphasize that our model is distinct from these Poisson-
 14364 Gamma models in that the number *and* location of such support points are estimated.

14365 If individuals were perfectly observable then the resulting point process of locations is
 14366 clearly a standard Poisson or Binomial (fixed N) cluster process or Neyman-Scott process.
 14367 If detection is uniform over space but imperfect, then the basic process is unaffected by
 14368 this random thinning. Our model can therefore be viewed formally as a Poisson (or
 14369 Binomial) cluster process model but one in which the thinning is non-uniform, governed
 14370 by the encounter model which dictates that thinning rate increases with distance from
 14371 the observation points. In addition, our inference objective is, essentially, to estimate the
 14372 number of parents in the underlying Poisson cluster process, where the observations are
 14373 biased by an incomplete sampling apparatus (points in space).

14374 As a model of a thinned point process, our model has much in common with classical
 14375 distance sampling models (Buckland et al., 2001). The main distinction is that our data
 14376 structure does *not* include observed distances, although the underlying observation model
 14377 is fundamentally the same as in distance sampling if there is only a single replicate sample
 14378 and s_i is defined as an individual’s location at an instant in time. For replicate samples,
 14379 our model preserves (latent) individuality across samples and traps which is not a feature
 14380 of distance sampling. We note that error in measurement of distance is not a relevant
 14381 consideration in our model, and we explicitly do not require the standard distance sampling
 14382 assumption that the probability of detection is 1 if an individual occurs at the survey
 14383 point. More importantly, distance sampling models cannot be applied to data from many
 14384 of the sampling designs for which our model is relevant. For example, many rare and
 14385 endangered species can only be effectively surveyed using methods such as hair snares and
 14386 camera traps that do not produce distance data (O’Connell et al., 2010).

18.9 CONCLUSION

14387 Concerns about “statistical independence” have prompted ecologists to design count-based
 14388 studies such that observed random variables can be regarded as *i.i.d.* outcomes (Hurlbert,
 14389 1984). Interestingly, this often proves impossible in practice, and elaborate methods have
 14390 been devised to model spatial dependence as a nuisance parameter. Our paper presents
 14391 a modeling framework that directly confronts this view by demonstrating that spatial
 14392 correlation carries information about the locations of individuals, which can be used to
 14393 estimate density even when individuals are unmarked and distance-related heterogeneity
 14394 exists in encounter probability.

14395 In this paper, we confronted one of the most difficult challenges faced in wildlife sam-

pling — estimation of density in the absence of data to distinguish among individuals. To do so, we developed a novel class of spatially-explicit models that applies to spatially organized counts, where the count locations or devices are located sufficiently close together so that individuals are exposed to encounter at multiple devices. This design yields correlation in the observed counts, and this correlation proves to be informative about encounter probability parameters and hence density. We note that sample locations in count-based studies are typically *not* organized close together in space because conventional wisdom and standard practice dictate that independence of sample units is necessary (Hurlbert, 1984). Our model suggests that in some cases it might be advantageous to deviate from the conventional wisdom if one is interested in direct inference about density. Of course, this is also known in the application of standard spatial capture-recapture models (Borchers and Efford, 2008) where individual identity is preserved across trap encounters, but it is seldom, if ever, considered in the design of more traditional count surveys.

Our model has broad relevance to an incredible number of animal sampling problems. Our motivating problem involved bird point counts where individual identity is typically not available. The model also applies to other standard methods used to sample unmarked populations, such as camera traps or even methods that yield sign (*e.g.* scat, track) counts indexed by space. However, results of our simulation study reveal some important limitations of the basic estimator applied to situations in which none of the individuals can be uniquely identified. In particular, posterior distributions are highly skewed in typical small to moderate sample size situations and posterior precision is low.

19

14417
14418

14419 SPATIAL MARK-RESIGHT MODELS FOR 14420 PARTIALLY IDENTIFIABLE 14421 POPULATIONS

14422 So far, we have dealt with the situation where all detected individuals are identifiable
14423 upon encounter, and in Chapt. 18 we introduced and developed an SCR model for non-
14424 identifiable populations, a spatial *non*-capture-recapture model, if you will. These two
14425 extremes are common in the study of animal populations with non-invasive sampling meth-
14426 ods. However, there is also an intermediate situation, where a part of the population is
14427 tagged or otherwise marked and can thus be identified upon recapture, while the untagged
14428 portion remains unidentified. In this situation so-called mark-resight models (Bartmann
14429 et al., 1987; Arnason et al., 1991; Neal et al., 1993) can be used to estimate population
14430 size and density combining data from both the marked and unmarked individuals.

14431 Traditionally, capture-recapture studies involve physical capture of individuals through-
14432 out the study; new individuals are marked on every re-capture occasion. This methodology
14433 is still widely applied in the study of species that are relatively easy to capture, such as
14434 small mammals, but can be very costly, logically challenging and risky when dealing
14435 with larger species. In contrast, in mark-resight studies a sample of individuals is captured
14436 and tagged (or otherwise marked) during a single marking event. Marking is followed by
14437 resighting surveys, upon which both the detection of marked and recognizable individuals
14438 and unmarked animals is recorded. Resighting surveys are usually non-invasive (hence the
14439 name resighting), so that they don't involve handling of animals. As such, mark-resight
14440 models have a major advantage over traditional capture-recapture models in that they
14441 only require individuals to be captured and handled once, during the initial marking.
14442 This reduces field costs and risks for the animals (and potentially the researchers).

14443 Mark-resight models have a set of underlying assumptions, most of which are analogous
14444 to those of capture-recapture models, e.g. demographic population closure (violation
14445 of geographic population closure can be accommodated by some models) and no loss
14446 or misidentification of marks (see also 5). Just like regular capture-recapture models,

14447 there are means to incorporate heterogeneity in capture probability. However, a new and
14448 essential assumption of mark-resight models is that the tagged (or otherwise identifiable)
14449 individuals are a representative sample of the study population, so that inference about
14450 detection can be made for the whole population from the tagged sample. This issue is
14451 usually addressed by using a different method for marking than for resighting, and by
14452 marking a random sample of the population.

14453 Owing to the advantages of mark-resight over capture-recapture, especially when dealing
14454 with hard-to-trap species, mark-resight is a popular tool in wildlife population studies.
14455 The method has been applied for decades and to a suite of species and survey techniques,
14456 ranging from banding and resighting Canada geese (Hestbeck and Malecki, 1989) to ear-
14457 tagging and camera-trapping grizzly bears (Mace et al., 1994) to paintball marking and
14458 areal resightings of large ungulates (Skalski et al., 2005).

14459 In this chapter we consider mark-resight in spatial context and develop a spatial mark-
14460 resight (SMR) model. To motivate this model development, imagine you conduct a live-
14461 trapping study during which you capture and mark a number of animals with individually
14462 recognizable tags. Subsequently, you go back out to the field and conduct resighting
14463 surveys on an array of locations, and during these resighting surveys you see some of your
14464 tagged individuals as well as new, untagged ones. Then, for the tagged animals you obtain
14465 the same form of spatially explicit individual encounter histories as you would in a regular
14466 SCR study. On top of that you obtain site (and occasion) specific counts of individuals you
14467 did not tag. Thus, spatial mark-resight is an SCR framework for populations where only
14468 part of the individuals can be identified and the major difference between SCR and SMR
14469 is how we include those counts of unmarked individuals in the model. In the following
14470 sections we first provide some background information on mark-resight and the types of
14471 data such surveys can provide. We then move on to the formal development of SMR
14472 models, which, as we will see, are hybrids of regular SCR models and the models for data
14473 without individual identity presented in Chapt. 18.

19.1 BACKGROUND

14474 19.1.1 Types of partial ID data

14475 Before we start exploring mark-resight approaches in more detail, we need a clear un-
14476 derstanding of what types of mark-resight data we can have, in order to appreciate and
14477 understand the different flavors of mark-resight models. In general, we have (at least) two
14478 sets of data: encounter histories for identifiable individuals i at trap j and occasion k ,
14479 y_{ijk} , and counts of unidentified records for each j and k , n_{jk} . Depending on the sampling
14480 technique, we can conceive of three slightly different types of partial ID data.

14481 **(1) Known number of tagged individuals** If you implement your resighting survey
14482 shortly after the marking session, you may be confident that none of the marked individuals
14483 has died or lost its mark. Under these circumstances you know that the number of marked
14484 individuals available for resighting, m , is equal to the number of individuals you tagged.
14485 Alternatively, tags might be radio-transmitters, allowing you to confirm the presence or
14486 absence of marked individuals in the resighting survey area using radio-telemetry (White
14487 and Shenk, 2001). In both cases, you know the number of marked individuals in the
14488 population you survey. In this situation, even though you may fail to resight some of the
14489 tagged individuals, since you know how many there are, you can simply assign those you

14490 never resighted all-zero encounter histories - in other words, contrary to regular capture-
14491 recapture models, in mark-resight models with a known number of tagged individuals, we
14492 can observe all-zero encounter histories. Under these circumstances, estimating N reduces
14493 to estimating the number of unmarked individuals, U .

14494 **(2) Unknown number of tagged individuals** If we suspect that some of the
14495 marks may have been lost between tagging and conducting the resighting samples, we
14496 obtain a slightly different type of mark-resight data. Here, we do not accurately know
14497 the number of marked individuals available for resighting. As a consequence, individuals
14498 have to be resighted at least once for us to know they are still tagged and alive and thus
14499 available for resighting. So, contrary to the situation where we know m and analogous to
14500 regular capture-recapture models, we cannot observe all-zero encounter histories of marked
14501 individual. Here, estimating N involves estimating both m and U .

14502 A special case of this kind of data can arise from camera trapping. Even when dealing
14503 with a species that has no spots or stripes, some individuals in the study population can
14504 have natural marks that make them identifiable on pictures, such as scars or some distinct
14505 coloration. Again, in this scenario an individual has to be photographed at least once to
14506 be known. Here, the fact that both the “marking” method and the subsequent resighting
14507 method are the same (although marking in this case does not involve any actual physical
14508 marking) can be cause for concern: our sample of “marked” individuals may not be a
14509 random sample of the population but consist of individuals that for some reason are more
14510 likely to be photographed. In that case, a basic assumption of the mark-resight model is
14511 violated.

14512 **(3) Unknown marked status** Finally, consider a scat or hair snare survey, where
14513 only a part of the sample is analyzed genetically (or DNA can only be extracted from
14514 a subset of samples due to sample quality). In this scenario, your n_{jk} can contain both
14515 completely unknown individuals that are not represented at all in Y , but it can also contain
14516 samples from individuals that we previously identified. The difference is that in the first
14517 two scenarios, part of the population of individuals is identifiable, while in the second
14518 scenario, part of the samples is identifiable. This type of data violates one of the basic
14519 assumptions of mark-resight models, namely, that tagged individuals are always correctly
14520 identified as such.

14521 To our knowledge there are currently no mark-resight models available that account for
14522 possible misidentification of the marking status of individuals (although some literature is
14523 available on misidentification of individuals in capture-recapture studies, e.g., Yoshizaki
14524 et al., 2009; Lukacs and Burnham, 2005; Link et al., 2010). In this chapter we will ignore
14525 this kind of data and focus instead on the two types of typical mark-resight data:

- 14526 (1) Known number of tagged individuals
14527 (2) Unknown number of tagged individuals,

14528 For both types of data a slightly different situation arises when in some instances we
14529 can only tell that an individual is tagged, but not who it is. You may be able to see that
14530 an individual is tagged but the identifying feature of the tag (a number or coloration)
14531 may have become unreadable, or may be hidden from view. In this case, in addition to
14532 your y_{ijk} and your n_{jk} you also have a number of sightings of tagged but unidentified
14533 individuals, say r_{jk} .

14534 **19.1.2 A short history of mark-resight models**

14535 Initially, mark-resight methods focused on radio-tagged individuals to estimate popula-
 14536 tion size (White and Shenk, 2001). Radio-collars provide a means of determining which
 14537 of the animals were in the study area and available for sampling, i.e. determining the
 14538 number of marked individuals in the population. Knowing this number was a prerequisite
 14539 for most earlier mark-resight approaches (White, 1996). The oldest mark-resight model
 14540 is the good old Lincoln-Petersen estimator, where individuals are marked and a single
 14541 resight/recapture occasion is carried out (Krebs, 1999). We need not identify individuals,
 14542 but only tell apart marked from unmarked individuals. Let m be the number of marked
 14543 individuals in the population, $m_{(R)}$ the number of marked individuals seen on the resight-
 14544 ing occasion, and $n_{(R)}$ the total number of marked and unmarked individuals observed
 14545 during resighting. Population size N is then estimated as

$$N = m \times n_{(R)} / m_{(R)}$$

14546 A suite of more elaborate models using individual capture histories over several re-
 14547 sighting occasions were developed in the 1980s and 90s and compiled into the program
 14548 NOREMARK (White, 1996). Apart from the basic model with known number of marked
 14549 individuals and no individual variation in resighting probabilities (joint hypergeometric
 14550 maximum likelihood estimator) (Bartmann et al., 1987; White and Garrot, 1990; Neal,
 14551 1990; Neal et al., 1993), NOREMARK contains models that account for lack of geographic
 14552 population closure (Neal et al., 1993), individual heterogeneity in resighting rates and
 14553 sampling with replacement (i.e. individuals can be seen more than once on any occasion,
 14554 (Minta and Mangel, 1989; Bowden, 1993)). A first mark-resight model allowing for an
 14555 unknown number of marked individuals was developed by Arnason et al. (1991).

14556 While many of these models perform well under certain situations, they are somewhat
 14557 limited: they do not allow for combining data across several surveys (McClintock et al.,
 14558 2006) and not all of them are likelihood-based or allow for different parameterizations
 14559 (e.g., including a time effect on detection), so that selection of the most appropriate
 14560 model cannot be based on standard approaches such as AIC, but is largely left up to
 14561 educated guesswork (McClintock et al., 2006). Recently, more flexible and generalized
 14562 likelihood-based mark-resight models have been developed. These models can account
 14563 for individual heterogeneity in detection, unknown number of marked individuals and
 14564 lack of geographical closure, as well as a less than 100% individual identification rate of
 14565 tagged individuals; they can be applied to sampling with and without replacement and
 14566 can combine data across several primary sampling occasions in a robust design type of
 14567 analysis (McClintock et al., 2009a,b). Since they are all likelihood-based, model selection
 14568 among different parameterizations and model averaging based on AIC is an option. Most
 14569 of these models have also been incorporated into the program **MARK** (McClintock and
 14570 White, 2012).

14571 For a detailed treatment of these different non-spatial mark-resight models, we refer
 14572 you to the original papers cited in the preceding paragraph. In short, these models are
 14573 based on the joint likelihood of two major model components: one describing the resight-
 14574 ing process of marked individuals (either using a Poisson or a Bernoulli observation model,
 14575 depending on whether sampling is with or without replacement), where resighting proba-
 14576 bilities can have both fixed effects to model individual and environmental covariates, and

14577 a random-effect component to accommodate variation in detection due to individual het-
 14578 erogeneity; and one describing the number of unmarked individuals observed (or, under a
 14579 Poisson observation model, the number of times unmarked individuals are observed), n_t
 14580 (t here and in the following description denotes a primary sampling occasion, for example,
 14581 a year or a season; for a single-season study we could easily drop this subscript) which are
 14582 approximated as a normal distribution (McClintock et al., 2006), or a normal distribution
 14583 left-truncated at 0 (McClintock et al., 2009a):

$$n_t \sim \text{Normal}(E(n_t), V(n_t))$$

14584 Although this is a simplification of the actual sampling process, McClintock et al. (2006)
 14585 found this normal distribution to be a satisfactory approximation, which allows N to enter
 14586 the model likelihood via $E(n_t)$ and $V(n_t)$.

14587 In the simplest model case without any variation in detection, the expected number
 14588 of resightings of unmarked individuals, $E(n_t)$, can be written as the number of unmarked
 14589 individuals times the expected number of detections of a single individual, which is the
 14590 mean or expected value of the underlying observation model:

$$E(n_t) = (N - m) * \theta \quad (19.1.1)$$

14591 where $\theta = K \times p$ for a Binomial observation model with K replicates and individual
 14592 detection probability p , or $\theta = \text{expected/average individual encounter rate } \lambda$ for a Poisson
 14593 observation model. Similarly, $V(n_t)$ depends on the underlying observation model and is
 14594 based on the parameters that determine the individual detection probability/encounter
 14595 rate. Combining these two components, N is directly incorporated into the joint likelihood
 14596 of the model.

14597 While these mark-resight models are very flexible, they share the shortcomings of
 14598 regular capture-recapture models when it comes to estimating population density (e.g.,
 14599 Chaps. 1 and 4). As long as resightings are collected across a network or array of loca-
 14600 tions, however, they come with the same spatial information as recaptures in a regular
 14601 SCR study. In the following sections we will consider mark-resight sampling in the frame-
 14602 work of spatial capture-recapture. We will look at models for both known and unknown
 14603 numbers of marked individuals, and for imperfect individual identification of marks. In the
 14604 spatial framework, most of the information on model parameters comes from the marked
 14605 individuals. But in sec. 19.5 we will see that, analogous to the models we developed in
 14606 the previous Chapt. 18, the spatial correlation in counts of unmarked individuals also
 14607 contributes information about detection and movement.

19.2 KNOWN NUMBER OF MARKED INDIVIDUALS

14608 We begin with the easiest situation: a known number of individuals constituting a random,
 14609 representative sample from the population are marked and a series of resight samples are
 14610 conducted following marking. No marks (or marked animals) are lost between marking
 14611 and resighting, all individuals are correctly identified as marked or unmarked, and marked
 14612 individuals are 100 % correctly identified to individual level.

14613 Recall from Chapt. 18 that without any individual identity, the observed counts at
 14614 trap j and occasion k , n_{jk} , represent the sum of all latent individual detections at j and

14615 $k, \sum_{i=1}^N y_{ijk}$, where y_{ijk} are the latent individual encounter histories which we include as
 14616 variables (or missing data) in our MCMC scheme. We can model these counts as

$$n_{jk} \sim \text{Poisson}(\Lambda_j)$$

14617 where

$$\Lambda_j = \sum_{i=1}^M (\lambda_{ij})$$

14618 Under this formulation we do not need to update the individual y_{ijk} in our model, which
 14619 is more efficient in terms of computing. However, we can also formulate the model as
 14620 conditional on the latent y_{ijk} . This is useful because if we have m individually known
 14621 animals in our study population, than those m y_{ijk} are no longer latent, but fully observed
 14622 and can easily be included in the analysis to provide information on detection parameters.

14623 The formulation conditional on y_{ijk} basically brings us back to the original SCR model,
 14624 where individual site and occasion specific counts, y_{ijk} , are modeled as

$$y_{ijk} \sim \text{Poisson}(\lambda_{ij})$$

14625 and

$$\lambda_{ij} = \lambda_0 \exp(-d_{ij}^2 / (2\sigma^2))$$

14626 Unobserved y_{ijk} are essentially missing data and have to be updated as part of the
 14627 MCMC procedure. We can do that by using their full conditional distribution, which is
 14628 multinomial with sample size n_{jk} :

$$y_{ujk} \sim \text{Multinomial}(n_{jk}, \lambda_{uj})$$

14629 where u is an index vector of the $M - m$ hypothetical unmarked individuals.

14630 While in the non-spatial mark-resight analysis known individuals provide direct information
 14631 about individual detection probability (or rate), in the spatial setting they also
 14632 inform σ . Including known individuals into the analysis helps estimate model parameters
 14633 more accurately and precisely. We will address the relationship between the number of
 14634 marked individuals and accuracy of the estimated parameters in sec. 19.5.

14635 19.2.1 MCMC for a spatial mark-resight model

14636 Implementing a spatial mark-resight model in **JAGS** is not trivial, since the program
 14637 does not accept partially observed multivariate nodes (in this case the partially observed
 14638 individual encounter histories which we model as coming from a multinomial distribution).
 14639 Therefore, knowing how to write your own MCMC algorithm comes in extremely handy.
 14640 You will find that we only have to make relatively simple modifications to the MCMC
 14641 code for the model without any individual identification presented in Chapt. 18, which,
 14642 in turn, has much in common with the algorithms we developed for regular SCR models
 14643 in Chapt. 17. Essentially, since we observe individual detections for the marked part
 14644 of the population, we have to update only the unobserved part of \mathbf{Y} , and modify the
 14645 updating steps for z_i and ψ , the parameters introduced by data augmentation, to reflect
 14646 some contribution to our knowledge of these parameters from the m marked individuals.

14647 First, we set up an array to hold \mathbf{Y} , fill the first m rows of the array with the m
 14648 observed individual encounter histories, then update \mathbf{Y} for the unknown individuals only
 14649 (note that the code is set up so that n_{jk} contains both pictures of marked **and** unmarked
 14650 individuals at j and k):

```
14651 # set up placeholders and create vectors for marked and unmarked
14652 Y <- array(NA, c(M, J, K))
14653 nMarked <- nrow(y)
14654 marked <- rep(FALSE, M)
14655 marked[1:nMarked] <- TRUE
14656 Y[1:nMarked, , ] <- y
14657 z[marked] <- 1
14658 Ydata <- !is.na(Y)
14659 for (j in 1:J) {
14660   for (k in 1:K) {
14661     if (y[j, k] == 0) {
14662       Y[, j, k] <- 0
14663       next
14664     }
14665     unmarked <- !Ydata[, j, k]
14666     nUnknown <- n[j, k] - sum(Y[!unmarked, j,k])
14667     if (nUnknown < 0)
14668       browser()
14669     probs <- lam[, j] * z
14670     probs <- probs[unmarked]
14671     probs <- probs/sum(probs)
14672     Y[unmarked, j, k] <- rmultinom(1, nUnknown, probs)
14673   }
14674 }
```

14675 When we know the number of marked individuals in the population estimating N
 14676 is reduced to estimating u . Thus, we only need to estimate the z_i for $M - m$ unknown
 14677 individuals and the updater for z_i becomes:

```
14678 zUps <- 0
14679 seen <- apply(Y > 0, 1, any)
14680 for (i in 1:M) {
14681   if (seen[i] | marked[i])
14682     next
14683   zcand <- ifelse(z[i] == 0, 1, 0)
14684   ll <- sum(dpois(Y[i, , ], lam[i, ] * z[i], log = TRUE))
14685   llcand <- sum(dpois(Y[i, , ], lam[i, ] * zcand,
14686                     log = TRUE))
14687   prior <- dbinom(z[i], 1, psi, log = TRUE)
14688   prior.cand <- dbinom(zcand, 1, psi, log = TRUE)
14689   if (runif(1) < exp((llcand + prior.cand) - (ll +
14690     prior))) {
```

```

14691      z[i] <- zcand
14692      zUps <- zUps + 1
14693      }
14694  }

```

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 $\sum z - m$:

```
14702  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`.

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:

```

14715  Y <- array(NA, c(M, J, K))
14716  #[...]
14717  for (j in 1:J) {
14718    for (k in 1:K) {
14719      if (y[j, k] == 0) {
14720        Y[, j, k] <- 0
14721        next
14722      }
14723      unmarked <- !Ydata[, j, k]
14724      nUnknown <- n[j, k] - sum(Y[!unmarked, j,k])
14725      if (nUnknown < 0)
14726        browser()
14727      probs <- lam[, j] * z
14728      probs <- probs[unmarked]
14729      probs <- probs/sum(probs)
14730      Y[unmarked, j, k] <- 0
14731      guys <- sample(which(unmarked), nUnknown, prob = probs)
14732      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

14733 }

14734 }

14735 **Example: Canada geese in North Carolina**

14736 We applied the spatial mark-resight model with a binomial encounter process to a
 14737 dataset of Canada goose resightings (Rutledge, 2012) XXXget full citation with LizXXX.
 14738 During the molt of 2008, 751 individual geese were captured and tagged with neck and
 14739 leg bands in Greensboro, North Carolina (Fig. 19.1). Geese were resighted at 87 different
 14740 locations on 81 resighting events over a period of 18 months. In addition to the banded
 14741 geese, the number of unmarked geese was recorded during each resighting event. Here,
 14742 we only looked at a subset of the data, from mid July to the end of October 2008, which
 14743 corresponds to the first part of the post-molt season, before migratory Canada geese arrive
 14744 in North Carolina. During this time frame, 746 of the 751 marked geese were known to
 14745 be alive. Of those, 654 were resighted 3994 times at 40 different sites. In addition, 7944
 14746 sightings of unmarked geese were recorded at 48 sites.

14747 In this model, we also allowed σ to vary between males and females. We augmented
 14748 the data set with $4500 - m$ all-zero encounter histories, ran 50000 MCMC iterations and
 14749 removed a burn-in of 1000 iterations. We provide all the data (`data('canadageese')`)
 14750 and functions (`pIDgeese`) for you to repeat this analysis but be aware that given the large
 14751 data set it will take days to do so. The **R** code to set up the data and run 5000 iterations
 14752 of the goose model is given as an example on the help page for `pIDgeese`. The model
 14753 results, including the derived parameter density (D) in individuals per km^2 are shown in
 14754 Tab. 19.2.2.

14755 We see that credible intervals of estimates are pretty narrow. Take, for example, σ for
 14756 males and females: Although they differ only by 0.08, there is barely any overlap between
 14757 the respective credible intervals, surely an effect of the large data set. The parameter ϕ in
 14758 this model is the probability of being a male, a measure of the sex ratio of the population,
 14759 which is close to 1:1.

19.3 UNKNOWN NUMBER OF MARKED INDIVIDUALS

14760 Now let us consider the case where we do not know the exact number of tagged individuals
 14761 available for resighting so that we have to capture an individual at least once to be sure
 14762 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)

14763 animals available for resighting, treating this number as unmarked is probably more realistic
14764 in most circumstances. As a consequence of not knowing the exact number of marked
14765 individuals, we cannot observe all-zero encounter histories. When using maximum likelihood
14766 inference, this situation requires a model where detection rates of known individuals
14767 are modeled using a zero-truncated distribution (McClintock et al., 2009a). If we did not
14768 account for the fact that zeros are unobservable, our estimates of detection rates would
14769 be artificially inflated and estimates of population size would be negatively biased.

14770 Working with zero-truncated distributions in a spatial mark-resight setting is less
14771 straight-forward than for non-spatial mark-resight. A marked individual only has to show
14772 up once, anywhere on the resighting array, for us to know that it is there. When resightings
14773 are pooled across the entire sampling grid, then the total individual counts $\sum_j y_{ij}$ have
14774 to be > 0 for all resighted individuals and a zero-truncated distribution can be used to
14775 model these counts. However, we are concerned with trap-specific encounters, y_{ij} , which
14776 can easily be 0 for a resighted individual, as long as a single y_{ij} is > 0 . Thus, the zero-
14777 truncation does not apply to the individual and trap specific counts we observe, but only

14778 to the sum of these counts over all traps.

14779 As an alternative to a zero-truncated distribution, in a Bayesian framework, we can
 14780 make use of data augmentation to estimate the number of marked individuals¹. In the
 14781 previous example, where we knew the number of marked individuals, we separate those
 14782 individuals from the augmented population by fixing their z_i at 1 and letting ψ refer only
 14783 to the unmarked population, $M - m$. All we have to do in the spatial mark-resight model
 14784 with unknown number of marked individuals is to let our marked individuals be part of
 14785 the augmented population again, analogous to the situation in regular SCR models:

```
14786     psi <- rbeta(1, 1 + sum(z), 1 + M - sum(z))
```

14787 Whether you have a known or an unknown number of marked individuals is included
 14788 as an option in **scrPID**.

14789 A simulation example

14790 For illustration purposes we simulated a data set with $N = 80$ individuals randomly
 14791 distributed across a state space of 10x10 units. Of those, we randomly choose 40 to be
 14792 marked and identifiable, and then simulate encounter data for both marked and unmarked
 14793 individuals on an 8x8 grid with unit spacing over $K = 5$ occasions, with $\sigma = 0.5$ and $\lambda_0 =$
 14794 0.5, adopting a Poisson encounter process. To do so we use the **scrbook** function **sim.data**,
 14795 which also allows you to create data sets from a Binomial observation process, known
 14796 number of marked individuals, and with telemetry locations (sec. 19.6) or individual
 14797 identification rate < 100 % (sec. 19.4). We analyzed the simulated data both assuming
 14798 we do not know the total number of marked animals in our state space, and assuming we
 14799 do know this number, using the **scrPID** function and running 20000 iterations. You can
 14800 repeat the analysis by executing the R code below.

```
14801 set.seed(2501)
14802
14803 #set input values
14804 N=80
14805 lam0=0.5
14806 knownID=40
14807 rat=0.8
14808 sigma=0.5
14809 K=5
14810
14811 #create grid and state space
14812 coords<-seq(0,7, 1)
14813 grid<-expand.grid(coords, coords)
14814 trapmat<-as.matrix(grid)
14815 buff<- 3*sigma
14816 xl<-min(trapmat[,1])-buff
14817 xu<-max(trapmat[,1])+buff
14818 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

```

14819 yu<-max(trapmat[,2])+buff
14820 xlims=c(xl, xu)
14821 ylims=c(yl,yu)
14822 area<-(xu-xl)*(yu-yl)
14823
14824 #simulate data
14825 dat<-sim.pID.data(N=N, K=K, sigma=sigma, lam0=lam0, knownID=knownID,
14826 X=trapmat, xlims=xlims, ylims=ylims, obsmod= "pois",
14827 nmarked="unknown", rat=1, tel =0, nlocs=0)
14828
14829 #create initial values function for scrPID, set M and tuning parameters
14830 inits<-function(){list(S=cbind(runif(M, xlims[1], xlims[2]),
14831 runif(M, ylims[1], ylims[2])), lam0=runif(1, 0.4, 0.6),
14832 sigma=runif(1, 0.4, 0.6), psi=runif(1, 0.4, 0.6))}
14833 M<-160
14834 delta=c(0.1, 0.01, 2)
14835
14836 #run model, first m=unknown, then m=known
14837 mod<-scrPID(n=dat$n, X=trapmat, y=dat$Yobs, M=M, obsmod = "pois",
14838 nmarked="unknown", niters=20000, xlims=xlims, ylims=ylims,
14839 inits=inits(), delta=delta ) )
14840 mod2<-scrPID(n=dat$n, X=trapmat, y=dat$Yobs, M=M, obsmod = "pois",
14841 nmarked="known", niters=20000, xlims=xlims, ylims=ylims,
14842 inits=inits(), delta=delta ) )
14843

```

14844 Looking at the data, we see that of the 40 marked animals, 26 were recorded at least
14845 once. In terms of data that means that in the second model, where we know m , we have
14846 14 observed all-zero encounter histories that we cannot use in the model where we assume
14847 m is not known. This reduction in data is reflected in the model results (Tab. 19.3). The
14848 estimate of N for the unknown- m model shows some positive bias, although the 95 % BCI
14849 still includes the true value of 80. Thus, while we can formally account for the fact that we
14850 often do not know the number of marked individuals in the state space, we clearly loose
14851 quite a bit of accuracy and precision. It would be an interesting little project to quantify
14852 this loss in accuracy and precision in a small simulation study.

19.4 IMPERFECT IDENTIFICATION OF MARKED INDIVIDUALS

14853 Often during resighting, it may be possible to see that an individual is tagged but impos-
14854 sible to determine its individual identity. In such a situation in addition to the y_{ijk} and
14855 n_{jk} , we also have site and occasion specific counts of marked but unidentified individuals,
14856 r_{jk} . Here, the individual encounter histories of marked animals are incomplete, and if we
14857 used these incomplete data to inform the detection parameter of the model, we would run
14858 the risk of underestimating detection/trap encounter rate and overestimating abundance.
14859 Some non-spatial mark-resight models do not require that marked animals be identified
14860 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

14861 individual level information means that we cannot accommodate heterogeneity in detection
 14862 (McClintock and White, 2012). In a spatial framework we could ignore marked and
 14863 unmarked status completely and apply the model by Chandler and Royle (In press) we
 14864 discussed in Chapt. 18. But, that would mean losing important information on individual
 14865 detection and movement. Therefore, being able to retain the individual identity of records
 14866 that can be identified while at the same time accounting for imperfect identification of
 14867 marked individuals is extremely useful.

14868 McClintock et al. (2009a,b) suggest an intuitive means of correcting for this bias in a
 14869 non-spatial model framework when dealing with a Poisson encounter model (or sampling
 14870 with replacement). When marked but unknown resightings are part of the data, the
 14871 expected number of records of unmarked individuals at time t , n_t , changes from Eq.
 14872 19.1.2 to:

$$E(n) = (N - m)\lambda + \eta/m$$

14873 Here, λ is the individual encounter rate estimated from the known resighted individuals
 14874 and η is the number of records of marked but unidentified individuals. So, because the
 14875 observed λ is known to be too low, the average number of unidentified pictures per known
 14876 individual is added as a correction factor. This procedure assumes that the inability to
 14877 identify a marked individual occurs at random throughout the population, which seems
 14878 to be a reasonable assumption under most circumstances.

14879 We can relatively easily translate this concept to our spatial mark-resight models. In
 14880 the spatial model framework we are interested in the individual and trap specific encounter
 14881 rate, λ_{ij} . Further, we do not look at the sum of all records of unmarked individuals, but
 14882 formulate the model conditional on the latent individual encounter histories. Thus, instead
 14883 of using η/m as a correction factor, we need something that applies at the individual and
 14884 trap level. If we take the sum of all correctly identified records of marked individuals,
 14885 $\sum y_c$ and divide it by the total number of records of marked individuals, $\sum y_m$, we get
 14886 the average rate of correct individual identification for marked individuals, say, c :

$$c = \sum y_c / \sum y_m$$

14887 We could then apply c as a correction factor for λ_0 for the marked individuals.

14888 A more formal, model-based way to specify c is by assuming that

$$\sum y_c \sim \text{Binomial}(\sum y_m, c)$$

14889 and estimating c as another model parameter, so that we account for the uncertainty about
 14890 it. If we choose an uninformative (and conjugate) beta(1, 1) prior for c , we can update it
 14891 directly from its full conditional distribution, which is beta($1 + \sum y_c, 1 + (\sum y_m - \sum y_c)$),
 14892 within our MCMC algorithm.

14893 For the marked individuals we can then multiply λ_0 with c to account for the fact that
 14894 we observe incomplete individual encounter histories. Since we don't have this identifica-
 14895 tion issue for unmarked individuals, their baseline trap encounter rate remains as before
 14896 simply λ_0 (or in other words, their c equals 1). Observe that now, in addition to assuming
 14897 that failure to identify tagged individuals occurs at random throughout the population,
 14898 we also assume that it occurs at random throughout space, i.e. our success of identifying
 14899 a tagged individual does not depend on the trap we encounter it in. Incomplete individual
 14900 identification of marked individuals is included as an option in the `scrPID` function and
 14901 we show an example of using c in an analysis in sec. 19.6.

14902 **Imperfect individual identification and unknown number of marks.** The ap-
 14903 proach described above works only if the number of marked individuals is known because,
 14904 in that case, we can observe the all-zero encounter histories of marked individuals and
 14905 know that all augmented individuals have to be unmarked individuals. If the number of
 14906 marked animals is unknown, on the other hand, some of the augmented individuals may
 14907 well be marked individuals we never observed. For those individuals we should multiply
 14908 λ_0 with c , but we don't know who (or how many) they are. As of this moment we have
 14909 not implemented a model with unknown number of marked individuals and imperfect
 14910 identification of marks. It seems like one strategy to tackle that problem would be to
 14911 estimate the number of marked and unmarked individuals separately, using two sets of
 14912 data augmentation (i.e., estimating ψ_{marked} and ψ_{unmarked}), but with shared detection
 14913 parameters, σ and λ_0 .

14914 As long as individuals are identified based on the same type of tags the assumption that
 14915 failure to identify marked individuals occurs at random throughout the population should
 14916 be valid. The assumption that failure to identify marked individuals occurs at random in
 14917 space could be violated, for example when spatially varying habitat conditions influence
 14918 the ability to recognize individual tags, or when an observer effect influences individual
 14919 identification rates. While we haven't ourselves experimented with it, we believe that the
 14920 approach described above could readily be extended to account for these differences. For
 14921 example, identification rates could be calculated separately for different observers, or be
 14922 modeled as functions of habitat covariates. As an alternative to the approach we present
 14923 here, model development could explore assigning records of marked but unidentified indi-
 14924 viduals to marked individuals in a fashion similar to how unmarked records are assigned
 14925 to hypothetical individuals in this model, namely, based on the location of the record and
 14926 the estimates of home range centers of marked individuals. While this is computationally
 14927 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?

14928 It is intuitive that having marked individuals in the study population should lead to more
 14929 accurate and precise parameter estimates than when no individuals are identifiable. To
 14930 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.

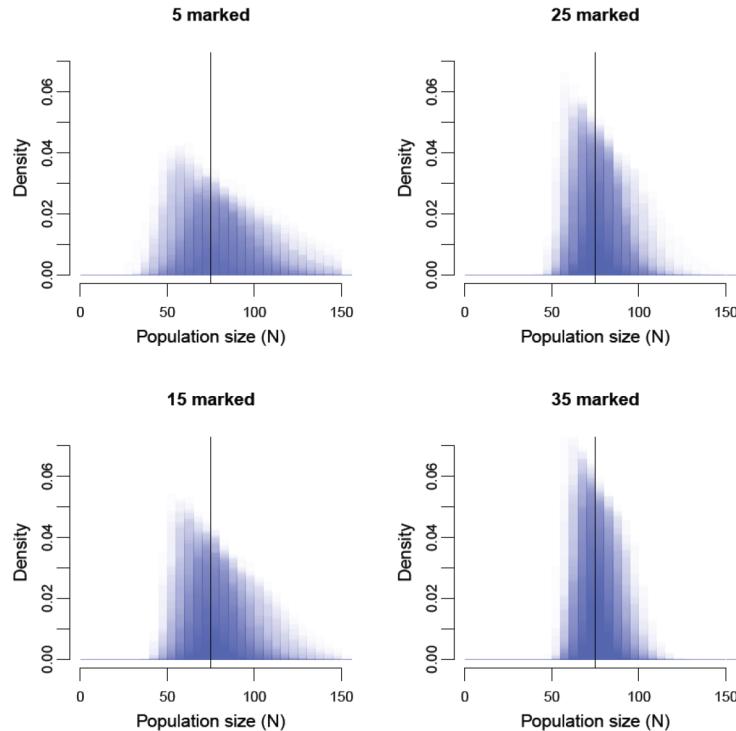


Figure 19.2. Overlaid posterior distributions of N from 100 simulations for four levels of marked individuals.

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:

```
14989 #ntot = number of telemetry-tagged individuals
14990 #locs = list of length ntot; each element is a matrix
```

```

14991 #with telemetry locations
14992 #telID = vector with identifier for telemetry-tagged
14993 #individuals
14994
14995 sigma.cand <- rnorm(1, sigma, delta[1])
14996 if (sigma.cand > 0) {
14997
14998 llsig<-llsig.cand<-rep(NA, ntot)
14999
15000 for (x in 1:ntot) {
15001 lls[x]<-sum(dmvnorm(x=locs[[x]],mean=c(S[telID[x],1],S[telID[x],2]),
15002 sigma=cbind(c(sigma^2,0), c(0,sigma^2)), log=T))
15003 lls.cand[x]<-sum(dmvnorm(x=locs[[x]],mean=c(S[telID[x],1],S[telID[x],2]),
15004 sigma=cbind(c(sigma.cand^2,0), c(0,sigma.cand^2)), log=T))
15005 }
15006 if(runif(1) < exp( sum(lls.cand) - sum(lls) ) ){
15007   sigma<-sigma.cand
15008   lam <- lam0*exp(-(D*D)/(2*sigma.cand*sigma.cand))
15009 }
15010 }

```

15011 For the s_i we use an analogous updater for the telemetry-tagged individuals and the
15012 regular updater for individuals without associated telemetry location information. A full
15013 example can be found in the **R** package **scrbook**, by calling **scrPID.tel**. Note that not
15014 all marked individuals need to be telemetry-tagged, but telemetry data used on the model
15015 should correspond to the period over which resighting surveys were conducted (as we
15016 discussed in Chapt. 5, both the s_i and σ should only be interpreted against the specific
15017 sampling period). Further, this approach of incorporating telemetry data into a spatial
15018 mark-resight model can easily be extended to update σ and s conditional on both resighting
15019 and telemetry data and applies equally to regular SCR models where all individuals are
15020 identifiable.

15021 **Example: Raccoons on the Outer Banks of North Carolina**

15022 Solmann et al. (2012b) applied a spatial mark-resight model with telemetry data to
15023 a camera-trap and radio-telemetry data set from the raccoon population on South Core
15024 Banks, a barrier island within Cape Lookout National Seashore, North Carolina. Between
15025 May and September 2007, 131 raccoons were marked with dog collars and large indi-
15026 vidual numbered cattle tags; 44 of these tagged individuals were equipped with radio
15027 collars. Collared individuals were located using a VHF receiver and antenna, and their
15028 locations were estimated approximately weekly. Twenty camera traps were set up along
15029 the length of South Core Banks and camera trapping data collected between October 1
15030 2007 to January 22 2008 constituted the resighting data in this analysis. During this
15031 period 104 marked individuals, 38 radio-collared, were alive and available for resighting
15032 with camera traps.

15033 The state-space \mathcal{S} was defined as the entire area of South Core Banks island. A
15034 change in the number of photocaptures over the course of the study suggested a variation
15035 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

15075 tagged this enables researchers to make optimal use of all existing data and obtain robust
15076 density estimates without the need for additional invasive methods. Just as SCR, the
15077 spatial mark-resight model framework is flexible to account for a variety of factors that
15078 may influence individual movement and detection, as well as survey-related parameters,
15079 and we saw one example for the Canada geese, where σ was sex-specific.

15080 Spatial mark-resight models are a fairly new development and much remains to be ex-
15081 plored. We mentioned the assignment of marked but unidentified records to actual marked
15082 individuals based on their spatial location, which provides some (though imperfect) infor-
15083 mation of their identity (sec. 19.4). Similarly, records where the marked status cannot be
15084 determined could potentially be included in the model as some form of overall correction
15085 factor on detection. GPS telemetry devices and their ability to collect location data with
15086 much higher frequency offer the opportunity to assign records of collared animals to indi-
15087 viduals based on how close to a given camera the collared individuals were, both in space
15088 and time. In this scenario, individual identity itself could be expressed probabilistically,
15089 leading to an SMR model accounting for potential misidentification. All these possible
15090 extensions can tailor SMR models to specific survey techniques. As such, the approach is
15091 applicable to a wide range of population estimation problems when dealing with animals
15092 that cannot be identified based on natural marks.

15093

15094

20 15095 2012: A SPATIAL CAPTURE-RECAPTURE 15096 ODYSSEY

15097 Capture recapture methods have been a cornerstone of ecological modeling and analysis
15098 for decades. Yet there are essentially no real capture-recapture data sets that come *without*
15099 auxiliary spatial information about location of capture (but sometimes such information
15100 is thrown into the trashcan).

15101 The big point is that we provide a framework for spatial analysis of animal populations
15102 from individual encounter data: MOVEMENT, SPACE USAGE, SPATIAL VARIATION
15103 IN DENSITY – much to be done: how do individuals interact? how is space usage
15104 changing over time, etc...

15105 Topics to discuss here:

15106 (1) Strauss process model (2) Need for general purpose software.... all of the spatial
15107 stuff + open populations in one big model. (3) Efficient computation is still an issue. (4)
15108 Fit and model selection will continue to be important practical issues.

20.1 10 THESIS OR DISSERTATION TOPICS

15109 Future research directions:
15110 Modeling dynamics of the point process. Transient individuals. Dispersal. Things like
15111 that.
15112 Calibration of GoF under meaningful alternatives
15113 Calibration of AIC/DIC and efficacy study
15114 Models for non-uniform point processes that exhibit clustering or repulsion
15115 no-marking model + RSF
15116 occupancy and counts data + SCR data (AOAS and Sollmann et al.)
15117 Spatial genetics – can use SCR to study gene flow, related things....
15118 SCR on dendritic networks (streams and trails).

20.2 THREE DIMESIONAL SPACE

15119 Throughout this book we have treated space as two-dimensional, meaning that activity
15120 centers are assumed to occur on the real plane. This approximation of reality is reasonable
15121 for many terrestrial species, but aquatic organisms, especially marine animals move about
15122 in three-dimensional space. Treating space as three-dimensional could also conceivably
15123 be useful in studies of flying organisms or species that use multiple strata of tall forests;
15124 however, we suspect that two dimensional models of space should suffice in such contexts.
15125 Regardless, a three-dimensional view of space requires that activity centers s_i are indexed
15126 by x, y, z coordinates. In theory, this presents no problem whatsoever. In practice, estima-
15127 tion based on integrated likelihood methods must involve a three-dimensional integration.
15128 This will clearly be more computationally demanding, but it should be possible using
15129 packages such as **R2Cuba**.

20.3 GREGARIOUS SPECIES

15130 Many social species move about in large groups rather than as single individuals. Even
15131 species regarded as solitary often join family groups for some portion of their life cycle.
15132 The consequences of gregariousness?? are x-fold....
15133 To account for this, we change our definition of s_i from the location of an individual's
15134 activity center, to the location of a group's activity center. We then expand our model to
15135 include a submodel for group size, and we can estimate both the density of group activity
15136 centers and total population size.

15137

Part V

15138

15139

Appendices

15140 **APPENDIX I - USEFUL SOFTWARE AND**
15141 **R PACKAGES**

15143 Throughout this book we have used a suite of software and R packages, all of which are
15144 freely available online. To make life a little easier for you, here we provide you with a list
15145 of all software and R packages, download links and some (hopefully) helpful tips regarding
15146 their installation.

20.4 WINBUGS

15147 Although **WinBUGS** (Gilks et al., 1994) is becoming increasingly obsolete with the
15148 faster and more flexible **OpenBUGS** and **JAGS**, there are still situations in which
15149 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
15150 and double-click on the .exe file and follow the installation instructions on the screen. On
15151 64 bit machines, according to the BUGS project you should download a zip file (from the
15152 same page) and unzip it into a folder of your choice. There are a couple of additional
15153 steps to make BUGS run. First, you need to obtain a key (which is free and valid for
15154 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
15155 to update the basic **WinBUGS** version to the most current one (which is from August
15156 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
15157 ready to use after quitting and re-opening it. Remember that **WinBUGS** only runs on
15158 Windows machines. Also, there appears to be a problem installing the program in Vista,
15159 although we have no personal experience with this.

20.4.1 WinBUGS through R

15160 While you can run **WinBUGS** as a standalone application, we recommend you access
15161 it from within **R** using the package **R2WinBUGS** (Sturtz et al., 2005), so you can conve-
15162 niently process your output, make graphs etc. **R2WinBUGS** also allows you to run mod-
15163 els in **OpenBUGS** (see below). You can install the package from within **R** directly
15164 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
15165 manual with some examples ('http://voterview.com/bayes_beach/R2WinBUGS.pdf').

20.5 OPENBUGS

15171 **OpenBUGS** is the up-to-date version of **WinBUGS** and can be downloaded here:
 15172 ''<http://www.openbugs.info/w/Downloads>'' (Windows, Mac and Linux versions are
 15173 available). The name '**OpenBUGS**' refers to the software being open source, so users
 15174 do not need to download a license key, like they have to for **WinBUGS** (although the
 15175 license key for **WinBUGS** is free and valid for life). For Windows, install by double-
 15176 clicking on the .exe file and following the instructions on the installer screen. Compared
 15177 to **WinBUGS**, **OpenBUGS** has more built-in functions. The method of how to deter-
 15178 mine the right updater for each model parameter has changed and the user can manually
 15179 control the MCMC algorithm used to update model parameters. Several other changes
 15180 have been implemented in **OpenBUGS** and a detailed list of differences between the two
 15181 **BUGS** versions, can be found at <http://www.openbugs.info/w/OpenVsWin>. We have
 15182 encountered convergence problems with simple scr models in this program. There is an
 15183 extensive help archive for both **WinBUGS** and **OpenBUGS** and you can subscribe to
 15184 a mailing list, where people pose and answer questions of how to use these programs at
 15185 <http://www.mrc-bsu.cam.ac.uk/bugs/overview/list.shtml>

15186 20.5.1 OpenBUGS through R

15187 Like **WinBUGS**, **OpenBUGS** can be used as a standalone application or through **R**.
 15188 There are several packages that allow **R** to interface with **OpenBUGS**, all of which can
 15189 be installed directly from a cran mirror:

15190 **R2WinBUGS**: One of the options in the `bugs()` call is `program`, which lets you specify either
 15191 **WinBUGS** or **OpenBUGS**. This is a convenient option because after having worked
 15192 through some of this book you will likely be familiar with the format of `bugs()` output
 15193 and other functions of the **R2WinBUGS** package.

15194 **R2openBUGS**: **R2openBUGS** (Sturtz et al., 2005) is very similar to, and actually based on,
 15195 **R2WinBUGS** and it is unclear to us what can be gained by using the former over the latter.
 15196 Arguments of the `bugs()` call differ slightly between the two packages and given that
 15197 **R2WinBUGS** allows for the use of both **OpenBUGS** and **WinBUGS** it is probably easiest
 15198 to stick with it.

15199 **BRugs**: **BRugs** (Thomas et al., 2006) can be installed from within **R** directly from a cran
 15200 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-
 15201 cess at ''<http://www.rni.helsinki.fi/openbugs/OpenBUGS/Docu/BRugs%20Manual.html>''.
 15202 **BRugs** has the convenient feature that all pieces of a **BUGS** analysis can be run
 15203 from within **R**, including checking the model syntax, something that requires opening the
 15204 **BUGS** GUI with other packages.

20.6 JAGS

15206 **JAGS** (Just Another Gibbs Sampler) (Plummer, 2003) runs scr models considerably faster
 15207 than **WinBUGS**, does not have the convergence problem with simple scr models we have

15208 encountered in **OpenBUGS** but similar to the latter program, is flexible and constantly
15209 updated. Writing a **JAGS** model is virtually identical to writing a **WinBUGS** model.
15210 However, some functions may have slightly different names and you can look up available
15211 functions and their use in the **JAGS** manual. One potential downside is that **JAGS** can
15212 be very particular when it comes to initial values. These may have to be set as close to
15213 truth as possible for the model to start. Although **JAGS** lets you run several parallel
15214 Markov chains, this characteristic interferes with the idea of using overdispersed initial
15215 values for the different chains. Also, we have found that when running models, sometimes
15216 **JAGS** crashes for unclear reasons, taking **R** down with it. Oftentimes, in order to make
15217 it run again you'll have to go through downloading and installing it again (remove the
15218 non-functioning version first).

15219 **JAGS** has a variety of functions that are not available in **WinBUGS**. For example,
15220 **JAGS** allows you to supply observed data for some deterministic functions of unobserved
15221 variables. In **BUGS** we cannot supply data to logical nodes. Another useful feature is
15222 that the adaptive phase of the model (the burn-in) is run separately from the sampling
15223 from the stationary Markov chains. This allows you to easily add more iterations to the
15224 adaptive phase if necessary without the need to start from 0. There are other, more
15225 subtle differences and there is an entire manual section on differences between **JAGS** and
15226 **OpenBUGS**.

15227 **JAGS** is available for download at '<http://sourceforge.net/projects/mcmc-jags/files/>', together with the R package **rjags** (Plummer, 2011), which allows running
15228 **JAGS** through **R**, user and installation manuals and examples. At this site **JAGS** is
15229 available for Windows and Mac; Linux binaries are distributed separately and you can
15230 find links to various sources here: '<http://mcmc-jags.sourceforge.net/>'. **JAGS**
15231 comes with a 32 bit and a 64 bit version and can be installed by double-clicking on the
15232 .exe file and following the instructions on the installer screen. For questions and prob-
15233 lems concerning **JAGS** there is a forum online at <http://sourceforge.net/projects/mcmc-jags/> forums/forum/610037.

15236 20.6.1 JAGS through R

15237 Unlike the two **BUGS** programs, **JAGS** does not have a GUI interface but a command
15238 line interface that can be used to run the program as a standalone application. **JAGS**
15239 will solely perform the MCMC simulation; analyzing and summarizing the output has to
15240 be done outside of **JAGS**. To run **JAGS** through **R** you have two options.

15241 **rjags**: As mentioned above, **rjags** (Plummer, 2011) can be found together with **JAGS**
15242 and was developed/is being maintained by the inventor of **JAGS**, which means it is
15243 guaranteed to stay up to date when/as **JAGS** changes. The package can be installed from
15244 a cran mirror and the help document can be accessed at '<http://cran.r-project.org/web/packages/rjags/rjags.pdf>'

15246 **R2jags**: Alternatively, the package **R2jags** (Su and Yajima, 2011) provides a means of
15247 accessing **JAGS** through **R**. We prefer **rjags** for the reason named above, as well as be-
15248 cause it stores data in a more memory-efficient way and has better **plot()** and **summary()**
15249 methods.

20.7 R

15250 At the time of the preparation of this list, **R** for Windows is at version 2.15.0, which can be
 15251 downloaded at <http://cran.r-project.org/bin/windows/base/> This site also contains help-
 15252 ful tips on how to install **R** in Windows Vista, how to update **R** packages etc. Installation
 15253 of **R** in Windows is straightforward: download the .exe file, double-click on it and follow
 15254 the instructions of the Windows installer. The later versions of **R** come with versions for
 15255 both 64 bit and 32 bit machines. The **R** site (''<http://mirrors.softliste.de/cran/>'')
 15256 has an extensive FAQ section Hornik (2011), which includes instructions on how to install
 15257 **R** on Unix and Mac computers.

20.7.1 R packages

15258 This section provides an alphabetical list of useful **R** packages. There is a large number
 15259 of **R** packages and by no means is this list intended to be complete in terms of what is
 15260 useful. Rather, we list packages that we are familiar with and that we employ at one point
 15261 or the other in this book. Unless explicitly stated otherwise, all packages can be installed
 15262 directly from within **R** trough a cran mirror.

15264 **adapt**: **adapt** (Genz et al., 2007) is a package for multidimensional numerical integration.
 15265 The package has been removed from the CRAN repository but can be obtained from
 15266 ''<http://cran.r-project.org/src/contrib/Archive/adapt/>''.

15267 **coda**: **coda** (Plummer et al., 2006) lets you summarize and perform diagnostics on mcmc
 15268 output. For a list and description of functions, see the manual at ''<http://cran.r-project.org/web/packages/coda/coda.pdf>''.

15270 **gdistance**: **gdistance** (van Etten, 2011) is a package for calculating distances and routes
 15271 on geographical grids and can be used to calculate least cost path surfaces. Manual at
 15272 ''<http://cran.r-project.org/web/packages/gdistance/gdistance.pdf>''.

15273 **igraph**: **igraph** (Csardi and Nepusz, 2006) provides routines for graphs and network anal-
 15274 ysis. Manual at ''<http://cran.r-project.org/web/packages/igraph/igraph.pdf>''.

15275 **inline**: **inline** (Sklyar et al., 2010) allows the user to define R functions with in-lined **C**,
 15276 **C++** or **Fortran** code. Manual at <http://cran.r-project.org/web/packages/inline/inline.pdf>.

15278 **maps**: **maptools** (?) is a library of maps. Manual at ''<http://cran.r-project.org/web/packages/maps/index.html>''.

15280 **maptools**: **maptools** (Lewin-Koh et al., 2011) provides a set of tools for reading and manip-
 15281 ulating spatial data, especially ESRI shapefiles. Manual at ''<http://cran.r-project.org/web/packages/maptools/maptools.pdf>''.

15283 **R2cuba**: **R2cuba** (Hahn et al., 2010) is another package for multidimensional integration.
 15284 Manual at ''<http://cran.r-project.org/web/packages/R2Cuba/R2Cuba.pdf>''.

15285 **raster**: **raster** (van Etten, 2012) provides functions for geographic analysis and modeling
15286 with raster data. Manual at '<http://cran.r-project.org/web/packages/raster/raster.pdf>'.

15288 **Rcpp**: **Rcpp** (Eddelbuettel and François, 2011) provides R functions as well as a C++ library
15289 which facilitate the integration of R and C++. Manual at <http://cran.r-project.org/web/packages/Rcpp/Rcpp.pdf>.

15291 **RcppArmadillo**: **RcppArmadillo** (François et al., 2011) is a templated C++ linear algebra
15292 library, integrating the **Armadillo** library and R. Manual at <http://cran.r-project.org/web/packages/RcppArmadillo/RcppArmadillo.pdf>.

15294 **reshape**: **reshape** (Wickham and Hadley, 2007) allows you to easily manipulate, summarize
15295 and reshape data. Manual at '<http://cran.r-project.org/web/packages/reshape/reshape.pdf>'.

15297 **rgeos**: **rgeos** (Bivand and Rundel, 2011) provides many useful functions for spatial operations
15298 such as intersecting or buffering spatial features. Manual at '<http://cran.r-project.org/web/packages/rgeos/rgeos.pdf>'.

15300 **SCRbayes**: (Russell et al., 2012)

15301 **secr**: **secr** (Efford et al., 2009a)

15302 **shapefiles**: **shapefiles** (Stabler, 2006) allows you to read and write ESRI shapefiles
15303 (i.e. shapefiles you would use in ArcGIS). Manual at '<http://cran.r-project.org/web/packages/shapefiles/shapefiles.pdf>'.

15305 **snow**, **snowfall**: **snow** (Tierney et al., 2011) and **snowfall** (Knaus, 2010) provide functionality
15306 for parallel computing. The latter is a more user-friendly wrapper around the former.
15307 Manuals at <http://cran.r-project.org/web/packages/snowfall/snowfall.pdf>
15308 and <http://cran.r-project.org/web/packages/snow/snow.pdf>.

15309 **sp**: **sp** (Pebesma and Bivand, 2011) is a package for plotting, selecting, subsetting etc.
15310 spatial data. **sp** and **spatstat** (see below) are complementary in many ways and data
15311 formats can be easily converted between the two packages. Manual at '<http://cran.r-project.org/web/packages/sp/sp.pdf>'.

15313 **SPACECAP**: (Gopalaswamy et al., 2012a)

15314 **spatstat**: **spatstat** (Baddeley and Turner, 2005) is an extensive package for analyzing
15315 spatial data. We use it, for example, to generate random points within a state space
15316 that cannot be described as a rectangle but consists of a (or several) arbitrary polygon(s).
15317 Manual at '<http://cran.r-project.org/web/packages/spatstat/spatstat.pdf>'.

15318 **unmarked**:

15319 #####

15320 #####

15321 **References**

15322 #####

BIBLIOGRAPHY

15323

15324

- 15325 Adriaensen, F., Chardon, J. P., De Blust, G., Swinnen, E., Villalba, S., Gulink, H.,
15326 and Matthysen, E. (2003), "The application of 'least-cost' modelling as a functional
15327 landscape model," *Landscape and urban planning*, 64, 233–247.
- 15328 Aitkin, M. (1991), "Posterior bayes factors," *Journal of the Royal Statistical Society. Series B (Methodological)*, 53, 111–142.
- 15329 Alho, J. M. (1990), "Logistic regression in capture-recapture models," *Biometrics*, 46,
15330 623–635.
- 15331 Alpízar-Jara, R. and Pollock, K. H. (1996), "A combination line transect and capture-
15332 recapture sampling model for multiple observers in aerial surveys," *Environmental and Ecological Statistics*, 3, 311–327.
- 15333 Amstrup, S. C., McDonald, T. L., and Manly, B. F. J. (2005), *Handbook of capture-
15334 recapture analysis*, Princeton Univ Pr.
- 15335 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.
- 15336 Arnason, A., Schwarz, C., and Gerrard, J. (1991), "Estimating closed population size and
15337 number of marked animals from sighting data," *Journal of Wildlife Management*, 55,
15338 716–730.
- 15339 Arnason, N. A. (1972), "Parameter estimates from mark-recapture experiments on two
15340 populations subject to migration and death," *Researches on Population Ecology*, 13,
15341 97–113.
- 15342 — (1973), "The estimation of population size, migration rates and survival in a stratified
15343 population," *Researches on Population Ecology*, 15, 1–8.
- 15344 Baddeley, A. and Turner, R. (2005), "Spatstat: an R package for analyzing spatial point
15345 patterns," *Journal of Statistical Software*, 12, 1–42, ISSN 1548-7660.
- 15346 Bales, S., Hellgren, E., Leslie Jr, D., and Hemphill Jr, J. (2005), "Dynamics of a recolonizing
15347 population of black bears in the Ouachita Mountains of Oklahoma," *Wildlife Society Bulletin*, 33, 1342–1351.
- 15348 Balme, G. A., Slotow, R., and Hunter, L. T. B. (2010), "Edge effects and the impact of non-
15349 protected areas in carnivore conservation: leopards in the Phinda–Mkuze Complex,
15350 South Africa," *Animal Conservation*, 13, 315–323.
- 15351 Bartmann, R. M., White, G. C., Carpenter, L. H., and Garrott, R. A. (1987), "Aerial
15352 Mark-Recapture Estimates of Confined Mule Deer in Pinyon-Juniper Woodland," *The Journal of Wildlife Management*, 51, 41–46.
- 15353 Bayne, E. and Hobson, K. (2002), "Annual survival of adult American Redstarts and
15354 Ovenbirds in the southern boreal forest," *The Wilson Bulletin*, 114, 358–367.

- 15361 Beier, P., Majka, D. R., and Spencer, W. D. (2008), "Forks in the road: choices in
15362 procedures for designing wildland linkages," *Conservation Biology*, 22, 836–851.
- 15363 Belant, J., Van Stappen, J., and Paetkau, D. (2005), "American black bear population
15364 size and genetic diversity at Apostle Islands National Lakeshore," *Ursus*, 16, 85–92.
- 15365 Berger, J., Liseo, B., and Wolpert, R. (1999), "Integrated likelihood methods for elimi-
15366 nating nuisance parameters," *Statistical Science*, 14, 1–28.
- 15367 Best, N. G., Ickstadt, K., and Wolpert, R. L. (2000), "Spatial Poisson Regression for
15368 Health and Exposure Data Measured at Disparate Resolutions," *Journal of the Amer-
15369 ican Statistical Association*, 95, 1076.
- 15370 Bibby, C. J. and Buckland, S. T. (1987), "Bias of bird census results due to detectability
15371 varying with habitat," *Acta Ecologica*, 8, 103–112.
- 15372 Bibby, C. J., Burgess, N. D., and Hill, D. A. (1992), "Bird census techniques: Academic
15373 Press," *London, UK*, 257.
- 15374 Bivand, R. and Rundel, C. (2011), *rgeos: Interface to Geometry Engine - Open Source
15375 (GEOS)*, r package version 0.1-8.
- 15376 Blair, W. (1940), "A study of prairie deer-mouse populations in southern Michigan,"
15377 *American Midland Naturalist*, 24, 273–305.
- 15378 Bolker, B. (2008), *Ecological models and data in R*, Princeton Univ Pr.
- 15379 Bondrup-Nielsen, S. (1983), "Density estimation as a function of live-trapping grid and
15380 home range size," *Canadian Journal of Zoology*, 61, 2361–2365.
- 15381 Borchers, D. L. (2012), "A non-technical overview of spatially explicit capture–recapture
15382 models," *Journal of Ornithology*, 152, 1–10.
- 15383 Borchers, D. L., Buckland, S. T., and Zucchini, W. (2002), *Estimating animal abundance:
15384 closed populations*, vol. 13, Springer Verlag.
- 15385 Borchers, D. L. and Efford, M. G. (2008), "Spatially explicit maximum likelihood methods
15386 for capture–recapture studies," *Biometrics*, 64, 377–385.
- 15387 Borchers, D. L., Zucchini, W., and Fewster, R. M. (1998), "Mark-recapture models for
15388 line transect surveys," *Biometrics*, 54, 1207–1220.
- 15389 Boulanger, J. and McLellan, B. (2001), "Closure violation in DNA-based mark-recapture
15390 estimation of grizzly bear populations," *Canadian Journal of Zoology*, 79, 642–651.
- 15391 Bowden, D. C. (1993), "A simple technique for estimating population size. Technical Re-
15392 port 93/12." Tech. rep., Department of Statistics, Colorado State University, Fort
15393 Collins, Colorado, USA.
- 15394 Brooks, S. P., Catchpole, E. A., and Morgan, B. J. T. (2000), "Bayesian Animal Survival
15395 Estimation," *Statistical Science*, 15, 357–376.
- 15396 Brownie, C., Hines, J. E., Nichols, J. D., Pollock, K. H., and Hestbeck, J. B. (1993),
15397 "Capture-recapture studies for multiple strata including non-Markovian transitions."
15398 *Biometrics*, 49, 1173–1187.
- 15399 Buckland, S., Anderson, D., Burnham, K., Laake, J., Borcher, D., and L, T. (2001), *In-
15400 troduction to distance sampling: estimating abundance of biological populations*, Oxford,
15401 UK: Oxford University Press.
- 15402 Buckland, S. T. (2004), *Advanced distance sampling*, Oxford University Press, USA.
- 15403 Burnham, K., Anderson, D., and Laake, J. (1980), "Estimation of density from line tran-
15404 sect sampling of biological populations," *Wildlife monographs*, 3–202.
- 15405 Burnham, K., Anderson, D., White, G., Brownie, C., and Pollock, K. (1987), "Design and
15406 analysis methods for fish survival experiments based on release-recapture," *American*

- 15407 *Fisheries Society Monograph 5. American Fisheries Society, Bethesda Maryland. 1987.*
15408 737.
- 15409 Burnham, K. P. and Anderson, D. R. (2002), *Model selection and multimodel inference: a practical information-theoretic approach*, Springer Verlag.
- 15410 Burnham, K. P. and Overton, W. S. (1978), "Estimation of the size of a closed population when capture probabilities vary among animals," *Biometrika*, 65, 625.
- 15411 Burt, W. (1943), "Territoriality and home range concepts as applied to mammals," *Journal of mammalogy*, 24, 346–352.
- 15412 Casella, G. and Berger, R. L. (2002), *Statistical inference*, Duxbury Press.
- 15413 Casella, G. and George, E. I. (1992), "Explaining the Gibbs sampler," *American Statistician*, 46, 167–174.
- 15414 Caswell, H. (1989), *Matrix population models*, Sinauer assoc. Sunderland.
- 15415 Chandler, R. B. and Royle, J. A. (In press), "Spatially-explicit models for inference about density in unmarked or partially marked populations," *Annals of Applied Statistics*.
- 15416 Chandler, R. B., Royle, J. A., and King, D. (2011), "Inference about density and temporary emigration in unmarked populations," *Ecology*, 92, 1429–1435.
- 15417 Clobert, J., Danchin, E., Dhondt, A., and Nichols, J. (2001), *Dispersal*, Oxford.
- 15418 Cochran, W. (2007), *Sampling techniques*, United States of America: John Wiley & Sons.
- 15419 Conde, D., Colchero, F., Zarza, H., Christensen, N., Sexton, J., Manterola, C., Chávez, C., Rivera, A., Azuara, D., and Ceballos, G. (2010), "Sex matters: Modeling male and female habitat differences for jaguar conservation," *Biological Conservation*, 143, 1980–1988.
- 15420 Converse, S., White, G., and Block, W. (2006a), "Small mammal responses to thinning and wildfire in ponderosa pine-dominated forests of the southwestern United States," *Journal of Wildlife Management*, 70, 1711–1722.
- 15421 Converse, S., White, G., Farris, K., and Zack, S. (2006b), "Small mammals and forest fuel reduction: national-scale responses to fire and fire surrogates," *Ecological Applications*, 16, 1717–1729.
- 15422 Converse, S. J. and Royle, J. A. (2012), "Dealing with incomplete and variable detectability in multi-year, multi-site monitoring of ecological populations," in *Design and Analysis of Long-term Ecological Monitoring Studies*, eds. Gitzen, R. R., Millspaugh, J. J., Cooper, A. B., and Licht, D. S., Cambridge University Press, pp. 426–442.
- 15423 Cooch, E. and White, G. (2006), "Program MARK: a gentle introduction," available online with the MARK programme, 7.
- 15424 Cormack, R. M. (1964), "Estimates of survival from the sighting of marked animals," *Biometrika*, 51, 429–438.
- 15425 Coull, B. A. and Agresti, A. (1999), "The Use of Mixed Logit Models to Reflect Heterogeneity in Capture-Recapture Studies," *Biometrics*, 55, 294–301.
- 15426 Cox, D. (1955), "Some statistical methods connected with series of events," *Journal of the Royal Statistical Society. Series B (Methodological)*, 17, 129–164.
- 15427 Cressie, N. (1991), *Statistics for spatial data*, Wiley Series in Probability and Mathematical Statistics.
- 15428 Csardi, G. and Nepusz, T. (2006), "The igraph software package for complex network research," *InterJournal, Complex Systems*, 1695.
- 15429 Cushman, S. A., Compton, B. W., and McGarigal, K. (2010), "Habitat fragmentation effects depend on complex interactions between population size and dispersal ability:

- 15453 Modeling influences of roads, agriculture and residential development across a range
15454 of life-history characteristics," in *Spatial complexity, informatics, and wildlife conser-*
15455 *vation*, eds. Cushman, S. A. and Huettmann, F., New York: Springer, chap. 20, pp.
15456 369–385.
- 15457 Cushman, S. A., McKelvey, K. S., Hayden, J., and Schwartz, M. K. (2006), "Gene flow in
15458 complex landscapes: testing multiple hypotheses with causal modeling," *The American*
15459 *Naturalist*, 168, 486–499.
- 15460 Dawson, D. K. and Efford, M. G. (2009), "Bird population density estimated from acoustic
15461 signals," *Journal of Applied Ecology*, 46, 1201–1209.
- 15462 DeGraaf, R. M. and Yamasaki, M. (2001), *New England wildlife: habitat, natural history,*
15463 *and distribution*, University Press of New England.
- 15464 DeSante, D. F., Burton, K. M., Saracco, J. F., and Walker, B. L. (1995), "Productiv-
15465 ity indices and survival rate estimates from MAPS, a continent-wide programme of
15466 constant-effort mist-netting in North America," *Journal of Applied Statistics*, 22, 935–
15467 948.
- 15468 Dice, L. R. (1938), "Some census methods for mammals," *Journal of Wildlife Management*,
15469 2, 119–130.
- 15470 — (1941), "Methods for estimating populations of mammals," *Journal of Wildlife Man-*
15471 *agement*, 5, 398–407.
- 15472 Diggle, P. J. (2003), *Statistical analysis of spatial point processes*, London: Arnold, 2nd
15473 ed.
- 15474 Dijkstra, E. W. (1959), "A note on two problems in connexion with graphs," *Numerische*
15475 *mathematik*, 1, 269–271.
- 15476 Dillon, A. and Kelly, M. (2007), "Ocelot *Leopardus pardalis* in Belize: the impact of trap
15477 spacing and distance moved on density estimates," *Oryx*, 41, 469–477.
- 15478 Dixon, P. (2002), "Bootstrap resampling," *Encyclopedia of environmetrics*.
- 15479 Dorazio, R. M. (2007), "On the choice of statistical models for estimating occurrence and
15480 extinction from animal surveys," *Ecology*, 88, 2773–2782.
- 15481 Dorazio, R. M. and Royle, J. A. (2003), "Mixture models for estimating the size of a closed
15482 population when capture rates vary among individuals," *Biometrics*, 59, 351–364.
- 15483 Durban, J. and Elston, D. (2005), "Mark-recapture with occasion and individual effects:
15484 abundance estimation through Bayesian model selection in a fixed dimensional parame-
15485 ter space," *Journal of agricultural, biological, and environmental statistics*, 10, 291–305.
- 15486 Eddelbuettel, D. and François, R. (2011), "Rcpp: Seamless R and C++ Integration,"
15487 *Journal of Statistical Software*, 40, 1–18.
- 15488 Efford, M. (2011), *secr: Spatially explicit capture-recapture models*, R package version
15489 2.3.1.
- 15490 Efford, M. G. (2004), "Density estimation in live-trapping studies," *Oikos*, 106, 598–610.
- 15491 Efford, M. G., Borchers, D. L., and Byrom, A. E. (2009a), "Density estimation by spatially
15492 explicit capture–recapture: likelihood-based methods," *Modeling demographic processes*
15493 *in marked populations*, 255–269.
- 15494 Efford, M. G. and Dawson, D. K. (2009), "Effect of distance-related heterogeneity on
15495 population size estimates from point counts," *The Auk*, 126, 100–111.
- 15496 — (2010), "SECR for acoustic data," .
- 15497 Efford, M. G., Dawson, D. K., and Borchers, D. L. (2009b), "Population density estimated
15498 from locations of individuals on a passive detector array," *Ecology*, 90, 2676–2682.

- 15499 Efford, M. G., Dawson, D. K., and Robbins, C. S. (2004), "DENSITY: software for
15500 analysing capture-recapture data from passive detector arrays," *Animal Biodiversity
15501 and Conservation*, 27, 217–228.
- 15502 Efford, M. G. and Fewster, R. M. (2012), "Estimating population size by spatially explicit
15503 capture–recapture," *Oikos*.
- 15504 Efford, M. G., Warburton, B., Coleman, M. C., and Barker, R. J. (2005), "A field test of
15505 two methods for density estimation," *Wildlife Society Bulletin*, 33, 731–738.
- 15506 Epps, C. W., Palsbøll, P. J., Wehausen, J. D., Roderick, G. K., Ramey II, R. R., and
15507 McCullough, D. R. (2005), "Highways block gene flow and cause a rapid decline in
15508 genetic diversity of desert bighorn sheep," *Ecology Letters*, 8, 1029–1038.
- 15509 Epps, C. W., Wehausen, J. D., Bleich, V. C., Torres, S. G., and Brashares, J. S. (2007),
15510 "Optimizing dispersal and corridor models using landscape genetics," *Journal of Applied
15511 Ecology*, 44, 714–724.
- 15512 F Dormann, C., M McPherson, J., B Araújo, M., Bivand, R., Bolliger, J., Carl, G.,
15513 G Davies, R., Hirzel, A., Jetz, W., Daniel Kissling, W., et al. (2007), "Methods to
15514 account for spatial autocorrelation in the analysis of species distributional data: a
15515 review," *Ecography*, 30, 609–628.
- 15516 Farnsworth, G. L., Pollock, K. H., Nichols, J. D., Simons, T. R., E., H. J., and R.,
15517 S. J. (2002), "A removal model for estimating detection probabilities from point-count
15518 surveys," *The Auk*, 119, 414–425.
- 15519 Fecske, D., Barry, R., Precht, F., Quigley, H., Bittner, S., and Webster, T. (2002), "Habitat
15520 use by female black bears in western Maryland," *Southeastern Naturalist*, 1, 77–92.
- 15521 Fieberg, J. (2007), "Utilization distribution estimation using weighted kernel density es-
15522 timators," *The Journal of wildlife management*, 71, 1669–1675.
- 15523 Fieberg, J. and Kochanny, C. (2005), "Quantifying home-range overlap: the importance
15524 of the utilization distribution," *Journal of Wildlife Management*, 69, 1346–1359.
- 15525 Fienberg, S. E., Johnson, M. S., and Junker, B. W. (1999), "Classical multilevel and
15526 Bayesian approaches to population size estimation using multiple lists," *Journal of the
15527 Royal Statistical Society of London A*, 163, 383–405.
- 15528 Fiske, I. J. and Chandler, R. B. (2011), "unmarked: An R package for fitting hierarchical
15529 models of wildlife occurrence and abundance," *Journal Of Statistical Software*, 43, 1–23.
- 15530 Forester, J. D., Ives, A. R., Turner, M. G., Anderson, D. P., Fortin, D., Beyer, H. L., Smith,
15531 D. W., and Boyce, M. S. (2007), "State-space models link elk movement patterns to
15532 landscape characteristics in Yellowstone National Park," *Ecological Monographs*, 77,
15533 285–299.
- 15534 Fortin, D., Beyer, H. L., Boyce, M. S., Smith, D. W., Duchesne, T., and Mao, J. S. (2005),
15535 "Wolves influence elk movements: behavior shapes a trophic cascade in Yellowstone
15536 National Park," *Ecology*, 86, 1320–1330.
- 15537 Foster, R. J. and Harmsen, B. J. (2012), "A critique of density estimation from camera-
15538 trap data," *The Journal of Wildlife Management*, 76, 224–236.
- 15539 François, R., Eddelbuettel, D., and Bates, D. (2011), *RcppArmadillo: Rcpp integration
15540 for Armadillo templated linear algebra library*, r package version 0.2.25.
- 15541 Frary, V., Duchamp, J., Maehr, D., and Larkin, J. (2011), "Density and distribution of
15542 a colonizing front of the American black bear *Ursus americanus*," *Wildlife Biology*, 17,
15543 404–416.
- 15544 Fujiwara, M., Anderson, K., Neubert, M., and Caswell, H. (2006), "On the estimation of

- 15545 dispersal kernels from individual mark-recapture data." *Environmental and Ecological
15546 Statistics*, 13, 183–197.
- 15547 García-Alaníz, N., Naranjo, E. J., and Mallory, F. F. (2010), "Hair-snares: A non-invasive
15548 method for monitoring felid populations in the Selva Lacandona, Mexico," *Tropical
15549 Conservation Science*, 3, 403–411.
- 15550 Gardner, B., Reppucci, J., Lucherini, M., and Royle, J. (2010a), "Spatially explicit in-
15551 ference for open populations: estimating demographic parameters from camera-trap
15552 studies," *Ecology*, 91, 3376–3383.
- 15553 Gardner, B., Royle, J. A., and Wegan, M. T. (2009), "Hierarchical models for estimating
15554 density from DNA mark-recapture studies," *Ecology*, 90, 1106–1115.
- 15555 Gardner, B., Royle, J. A., Wegan, M. T., Rainbolt, R. E., and Curtis, P. D. (2010b),
15556 "Estimating black bear density using DNA data from hair snares," *The Journal of
15557 Wildlife Management*, 74, 318–325.
- 15558 Garshelis, D. L. and Hristienko, H. (2006), "State and provincial estimates of American
15559 black bear numbers versus assessments of population trend," *Ursus*, 17, 1–7.
- 15560 Gelfand, A. and Smith, A. (1990), "Sampling-based approaches to calculating marginal
15561 densities," *Journal of the American statistical association*, 85, 398–409.
- 15562 Gelman, A. (2006), "Prior distributions for variance parameters in hierarchical models,"
15563 *Bayesian analysis*, 1, 515–533.
- 15564 Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004), *Bayesian data analysis,
15565 second edition.*, Bocan Raton, Florida, USA: CRC/Chapman & Hall.
- 15566 Gelman, A., Meng, X. L., and Stern, H. (1996), "Posterior predictive assessment of model
15567 fitness via realized discrepancies," *Statistica Sinica*, 6, 733–759.
- 15568 Gelman, A. and Rubin, D. B. (1992), "Inference from iterative simulation using multiple
15569 sequences," *Statistical Science*, 7, 457–511.
- 15570 Geman, S. and Geman, D. (1984), "Stochastic relaxation, Gibbs distributions, and the
15571 Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine
15572 Intelligence*, PAMI-6, 721–741.
- 15573 Genz, A. S., Meyer, M. R., Lumley, T., and Maechler, M. (2007), "The adapt Package. R
15574 package version 1.0-4."
- 15575 Gerlach, G. and Musolf, K. (2000), "Fragmentation of landscape as a cause for genetic
15576 subdivision in bank voles," *Conservation Biology*, 14, 1066–1074.
- 15577 Gilks, W. and Wild, P. (1992), "Adaptive rejection sampling for Gibbs sampling," *Applied
15578 Statistics*, 41, 337–348.
- 15579 Gilks, W. R., Thomas, A., and Spiegelhalter, D. J. (1994), "A Language and Program for
15580 Complex Bayesian Modelling," *Journal of the Royal Statistical Society. Series D (The
15581 Statistician)*, 43, 169–177, ArticleType: primary_article / Issue Title: Special Issue:
15582 Conference on Practical Bayesian Statistics, 1992 (3) / Full publication date: 1994 /
15583 Copyright 1994 Royal Statistical Society.
- 15584 Gimenez, O., Rossi, V., Choquet, R., Dehais, C., Doris, B., Varella, H., Vila, J. P., and
15585 Pradel, R. (2007), "State-space modelling of data on marked individuals," *Ecological
15586 Modelling*, 206, 431–438.
- 15587 Gopalaswamy, A. M. (2012), "Capture-recapture models, spatially explicit," in *Encyclo-
15588 pedia of Environmentrics Second Edition*, eds. El-Shaarawi, A. H. and Piegorsch, W.,
15589 John Wiley and Sons Ltd, Chichester, UK.
- 15590 Gopalaswamy, A. M., Royle, A. J., Hines, J., Singh, P., Jathanna, D., Kumar, N. S., and

- 15591 Karanth, K. U. (2012a), "Program SPACECAP: software for estimating animal density
15592 using spatially explicit capturerecapture models," *Methods in Ecology and Evolution*,
15593 online early, r package version 1.0.4.
- 15594 Gopalaswamy, A. M., Royle, J. A., Delampady, M., Nichols, J. D., Karanth, K. U., and
15595 Macdonald, D. W. (2012b), "Density estimation in tiger populations: combining infor-
15596 mation for strong inference," *Ecology*, 93, 1741–1751.
- 15597 Gruber, D. and White, M. (1983), "Black bear food habits in Yosemite National Park,"
15598 *Bears: Their Biology and Management*, 5, 1–10.
- 15599 Greig-Smith, P. (1964), *Quantitative plant ecology*, Butterworths (Washington).
- 15600 Grimm, V., Revilla, E., Berger, U., Jeltsch, F., Mooij, W., Railsback, S., Thulke, H.-
15601 H., Weiner, J., Wiegand, T., and DeAngelis, D. (2005), "Pattern-oriented modeling of
15602 agent-based complex systems: Lessons from ecology." *Science*, 310, 987–991.
- 15603 Hahn, T., Bouvier, A., and Kiêu, K. (2010), *R2Cuba: Multidimensional Numerical Inte-
15604 gration*, r package version 1.0-6.
- 15605 Hall, R. J., Henry, P. F. P., and Bunck, C. M. (1999), "Fifty-year trends in a box turtle
15606 population in Maryland," *Biological Conservation*, 88, 165–172.
- 15607 Hanski, I. A. (1999), *Metapopulation Ecology*, Oxford Univiversity Press.
- 15608 Hastings, W. (1970), "Monte Carlo sampling methods using Markov chains and their
15609 applications," *Biometrika*, 57, 97–109.
- 15610 Hawkins, C. E. and Racey, P. A. (2005), "Low population density of a tropical forest
15611 carnivore, Cryptoprocta ferox: implications for protected area management," *Oryx*, 39,
15612 35–43.
- 15613 Hayes, R. J. and Buckland, S. T. (1983), "Radial distance models for the line transect
15614 method." *Biometrics*, 39, 29–42.
- 15615 Hayne, D. (1950), "Apparent home range of Microtus in relation to distance between
15616 traps," *Journal of mammalogy*, 31, 26–39.
- 15617 Hayne, D. W. (1949), "An Examination of the Strip Census Method for Estimating Animal
15618 Populations," *The Journal of Wildlife Management*, 13, 145–157.
- 15619 Hedley, S. L., Buckland, S. T., and Borchers, D. L. (1999), "Spatial modelling from line
15620 transect data," *Journal of Cetacean Research and Management*, 1, 255–264.
- 15621 Hestbeck, J. B. and Malecki, R. A. (1989), "Mark-resight estimate of Canada Goose
15622 midwinter numbers," *Journal of Wildlife Management*, 53, 749–752.
- 15623 Higdon, D. (1998), "A process-convolution approach to modelling temperatures in the
15624 North Atlantic Ocean," *Environmental and Ecological Statistics*, 5, 173–190.
- 15625 Hobbs, N. T. (2011), "An Ecological Modeler's Primer on JAGS," .
- 15626 Holdenried, R. (1940), "A population study of the long-eared chipmunk (*Eutamias quadri-
15627 maculatus*) in the central Sierra Nevada," *Journal of Mammalogy*, 21, 405–411.
- 15628 Hooten, M., Johnson, D., Hanks, E., and Lowry, J. (2010), "Agent-based inference for
15629 animal movement and selection," *Journal of agricultural, biological, and environmental
15630 statistics*, 15, 523–538.
- 15631 Hooten, M. and Wikle, C. (2010), "Statistical agent-based models for discrete spatio-
15632 temporal systems." *Journal of the American Statistical Association*, 105, 236–248.
- 15633 Hornik, K. (2011), "The R FAQ," ISBN 3-900051-08-9.
- 15634 Huggins, R. M. (1989), "On the statistical analysis of capture experiments," *Biometrika*,
15635 76, 133.
- 15636 Hurlbert, S. H. (1984), "Pseudoreplication and the design of ecological field experiments,"

- 15637 *Ecological monographs*, 54, 187–211.
- 15638 Illian, J., Penttinen, A., Stoyan, H., and Stoyan, D. (2008), *Statistical analysis and modelling of spatial point patterns*, Wiley.
- 15640 Ivan, J. (2012), “Density, demography, and seasonal movements of snowshoe hares in central Colorado,” Ph.D. thesis, Colorado State University.
- 15641 Ivan, J., White, G., and Shenk, T. (2013a), “Using auxiliary telemetry information to estimate animal density from capture-recapture data,” *Ecology*.
- 15644 — (2013b), “Using simulation to compare methods for estimating density from capture-recapture data,” *Ecology*.
- 15646 Jackson, R., Roe, J., Wangchuk, R., and Hunter, D. (2006), “Estimating Snow Leopard Population Abundance Using Photography and Capture-Recapture Techniques,” *Wildlife Society Bulletin*, 34, 772–781.
- 15649 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.
- 15651 Jhala, Y. V., Qureshi, Q., Gopal, R., and R., S. P. (2011), “Status of tigers, co-predators and prey in India.” Tech. rep., National Tiger Conservation Authority, Govt. of India, New Delhi and Wildlife Institute of India, Dehradun.
- 15654 Johnson (2010), “A Model-Based Approach for Making Ecological Inference from Distance Sampling Data,” *Biometrics*, 66, 310318.
- 15656 Johnson, D. (1980), “The comparison of usage and availability measurements for evaluating resource preference,” *Ecology*, 61, 65–71.
- 15658 Johnson, D. H. (1999), “The insignificance of statistical significance testing,” *The journal of wildlife management*, 63, 763–772.
- 15660 Johnson, D. S., Thomas, D. L., Ver Hoef, J. M., and Christ, A. (2008), “A general framework for the analysis of animal resource selection from telemetry data,” *Biometrics*, 64, 968–976.
- 15663 Jolly, G. M. (1965), “Explicit estimates from capture-recapture data with both death and dilution-stochastic model,” *Biometrika*, 52, 225–247.
- 15665 Jonsen, I. D., Flemming, J. M., and Myers, R. A. (2005), “Robust state-space modeling of animal movement data,” *Ecology*, 86, 2874–2880.
- 15667 Kadane, J. B. and Lazar, N. A. (2004), “Methods and criteria for model selection,” *Journal of the American Statistical Association*, 99, 279–290.
- 15669 Karanth, K. U. (1995), “Estimating tiger *Panthera tigris* populations from camera-trap data using capture–recapture models,” *Biological Conservation*, 71, 333–338.
- 15671 Karanth, K. U. and Nichols, J. D. (1998), “Estimation of Tiger Densities in India Using Photographic Captures and Recaptures,” *Ecology*, 79, 2852–2862.
- 15673 — (2000), “Ecological status and conservation of tigers in India,” *WCS, US Fish and Wildlife Service. Centre for Wildlife Studies. Bangalore, India*, 123.
- 15675 — (2002), *Monitoring tigers and their prey: a manual for researchers, managers and conservationists in Tropical Asia*, Bangalore, India: Centre for Wildlife Studies.
- 15677 Kass, R. and Wasserman, L. (1996), “The selection of prior distributions by formal rules,” *Journal of the American Statistical Association*, 91, 1343–1370.
- 15679 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.
- 15681 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 Data in the Presence of Individual and Temporal Covariates and Model Uncertainty," *Biometrics*, 64, 1187–1195.
- Knaus, J. (2010), *snowfall: Easier cluster computing (based on snow)*., r package version 1.84.
- Koehler, G. M. and Pierce, D. J. (2003), "Black Bear Home-Range Sizes in Washington: Climatic, Vegetative, and Social Influences," *Journal of Mammalogy*, 84, 81–91, Article-Type: research-article / Full publication date: Feb., 2003 / Copyright 2003 American Society of Mammalogists.
- Kohn, M., York, E., Kamradt, D., Haught, G., Sauvajot, R., and Wayne, R. (1999), "Estimating population size by genotyping faeces," *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 266, 657–663.
- Krebs, C. J. (1999), *Ecological methodology*, Menlo Park, CA: Benjamin/Cummings.
- Kucera and Barrett (2011), "missing," missing, missing.
- Kuo, L. and Mallick, B. (1998), "Variable selection for regression models," *Sankhy: The Indian Journal of Statistics, Series B*, 60, 65–81.
- Laird, N. M. and Ware, J. H. (1982), "Random-effects models for longitudinal data," *Biometrics*, 38, 963–974.
- Langtimm, C. A., Dorazio, R. M., Stith, B. M., and Doyle, T. J. (2011), "New aerial survey and hierarchical model to estimate manatee abundance," *The Journal of Wildlife Management*, 75, 399–412.
- Le Cam, L. (1990), "Maximum likelihood: an introduction," *International Statistical Review/Revue Internationale de Statistique*, 58, 153–171.
- Lebreton, J. D., Burnham, K. P., Clobert, J., and Anderson, D. R. (1992), "Modeling Survival and Testing Biological Hypotheses Using Marked Animals: A Unified Approach

- 15729 with Case Studies," *Ecological Monographs*, 62, 67–118.
- 15730 Lele, S. R. and Keim, J. L. (2006), "Weighted distributions and estimation of resource
15731 selection probability functions," *Ecology*, 87, 3021–3028.
- 15732 Lele, S. R., Moreno, M., and Bayne, E. (2012), "Dealing with detection error in site
15733 occupancy surveys: What can we do with a single survey?" *Journal of Plant Ecology*.
- 15734 Lewin-Koh, N. J., Bivand, R., contributions by Edzer J. Pebesma, Archer, E., Baddeley,
15735 A., Bibiko, H.-J., Dray, S., Forrest, D., Friendly, M., Giraudeau, P., Golicher, D., Rubio,
15736 V. G., Hausmann, P., Hufthammer, K. O., Jagger, T., Luque, S. P., MacQueen, D.,
15737 Niccolai, A., Short, T., Stabler, B., and Turner, R. (2011), *maptools: Tools for reading
15738 and handling spatial objects*, r package version 0.8-10.
- 15739 Link, W. (2003), "Nonidentifiability of Population Size from Capture-Recapture Data
15740 with Heterogeneous Detection Probabilities," *Biometrics*, 59, 1123–1130.
- 15741 Link, W. A. and Barker, R. J. (1994), "Density Estimation Using the Trapping Web
15742 Design: A Geometric Analysis," *Biometrics*, 50, 733–745.
- 15743 — (2006), "Model weights and the foundations of multimodel inference," *Ecology*, 87,
15744 2626–2635.
- 15745 — (2010), *Bayesian Inference: With Ecological Applications*, London, UK: Academic
15746 Press.
- 15747 Link, W. A. and Eaton, M. J. (2011), "On thinning of chains in MCMC," *Methods in
15748 Ecology and Evolution*, 3, 112–115.
- 15749 Link, W. A., Yoshizaki, J., Bailey, L. L., and Pollock, K. H. (2010), "Uncovering a latent
15750 multinomial: analysis of markrecapture data with misidentification," *Biometrics*, 66,
15751 178–185.
- 15752 Liu and Wu (1999), "Parameter expansion for data augmentation," *J Am Stat Assoc*, 94,
15753 1264–1274.
- 15754 Lukacs, P. M. and Burnham, K. P. (2005), "Estimating population size from DNA-based
15755 closed capture-recapture data incorporating genotyping error," *Journal of Wildlife
15756 Management*, 69, 396–403.
- 15757 Lunn, D., Spiegelhalter, D., Thomas, A., and Best, N. (2009), "The BUGS project: Evo-
15758 lution, critique, and future directions," *Statistics in Medicine*, 28, 3049–3067.
- 15759 Lyons, A., Gaines, W., and Servheen, C. (2003), "Black bear resource selection in the
15760 northeast Cascades, Washington," *Biological Conservation*, 113, 55–62.
- 15761 Mace, R., Minta, S., Manley, T., and Aune, K. (1994), "Estimating grizzly bear population
15762 size using camera sightings," *Wildlife Society Bulletin*, 22, 74–83.
- 15763 MacEachern, S. N. and Berliner, L. M. (1994), "Subsampling the Gibbs sampler," *Amer-
15764 ican Statistician*, 48, 188–190.
- 15765 MacKay, P., Smith, D. A., Long, R. A., Parker, M., Long, R. A., MacKay, P., Zielinski,
15766 W. J., and Ray, J. C. (2008), "Scat detection dogs," *Noninvasive survey methods for
15767 carnivores*, 183–222.
- 15768 MacKenzie, D. I., Nichols, J. D., Lachman, G. B., Droege, S., Royle, J. A., and Langtimm,
15769 C. A. (2002), "Estimating site occupancy rates when detection probabilities are less than
15770 one," *Ecology*, 83, 2248–2255.
- 15771 MacKenzie, D. I., Nichols, J. D., Royle, J. A., Pollock, K. H., Bailey, L. L., and Hines,
15772 J. E. (2006), *Occupancy estimation and modeling: inferring patterns and dynamics of
15773 species occurrence*, Academic Press.
- 15774 Maffei, L. and Noss, A. J. (2008), "How small is too small? Camera trap survey areas and

- density estimates for ocelots in the Bolivian Chaco," *Biotropica*, 40, 71–75.
- Magoun, A. J., Long, C. D., Schwartz, M. K., Pilgrim, K. L., Lowell, R. E., and Valkenburg, P. (2011), "Integrating motion-detection cameras and hair snags for wolverine identification," *The Journal of Wildlife Management*, 75, 731–739.
- Manel, S., Schwartz, M. K., Luikart, G., and Taberlet, P. (2003), "Landscape genetics: combining landscape ecology and population genetics," *Trends in Ecology & Evolution*, 18, 189–197.
- Manly, B., McDonald, L., Thomas, D., McDonald, T., and Erickson, W. (2002), *Resource selection by animals: statistical design and analysis for field studies*, Springer, 2nd ed.
- Marques, T., Thomas, L., Ward, J., DiMarzio, N., and Tyack, P. (2009), "Estimating cetacean population density using fixed passive acoustic sensors: An example with Blainvilles beaked whales," *The Journal of the Acoustical Society of America*, 125, 1982.
- Marques, T. A., Thomas, L., and Royle, J. A. (2011), "A hierarchical model for spatial capture-recapture data: Comment," *Ecology*, 92, 526–528.
- Mayor, S., Schneider, D., Schaefer, J., and Mahoney, S. (2009), "Habitat selection at multiple scales," *Ecoscience*, 16, 238–247.
- McCarthy, M. A. (2007), *Bayesian Methods for Ecology*, Cambridge: Cambridge University Press.
- McClintock, B. and Hoeting, J. (2010), "Bayesian analysis of abundance for binomial sighting data with unknown number of marked individuals," *Environmental and Ecological Statistics*, 17, 317–332.
- McClintock, B., King, Thomas, Matthiopoulos, McConnell, and Morales (2012), "A general discrete-time modeling framework for animal movement using multi-state random walks," *Ecological Monographs*, 82, 335–349.
- McClintock, B. and White, G. (2012), "From NOREMARK to MARK: software for estimating demographic parameters using mark-resight methodology," *Journal of Ornithology*, 152, 641–650.
- McClintock, B., White, G., Antolin, M., and Tripp, D. (2009a), "Estimating abundance using mark-resight when sampling is with replacement or the number of marked individuals is unknown," *Biometrics*, 65, 237–246.
- McClintock, B., White, G., and Burnham, K. (2006), "A robust design mark-resight abundance estimator allowing heterogeneity in resighting probabilities," *Journal of agricultural, biological, and environmental statistics*, 11, 231–248.
- McClintock, B. T., White, G. C., Burnham, K. P., and Pryde, M. A. (2009b), "A generalized mixed effects model of abundance for mark-resight data when sampling is without replacement," in *Modeling demographic processes in marked populations*, ed. Thomson, D., New York: Springer, pp. 271–289.
- McCullagh, P. and Nelder, J. A. (1989), *Generalized linear models*, Chapman & Hall/CRC.
- McRae, B. H. and Beier, P. (2007), "Circuit theory predicts gene flow in plant and animal populations," *Proceedings of the National Academy of Sciences*, 104, 19885.
- McRae, B. H., Dickson, B. G., Keitt, T. H., and Shah, V. B. (2008), "Using circuit theory to model connectivity in ecology, evolution, and conservation," *Ecology*, 89, 2712–2724.
- Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., Teller, E., et al. (1953), "Equation of state calculations by fast computing machines," *The journal of chemical physics*, 21, 1087–1092.

- 15821 Metropolis, N. and Ulam, S. (1949), "The Monte Carlo method," *Journal of the American
15822 Statistical Association*, 44, 335–341.
- 15823 Millar, R. B. (2009), "Comparison of hierarchical Bayesian models for overdispersed count
15824 data using DIC and Bayes' Factors," *Biometrics*, 65, 962–969.
- 15825 Mills, L. S., Citta, J. J., Lair, K. P., Schwartz, M. K., and Tallmon, D. A. (2000), "Es-
15826 timating animal abundance using noninvasive DNA sampling: promise and pitfalls,"
15827 *Ecological Applications*, 10, 283–294.
- 15828 Minta, S. and Mangel, M. (1989), "A Simple Population Estimate Based on Simulation
15829 for Capture-Recapture and Capture-Resight Data," *Ecology*, 70, 1738–1751.
- 15830 Møller, J. and Waagepetersen, R. P. (2004), *Statistical inference and simulation for spatial
15831 point processes*, Chapman & Hall/CRC.
- 15832 Mohr, C. (1947), "Table of equivalent populations of North American small mammals,"
15833 *American midland naturalist*, 37, 223–249.
- 15834 Moldenhauer, R. R. and Regelski, D. J. (1996), "Northern Parula (*Parula americana*),"
15835 in *The Birds of North America Online*, ed. Poole, A., Ithaca, NY: Cornell Lab of
15836 Ornithology.
- 15837 Molinari-Jobin, A., Kéry, M., Marboutin, E., Marucco, F., Zimmermann, F., Molinari, P.,
15838 Frick, H., Wölfl, S., Bled, F., Breitenmoser-Würsten, C., Fuxjäger, C., Huber, T., I.,
15839 K., Kos, I., Manfred Wölfl, M., and Breitenmoser, U. (2013), "Mapping range dynamics
15840 from opportunistic data: spatio-temporal distribution modeling of lynx *Lynx lynx* L.
15841 in the Alps," *Biological Conservation*, xx, xxxx–xxxx.
- 15842 Mollet, P., Kéry, M., Gardner, B., Pasinelli, G., and A, R. J. (2012), "Population size
15843 estimation for capercaille (*Tetrao urogallus* L.) using DNA-based individual recognition
15844 and spatial capture-recapture models," missing, missing, missing.
- 15845 Morrison, M. L., Strickland, M. D., Block, W. M., Collier, B. A., and Peterson, M. J.
15846 (2008), *Wildlife Study Design*, Springer.
- 15847 Muller, W. G. (2007), *Collecting spatial data: Optimum design of experiments for random
15848 fields*, Springer.
- 15849 Neal, A., White, G., Gill, R., Reed, D., and Olterman, J. (1993), "Evaluation of mark-
15850 resight model assumptions for estimating mountain sheep numbers," *Journal of Wildlife
15851 Management*, 57, 436–450.
- 15852 Neal, A. K. (1990), "Evaluation of mark-resight population estimates using simulations
15853 and field data from mountain sheep." M.S. thesis, Colorado State University, Fort
15854 Collins, Colorado, USA.
- 15855 Neal, R. (2003), "Slice sampling," *Annals of Statistics*, 31, 705–741.
- 15856 Nelder, J. A. and Wedderburn, R. W. M. (1972), "Generalized linear models," *Journal of
15857 the Royal Statistical Society. Series A (General)*, 135, 370–384.
- 15858 Nichols, J. D., Hines, J. E., Sauer, J. R., Fallon, F. W., Fallon, J. E., and Heglund,
15859 P. J. (2000), "A double-observer approach for estimating detection probability and
15860 abundance from point counts," *The Auk*, 117, 393–408.
- 15861 Nichols, J. D. and Karanth, K. U. (2002), "Statistical concepts: assessing spatial distri-
15862 butions," in *Monitoring tigers and their prey: a manual for researchers, managers and
15863 conservationists in Tropical Asia*, eds. Karanth, K. U. and Nichols, J. D., Bangalore,
15864 India: Centre for Wildlife Studies, pp. 29–38.
- 15865 Niemi, A. and Fernández, C. (2010), "Bayesian Spatial Point Process Modeling of Line
15866 Transect Data," *Journal of Agricultural, Biological, and Environmental Statistics*, 15,

- 15867 327–345.
- 15868 Norris, J. L. and Pollock, K. H. (1996), “Nonparametric MLE under two closed capture–
15869 recapture models with heterogeneity,” *Biometrics*, 52, 639–649.
- 15870 O’Brien, T. (2011), “Abundance, density and relative abundance: A conceptual frame-
15871 work,” in *Camera traps in animal ecology: methods and analyses*, eds. O’Connel, A.
15872 F. J., Nichols, J. D., and Karanth, U., Tokyo, Japan: Springer Verlag, pp. 71–96.
- 15873 O’Connell, A. F., Nichols, J. D., and Karanth, U. K. (2010), *Camera traps in animal
15874 ecology: Methods and analyses*, Springer.
- 15875 O’Hara, R. and Sillanpää, M. (2009), “A review of Bayesian variable selection methods:
15876 what, how and which,” *Bayesian Analysis*, 4, 85–118.
- 15877 Otis, D. L., Burnham, K. P., White, G. C., and Anderson, D. R. (1978), “Statistical
15878 inference from capture data on closed animal populations,” *Wildlife monographs*, 3–
15879 135.
- 15880 Ovaskainen, O. (2004), “Habitat-specific movement parameters estimated using mark–
15881 recapture data and a diffusion model,” *Ecology*, 85, 242–257.
- 15882 Ovaskainen, O., Rekola, H., Meyke, E., and Arjas, E. (2008), “Bayesian methods for
15883 analyzing movements in heterogeneous landscapes from mark-recapture data,” *Ecology*,
15884 89, 542–554.
- 15885 Parmenter, R. R. and MacMahon, J. A. (1989), “Animal Density Estiamtion Using a
15886 Trapping Web Design: Field Validation Experiments,” *Ecology*, 70, 169–179.
- 15887 Parmenter, R. R., Yates, T. L., Anderson, D. R., Burnham, K. P., Dunnum, J. L., Franklin,
15888 A. B., Friggens, M. T., Lubow, B. C., Miller, M., Olson, G. S., and Others (2003),
15889 “Small-mammal density estimation: A field comparison of grid-based vs. web-based
15890 density estimators,” *Ecological Monographs*, 73, 1–26.
- 15891 Patterson, T., Thomas, L., Wilcox, C., Ovaskainen, O., and Matthiopoulos, J. (2008),
15892 “State-space models of individual animal movement,” *Trends in ecology & evolution*,
15893 23, 87–94.
- 15894 Paviolo, A., De Angelo, C., Di Blanco, Y., and Di Bitetti, M. (2008), “Jaguar *Panthera
15895 onca* population decline in the Upper Paraná Atlantic Forest of Argentina and Brazil,”
15896 *Oryx*, 42, 554.
- 15897 Paviolo, A., Di Blanco, Y., De Angelo, C., and Di Bitetti, M. (2009), “Protection af-
15898 fects the abundance and activity patterns of pumas in the Atlantic Forest,” *Journal of
15899 Mammalogy*, 90, 926–934.
- 15900 Pebesma, E. and Bivand, R. (2011), *Package ‘sp’*, r package version 0.9-91.
- 15901 Pledger, S. (2004), “Unified maximum likelihood estimates for closed capture–recapture
15902 models using mixtures,” *Biometrics*, 56, 434–442.
- 15903 Plummer, M. (2003), “JAGS: A program for analysis of Bayesian graphical models us-
15904 ing Gibbs sampling,” in *Proceedings of the 3rd International Workshop on Distributed
15905 Statistical Computing (DSC 2003). March*, pp. 20–22.
- 15906 —— (2009), “rjags: Bayesian graphical models using mcmc. R package version 1.0 3-12,” .
- 15907 —— (2011), “rjags: Bayesian graphical models using mcmc. R package version 3-5,” .
- 15908 Plummer, M., Best, N., Cowles, K., and Vines, K. (2006), “CODA: Convergence Diagnosis
15909 and Output Analysis for MCMC,” *R News*, 6, 7–11.
- 15910 Pollock, K. H. (1982), “A Capture-Recapture Design Robust to Unequal Probability of
15911 Capture.” *Journal of Wildlife Management*, 46, 752–757.
- 15912 Pollock, K. H., Nichols, J. D., Brownie, C., and Hines, J. E. (1990), “Statistical inference

- 15913 for capture-recapture experiments," *Wildlife monographs*, 3–97.
15914 Porneluzi, P. A. and Faaborg, J. (1999), "Season long fecundity, survival, and viability
15915 of Ovenbirds in fragmented and unfrangmented landscapes," *Conservation Biology*, 13,
15916 1151–1161.
15917 Pradel, R. (1996), "Utilization of Capture-Mark-Recapture for the Study of Recruitment
15918 and Population Growth Rate," *Biometrics*, 52, 703–709.
15919 Raabe, J. (2012), "Factors Influencing Distribution and Survival of Migratory Fishes Fol-
15920 lowing Multiple Low-Head Dam Removals on a North Carolina River. PhD disserta-
15921 tion." Ph.D. thesis, North Carolina State University, Raleigh, NC.
15922 Rathbun, S. (1996), "Estimation of Poisson intensity using partially observed concomitant
15923 variables," *Biometrics*, 52, 226–242.
15924 Rathbun, S. and Cressie, N. (1994), "A space-time survival point process for a longleaf
15925 pine forest in southern Georgia," *Journal of the American Statistical Association*, 89,
15926 1164–1174.
15927 Rathbun, S., Shiffman, S., and Gwaltney, C. (2007), "Modelling the effects of partially
15928 observed covariates on Poisson process intensity," *Biometrika*, 94, 153–165.
15929 Reich, B. J., Gardner, B., and Wilting, A. (2012), "A spatial capture-recapture model for
15930 territorial species," *Biometrics in review*, xx, xx–xxx.
15931 Reynolds, D. and Beecham, J. (1980), "Home range activities and reproduction of black
15932 bears in west-central Idaho," *Bears: Their Biology and Management*, 4, 181–190.
15933 Robert, C. P. and Casella, G. (2004), *Monte Carlo statistical methods*, New York, USA:
15934 Springer.
15935 — (2010), *Introducing Monte Carlo Methods with R*, New York, USA: Springer.
15936 Roberts, G. O. and Rosenthal, J. S. (1998), "Optimal scaling of discrete approximations
15937 to Langevin diffusions," *Journal of the Royal Statistical Society: Series B (Statistical
15938 Methodology)*, 60, 255–268.
15939 Royle, J., Kéry, M., and Guélat, J. (2011a), "Spatial capture-recapture models for search-
15940 encounter data," *Methods in Ecology and Evolution*, 2, 602–611.
15941 Royle, J. A. (2004a), "Generalized estimators of avian abundance from count survey data,"
15942 *Animal Biodiversity and Conservation*, 27, 375–386.
15943 — (2004b), "N-Mixture Models for Estimating Population Size from Spatially Replicated
15944 Counts," *Biometrics*, 60, 108–115.
15945 — (2006), "Site occupancy models with heterogeneous detection probabilities," *Biomet-
15946 rics*, 62, 97–102.
15947 — (2008), "Modeling individual effects in the Cormack–Jolly–Seber model: a state–space
15948 formulation," *Biometrics*, 64, 364–370.
15949 — (2009a), "Analysis of capture–recapture models with individual covariates using data
15950 augmentation," *Biometrics*, 65, 267–274.
15951 — (2009b), "Analysis of capture-recapture models with individual covariates using data
15952 augmentation," *Biometrics*, 65, 267–274.
15953 Royle, J. A. and Chandler, R. B. (2012), "Integrating Resource Selection Information with
15954 Spatial Capture-Recapture," *arXiv preprint arXiv:1207.3288*.
15955 Royle, J. A., Chandler, R. B., Gazenski, K. D., and Graves, T. A. (2012a), "Spatial
15956 capture-recapture for jointly estimating population density and landscape connectivity,"
15957 *Ecology*, to appear.
15958 Royle, J. A., Chandler, R. B., Sun, C., and Fuller, A. (2012b), "Integrating Resource

- 15959 Selection Information with Spatial Capture-Recapture," *needs updated MEE*.
15960 Royle, J. A. and Converse, S. J. (in review), "Hierarchical spatial capture-recapture mod-
15961 ells: Modeling population density from replicated capture-recapture experiments," *Ecol-*
15962 *ogy*.
15963 Royle, J. A., Converse, S. J., and Link, W. A. (2012c), "Data Augmentation for Hierar-
15964 chical Capture-recapture Models," *arXiv preprint arXiv:1211.5706*.
15965 Royle, J. A., Dawson, D. K., and Bates, S. (2004), "Modeling abundance effects in distance
15966 sampling," *Ecology*, 85, 1591–1597.
15967 Royle, J. A. and Dorazio, R. M. (2006), "Hierarchical models of animal abundance and
15968 occurrence," *Journal of Agricultural, Biological, and Environmental Statistics*, 11, 249–
15969 263.
15970 —— (2008), *Hierarchical modeling and inference in ecology: the analysis of data from pop-*
15971 *ulations, metapopulations and communities*, Academic Press.
15972 —— (2012), "Parameter-expanded data augmentation for Bayesian analysis of capture-
15973 recapture models," *Journal of Ornithology*, 152, S521–S537.
15974 Royle, J. A., Dorazio, R. M., and Link, W. A. (2007), "Analysis of multinomial models with
15975 unknown index using data augmentation," *Journal of Computational and Graphical
15976 Statistics*, 16, 67–85.
15977 Royle, J. A. and Dubovský, J. A. (2001), "Modeling spatial variation in waterfowl band-
15978 recovery data," *The Journal of wildlife management*, 65, 726–737.
15979 Royle, J. A. and Gardner, B. (2011), "Hierarchical models for estimating density from trap-
15980 ping arrays," in *Camera traps in animal ecology: methods and analyses*, eds. O'Connel,
15981 A. F. J., Nichols, J. D., and Karanth, U., Tokyo, Japan: Springer Verlag, pp. 163–190.
15982 Royle, J. A., Karanth, K. U., Gopalaswamy, A. M., and Kumar, N. S. (2009a), "Bayesian
15983 inference in camera trapping studies for a class of spatial capture-recapture models,"
15984 *Ecology*, 90, 3233–3244.
15985 Royle, J. A. and Kéry, M. (2007), "A Bayesian state-space formulation of dynamic occu-
15986 pancy models," *Ecology*, 88, 1813–1823.
15987 Royle, J. A. and Link, W. A. (2006), "Generalized site occupancy models allowing for
15988 false positive and false negative errors," *Ecology*, 87, 835–841.
15989 Royle, J. A., Magoun, A. J., Gardner, B., Valkenburg, P., and Lowell, R. E. (2011b),
15990 "Density estimation in a wolverine population using spatial capture–recapture models,"
15991 *The Journal of Wildlife Management*, 75, 604–611.
15992 Royle, J. A. and Nichols, J. D. (2003), "Estimating abundance from repeated presence-
15993 absence data or point counts," *Ecology*, 84, 777–790.
15994 Royle, J. A., Nichols, J. D., Karanth, K. U., and Gopalaswamy, A. M. (2009b), "A
15995 hierarchical model for estimating density in camera-trap studies," *Journal of Applied
15996 Ecology*, 46, 118–127.
15997 Royle, J. A. and Young, K. V. (2008), "A Hierarchical Model For Spatial Capture-
15998 Recapture Data," *Ecology*, 89, 2281–2289.
15999 Russell, R. E., Royle, J. A., Desimone, R., Schwartz, M. K., Edwards, V. L., Pilgrim,
16000 K. P., and McKelvey, K. S. (2012), "Estimating abundance of mountain lions from
16001 unstructured spatial samples," *Journal of Wildlife Management*.
16002 Rutledge, M. (2012), "XXX to be determined XXX," Ph.D. thesis, North Carolina State
16003 University.
16004 Sadeghpour, M. and Ginnett, T. (2011), "Habitat selection by female American black

- 16005 bears in northern Wisconsin," *Ursus*, 22, 159–166.
16006 Sæther, B.-E. and Bakke, Ø. (2000), "Avian life history variation and contribution of
16007 demographic traits to the population growth rate," *Ecology*, 81, 642653.
16008 Saïd, S. and Servanty, S. (2005), "The influence of landscape structure on female roe deer
16009 home-range size," *Landscape ecology*, 20, 1003–1012.
16010 Salom-Pérez, R., Carrillo, E., Sáenz, J., and Mora, J. (2007), "Critical condition of the
16011 jaguar *Panthera onca* population in Corcovado National Park, Costa Rica," *Oryx*, 41,
16012 51.
16013 Sanathanan, L. (1972), "Estimating the size of a multinomial population," *The Annals of
16014 Mathematical Statistics*, 43, 142–152.
16015 Schofield, M. and Barker, R. (2008), "A unified capture-recapture framework," *Journal of
16016 agricultural, biological, and environmental statistics*, 13, 458–477.
16017 Schwartz, M. K., Copeland, J. P., Anderson, N. J., Squires, J. R., Inman, R. M., McKelvey,
16018 K. S., Pilgrim, K. L., Waits, L. P., and Cushman, S. A. (2009), "Wolverine gene flow
16019 across a narrow climatic niche," *Ecology*, 90, 3222–3232.
16020 Schwartz, M. K. and Monfort, S. L. (2008), *Genetic and endocrine tools for carnivore
16021 surveys*, Island Press Washington, DC, USA.
16022 Schwarz, C. J. and Arnason, A. N. (1996), "A General Methodology for the Analysis of
16023 Capture-Recapture Experiments in Open Populations," *Biometrics*, 52, 860–873.
16024 — (2005), "Jolly-Seber Models in MARK," in *Program MARK: A Gentle Introduction*,
16025 5th ed., eds. Cooch, E. G. and White, G.
16026 Schwarz, C. J., Bailey, R. E., Irvine, J. R., and Dalziel, F. C. (1993), "Estimating salmon
16027 spawning escapement using capture-recapture methods," *Canadian Journal of Fisheries
16028 and Aquatic Sciences*, 50, 1181–1191.
16029 Seber, G. A. F. (1965), "A Note on the Multiple-Recapture Census," *Biometrika*, 52,
16030 249–259.
16031 — (1982), *The estimation of animal abundance and related parameters*, Macmillan Pub-
16032 lishing Co.
16033 Sepúlveda, M. A., Bartheld, J. L., Monsalve, R., Gómez, V., and Medina-Vogel, G. (2007),
16034 "Habitat use and spatial behaviour of the endangered Southern river otter (*Lontra
16035 provocax*) in riparian habitats of Chile: conservation implications," *Biological Conserva-
16036 tion*, 140, 329–338.
16037 Shirk, A. J., Wallin, D. O., Cushman, S. A., Rice, C. G., and Warheit, K. I. (2010),
16038 "Inferring landscape effects on gene flow: a new model selection framework," *Molecular
16039 ecology*, 19, 3603–3619.
16040 Sillett, S., Chandler, R. B., Royle, J. A., Kéry, M., and Morrison, S. (2012), "Hierarchical
16041 distance sampling models to estimate population size and habitat-specific abundance
16042 of an island endemic," *Ecological Applications*.
16043 Sillett, T., Rodenhouse, N., and Holmes, R. (2004), "Experimentally reducing neighbor
16044 density affects reproduction and behavior of a migratory songbird," *Ecology*, 85, 2467–
16045 2477.
16046 Simons, T. R., Pollock, K. H., Wettroth, J. M., Alldredge, M. W., Pacifici, K., and
16047 Brewster, J. (2009), "Sources of Measurement Error, Misclassification Error, and Bias
16048 in Auditory Avian Point Count Data," in *Modeling Demographic Processes In Marked
16049 Populations*, eds. Thomson, D. L., Cooch, E. G., and Conroy, M. J., Boston, MA:
16050 Springer US, vol. 3, pp. 237–254.

- 16051 Skalski, J. R., Millspaugh, J. J., and Spencer, R. D. (2005), "Population estimation and
16052 biases in paintball, mark-resight surveys of elk," *Journal of Wildlife Management*, 69,
16053 1043–1052.
- 16054 Sklyar, O., Murdoch, D., Smith, M., Eddelbuettel, D., and François, R. (2010), *inline:*
16055 *Inline C, C++, Fortran function calls from R*, r package version 0.3.8.
- 16056 Smith, M. H., Blessing, R., Chelton, J. G., Gentry, J. B., Golley, F. B., and McGinnis,
16057 J. T. (1971), "Determining density for small mammal populations using a grid and
16058 assessment lines," *Acta Theriologica*, 16, 105–125.
- 16059 Soisalo, M. K. and Cavalcanti, S. M. C. (2006), "Close-up Space in "Radio-telemetry","
16060 *Biological Conservation*, 129, 487–496.
- 16061 Sollmann, R., Furtado, M. M., Gardner, B., Hofer, H., Jacomo, A. T. A., Trres, N. M., and
16062 Silveira, L. (2011), "Improving density estimates for elusive carnivores: Accounting for
16063 sex-specific detection and movements using spatial capture-recapture models for jaguars
16064 in central Brazil," *Biological Conservation*, 144, 1017–1024.
- 16065 Sollmann, R., Gardner, B., and Belant, J. L. (2012a), "How Does Spatial Study Design
16066 Influence Density Estimates from Spatial Capture-Recapture Models?" *PloS one*, 7,
16067 e34575.
- 16068 Sollmann, R., Gardner, B., Chandler, R. B., Shindle, D., Onorato, D. P., Royle, J. A., and
16069 O'Connell, A. F. (in revision), "Using multiple data sources provides density estimates
16070 for endangered Florida panther," *Journal of Applied Ecology*.
- 16071 Sollmann, R., Gardner, B., Parsons, A., Stocking, J., McClintock, B., Simons, T., Pol-
16072 lock, K., and O'Connell, A. (2012b), "A spatial mark-resight model augmented with
16073 telemetry data," *Ecology*.
- 16074 Sólymos, P., Lele, S., and Bayne, E. (2012), "Conditional likelihood approach for analyzing
16075 single visit abundance survey data in the presence of zero inflation and detection error,"
16076 *Environmetrics*, 23, 197–205.
- 16077 Spear, S. F., Balkenhol, N., Fortin, M. J., Mcrae, B. H., and Scribner, K. (2010), "Use
16078 of resistance surfaces for landscape genetic studies: considerations for parameterization
16079 and analysis," *Molecular Ecology*, 19, 3576–3591.
- 16080 Spiegelhalter, D., Best, N., Carlin, B., and Van Der Linde, A. (2002), "Bayesian mea-
16081 sures of model complexity and fit," *Journal of the Royal Statistical Society: Series B
16082 (Statistical Methodology)*, 64, 583–639.
- 16083 Stabler, B. (2006), *shapefiles: Read and Write ESRI Shapefiles*, r package version 0.6.
- 16084 Stanley, T. and Burnham, K. (1999), "A closure test for time-specific capture-recapture
16085 data," *Environmental and Ecological Statistics*, 6, 197–209.
- 16086 Stevens Jr, D. and Olsen, A. (2004), "Spatially balanced sampling of natural resources,"
16087 *Journal of the American Statistical Association*, 99, 262–278.
- 16088 Stoyan, D. and Penttinen, A. (2000), "Recent Applications of Point Process Methods in
16089 Forestry Statistics," *Statistical Science*, 15, 61–78.
- 16090 Sturtz, S., Ligges, U., and Gelman, A. (2005), "R2WinBUGS: A Package for Running
16091 WinBUGS from R," *Journal of Statistical Software*, 12, 1–16.
- 16092 Su, Y.-S. and Yajima, M. (2011), *R2jags: A Package for Running jags from R*, r package
16093 version 0.02-17.
- 16094 Sun, C. C. (in prep), "Population estimation, genetic diversity, and structure of black
16095 bears in southwestern New York," Master's thesis, Cornell University.
- 16096 Taberlet, P. and Bouvet, J. (1992), "Bear conservation genetics," *Nature*, 358, 197–197.

- 16097 Tanner, M. A. and Wong, W. H. (1987), "The calculation of posterior distributions by
16098 data augmentation," *J Am Stat Assoc*, 82, 528–540.
- 16099 Thomas, A., O'Hara, B., Ligges, U., and Sturtz, S. (2006), "Making BUGS Open," *R
16100 News*, 6, 12–17.
- 16101 Thomas, M. (1949), "A generalization of Poisson's binomial limit for use in ecology,"
16102 *Biometrika*, 36, 18–25.
- 16103 Thompson, C., Royle, J. A., and Garner, J. (2012), "A framework for inference about
16104 carnivore density from unstructured spatial sampling of scat using detector dogs," *The
16105 Journal of Wildlife Management*, 76, 863–871.
- 16106 Thompson, S. K. (2002), *Sampling*, New York: Wiley.
- 16107 Tierney, L., Rossini, A. J., Li, N., and Sevcikova, H. (2011), *snow: Simple Network of
16108 Workstations*, r package version 0.3–7.
- 16109 Tilman, D. and Kareiva, P. (1997), *Spatial ecology: the role of space in population dynamics
16110 and interspecific interactions*, vol. 30, Princeton University Press.
- 16111 Tischendorf, L. and Fahrig, L. (2000), "On the usage and measurement of landscape
16112 connectivity," *Oikos*, 90, 7–19.
- 16113 Tischendorf, L., Grez, A., Zaviezo, T., Fahrig, L., and ALLIANCE, R. (2005), "Mechanisms
16114 affecting population density in fragmented habitat," *Ecology and Society*.
- 16115 Tobler, M., Carrillo-Percastegui, S., Leite Pitman, R., Mares, R., and Powell, G. (2008),
16116 "An evaluation of camera traps for inventorying large-and medium-sized terrestrial
16117 rainforest mammals," *Animal Conservation*, 11, 169–178.
- 16118 Tobler, M. W., Hibert, F., Debeir, L., and Hansen, C. (2012), "Density and sustainable
16119 harvest estimates for the lowland tapir in the Amazon of French Guiana using a spatial
16120 capture-recapture model," *none yet*, none, none.
- 16121 Tracy, J. A. (2006), "Individual-based movement modeling as a tool for conserving con-
16122 nectivity," *Connectivity conservation*.
- 16123 Trolle, M. and Kéry, M. (2003), "Estimation of ocelot density in the Pantanal using
16124 capture-recapture analysis of camera-trapping data," *Journal of Mammalogy*, 84, 607–
16125 614.
- 16126 — (2005), "Camera-trap study of ocelot and other secretive mammals in the northern
16127 Pantanal," *Mammalia*, 69, 409–416.
- 16128 Tufto, J., Andersen, R., and Linnell, J. (1996), "Habitat use and ecological correlates
16129 of home range size in a small cervid: the roe deer," *Journal of Animal Ecology*, 65,
16130 715–724.
- 16131 Tyre, A. J., Tenhumberg, B., Field, S. A., Niejalke, D., Parris, K., and Possingham,
16132 H. P. (2003), "Improving precision and reducing bias in biological surveys: estimating
16133 false-negative error rates," *Ecological Applications*, 13, 1790–1801.
- 16134 Valiere, N. and Taberlet, P. (2000), "Urine collected in the field as a source of DNA for
16135 species and individual identification," *Molecular Ecology*, 9, 2150–2152.
- 16136 van Etten, J. (2011), *Package gdistance*, r package version 1.1–2.
- 16137 van Etten, R. J. H. . J. (2012), *raster: Geographic analysis and modeling with raster data*,
16138 r package version 1.9–67.
- 16139 Venables, W. and Ripley, B. (2002), *Modern applied statistics with S*, Springer verlag.
- 16140 Venables, W., Smith, D., and Team, R. D. C. (2012), "An introduction to R," .
- 16141 Ver Hoef, J. and Boveng, P. (2007), "Quasi-poisson vs. negative binomial regression: How
16142 should we model overdispersed count data?" *Ecology*, 88, 2766–2772.

- 16143 Ver Hoef, J. M. (2012), "Who Invented the Delta Method?" *The American Statistician*,
16144 66, 124–127.
- 16145 Wallace, R. B., Gomez, H., Ayala, G., and Espinoza, F. (2003), "Camera trapping for
16146 jaguar (*Panthera onca*) in the Tuichi Valley, Bolivia," *Journal of Neotropical Mammal-*
16147 *ogy*, 10, 133–139.
- 16148 Warton, D. and Shepherd, L. (2010), "Poisson point process models solve the pseudo-
16149 absence problem for presence-only data in ecology," *The Annals of Applied Statistics*,
16150 4, 1383–1402.
- 16151 Wegan, M., Curtis, P., Rainbolt, R., and Gardner, B. (2012), "Temporal sampling frame
16152 selection in DNA-based capture mark-recapture investigations." *Ursus*, 23, 42 – 51.
- 16153 Wegan, M. T. (2008), "Aversive conditioning, population estimation, and habitat prefer-
16154 ence of black bears (*Ursus americanus*) on Fort Drum Military Installation in northern
16155 New York," Ph.D. thesis, Cornell University, Jan.
- 16156 Wegge, P., Pokheral, C. P., and Jnawali, S. R. (2004), "Effects of trapping effort and trap
16157 shyness on estimates of tiger abundance from camera trap studies," *Animal Conserva-*
16158 *tion*, 7, 251–256.
- 16159 White, G. (1996), "NOREMARK: population estimation from mark-resighting surveys,"
16160 *Wildlife Society Bulletin*, 24, 50–52.
- 16161 White, G. and Bennetts, R. (1996), "Analysis of frequency count data using the negative
16162 binomial distribution," *Ecology*, 77, 2549–2557.
- 16163 White, G. and Shenk, M. (2001), "Population estimation with radio-marked individuals." in
16164 *Radio tracking adn animal populations*, eds. Millspaugh, J. and Marzluff, J., San Diego,
16165 USA: Academic Press, pp. 329–350.
- 16166 White, G. C., Anderson, D. R., Burnham, K. P., and Otis, D. (1982), *Capture-recapture*
16167 *and removal methods for sampling closed populations*, Los Alamos: Los Alamos National
16168 Laboratory.
- 16169 White, G. C. and Garrot, R. (1990), *Analysis of wildlife radiolocation data*, New York,
16170 USA: Academic Press.
- 16171 White, G. C. and Shenk, T. M. (2000), *Population estimation with radio-marked animals*,
16172 San Diego, California: Academic Press.
- 16173 Whitman, J., Ballard, W., and Gardner, C. (1986), "Home range and habitat use by
16174 wolverines in southcentral Alaska," *The Journal of wildlife management*, 460–463.
- 16175 Wickham and Hadley (2007), "Reshaping data with the reshape package," *Journal of*
16176 *Statistical Software*, 21.
- 16177 Williams, B. K., Nichols, J. D., and Conroy, M. J. (2002), *Analysis and management of*
16178 *animal populations: modeling, estimation, and decision making*, Academic Pr.
- 16179 Wilson, K. R. and Anderson, D. R. (1985a), "Evaluation of a Density Estimator Based
16180 on a Trapping Web and Distance Sampling Theory," *Ecology*, 66, 1185–1194.
- 16181 — (1985b), "Evaluation of Two Density Estimators of Small Mammal Population Size,"
16182 *Journal of Mammalogy*, 66, 13–21.
- 16183 With, K. and Crist, T. (1995), "Critical thresholds in species' responses to landscape
16184 structure," *Ecology*, 76, 2446–2459.
- 16185 Wolpert, R. L. and Ickstadt, K. (1998), "Poisson/gamma random field models for spatial
16186 statistics," *Biometrika*, 85, 251 –267.
- 16187 Woods, J. G., Paetkau, D., Lewis, D., McLellan, B. N., Proctor, M., and Strobeck, C.
16188 (1999), "Genetic tagging of free-ranging black and brown bears," *Wildlife Society Bul-*

- 16189 *letin*, 27, 616–627.
- 16190 Worton, B. (1989), “Kernel methods for estimating the utilization distribution in home-
16191 range studies,” *Ecology*, 70, 164–168.
- 16192 Wright, J., Barker, R., Schofield, M., Frantz, A., Byrom, A., and Gleeson, D. (2009), “In-
16193 corporating Genotype Uncertainty into Mark–Recapture-Type Models For Estimating
16194 Abundance Using DNA Samples,” *Biometrics*, 65, 833–840.
- 16195 Yang, H. C. and Chao, A. (2005), “Modeling Animals’ Behavioral Response by Markov
16196 Chain Models for Capture–Recapture Experiments,” *Biometrics*, 61, 1010–1017.
- 16197 Yoshizaki, J., Pollock, K. H., Brownie, C., and Webster, R. A. (2009), “Modeling misiden-
16198 tification errors in capture-recapture studies using photographic identification of evolv-
16199 ing marks,” *Ecology*, 90, 3–9.
- 16200 Zeller, K., McGarigal, K., and Whiteley, A. (2012), “Estimating landscape resistance to
16201 movement: a review,” *Landscape Ecology*, 27, 777–797.
- 16202 Zuur, A. F., Ieno, E. N., Walker, N. J., Saveliev, A. A., and Smith, G. M. (2009), *Mixed
16203 effects models and extensions in ecology with R*, Springer Verlag.
- 16204 Zylstra, E., Steidl, R., and Swann, D. (2010), “Evaluating survey methods for monitoring
16205 a rare vertebrate, the Sonoran desert tortoise,” *The Journal of Wildlife Management*,
16206 74, 1311–1318.