

¹

² **Spatial Capture-Recapture**

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

⁷ USGS Patuxent Wildlife Research Center
⁸ North Carolina State University

⁹ ACADEMIC PRESS

CONTENTS

10

11

¹⁴ Acknowledgements
¹⁵

¹⁶

Part I

¹⁷

¹⁸

Background and Concepts

19
20

21

INTRODUCTION

22 Space plays a vital role in virtually all ecological processes (???). The spatial ar-
23 rangement of habitat can influence movement patterns during dispersal, habitat
24 selection, and survival. The distance between an organism and its competitors and
25 prey can influence activity patterns and foraging behavior. Further, understanding
26 distribution and spatial variation in abundance is necessary in the conservation and
27 management of populations. The inherent spatial aspect of *sampling* populations
28 also plays an important role in ecology as it strongly affects, and biases, how we
29 observe population structure (????). However, despite the central role of space and
30 spatial processes to both understanding population dynamics and how we observe
31 or sample populations, a coherent framework that integrates these two aspects of
32 ecological systems has not been fully realized either conceptually or methodologi-
33 cally.

34 Capture-recapture methods represent perhaps the most common technique for
35 studying animal populations, and their use is growing in popularity due to recent
36 technological advances that provide mechanisms to study many taxa which before
37 could not be studied efficiently, if at all. However, a major deficiency of classical
38 capture-recapture methods is that they do not admit the spatial structure of either
39 ecological processes that give rise to encounter history data, nor the spatial aspect
40 of collecting these data. While many technical limitations of this lack of spatial
41 explicitness have been recognized for decades (??), it has only been very recent
42 (??) that spatially explicit capture-recapture methods – those which accommodate
43 space – have been developed.

44 Spatial capture-recapture (SCR) methods resolve a host of technical problems
45 that arise in applying capture-recapture methods to animal populations. However,
46 SCR models are not merely an extension of technique. Rather, they represent a
47 much more profound development in that they make ecological processes explicit in

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

1.1 THE STUDY OF POPULATIONS BY CAPTURE-RECAPTURE

In the fields of conservation, management, and general applied ecology, information about abundance or density of populations and their vital rates is a basic requirement. To that end, a huge variety of statistical methods have been devised, and as we noted already, the most well-developed are collectively known as capture-recapture (or capture-mark-recapture) methods. For example, the volumes by ?, ?, ?, ?, ?, ?, and ? are largely synthetic treatments of such methods, and contributions on modeling and estimation using capture-recapture are plentiful in the peer-reviewed ecology literature.

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

Capture-recapture has been important in studies of animal populations for many decades, and its importance is growing dramatically in response to technological advances that improve our ability and efficiency to obtain encounter history data. Historically, such information was obtainable using methods requiring physical capture of individuals. However, new methods do not require physical capture or handling of individuals. A large number of passive detection devices produce individual encounter history data including camera traps (??), acoustic recording devices (?), and methods that obtain DNA samples such as hair snares for bears, scent posts for many carnivores, and related methods which allow DNA to be extracted from scat, urine or animal tissue in order to identify individuals. This book is concerned with how such data can be used to carry out inference about animal abundance or density, and other parameters such as survival, recruitment, resource selection, and movement using new classes of capture-recapture models which utilize auxiliary spatial information related to the encounter process. We refer to such methods as spatial capture-recapture (SCR) models¹.

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

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

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

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

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

1.2.1 Camera trapping

Considerable recent work has gone into the development of camera-trapping methodologies. For a historical overview of this method see ? and ?. Several recent synthetic works have been published including ?, and an edited volume by ? devoted solely to camera trapping concepts and methods. As a method for estimating abundance, some of the earliest work that relates to the use of camera trapping data in capture-recapture models originates from Karanth and colleagues (??).

In camera trapping studies, cameras are often situated along trails or at baited stations and individual animals are photographed and subsequently identified either manually by a person sitting behind a computer, or sometimes now using specific

⁵prefer the more concise term.

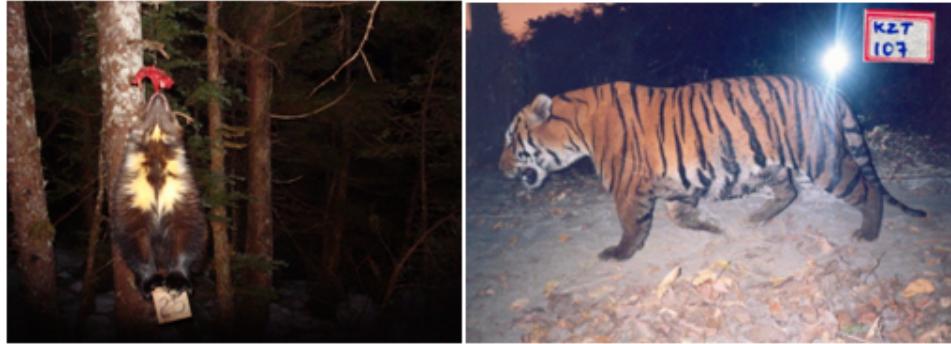


Figure 1.1. Left: Wolverine being encounter by a camera trap (*Photo credit: Audrey Magoun*). Right: Tiger encountered by camera trap (*Photo credit: Ullas Karanth*).

121 identification software. Camera trapping methods are widely used for species that
 122 have unique stripe or spot patterns such as tigers (??), ocelots (*Leopardus pardalis*
 123 ; ??)), leopards (*Panthera pardus*; ??), and many other cat species. Camera traps
 124 are also used for other species such as wolverines (*Gulo gulo*; ??), and even species
 125 that are less easy to identify uniquely such as mountain lions (*Puma concolor*, ??))
 126 and coyotes (*Canis latrans*, ??). We note that even for species that are not readily
 127 identified by pelage patterns, it might be efficient to use camera traps in conjunction
 128 with spatial capture-recapture models to estimate density (see Chaps. ?? and ??).

129 **1.2.2 DNA sampling**

130 DNA obtained from hair, blood or scat is now routinely used to obtain individual
 131 identity and encounter history information about individuals (?????). A common
 132 method is based on the use of “hair snares” (Fig. 1.2) which are widely used to
 133 study bear populations (????). A sample of hair is obtained as individuals pass
 134 under or around barbed-wire (or other physical mechanism) to take bait. Hair
 135 snares and scent sticks have also been used to sample felid populations (??) and
 136 other species. Research has even shown that DNA information can be extracted
 137 from urine deposited in the wild (e.g., in snow; see ?) and as a result this may
 138 prove another future data collection technique where SCR models are useful.

139 **1.2.3 Acoustic sampling**

140 Many studies of birds (?), bats, and whales (?) now collect data using devices that
 141 record vocalizations. When vocalizations can be identified by individual from mul-
 142 tiple recording devices, spatial encounter histories are produced that are amenable

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

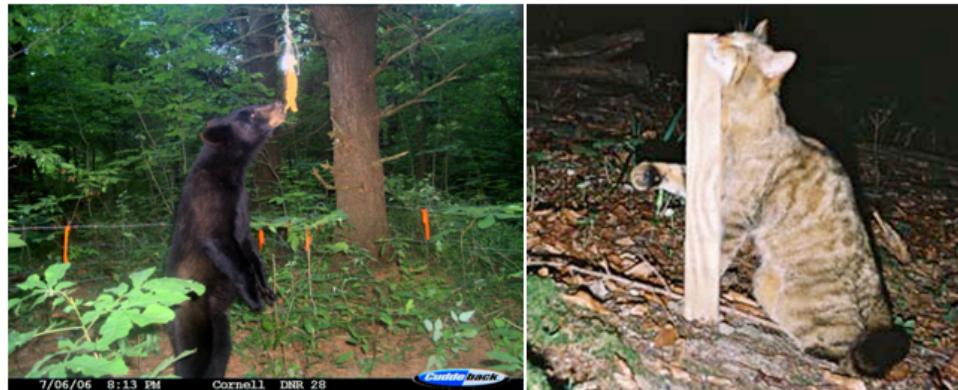


Figure 1.2. Left: Black bear in a hair snare (*Photo credit: M. Wegan*) Right: European wildcat loving on a scent stick (*Photo credit: Darius Weber*)



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

¹⁴³ to the application of SCR models (??). Recently, these ideas have been applied to
¹⁴⁴ data on direction or distance to vocalizations by multiple simultaneous observers
¹⁴⁵ and related problems (D. Borchers, ISEC 2012 presentation).

¹⁴⁶ **1.2.4 Search-encounter methods**

¹⁴⁷ There are other methods which don't fall into a nice clean taxonomy of "devices".
¹⁴⁸ Spatial encounter histories are commonly obtained by conducting manual searches
¹⁴⁹ of geographic sample units such as quadrats, transects or road or trail networks.
¹⁵⁰ For example, DNA-based encounter histories can be obtained from scat samples
¹⁵¹ located along roads or trails or by specially trained dogs (?) searching space (Fig.
¹⁵² 1.3). This method has been used in studies of martens, fishers (?), lynx, coyotes,
¹⁵³ birds (?), and many other species. A similar data structure arises from the use of
¹⁵⁴ standard territory or spot mapping of birds ? or area sampling in which space is
¹⁵⁵ searched by observers to physically capture individuals. This is common in surveys
¹⁵⁶ that involve reptiles and amphibians, e.g., we might walk transects picking up box
¹⁵⁷ turtles (?), or desert tortoises (?), or search space for lizards (?).

¹⁵⁸ These methods don't seem like normal capture-recapture in the sense that the
¹⁵⁹ encounter of individuals is not associated with specific trap location, but SCR
¹⁶⁰ models are equally relevant for analysis of such data as we discuss in Chapt. ??.

1.3 CAPTURE-RECAPTURE FOR MODELING ENCOUNTER PROBABILITY

¹⁶¹ We briefly introduced techniques used for the study of animal populations. These
¹⁶² methods produce individual encounter history data, a record of where and when
¹⁶³ each individual was captured. We refer to this as a *spatial encounter history*. Historically,
¹⁶⁴ auxiliary spatial information has been ignored, and encounter history data
¹⁶⁵ have been *summarized* to simple "encounter or not" for the purpose of applying
¹⁶⁶ ordinary CR models. The basic problem with these ordinary (or "non-spatial")
¹⁶⁷ capture-recapture models is they don't have any sense of space in them, the spatial
¹⁶⁸ information is summarized out of the data set, so we aren't able to use such mod-
¹⁶⁹ els for studying things such as movement, or resource selection, etcdots. Instead,
¹⁷⁰ ordinary capture-recapture models usually resort to models of "encounter prob-
¹⁷¹ ability," which is a nuisance parameter, seldom of any ecological relevance. We
¹⁷² show an example here that is in keeping with the classical application of ordinary
¹⁷³ capture-recapture models.

¹⁷⁴ **1.3.1 Example: Fort Drum bear study**

¹⁷⁵ Here we confront the simplest possible capture-recapture problem – but one of great
¹⁷⁶ applied interest – estimating density from a standard capture-recapture study. We
¹⁷⁷ 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 (?, see also Chapt. ?? 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 ? and ?, and we use the data in a number of analyses in later chapters.

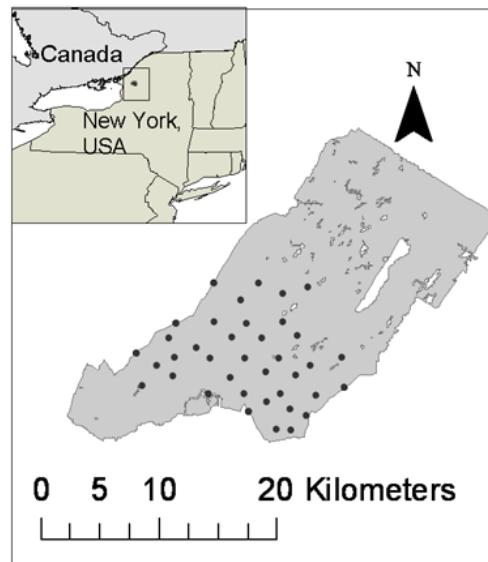


Figure 1.4. Locations of hair snares on Fort Drum, New York, operated during the summer of 2006 to sample black bears.

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

The key issue in using such a closed population model regards how we should

197 interpret this estimate of $N = 49.19$ bears. Does it represent the entire population
198 of Fort Drum? Certainly not – the trapping array covers less than half of Fort
199 Drum as we see in Fig. 1.4. So to get at the total bear population size of Fort
200 Drum, we would have to convert our \hat{N} to an estimate of density and extrapolate.
201 To get at density, then, should we assert that N applies to the southern half of
202 Fort Drum below some arbitrary line? Surely bears move on and off of Fort Drum
203 without regard to hypothetical boundaries. Without additional information there
204 is simply no way of converting this estimate of N to density, and hence it is really
205 not meaningful biologically. To resolve this problem, we will adopt the customary
206 approach of converting N to D by buffering the convex hull around the trap array.
207 The convex hull has area 157.135 km^2 . We follow ? in buffering the convex hull of
208 the trap array by the radius of the mean female home range size.

209 The mean female home range radius was estimated (?) for this study region to
210 be 2.19 km , and the area of the convex hull buffered by 2.19 km is 277.01 km^2 . (**R**
211 commands to compute the convex hull, buffer it, and compute the area are given
212 in the **R** package **scrbook** which accompanies the book). Hence, the estimated
213 density here is approximately 0.178 bears/km^2 using the estimated population size
214 obtained by model M_0 . We could assert that the problem has been solved, go home,
215 and have a beer. But then, on the other hand, maybe we should question the use
216 of the estimated home range radius – after all, this is only the female home range
217 radius and the home ranges change for many reasons. Instead, we may decide to
218 rely on a buffer width based on one-half mean maximum distance moved (MMDM)
219 estimated from the actual hair snare data as is more customary (?). In that case
220 the buffer width is 1.19 km , and the resulting estimated density is increased to
221 0.225 bears/km^2 about 27 % larger. But wait – some studies actually found the
222 full MMDM (?) to be a more appropriate measure of movement (e.g. ?). So maybe
223 we should use the full MMDM which is 2.37 km , pretty close to the telemetry-based
224 estimate and therefore providing a similar estimate of density (0.171 bears/km^2).
225 So in trying to decide how to buffer our trap array we have already generated 3
226 density estimates. The crux of the matter is obvious: Although it is intuitive that
227 N should scale with area – the number of bears should go up as area increases
228 and go down as area decreases – in this ad hoc approach of accounting for animal
229 movement N remains the same, no matter what area we assert was sampled. The
230 number of bears and the area they live in are not formally tied together within
231 the model, because estimating N and estimating the area N relates to are two
232 completely independent analytical steps which are unrelated to one another by a
233 formal model.

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

240 phenomenon. To illustrate this phenomenon, here are the number of traps that each
 241 individual was encountered in:

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

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

263 So what are we left with? Our density estimates span a range from 0.17 to
 264 0.43 bears/km² depending on which estimator of N we use and what buffer strip
 265 we apply. Should we feel strongly about one or the other? Which buffer should
 266 we prefer? AIC favors model M_h , but did it adequately account for the differ-
 267 ences in exposure of individuals to the trap array? Are we happy with a purely
 268 phenomenological model for heterogeneity? It assumes that all individuals are in-
 269 dependent and identically distributed (*iid*) draws from some distribution, but does
 270 not account for the explicit mechanism of induced heterogeneity. And, further, we
 271 have information about that (trap of capture) which model M_h ignores. And if we
 272 choose one type of buffer, how do we compare our density estimates to those from
 273 other studies that may opt for a different kind of buffer? The fact that N does not
 274 scale with A , as part of the model, renders this choice arbitrary.

275 **1.3.2 Inadequacy of non-spatial capture-recapture**

276 The parameter N (population size) in an ordinary capture-recapture model is func-
 277 tionally unrelated to any notion of sample area, and so we are left taking arbitrary
 278 guesses at area, and matching it up with estimates of N from different models that
 279 do not have any explicit biological relevance. Clearly, there is not a compelling

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 (?).

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

solution to be derived from this “estimate N and conjure up a buffer” approach and we are left not much wiser about bear density at Fort Drum than we were before we conducted this analysis, and certainly not confident in our assessments. Closed population models are not integrated with any ecological theory, so our N is not connected to the specific landscape in any explicit way.

The capture-recapture models that we used apply to truly closed populations – a population of goldfish in a fish bowl. Yet here we are applying them to a population of bears that inhabit a rich two-dimensional landscape of varied habitats, exposed to trapping by an irregular and sparse array of traps. It seems questionable that the same model that is completely sensible for a population of goldfish in a bowl, should also be the right model for this population of bears distributed over a broad landscape. Ordinary capture-recapture methods are distinctly non-spatial. They don’t admit spatial indexing of either sampling (the observation process) or of individuals (the ecological process). This leads immediately to a number of practical deficiencies: (1) Ordinary CR models do not provide a coherent basis for estimating density, a problem we struggled with in the black bear study. (2) Ordinary CR model and sampling methods *induce* a form of heterogeneity that can only at best be approximated by classical models of latent heterogeneity. SCR models formally accommodate heterogeneity due to the juxtaposition of individuals with the encounter devices. (3) Ordinary CR models do not accommodate trap-level covariates which exist in a large proportion of real studies; (4) Ordinary CR models do not accommodate formal consideration of any spatial process that gives rise to the observed data.

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

1.4 HISTORICAL CONTEXT: A BRIEF SYNOPSIS

Spatial capture-recapture is a relatively new methodological development, at least with regard to formal estimation and inference. However, the basic problems that motivate the need for formal spatially-explicit models have been recognized for

³⁰⁹ decades and quite a large number of ideas have been proposed to deal with these
³¹⁰ problems. We review some of these ideas here.

³¹¹ **1.4.1 Buffering**

³¹² The standard approach to estimating density even now is to estimate N using
³¹³ conventional closed population models (?) and then try to associate with this estimate
³¹⁴ some specific sampled area, say A , the area which is contributing individuals to the
³¹⁵ population for which N is being estimated. The strategy is to define A by placing
³¹⁶ a buffer of say W around the trap array or some polygon which encloses the trap
³¹⁷ array. The historical context is succinctly stated by (?) from which we draw this
³¹⁸ description:

³¹⁹ “At its most simplistic, A may be described by a concave polygon defined by con-
³²⁰ necting the outermost trap locations (A_{tp} ; ?). This assumes that animals do not
³²¹ move from outside the bounded area to inside the area or vice versa. Unless the
³²² study is conducted on a small island or a physical barrier is erected in the study area
³²³ to limit movement of animals, this assumption is unlikely to be true. More often, a
³²⁴ boundary area of width W (A_w) is added to the area defined by the polygon A_{tp} to
³²⁵ reflect the area beyond the limit of the traps that potentially is contributing animals
³²⁶ to the abundance estimate (?). The sampled area, also known as the effective area,
³²⁷ is then $A(W) = A_{tp} + A_w$. Calculation of the buffer strip width (W) is critical to the
³²⁸ estimation of density and is problematic because there is no agreed upon method of
³²⁹ estimating W . Solutions to this problem all involve ad hoc methods that date back
³³⁰ to early attempts to estimate abundance and home ranges based on trapping grids
³³¹ (see ?). ? first drew attention to this problem in small mammal studies and recom-
³³² mended using one-half the diameter of an average home range. Other solutions have
³³³ included use of inter-trap distances (??), mean movements among traps, maximum
³³⁴ movements among traps (??), nested grids (?), and assessment lines (?).”

³³⁵ The idea of using $1/2$ mean maximum distance moved (“MMDM” ?) to create
³³⁶ a buffer strip seems to be the standard approach even today, presumably justified
³³⁷ by Dice’s suggestion to use $1/2$ the home range diameter, with the mean over
³³⁸ individuals of the maximum distance moved being an estimator of home range
³³⁹ diameter. Alternatively, some studies have used the full MMDM (e.g. ?), because
³⁴⁰ the trap array might not provide a full coverage of the home range (home ranges near
³⁴¹ the edge should be truncated) and so $1/2$ MMDM should be biased smaller than the
³⁴² home range radius. And, sometimes home range size is estimated by telemetry (??).
³⁴³ Use of MMDM summaries to estimate home range radius is usually combined with
³⁴⁴ an AIC-based selection from among the closed-population models in ? which most
³⁴⁵ often suggests heterogeneity in detection (model M_h). Almost all of these early
³⁴⁶ methods were motivated by studies of small mammals using classical “trapping
³⁴⁷ grids” but, more recently, their popularity in the study of wildlife populations has
³⁴⁸ increased with the advent of new technologies, especially related to non-invasive
³⁴⁹ sampling methods such as camera trapping. In particular, the series of papers by
³⁵⁰ Karanth and Nichols (???) has led to fairly widespread adoption of these ideas.

351 **1.4.2 Temporary emigration**

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

369 Another interesting idea is that of using some summary of “average location” as
370 an individual covariate in standard capture-recapture models. ? use distance-to-
371 edge (DTE) as a covariate in the Huggins-Alho type of model. ? uses this approach
372 in conjunction with an adjustment to the estimated N obtained by estimating the
373 proportion of time individuals are “on the area formally covered by the grid” using
374 radio telemetry. We do not dwell too much on these different variations but we
375 do note that the use of DTE as an individual covariate amounts to some kind
376 of intermediate model between simple closed population models and fully spatial
377 capture-recapture models, which we address directly in Chapt. ??.

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

1.5 EXTENSION OF CLOSED POPULATION MODELS

387 The deficiency with classical closed population models is that they have no spatial
388 context. N is just an integer parameter that applies equally well to estimating the
389 number of unique words in a book, the size of some population that exists in a

390 computer, or a bucket full of goldfish. The question of *where* the N items belong
391 is central both to interpretation of data and estimates from all capture-recapture
392 studies and, in fact, to the construction of spatial capture-recapture models con-
393 sidered in this book. Surely it must matter whether the N items exist as words in
394 a book, or goldfish in a bowl, or tigers in a patch of forest! That classical closed
395 population models have no spatial context leads to a number of conceptual and
396 methodological problems or limitations as we have encountered previously. More
397 important, ecologists seldom care only about N – space is often central to objec-
398 tives of many population studies – movement, space usage, resource selection, how
399 individuals are distributed in space and in response to explicit factors related to
400 landuse or habitat. Because space is central to so many real problems, this is proba-
401 bly the number 1 reason that many ecologists don't bother with capture-recapture.
402 They haven't seen capture-recapture methods as being able to solve their problems.
403 Thus, the essential problem is that classical closed population models are too sim-
404 ple – they ignore the spatial attribution of traps and encounter events, movement
405 and variability in exposure of individuals to trap proximity. These problems can be
406 addressed formally by the development of more general capture-recapture models.

407 **1.5.1 Towards spatial explicitness: Efford's formulation**

408 The solution to the various issues that arise in the application of ordinary capture-
409 recapture models is to extend the closed population model so that N becomes
410 spatially explicit. ? was the first to formalize an explicit model for spatial capture-
411 recapture problems in the context of trapping arrays. He adopted a Poisson point
412 process model to describe the distribution of individuals and essentially a distance
413 sampling formulation of the observation model which describes the probability of
414 detection as a function of individual location, regarded as a latent variable gov-
415 erned by the point process model. While earlier (and contemporary) methods of
416 estimating density from trap arrays have been ad hoc in the sense of lacking a for-
417 mal description of the spatial model, Efford achieved a formalization of the model,
418 describing explicit mechanisms governing the spatial distribution of individuals and
419 how they are encountered by traps, but adopted a more or less ad hoc framework
420 for inference under that spatial model using a simulation based method known as
421 inverse prediction (?).

422 Recently, there has been a flurry of effort devoted to formalizing inference un-
423 der this model-based framework for the analysis of spatial capture-recapture data
424 (????). There are two distinct lines of work which adopt the model-based formu-
425 lation in terms of the underlying point process but differ primarily by the manner
426 in which inference is achieved. One approach (?) uses classical inference based on
427 likelihood (see Chapt. ??), and the other (?) adopts a Bayesian framework for
428 inference (Chaps. ?? and ??).

429 1.5.2 Abundance as the aggregation of a point process

430 Spatial point process models represent a major methodological theme in spatial
431 statistics (?) and they are widely applied as models for many ecological phenomena
432 (??). Point process models apply to situations in which the random variable in
433 question represents the locations of events or objects: trees in a forest, weeds in
434 a field, bird nests, etc... As such, it seems natural to describe the organization
435 of individuals in space using point process models. SCR models represent the
436 extension of ordinary capture-recapture by augmenting the model with a point
437 process to describe individual locations.

438 Specifically, let $\mathbf{s}_i; i = 1, 2, \dots, N$ be the locations of all individuals in the popu-
439 lation. One of the key features of SCR models is that the point locations are latent,
440 or unobserved, and we only obtain imperfect information about the point locations
441 by observing individuals at trap or observation locations. Thus, the realized loca-
442 tions of individuals represent a type of “thinned” point process, where the thinning
443 mechanism is not random but, rather, biased by the observation mechanism. It is
444 also natural to think about the observed point process as some kind of a compound
445 or aggregate point process with a set of “parent” nodes being the locations of in-
446 dividual home ranges or their centroids, and the observed locations as “offspring”
447 - i.e., a Poisson cluster process (PCP). In that context, density estimation in SCR
448 models is analogous to estimating the number of parents of a Poisson cluster process
449 (?).

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

459 1.5.3 The activity center concept

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

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

482 **1.5.4 The state-space**

483 Once we introduce the collection of activity centers, $\mathbf{s}_i; i = 1, 2, \dots, N$, then the
 484 question “what are the possible values of \mathbf{s} ?” needs to be addressed because the
 485 individual \mathbf{s}_i are *unknown*. As a technical matter, we will regard them as random
 486 effects and in order to apply standard methods of statistical inference we need to
 487 provide a distribution for these random effects. In the context of the point process
 488 model, the possible values of the point locations referred to as the “state-space” of
 489 the point process and this is some region or set of points which we will denote by
 490 \mathcal{S} . This is analogous to what is sometimes called the *observation window* for \mathbf{s} in
 491 the point process literature. The region \mathcal{S} serves as a prior distribution for \mathbf{s}_i (or,
 492 equivalently, the random effects distribution). In animal studies, as a description
 493 of where individuals that could be captured are located, it includes our study area,
 494 and should accommodate all individuals that could have been captured in the study
 495 area. In the practical application of SCR models, in most cases estimates of density
 496 will be relatively insensitive to choice of state-space which we discuss further in
 497 Chapt. ?? and elsewhere.

498 **1.5.5 Abundance and density**

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

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

504 A deficiency with some classical methods of “adjustment” is they attempted to
 505 prescribe something like a state-space - a “sampled area” - except absent any pre-

506 cise linkage of individuals with the state-space. SCR models formalize the linkage
507 between individuals and space and, in doing so, provide an explicit definition of N
508 associated with a well-defined spatial region, and hence density. That is, the pro-
509 vide a model in which N scales, as part of the model, with the size of the prescribed
510 state-space. In a sense, the whole idea of SCR models is that by defining a point
511 process and its state-space \mathcal{S} , this gives context and meaning to N which can be
512 estimated directly for that specific state-space. Thus, it is fixing \mathcal{S} that resolves
513 the problem of “unknown area” that we have previously discussed.

1.6 CHARACTERIZATION OF SCR MODELS

514 Formulation of capture-recapture models conditional on the latent point process is
515 the critical and unifying element of *all* SCR models. However, SCR models differ
516 in how the underlying process model is formulated, and its complexity. Most of the
517 development and application of SCR models has focused on their use to estimate
518 density and touting the fact that they resolve certain specific technical problems
519 related to the use of ordinary capture-recapture models. This is achieved with a sim-
520 ple process model being a basic point process of independently distributed points.
521 At the same time, there are models of CR data that focus exclusively on *movement*
522 modeling, or models with explicit dynamics (??). Conceptually, these are akin to
523 spatial versions of so-called Cormack-Jolly-Seber (CJS) models in the traditional
524 capture-recapture literature, except they involve explicit mathematical models of
525 movement based on diffusion or Brownian motion. Finally, there are now a very
526 small number of papers that focus on *both* movement and density simultaneously
527 (???) or population dynamics and density (?).

528 A key thing is that these models, whether focused just on density, or just on
529 movement, or both, are similar models in terms of the underlying concepts, the
530 latent structure, and the observation model. They differ primarily in terms of the
531 ecological focus. Understanding movement is an important topic in ecology, but
532 models that strictly focus on movement will be limited by two practical consid-
533 erations: (1) most capture-recapture data e.g., by camera trapping or whatever,
534 produces only a few observations of each individual (between 1-5 would be typi-
535 cal). So there is not too much information about complex movement models. (2)
536 Typically people have an interest in density of individuals and therefore we need
537 models that can be extrapolated from the sample to the unobserved part of the
538 population. That said, there are clearly some cases where more elaborate move-
539 ment models should come into play. If one has some telemetry data in addition to
540 SCR then there is additional information on fine-scale movements that should be
541 useful.

1.7 SUMMARY AND OUTLOOK

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

551 As we have shown already, SCR models are not simply an extension of a tech-
552 nique to resolve certain technical problems. Rather, they provide a coherent, flex-
553 ible framework for making ecological processes explicit in models of individual en-
554 counter history data, and for studying animal populations processes such as individ-
555 ual movement, resource selection, space usage, population dynamics, and density.
556 Historically, researchers studied these questions independently, using ostensibly un-
557 related study designs and statistical procedures. For example, resource selection
558 function (RSF) models for resource selection, state-space models for movement,
559 density using closed capture-recapture methods, and population dynamics with
560 various “open” capture-recapture models. SCR can bring all of these problems
561 together into a single unified framework for modeling and inference. Most impor-
562 tantly, spatial capture-recapture models promise the ability to integrate explicit
563 ecological theories directly into the models so that we can directly test hypotheses
564 about either space usage (e.g., Chapt. ??), landscape connectivity (Chapt. ??),
565 movement, or spatial distribution (Chapt. ??). We imagine that, in the near future,
566 SCR models will include point process models that allow for interactions among in-
567 dividuals such as inhibition or clustering (?). In the following chapters we develop a
568 comprehensive synthesis and extension of spatial capture-recapture models as they
569 presently exist, and we suggest areas of future development and needed research.

570
571

2

572

STATISTICAL MODELS AND SCR

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

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

590 The material in the beginning of this chapter is explained in numerous other
591 texts. Technical treatments that emphasize ecological problems are given by ?, ?,
592 and ?, to name just a few. A very accessible introduction to some of the topics
593 covered in this chapter is presented in Chapt. 3 of ?. With all these resources,
594 one might wonder why we bother rehashing these concepts here. Our motivation is
595 two-fold: first, we wish to develop this material using examples relevant to spatial
596 capture-recapture, and second, we find that most introductory texts are not accom-
597 panied by code that can be helpful to the novice. We therefore attempt to present
598 simple **R** code throughout this chapter so that those who struggle with equations

599 and mathematical notation can learn by doing. As mentioned in the Preface, we
600 rely on **R** because it provides tremendous flexibility for analyzing data and because
601 it is free. We do not, however, try to explain how to use **R** because there are so
602 many good references already, including ???.

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

2.1 RANDOM VARIABLES AND PROBABILITY DISTRIBUTIONS

609 2.1.1 Stochasticity in ecology

610 Few ecological processes can be described using purely deterministic models, and
611 thus we need a formal method for drawing conclusions from data while acknowl-
612 edging the stochastic nature of ecological systems. This is the role of statistical
613 inference, which is founded on the laws of probability. For our purposes, it suffices
614 to be familiar with a small number of concepts from probability theory—the most
615 important of which is the concept of a random variable, say X . A random variable
616 is a variable whose realized value is the outcome of some stochastic process. To
617 be more precise, a random variable is characterized by a function that describes
618 the probability of observing the value x . This probability function can be written
619 $\Pr(X = x|\theta)$ where θ is a parameter, or set of parameters of the function. If x is
620 discrete, e.g. binary or integer, then we call the probability function a probability
621 mass function (pmf). If x is continuous, the function is called a probability density
622 function (pdf).

623 To clarify the concept of a random variable, let X be the number of American
624 shad (*Alosa sapidissima*) caught after $K = 20$ casts at the shad hole on Deerfield
625 River in Massachusetts. Suppose that we had a good day and caught $x = 7$ fish.
626 If there were no random variation at play, we would say that the probability of
627 catching a fish, which we will call p , is $p = 7/20 = 0.35$, and we would always
628 expect to catch 7 shad after 20 casts. In other words, our deterministic model is
629 $x = 0.35 \times K$. In reality, however, we can be pretty sure that this deterministic
630 model would not be very good. Even if we knew for certain that $p \equiv 0.35$, we would
631 expect some variation in the number of fish caught on repeated fishing outings.
632 To describe this variation, we need a model that acknowledges uncertainty (i.e.,
633 stochasticity), and specifically we need a model that describes the probability of
634 catching x fish given K and p , $\Pr(X = x|K, p)$. Since x is discrete, not continuous,
635 we need a pmf. Before contemplating which pmf is most appropriate in this case,
636 we need to first mention a few issues related to notation.

637 Statisticians make things easier for themselves, and more complicated for ev-

638 eryone else, by using different notation for probability distributions. Sometimes
 639 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
 640 $\pi(X|K, p)$ or $\mathbb{P}(X|K, p)$ or $[X|K, p]$ or even just $[X]!$. Just remember that these
 641 expressions all have the same meaning—they are all probability distributions that
 642 tell us the probability of observing any possible realization of the random variable
 643 X . In this book, we will almost always use bracket notation (the last two examples
 644 above) to represent arbitrary probability distributions. Hence, from here on out,
 645 when you see $[X|K, p]$, just remember that this is equivalent to the more traditional
 646 expression $\Pr(X = x|K, p)$. In addition, from here on, to achieve a more concise
 647 presentation, we will no longer use uppercase letters to denote random variables
 648 and lowercase letters for realized values. Rather, we will define a random vari-
 649 able by some symbol (x , N , etc...) and let the context determine whether we are
 650 talking about the random variable itself, or realized values of it. In some limited
 651 cases, we will want upper- and lower-case letters to represent different variables.
 652 For example, we will often let N denote population size and n denote the number
 653 of individuals actually detected.

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

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

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

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

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

669 Note that once we choose a probability distribution, we have chosen a model. In
 670 our example, we have specified our model as $x \sim \text{Binomial}(K, p)$, and because we
 671 are assuming that the parameters are known, we can make probability statements
 672 about future outcomes. Continuing with our fish example, we might want to know
 673 the probability of catching $x = 7$ again after $K = 20$ casts on a future fishing
 674 outing, assuming that we know $p = 0.35$. Evaluating the binomial pmf returns a
 675 probability of approximately 0.18, as show using this bit of **R** code:

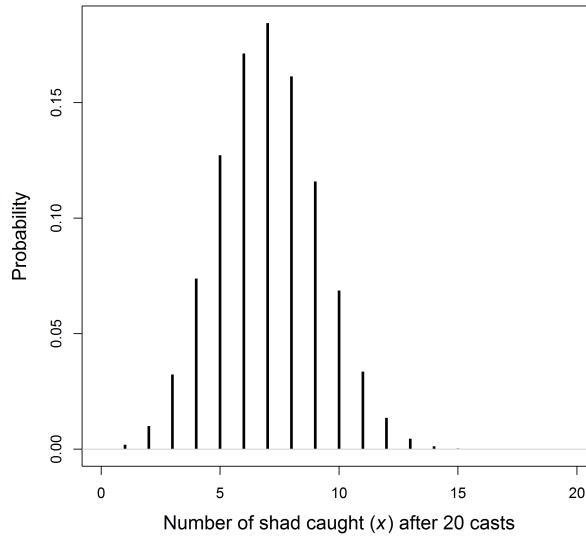


Figure 2.1. The binomial probability mass function with $N = 20$ and $p = 0.35$.

```
676 > dbinom(7, 20, 0.35)
677 [1] 0.1844012
```

678 By definition, the pmf allows us to evaluate the probability of observing any x given
 679 $K = 20$ and $p = 0.35$, thus the distribution of the random variable can be visualized
 680 by evaluating it for all values of x that have non-negligible probabilities, as can be
 681 easily done in R:

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

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

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

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

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

690 **2.1.2 Properties of probability distributions**

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

```
699 > integrate(dnorm, -Inf, Inf, mean=0, sd=1)$value
700 [1] 1
701 > sum(dbinom(0:5, size=5, p=0.1))
702 [1] 1
```

703 This requirement is important to remember when one develops a non-standard
 704 probability distribution. For example, in Chapt. ?? and ??, we work with resource
 705 selection functions whose probability density function is not one that is pre-defined
 706 in software packages such as **R** or **BUGS**.

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

```
714 > mean(rbinom(10000, 20, 0.3))
715 [1] 6.9865
```

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

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

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

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

```

726 > integrate(function(x) x*dnorm(x, 3, 1), -Inf, Inf)
727 3 with absolute error < 0.00033

```

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

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

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

```

736 > sum(dbinom(0:100, 20, 0.35)*0:100)
737 [1] 7

```

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

742 The ability to model the expected value of a random variable gets us very far,
 743 but we also need a model for the variance of the random variable. The variance
 744 describes the amount of variation around the expected value. Specifically, $\text{Var}(x) =$
 745 $\mathbb{E}((x - \mathbb{E}(x))^2)$. Clearly, if the variance is zero, the variable is not random as
 746 there is no uncertainty in its outcome. For some distributions, notably the normal
 747 distribution, the variance is a parameter to be estimated. Thus, in ordinary linear
 748 regression, we estimate both the expected value $\mu = \mathbb{E}(x)$, which may be a function
 749 of covariates, and the variance σ^2 , or similarly the residual standard error σ . For
 750 other distributions, the variance is not an explicit parameter to be estimated, and
 751 instead, the mean to variance ratio is fixed. In the case of the Poisson distribution,
 752 the mean is equal to the variance, $\mathbb{E}(x) = \text{Var}(x) = \lambda$. A similar situation is true
 753 for the binomial distribution—the variance is determined by the two parameters K
 754 and p , $\text{Var}(x) = Kp(1-p)$. In our earlier example with $K = 20$ and $p = 0.35$, the
 755 variance is 4.55. Toying around with these ideas using random number generators
 756 may be helpful. Here is some code to illustrate some of these basic concepts:

```

757 > 20*0.35*(1-0.35)                      # Exact variance, Var(x)
758 [1] 4.55
759 > x <- rbinom(100000, 20, 0.35)
760 > mean((x-mean(x))^2)                   # Monte Carlo approximation
761 [1] 4.545525

```

2.2 COMMON PROBABILITY DISTRIBUTIONS

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

770 **2.2.1 The binomial distribution**

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

```
783 > n <- rbinom(3, size=10, prob=0.5) # Generate 3 binomial outcomes
784 > n                                     # Display the 3 values
785 [1] 6 4 8
```

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

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

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

```
793 > n <- 5; N <- 10; p <- 0.5
794 > factorial(N)/(factorial(n)*factorial(N-n))*p^n*(1-p)^(N-n)
795 > exp(lgamma(N+1) - (lgamma(n+1) + lgamma(N-n+1)))*p^n*(1-p)^(N-n)
796 > choose(N, n)*p^n*(1-p)^(N-n)
```

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

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

801 Now that we know how to simulate binomial data and compute the probabilities
 802 of observing any particular outcome n , conditional on the parameters N and
 803 p , we can contemplate the relevance of the binomial distribution in spatial capture-
 804 recapture models. One important application of the binomial distribution is as a
 805 model encounter frequencies. Indeed, one of the most important encounter models
 806 in SCR will be referred to as the “binomial encounter model”, in which the number
 807 of times individual i is captured at “trap” j after K survey occasions is modeled as
 808 $y_{ij} \sim \text{Binomial}(K, p_{ij})$. Here, p_{ij} is the encounter probability determined, in part,
 809 by the distance between an animal’s activity center and the trap location. This
 810 binomial encounter model is described in detail in Sec. ???. Another important ap-
 811 plication of the binomial distribution is as a prior for the population size parameter
 812 in Bayesian analyses, as is discussed in Chapt. ??.

813 2.2.2 The Bernoulli distribution

814 Above, we showed 3 alternatives to `dbinom` for evaluating the binomial pmf. These
 815 three commands differed only in how they computed the binomial coefficient, which
 816 we needed because of the numerous ways in which we could observe $n = 5$ given
 817 $N = 10$. To conceptualize this, let y_i be a binary variable indicating if individual i
 818 was detected or not. Hence, given that 5 individuals were detected, the vector of
 819 individual detections could be something like $\mathbf{y} = (0, 0, 1, 1, 1, 1, 0, 0, 0)$, indicating
 820 that we detected individuals 3-7 but not 1-2 or 8-10. For $N = 10$ and $n = 5$,
 821 the binomial coefficient tells us that there are 252 possible vectors \mathbf{y} with 5 ones.
 822 However, when $N \equiv 1$, this term drops from the pmf and the result is the pmf for
 823 the Bernoulli distribution. That is, the Bernoulli distribution is simply the binomial
 824 distribution when $N \equiv 1$. Alternatively, we could say that the binomial distribution
 825 is the outcome of N iid Bernoulli trials. We use the standard abbreviation “iid”
 826 to mean *independent, identically distributed*.

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

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

counts of chestnut-sided warblers as

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

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

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

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

2.2.3 The multinomial and categorical distributions

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

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

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

870 Going back to our chestnut-sided warbler example, suppose the 10 individuals
 871 are marked and we make $K = 2$ visits to the site such that there are $2^K = 4$ pos-
 872 sible encounter histories: (11, 10, 01, 00), where, for example, “10” is the encounter
 873 history for an individual detected on the first visit but not the second. If $p = 1$,
 874 then the encounter history for each of the 10 individuals must be “11”. That is, we
 875 would detect each individual on both occasions. In this case, we the data would be:
 876 $\mathbf{h} = (10, 0, 0, 0)$, which indicates that all 10 warblers had the first encounter history.
 877 The corresponding cell probabilities would be $\boldsymbol{\pi} = (1, 0, 0, 0)$. What about the sit-
 878 uation where $p < 1$, e.g. $p = 0.3$? In this case, the probability of observing the
 879 capture history “11” (detected on both occasions) is $p \times p = 0.3 \times 0.3 = 0.09$. The
 880 probability of observing “10” is $p \times (1 - p) = 0.21$. Following this logic, the vector
 881 of cell probabilities is $\boldsymbol{\pi} = (0.09, 0.21, 0.21, 0.49)$. We can simulate data under this
 882 model as follows:

```
883 > caphist.probs <- c("11"=0.09, "10"=0.21, "01"=0.21, "00"=0.49)
884 > drop(rmultinom(1, 10, caphist.probs))
885 11 10 01 00
886 0 3 2 5
```

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

891 As in non-spatial capture-recapture studies, the multinomial distribution turns
 892 out to be very important in spatial capture-recapture studies. However, N is not
 893 defined as population size. Rather, we use the multinomial distribution when an
 894 individual can only be captured in a single trap during an occasion. Thus $N = 1$
 895 and the cell probabilities are the probabilities of being captured in each trap. A
 896 thorough discussion of this point can be found in Chapt. ???. Another application
 897 of the multinomial distribution in SCR models is discussed in Chapt. ?? where we
 898 discuss how to model the probability that an individual’s activity center is located
 899 in one of the cells of a raster defining the spatial region of interest.

900 Just as the Bernoulli distribution is the elemental form of the binomial distri-
 901 bution (being the case $N = 1$), the categorical distribution is essentially equivalent
 902 to the multinomial distribution with size parameter $N \equiv 1$. The only difference is
 903 that, rather than returning a vector with a single element equal to 1, it returns the
 904 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}

905 of a multinomial distribution with $N = 1$, then the categorical outcome would be
 906 3 because the 1 is located in third position in the vector. Thus, in spatial capture-
 907 recapture models, we might use either the multinomial distribution with $N = 1$
 908 or the categorical distribution. The various **BUGS** engines describe the categor-
 909 ical distribution by the declaration **dcat** and, in **R**, we can simulate categorical
 910 outcomes using the function **sample** or as so:

```
911 > which(rmultinom(1, 1, c(0.1, 0.7, 0.2)) == 1)
912 [1] 2
```

913 2.2.4 The Poisson distribution

914 The Poisson distribution is the canonical model for count data in ecology. More
 915 generally, the Poisson distribution is a model for random variables taking on non-
 916 negative, integer values. Although it is a simple model having just one parameter,
 917 $\lambda = \mathbb{E}(x) = \text{Var}(x)$, its applications are highly diverse, including as a model of
 918 spatial variation in abundance or as a model for the frequency of behaviors over
 919 time. Just as logistic regression is the standard generalized linear model (GLM)
 920 used to model binary data, Poisson regression is the default GLM for modeling
 921 count data and variation in λ .

922 The Poisson distribution is related to both the binomial and multinomial distri-
 923 butions, and the following three bits of trivia are occasionally worth knowing. First,
 924 it is the limit of the binomial distribution as $N \rightarrow \infty$ and $p \rightarrow 0$, which means that
 925 for high values of N and low values of p , $\text{Poisson}(N \times p)$ is approximately equal
 926 to $\text{Binomial}(N, p)$. Second, if $\{n_1 \sim \text{Poisson}(\lambda_1), \dots, n_K \sim \text{Poisson}(\lambda_K)\}$ then the
 927 vector of counts is multinomial, $\{n_1, \dots, n_K\} \sim \text{Multinomial}(\sum_k n_k, \{\frac{\lambda_1}{\sum_k \lambda_k}, \dots, \frac{\lambda_K}{\sum_k \lambda_k}\})$.
 928 Third, the sum of two Poisson random variables $x_1 \sim \text{Poisson}(\lambda_1)$ and $x_2 \sim$
 929 $\text{Poisson}(\lambda_2)$ is also Poisson: $x_1 + x_2 \sim \text{Poisson}(\lambda_1 + \lambda_2)$.

930 The Poisson distribution has two important uses in spatial capture-recapture
 931 models: (1) as a prior distribution for the population size parameter N , and (2) as a
 932 model for the frequency of captures in a trap. In the first context, the Poisson prior
 933 for N results in a Poisson point process for the location of the N activity centers
 934 in the region of interest. This topic is discussed in Chapt. ?? and Chapt. ???. The
 935 second use of the Poisson distribution in spatial capture-recapture is to describe
 936 data from sampling methods in which an individual can be detected multiple times
 937 at a trap during a single occasion. For example, in camera trapping studies we
 938 might obtain multiple pictures of the same individual at a trap during a single
 939 sampling occasion. Thus, λ in this case would be defined as the expected number
 940 of detections or captures per occasion.

941 2.2.5 The uniform distribution

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

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

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

```
959 D <- 100      # points per unit area
960 A <- 1        # Area of unit square
961 N <- rpois(1, D*A)
962 plot(s <- cbind(runif(N), runif(N)))
```

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

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

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

970 2.2.6 Other distributions

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

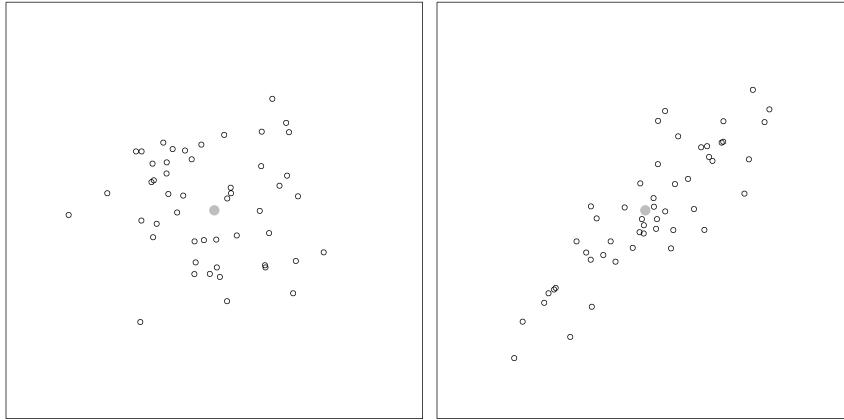


Figure 2.2. Two realized point patterns from the bivariate normal distribution.

981 a cautionary note, is provided in Sec. ???. Be aware that although the normal
 982 distribution is typically parameterized in terms of the variance parameter σ^2 , in
 983 the **BUGS** language, the inverse of the variance, or precision, is used instead,
 984 $\tau = 1/\sigma^2$. In **R**, the `dnorm` function requires the standard deviation σ , rather than
 985 the variance σ^2 .

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

```

995 library(mvtnorm)
996 set.seed(3)
997 mu <- c(0,0)
998 Sigma <- matrix(c(1, .9, .9, 1), 2, 2)
999 X1 <- cbind(rnorm(50, mu[1], Sigma[1,1]), # No correlation (rho=0)
1000           rnorm(50, mu[2], Sigma[2,2]))
1001 X2 <- rmvnorm(50, mu, Sigma)           # rho=0.9

```

1002 Fig. 2.2 shows the simulated points.

1003 Several of the parameters in capture-recapture models do not have infinite sup-
 1004 port, but instead are probabilities restricted to the range $[0, 1]$, or are positive

1005 valued living between zero and ∞ . The beta distribution is the standard prior
 1006 used for probabilities because it can be used to express either a lack of knowledge
 1007 or very precise knowledge about a parameter. For example, a Beta(1, 1) distribu-
 1008 tion is equivalent to a Uniform(0, 1) distribution. However, unlike the uniform
 1009 distribution, the beta distribution can be used as an informative prior; for exam-
 1010 ple if published estimates of detection probability exist we can choose parameters
 1011 of the beta distribution to reflect that. To gain some familiarity with the beta
 1012 distribution, execute the following R commands:

```
1013 curve(dbeta(x, 1, 1), col="black", ylim=c(0,5))
1014 curve(dbeta(x, 10, 10), col="blue", add=TRUE)
1015 curve(dbeta(x, 10, 20), col="darkgreen", add=TRUE)
```

1016 Other parameters in SCR models are continuous but positive-valued and can be
 1017 modeled using the gamma distribution. As with the beta distribution, the gamma
 1018 distribution is typically favored over the uniform distribution when one is interested
 1019 in using an informative prior. It is also frequently used as a vague prior for the
 1020 inverse of variance parameters, but it is wise to compare this prior to a uniform to
 1021 assess its influence on the posterior.

2.3 STATISTICAL INFERENCE AND PARAMETER ESTIMATION

1022 If the parameters of a statistical model were known with absolute certainty, then it
 1023 would be possible to use pdfs and pmfs to make direct probability statements about
 1024 unknowns such as future outcomes. However, we almost never know the actual
 1025 values of parameters, and instead we have to estimate them from observations
 1026 (i.e., data). Our inferences must then acknowledge the uncertainty associated with
 1027 our imperfect knowledge of the parameters. Doing so is most often accomplished
 1028 using one of two approaches: classical (frequentist) inference or Bayesian inference.
 1029 These two modes of inference regard the uncertainty about parameters in entirely
 1030 different ways. In the next chapter, we will review some of the important concepts
 1031 in Bayesian inference, so here, we will focus on the frequentist perspective.

1032 Suppose we count oak trees at J sites, and the resulting data $\{y_1, \dots, y_J\}$ can
 1033 be assumed to be *iid* outcomes from some distribution, such as the Poisson with
 1034 unknown parameter λ . We want to estimate this parameter. In classical inference,
 1035 the only uncertainty about λ is that attributable to sampling. For instance, we can
 1036 imagine repeatedly sampling the population (sites in this example) and obtaining
 1037 sample-specific estimates of λ . Typically, we entertain the idea that there are an
 1038 infinite number of possible samples and so we could obtain an infinite number of
 1039 estimates: $\{\hat{\lambda}_1, \hat{\lambda}_2, \dots, \hat{\lambda}_\infty\}$. If these estimates are produced using the method
 1040 of maximum likelihood, and as n tends to infinity, the distribution of estimates,
 1041 called the sampling distribution, will be normally distributed with $E(\hat{\lambda}) = \lambda$. The
 1042 standard deviation of the sampling distribution is called the standard error, which
 1043 can also be estimated as part of the maximum likelihood procedure. Of course, we

1044 almost always have just a single sample of data, and hence a single $\hat{\lambda}$ and a single
 1045 estimate of the standard error. However, under the assumption of a normally
 1046 distributed sampling distribution, we can construct a confidence interval that will
 1047 include the true value of λ with coverage probability $1 - \alpha$, where α is a prescribed
 1048 value like 0.05. An important point is that there is no uncertainty associated with
 1049 the actual parameter—it is regarded as a fixed value, and hence probability is only
 1050 used to characterize the estimator via its sampling distribution.

1051 Maximum likelihood is heuristically a method of finding the most “likely” value
 1052 of λ , given the observed data, and of characterizing the variance of the sampling dis-
 1053 tribution. Of course, it also applies to cases where the observations are multivariate,
 1054 or the probability distribution is a function of multiple parameters. Endless num-
 1055 bers of textbooks and online resources are available for those interested in a detailed
 1056 explanation of maximum likelihood. For our purposes, we wish to keep it simple
 1057 and focus on *how* to do it. The first step is to define the likelihood function, which
 1058 is the joint distribution of the data regarded as a function of the parameter(s). If
 1059 the joint distribution of the observations is denoted by $[y_1, y_2, \dots, y_n | \lambda]$, we usually
 1060 denote the likelihood by flipping the arguments: $\mathcal{L}(\lambda | \mathbf{y}) = [\lambda | y_1, y_2, \dots, y_n]$.

1061 If the observations are *iid*, the likelihood simplifies to

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

1062 where $[y_i | \lambda]$ is a probability distribution, like those discussed in the previous sec-
 1063 tions. For example, if y_i is Poisson distributed, then $[y_i | \lambda] = \text{Poisson}(\lambda) = \frac{\lambda^{y_i} e^{-\lambda}}{y_i!}$.
 1064 Although likelihoods are typically shown on the natural scale, we almost always
 1065 maximize the logarithm of the likelihood to avoid computational problems that
 1066 arise when multiplying very small probabilities. Thus, we rewrite Eq. 2.3.1 as

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

1067 Here is some simple **R** code to simulate independent Poisson outcomes and esti-
 1068 mate λ (as though we did not know it) using the method of maximum likelihood.
 1069 Actually, we will minimize the negative log-likelihood because it is equivalent and
 1070 is the default for **R**’s optimizers like `optim` and `nls`.

```
1071 > lambda <- 3                      # Actual parameter value
1072 > y1 <- rpois(100, lambda)        # Realized values (data)
1073 > negLogLike1 <- function(par) -sum(dpois(y1, par, log=TRUE))
1074 > starting.value <- c('lambda'=1)
1075 > optim(starting.value, negLogLike1)$par # MLE
1076   lambda
1077 3.039844
```

1078 Explicitly maximizing the likelihood, numerically, isn't actually necessary here because
 1079 the MLE of λ is given by the mean of the observations. A more interesting
 1080 example is when there are covariates of λ . For example, suppose λ is a function of
 1081 elevation and vegetation height according to: $\log(\lambda_i) = \beta_0 + \beta_1 ELEV_i + \beta_2 VEGHT_i$.
 1082 This is a standard Poisson regression problem, with likelihood:

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

1083 This likelihood is almost identical to the previous one except that λ is now a
 1084 function, and so we need to estimate the parameters of the function, i.e. the β 's.
 1085 Some code to fit this model to simulated data is shown here:

```
1086 > nsites <- 100
1087 > elevation <- rnorm(100)
1088 > veght <- rnorm(100)
1089 > beta0 <- 1
1090 > beta1 <- -1
1091 > beta2 <- 0
1092 > lambda <- exp(beta0 + beta1*elevation + beta2*vegght)
1093 > y2 <- rpois(nsites, lambda)
1094 > negLogLike2 <- function(pars) {
1095   +   beta0 <- pars[1]
1096   +   beta1 <- pars[2]
1097   +   beta2 <- pars[3]
1098   +   lambda <- exp(beta0 + beta1*elevation + beta2*vegght)
1099   +   -sum(dpois(y2, lambda, log=TRUE))
1100   +
1101 > starting.values <- c('beta0'=0, 'beta1'=0, 'beta2'=0)
1102 > optim(starting.values, negLogLike2)$par
1103   beta0      beta1      beta2
1104 0.98457756 -1.03025173 -0.01218292
```

1105 We see that the maximum likelihood estimates (MLEs) are very close to the true
 1106 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:

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

1107 This is an example of a mixed effects model or a hierarchical model. How do
 1108 we estimate the parameters of a model that includes random effects? Earlier the
 1109 likelihood function was written as the product of probabilities determined by a
 1110 single pmf or pdf, $[y|\lambda]$, but now we have an additional random variable, and we
 1111 are forced to think about conditional relationships, because y depends upon β_i and
 1112 β_i depends upon other parameters, specifically μ and σ^2 . This type of conditional
 1113 dependence among parameters is the essence of hierarchical models, and statistical
 1114 analysis of hierarchical models requires that we discuss joint distributions, marginal
 1115 distributions and conditional distributions. These concepts will be used extensively
 1116 in Chapt. ?? where we demonstrate how to estimate parameters of hierarchical
 1117 models using maximum likelihood.

2.4 JOINT, MARGINAL, AND CONDITIONAL DISTRIBUTIONS

1118 So far we have restricted our attention to situations in which we wish to make
 1119 inference about a single random variable. However, in ecology, we often are inter-
 1120 ested in multiple random variables and how they are related. Let Y be a random
 1121 variable that may or may not be independent of X (here again we will distinguish
 1122 between random variables and realized values for conceptual clarity). Inference
 1123 about these two random variables can be made using the joint, marginal, or condi-
 1124 tional distributions—or, we may make use of all of them depending on the question
 1125 being asked. In the case of discrete random variables, the joint distribution is the
 1126 probability that X takes on the value x and that Y takes on the value y , which
 1127 is written $[X = x, Y = y]$. To clarify this concept, let's go back to our original
 1128 example where X was the number of fish caught after 20 casts, which we said
 1129 was an *iid* binomial random variable. Now, let's suppose that X depends on the
 1130 random variable Y , which is the number of other fisherman at the hole. Specifi-
 1131 cally, let's say that the probability of catching a fish p is related to Y according
 1132 to $\text{logit}(p) = -0.6 + -2y$. Furthermore, let's make the intuitive assumption that
 1133 the number of fishermen at the hole is a Poisson random variable with mean 0.6,
 1134 i.e. $Y \sim \text{Poisson}(0.6)$. Our model is now fully specified, and so we can answer the
 1135 question: “what is the probability of catching x fish and of there being y fishermen
 1136 at the hole”. This joint distribution is given by the product of the binomial pmf
 1137 (with p determined by y) and the Poisson pmf with $\lambda = 0.6$. The following R code
 1138 creates the joint distribution.

```
1139 > X <- 0:20 # All possible values of X
1140 > Y <- 0:10 # All possible values of Y
1141 > lambda <- 0.6
```

```

1142 > p <- plogis(-0.62 + -2*Y) # p as function of Y
1143 > round(p,2)
1144 [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
1145 > joint <- matrix(NA, length(X), length(Y))
1146 > rownames(joint) <- paste("X=", X, sep="")
1147 > colnames(joint) <- paste("Y=", Y, sep="")
1148 >
1149 > # Joint distribution [X,Y]
1150 > for(i in 1:length(Y)) {
1151 +   joint[,i] <- dbinom(X, 20, p[i]) * dpois(Y[i], lambda)
1152 + }
1153 > round(joint,2)
1154   Y=0  Y=1  Y=2  Y=3  Y=4  Y=5  Y=6  Y=7  Y=8  Y=9  Y=10
1155 X=0  0.00 0.08 0.08 0.02  0  0  0  0  0  0  0
1156 X=1  0.00 0.12 0.02 0.00  0  0  0  0  0  0  0
1157 X=2  0.01 0.08 0.00 0.00  0  0  0  0  0  0  0
1158 X=3  0.02 0.04 0.00 0.00  0  0  0  0  0  0  0
1159 X=4  0.04 0.01 0.00 0.00  0  0  0  0  0  0  0
1160 X=5  0.07 0.00 0.00 0.00  0  0  0  0  0  0  0
1161 X=6  0.09 0.00 0.00 0.00  0  0  0  0  0  0  0
1162 X=7  0.10 0.00 0.00 0.00  0  0  0  0  0  0  0
1163 X=8  0.09 0.00 0.00 0.00  0  0  0  0  0  0  0
1164 X=9  0.06 0.00 0.00 0.00  0  0  0  0  0  0  0
1165 X=10 0.04 0.00 0.00 0.00  0  0  0  0  0  0  0
1166 X=11 0.02 0.00 0.00 0.00  0  0  0  0  0  0  0
1167 X=12 0.01 0.00 0.00 0.00  0  0  0  0  0  0  0
1168 X=13 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
1169 X=14 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
1170 X=15 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
1171 X=16 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
1172 X=17 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
1173 X=18 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
1174 X=19 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0
1175 X=20 0.00 0.00 0.00 0.00  0  0  0  0  0  0  0

```

1176 This matrix tells us the probability of all possible combinations of x and y , and
 1177 we see that the most likely value is $(X = 1, Y = 1)$, i.e. we will catch 1 fish and
 1178 there will be 1 other fisherman. This matrix also demonstrates the law of total
 1179 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.???. Here is some **R** code to compute the marginal distribution of X , i.e. the probability of catching $X = x$ fish:

```
1185 > margX <- rowSums(joint)
1186 > round(margX, 2)
1187   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
1188 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
1189 X=15  X=16  X=17  X=18  X=19  X=20
1190 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 .

```
1193 > XgivenY <- joint/matrix(margY, nrow(joint), ncol(joint), byrow=TRUE)
1194 > round(XgivenY, 2)
1195   Y=0  Y=1  Y=2  Y=3  Y=4  Y=5  Y=6  Y=7  Y=8  Y=9  Y=10
1196   X=0  0.00 0.25 0.82 0.97  1  1  1  1  1  1  1
1197   X=1  0.00 0.36 0.16 0.03  0  0  0  0  0  0  0
1198   X=2  0.01 0.25 0.02 0.00  0  0  0  0  0  0  0
1199   X=3  0.03 0.11 0.00 0.00  0  0  0  0  0  0  0
1200   X=4  0.07 0.03 0.00 0.00  0  0  0  0  0  0  0
1201   X=5  0.13 0.01 0.00 0.00  0  0  0  0  0  0  0
1202   X=6  0.17 0.00 0.00 0.00  0  0  0  0  0  0  0
1203   X=7  0.18 0.00 0.00 0.00  0  0  0  0  0  0  0
```

```

1206 X=8 0.16 0.00 0.00 0.00 0 0 0 0 0 0 0
1207 X=9 0.12 0.00 0.00 0.00 0 0 0 0 0 0 0
1208 X=10 0.07 0.00 0.00 0.00 0 0 0 0 0 0 0
1209 X=11 0.03 0.00 0.00 0.00 0 0 0 0 0 0 0
1210 X=12 0.01 0.00 0.00 0.00 0 0 0 0 0 0 0
1211 X=13 0.00 0.00 0.00 0.00 0 0 0 0 0 0 0
1212 X=14 0.00 0.00 0.00 0.00 0 0 0 0 0 0 0
1213 X=15 0.00 0.00 0.00 0.00 0 0 0 0 0 0 0
1214 X=16 0.00 0.00 0.00 0.00 0 0 0 0 0 0 0
1215 X=17 0.00 0.00 0.00 0.00 0 0 0 0 0 0 0
1216 X=18 0.00 0.00 0.00 0.00 0 0 0 0 0 0 0
1217 X=19 0.00 0.00 0.00 0.00 0 0 0 0 0 0 0
1218 X=20 0.00 0.00 0.00 0.00 0 0 0 0 0 0 0

```

1219 Note that we have 11 probability distributions for X , one for each possible value
 1220 of Y , and each pmf sums to unity as it should. Note also that if you show up at
 1221 the hole and there are > 2 fisherman, your chance of catching a fish is very low.
 1222 Go home. These concepts are explained in more detail in other texts such as ?, ?,
 1223 and ?, but hopefully, the code shown here complements the equations and makes
 1224 it easier for 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:

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

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

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

1225 From here on out, when you see such notation, you should immediately grasp
 1226 the fact that Y is a random variable independent of X , but X depends upon
 1227 Y through p . Now you have the tools to make probability statements about the
 1228 random variables in this system. The one caveat faced in reality is that we typically
 1229 do not know the values of the parameters, and instead we have to estimate them.
 1230 Maximum likelihood methods for hierarchical models are covered in Chapt. ??

2.5 HIERARCHICAL MODELS AND INFERENCE

1231 The term hierarchical modeling (or hierarchical model) has become something of
 1232 a buzzword over the last decade with hundreds of papers published in ecological
 1233 journals using that term. So then, what exactly is a hierarchical model, anyhow?
 1234 Obviously, this term stems from the root “hierarchy” which means:

1235 **Definition:** *hierarchy* (noun) – a series of ordered groupings of people or things
 1236 within a system;

1237 In the case of a hierarchical model (hierarchical being the adjective form of hi-
 1238 erarchy), the “things” are probability distributions, and they are ordered according
 1239 to their conditional probability structure. Thus, a hierarchical model is *an ordered*
 1240 *series of models, ordered by their conditional probability structure.*

1241 A canonical hierarchical model in ecology is this elemental model of species
 1242 occurrence or distribution (???:)

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

1243

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

1244 where y_i = observation of presence/absence at a site i and z_i = occurrence status
 1245 ($z_i = 1$ if a species occurs at site i and $z_i = 0$ if not). Note that if $p = 1$, then we
 1246 would perfectly observe z and the model would no longer be hierarchical—it would
 1247 be a simple logistic regression model. Note also that this hierarchical model has an
 1248 important conceptual distinction between other types of classical multi-level models
 1249 such as repeated measures on subjects, in that z_i is an actual state of nature. In
 1250 that sense, z is a random variable that is the outcome of a “real” process. ? used
 1251 the term *explicit* hierarchical model to describe this type of model to distinguish
 1252 from hierarchical models (*implicit* hierarchical models) where the latent variables
 1253 don’t correspond to an actual state of nature—but rather just soak up variation
 1254 that is unmodeled by explicit elements of the model. At best, latent variables in
 1255 such models are surrogates for something of ecological relevance (“time effects”,
 1256 “space effects” etc.).

1257 With these examples, we expand on our definition of a hierarchical model as we
 1258 will use it in this book:

1259 **Definition: Hierarchical Model:** A model with explicit component models that de-
 1260 scribe variation in the data due to (spatial/temporal) variation in *ecological process*,
 1261 and due to *imperfect observation* of the process.

1262 Most models considered in this book describe the encounter of individuals con-
 1263 ditional on the “activity center” of the individual, which is a latent variable (i.e.,
 1264 unobserved random effect). The definition of an activity center will be context-
 1265 dependent as discussed in Chapt. ??, but often it can be thought of as an indi-
 1266 vidual’s home range center. The collection of these latent variables represents the
 1267 outcome of an ecological process describing how individuals distribute themselves
 1268 over the landscape. Moreover, how individuals are encountered in traps is, in some
 1269 cases, the result of a model governing movement. As such, these models are ex-
 1270 amples of hierarchical models that contain formal model components representing
 1271 both ecological process and also the observation of that process. That is, they are
 1272 explicit hierarchical models (?) as opposed to implicit hierarchical models.

2.6 CHARACTERIZATION OF SCR MODELS

1273 For the purposes of this book, an SCR model is any “individual encounter model”
 1274 (not just “capture-recapture”!) where auxiliary spatial information is also obtained.
 1275 To be more precise we could as well use the term “spatial capture and/or recap-
 1276 ture” but that is slightly unwieldy and, besides, it also abbreviates to SCR. The
 1277 class of SCR models includes traditional capture-recapture models with auxiliary
 1278 spatial information and even some models that do not even require “recapture”
 1279 (e.g., distance sampling). There is even a class of models (Chapt. ??) which don’t
 1280 require capture or unique identification of individuals.

1281 Conceptually, SCR models involve a collection of random variables, \mathbf{s} , \mathbf{u} and
 1282 y where \mathbf{s} is the activity center, or home range center, \mathbf{u} is the location of the
 1283 individual at the time of sampling, which we may think of as a realization from some
 1284 movement model, and y is the “response variable”—what the observer records. For
 1285 example, $y = 1$ means “detected” and $y = 0$ means “not detected”, but many other
 1286 types of responses are possible (Chapt. ??). A broad class of models for estimating
 1287 density are unified by a hierarchical model involving explicit models for animal
 1288 activity centers \mathbf{s} , movement outcomes \mathbf{u} , and encounter data y . In some cases, we
 1289 don’t observe y but rather summaries of y , say $n(y)$, yet it might be convenient
 1290 in such cases to retain an explicit focus on y in terms of model construction. We
 1291 thus introduce a sequence of models—a hierarchical model—to relate these random
 1292 variables, which can be written as

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

1293 Every model we talk about in this book has a subset of these components although
 1294 we never fit the full model because we have not encountered a situation requiring
 1295 that we do so. However, a detailed description of this model and its various com-
 1296 ponents is the subject of this book, and we will not pretend to condense hundreds
 1297 of pages of material into the next few paragraphs. However, we give a cursory
 1298 overview here to whet the appetite and provide some indication of where we are
 1299 going. Don’t worry if some of this material doesn’t sink in just yet—we will walk
 1300 through it slowly in the subsequent chapters.

1301 Let’s begin with the model $[\mathbf{s}]$ that describes the distribution of the activity
 1302 centers of each animal in the spatial region \mathcal{S} (the state-space as we called it previ-
 1303 ously). As will be explained in Chapt. ?? and Chapt. ??, $[\mathbf{s}]$ defines a spatial point
 1304 process, which may be inhomogeneous if there exists spatial variation in density, or
 1305 it may be homogeneous if density is constant throughout \mathcal{S} . In the later case, we can
 1306 write $[\mathbf{s}] = \text{Uniform}(\mathcal{S})$, which is to say that the N activity centers are uniformly
 1307 distributed in the polygon \mathcal{S} . A point process is also a model for the number of indi-
 1308 viduals in the population N . So we could write $[\mathbf{s}|\mu]$ where μ is an intensity param-
 1309 eter defined as the number of points per unit area. In other words, μ is population
 1310 density, and we often model population size as either $N \sim \text{Poisson}(\mu A(\mathcal{S}))$, where
 1311 $A(\mathcal{S})$ is the area of the state-space; or, $N \sim \text{Binomial}(M, \psi)$ where $\psi = \mu A(\mathcal{S})/M$

1312 and M is some large integer used simply as a convenience measure when conducting
 1313 Bayesian analysis. As it turns out, there is very little practical difference in the
 1314 Poisson prior versus a binomial models for N (Chapt. ??).

1315 The model $[\mathbf{u}|\mathbf{s}]$ describes the locations of animals conditional on their activity
 1316 center. In the original formulation of SCR models (?), this model component was
 1317 intentionally ignored. Indeed when movement is not of direct interest, or when \mathbf{s} is
 1318 defined in a way not related to a home range center, it may be preferable to ignore
 1319 this model component (?). In other cases, we might use an explicit model, such as
 1320 the bivariate normal model (?).

1321 The third component of the model, $[y|\mathbf{u}]$, describes how the observed data—the
 1322 so-called capture-histories—arise conditional on the locations of animals. However,
 1323 as mentioned previously, most SCR models do not contain a movement model, and
 1324 thus, we typically entertain the model $[y|\mathbf{s}]$ instead of $[y|\mathbf{u}]$. This encounter model
 1325 generally has at least two parameters, say p_0 and σ , describing the probability of
 1326 capturing or detecting an individual given the distance between \mathbf{s} and the trap.
 1327 The most basic model is often called the half-normal model, although we typically
 1328 refer to it as the Gaussian model since, in two-dimensional space, it is the kernel
 1329 of a bivariate normal distribution. The model is $p_{ij} = p_0 \exp(-\|\mathbf{x}_j - \mathbf{s}_i\|/(2\sigma^2))$
 1330 where p_0 is the capture probability when the activity center occurs at the trap
 1331 location \mathbf{x}_j , and σ is a spatial scale parameter determining how rapidly capture
 1332 probability declines with distance. One common design leads to the model $[y_{ij}|\mathbf{s}_i] =$
 1333 Bernoulli(p_{ij}). Chapt. ?? and Chapt. ?? describe many other possible encounter
 1334 models.

1335 When individuals are marked by biologists or have natural markings permitting
 1336 individual recognition, y_{ij} is the observed data. However, some or all of the
 1337 individuals cannot be uniquely identified, then we cannot record this individual-
 1338 specific encounter history data. Instead, the data might be simply the number of
 1339 detections at a trap or perhaps binary detection/non-detection data at each trap on
 1340 each survey occasion. We call this reduced information data $n(y)$, and Chapt. ??
 1341 and Chapt. ?? describe models for $[n(y)|y]$ that still allow for density estimation.
 1342 The basic strategy is to view y as “missing data” and to use the spatial correlation
 1343 in the counts, or other sources of information, to provide information about these
 1344 latent encounter histories.

1345 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:

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

1345 These “assumptions” are statistical statements of three basic hypotheses that (1)

1346 population size N is Poisson distributed (2) activity centers are uniformly dis-
 1347 tributed in two-dimensional space, and (3) capture probability is a function of the
 1348 distance between the activity and the trap. Each of these model components can
 1349 be modified as needed to match specific hypotheses, study designs, and data struc-
 1350 tures. For example, spatial variation in abundance or density can be easily modeled
 1351 as a function of habitat covariates (Chapt. ??).

1352 We realize that many the model description in Eq. 2.6.2 may not be self-evident
 1353 to some ecologists. However, it is absolutely essential that one can understand
 1354 such a model description—not just for being able to read this book, but also for
 1355 understanding any statistical model in ecology. One of the best ways of familiarizing
 1356 oneself with this notation is to translate it into **R** code that simulates outcomes
 1357 from the model. The following code is an example.

```
1358 set.seed(36372)
1359 Area <- 1                                # area of state-space (unit square)
1360 x <- cbind(rep(seq(.1,.9,.2), each=5),    # trap locations
1361             rep(seq(.1,.9,.2), times=5))
1362 p0 <- 0.3                                 # baseline capture probability
1363 sigma <- 0.05                             # Gaussian scale parameter
1364 mu <- 50                                  # population density
1365 N <- rpois(1, mu*Area)                   # population size
1366 s <- cbind(runif(N, 0, 1),                # activity centers in unit square
1367               runif(N, 0, 1))
1368 K <- 5
1369 y <- matrix(NA, N, nrow(x))              # capture data
1370 for(i in 1:N) {
1371   d.ij <- sqrt((x[,1] - s[i,1])^2 +      # distance between x and s[i]
1372                 (x[,2] - s[i,2])^2)
1373   p.ij <- p0*exp(-d.ij^2 / (2*sigma^2)) # capture probability
1374   y[i,] <- rbinom(nrow(x), K, p.ij)       # capture history for animal i
1375 }
```

1376 Fig. 2.3 shows the results of this simulation from a basic, yet very useful, SCR
 1377 model.

1378 Having briefly explained each of the model components in Eq. 2.6.1, and hav-
 1379 ing shown how a subset of these components results in a basic SCR model, we
 1380 can now discuss other relevant arrangements. Examples include: (1) Classical dis-
 1381 tance sampling (??), (2) Spatial capture-recapture models with fixed arrays of traps
 1382 (??????), and (3) Search-encounter models (??). We will now elaborate on some
 1383 of these distinctions.

- 1384 1. **Distance sampling.** The last 2 stages of the hierarchy are confounded
 1385 (implicitly) and so analysis is based on the model $[y|\mathbf{u}][\mathbf{u}]$. The “process
 1386 model” is that of “uniformity”: $\mathbf{u} \sim \text{Uniform}(\mathcal{S})$.
- 1387 2. **Spatial capture-recapture model with a fixed array of traps.** SCR
 1388 models appear to have little in common with distance sampling because ob-

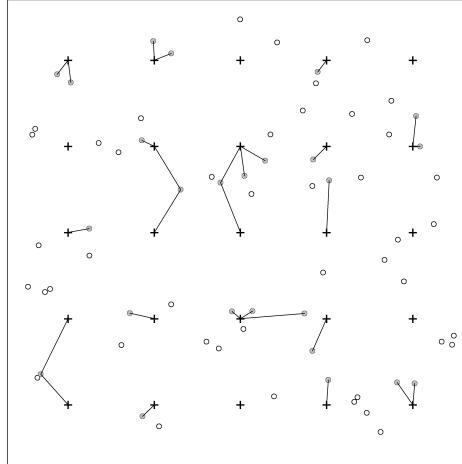


Figure 2.3. Population of $N = 69$ home-range centers (s , circles) and 25 trap locations (x , crosses). Lines connect activity centers to the traps where the individuals were detected. As in many SCR models, movement outcomes (u) are ignored.

1389 servations are made only at a pre-defined set of discrete locations—where
 1390 traps are placed. However, the models are closely related in terms of our
 1391 hierarchical representation above. In SCR models based on fixed arrays, we
 1392 cannot estimate both $\Pr(y = 1|u)$ and $\Pr(u|s)$ —the probability that an in-
 1393 dividual “moves to u ” cannot be separated from the probability that it is
 1394 detected given that it moves to u , because of the fact that the observation
 1395 locations are fixed by design. Formally, such SCR models confound $[y|u]$
 1396 with $[u|s]$ so that the observation model arises as:

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

1397 This confounding happens because SCR sampling is spatially biased—restricted
 1398 to a fixed pre-determined set of locations. Conversely, distance sampling
 1399 confounds $[u|s][s]$ because, essentially, there is only a single realization of the
 1400 encounter process. It is probably reasonable to assume that $\Pr(y = 1|u) = 1$
 1401 or at least it is locally constant for most devices (e.g., cameras, etc..), and
 1402 thus the detection model will have the interpretation in terms of movement
 1403 (see Chapt. ?? and ??).

1404 3. **Search-encounter models.** What we call “search-encounter” models (??)
 1405 are kind of a hybrid model combining features of SCR models and fea-
 1406 tures of distance sampling. Like distance sampling they allow for encoun-

1407 ters in continuous space which provide direct observations from $[\mathbf{u}|\mathbf{s}]$. Thus,
1408 the hierarchical model is fully identified. These models are described in
1409 Chapt. chapt.search-encounter.

2.7 SUMMARY AND OUTLOOK

1410 Spatial capture-recapture models are hierarchical models, and hierarchical models
1411 are models of multiple random variables that are conditionally related. It is there-
1412 fore important that the basic rules of modeling random variables are understood,
1413 and we hope that this chapter has made some of the basic concepts accessible to
1414 ecologists with rudimentary background in statistics. If some of this material still
1415 seems difficult to grasp, we recommend working with the provided **R** code, which
1416 is perhaps the best way of making the equations more tangible.

1417 In some respects, it is possible to understand the jist of SCR without knowing
1418 anything about marginal and conditional relationships. One can always fit models
1419 using canned software and interpret the output without understanding the guts of
1420 the model or the details of the estimation process. For some applied ecologists,
1421 this may be perfectly fine, and this book is meant to be useful for both statistical
1422 novices and ecologists with more advanced quantitative skills. In most chapters, we
1423 begin with a basic conceptual discussion, then we explain the technical details that
1424 require an understanding of the concepts in this chapter, and finally we end with
1425 one or more worked examples. For those not interested in the technical details,
1426 we recommend focusing on the chapter introductions and the examples. However,
1427 taking the time to understand the concepts presented in this chapter can only
1428 increase one's ability to tackle the unique and complex problems that often present
1429 themselves when modeling spatial and temporal aspects of population dynamics.

1430
1431

1432

3

GLMS AND BAYESIAN ANALYSIS

1433 A major theme of this book is that spatial capture-recapture models are, for the
1434 most part, just generalized linear models (GLMs) wherein the covariate, distance
1435 between trap and home range center, is partially or fully unobserved – and therefore
1436 regarded as a random effect. Outside of capture-recapture, such models are usually
1437 referred to as generalized linear mixed models (GLMMs) and, therefore, SCR mod-
1438 els can be thought of as a specialized type of GLMM. Naturally then, we should
1439 consider analysis of these slightly simpler models in order to gain some experience
1440 and, hopefully, develop a better understanding of spatial capture-recapture models.

1441 In this chapter, we consider classes of GL(M)Ms – Poisson and binomial (i.e.,
1442 logistic regression) models – that will prove to be enormously useful in the analysis
1443 of capture-recapture models of all kinds. Many readers are likely familiar with these
1444 models already because they are among the most useful models in ecology and,
1445 as such, have received considerable attention in many introductory and advanced
1446 texts. We focus on them here in order to introduce the readers to the analysis of
1447 such models in **R** and **WinBUGS** or **JAGS**, which we will translate directly to
1448 the analysis of SCR models in subsequent chapters.

1449 Bayesian analysis is convenient for analyzing GL(M)Ms because it allows us to
1450 work directly with the conditional model – i.e., the model that is conditional on
1451 the random effects, using computational methods known as Markov chain Monte
1452 Carlo (MCMC). Learning how to do Bayesian analysis of GLMs and GLMMs using
1453 the **BUGS** language is, in part, the purpose of this chapter. We focus here on
1454 the use of **WinBUGS** because it is the most popular “**BUGS** engine”. However,
1455 later in the book we transition to another popular **BUGS** engine known as **JAGS**
1456 (?) which stands for *Just Another Gibbs Sampler*. For most of our purposes, the
1457 specification of models in either platform is the same, but **JAGS** is under active
1458 development at the present time while **WinBUGS** no longer is, having transitioned

1459 to **OpenBUGS** (?) which is still in active development. While we use **BUGS** of
 1460 one sort or another to do the Bayesian computations, we organize and summarize
 1461 our data and execute **WinBUGS** or **JAGS** from within **R** using the packages
 1462 **R2WinBUGS** (?), **R2jags** (?) or **rjags** (?). ?, and ? provide excellent and accessible
 1463 introductions to the basics of Bayesian analysis and GL(M)Ms using **WinBUGS**.
 1464 We don't want to be too redundant with those books and so we avoid a detailed
 1465 treatment of Bayesian methodology and software usage - instead just providing
 1466 a cursory overview so that we can move on and attack the problems we're most
 1467 interested in related to spatial capture-recapture. In addition, there are a number
 1468 of texts that provide general introductions to Bayesian analysis, MCMC, and their
 1469 applications in ecology including ?, ?, ?, and ?.

1470 While this chapter is about Bayesian analysis of GL(M)Ms, such models are
 1471 routinely analyzed using likelihood methods too. Later in this book (Chapt. ??),
 1472 we will use likelihood methods to analyze SCR models but, for now, we concentrate
 1473 on providing a basic introduction to Bayesian analysis because that is the approach
 1474 we will use in a majority of cases in later chapters.

3.1 GLMS AND GLMMS

1475 We have asserted already that SCR models work out most of the time to be vari-
 1476 ations of GL(M)Ms. You might therefore ask: What are these GLM and GLMM
 1477 models, anyhow? These models are covered extensively in many very good applied
 1478 statistics books and we refer the reader elsewhere for a detailed introduction. The
 1479 classical references for GLMs are ? and ?. In addition, we think ?, ?, and ? are all
 1480 accessible treatments. Here, we'll give the 1 minute treatment of GL(M)Ms, not
 1481 trying to be complete but rather only to preserve a coherent organization to the
 1482 book.

1483 The GLM is an extension of standard linear models allowing the response vari-
 1484 able to have some distribution from the exponential family of distributions. This
 1485 includes the normal distribution but also others such as the Poisson, binomial,
 1486 gamma, exponential, and many more. In addition, GLMs allow the response vari-
 1487 able to be related to the predictor variables (i.e., covariates) using a link function,
 1488 which is usually nonlinear. The GLM consists of three components:

- 1489 1. A probability distribution for the dependent (or response) variable y , from the
 1490 exponential family of probability distributions.
- 1491 2. A "linear predictor" $\eta = \beta_0 + x\beta_1$, where x is a predictor variable (i.e., a covari-
 1492 ate).
- 1493 3. A link function g that relates the expected value of y , $\mathbb{E}(y)$, to the linear predic-
 1494 tor, $\mathbb{E}(y) = \mu = g^{-1}(\eta)$. Therefore $g(\mathbb{E}(y)) = \eta = \beta_0 + x\beta_1$.

1495 A key aspect of GLMs is that $g(\mathbb{E}(y))$ is assumed to be a linear function of the
 1496 predictor variable(s), here x , with unknown parameters, here β_0 and β_1 , to be
 1497 estimated. In standard GLMs, the variance of y is a function V of the mean of y :

1498 $\text{Var}(y) = V(\mu)$ (see below for examples). As an example, a Poisson GLM posits
 1499 that $y \sim \text{Poisson}(\lambda)$ with $\mathbb{E}(y) = \lambda$ and usually the model for the mean is specified
 1500 using the *log link function* by

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

1501 The variance function is $V(y_i) = \lambda_i$. To see how a Poisson GLM works, use the **R**
 1502 code below to simulate some data and then estimate the parameters:

```
1503 > set.seed(13)
1504 > n <- 100          # set sample size
1505 > beta0 <- -2       # set intercept term
1506 > beta1 <- 1.5      # set coefficient
1507 > x <- rnorm(n, 0,1) # generate a predictor variable, x
1508
1509 > linpred <- beta0 + beta1*x # calculate linear predictor of E(y)
1510 > y <- rpois(n, exp(linpred)) # generate observations from model
```

1511 The **R** function `glm()` fits a GLM to the data we just generated and returns estimates of
 1512 β_0 and β_1 , which we see are fairly close to the data generating values above:

1513 > glm(y ~ 1 + x, family='poisson') # the fit model

1514 This produces the output:

```
1515 Call: glm(formula = y ~ 1 + x, family = "poisson")
1516
1517 Coefficients:
1518   (Intercept)           x
1519     -2.007            1.446
1520
1521 [... some output deleted ...]
```

1522 In this summary output, the maximum likelihood estimates (MLEs) of the regression
 1523 parameters β_0 and β_1 are labeled “**Coefficients**.” We see that these are not too different
 1524 from the data-generating values (-2 and 1.5, respectively).

1525 The binomial GLM posits that $y_i \sim \text{Binomial}(K, p)$ where K is the fixed sample size
 1526 parameter and $\mathbb{E}(y_i) = K \times p_i$. Usually the model for the mean is specified using the *logit*
 1527 *link function* according to

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

1528 Where $\text{logit}(p) = \log(p/(1-p))$. The inverse-logit function, consequently, is $\text{logit}^{-1}(p) =$
 1529 $\exp(p)/(1+\exp(p))$.

1530 A GLMM is the extension of GLMs to accommodate “random effects”. Often this
 1531 involves adding a normal random effect to the linear predictor. One simple example is
 1532 using a random intercept, α :

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

1533 where

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

1534 Many other probability distributions and formulations of the linear predictor might be
 1535 considered. GLMMs are enormously useful in ecological modeling applications for mod-
 1536 eling variation due to subjects, observers, spatial or temporal stratification, clustering,
 1537 and dependence that arises from any kind of group structure and, of course, because SCR
 1538 models prove to be a type of GLM with a random effect, but one that does not enter the
 1539 mean linearly.

3.2 BAYESIAN ANALYSIS

1540 Bayesian analysis is less familiar to many ecological researchers because they are often
 1541 educated only in the classical statistical paradigm of frequentist inference. But advances
 1542 in technology and increasing exposure to the benefits of Bayesian analysis are fast mak-
 1543 ing Bayesians out of people or at least making Bayesian analysis an acceptable, general
 1544 alternative to classical, frequentist inference.

1545 Conceptually, the main thing about Bayesian inference is that it uses probability
 1546 directly to characterize uncertainty about things we don't know. "Things", in this case,
 1547 are parameters of models and, just as it is natural to characterize uncertain outcomes of
 1548 stochastic processes using probability, it seems natural also to characterize information
 1549 about unknown parameters using probability. At least this seems natural to us and, we
 1550 think, most ecologists either explicitly adopt that view or tend to fall into that point
 1551 of view naturally. Conversely, frequentists use probability in many different ways, but
 1552 never to characterize uncertainty about parameters¹. Instead, frequentists use probability
 1553 to characterize the behavior of *procedures* such as estimators or confidence intervals (see
 1554 below). It is surprising that people readily adopt a philosophy of statistical inference in
 1555 which the things you don't know (i.e., parameters) should *not* be regarded as random
 1556 variables, so that, as a consequence, one cannot use probability to characterize one's state
 1557 of knowledge about them.

1558 **3.2.1 Bayes' rule**

1559 As its name suggests, Bayesian analysis makes use of Bayes' rule in order to make direct
 1560 probability statements about model parameters. Given two random variables z and y ,
 1561 Bayes' rule relates the two conditional probability distributions $[z|y]$ and $[y|z]$ by the
 1562 relationship:

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

1563 Bayes' rule itself is a mathematical fact and there is no debate in the statistical community
 1564 as to its validity and relevance to many problems. Generally speaking, these distributions
 1565 are characterized as follows: $[y|z]$ is the conditional probability distribution of y given z ,
 1566 $[z]$ is the marginal distribution of z and $[y]$ is the marginal distribution of y . In the context
 1567 of Bayesian inference we usually associate specific meanings in which $[y|z]$ is thought of
 1568 as "the likelihood", $[z]$ as the "prior" and so on. We leave this for later because here the
 1569 focus is on this expression of Bayes' rule as a basic fact of probability.

1570 As an example of a simple application of Bayes' rule, consider the problem of deter-
 1571 mining species presence at a sample location based on imperfect survey information. Let

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

1572 z be a binary random variable that denotes species presence ($z = 1$) or absence ($z = 0$),
 1573 let $\Pr(z = 1) = \psi$ where ψ is usually called occurrence probability, “occupancy” (?) or
 1574 “prevalence”. Let y be the *observed* presence ($y = 1$) or absence ($y = 0$) (or, strictly
 1575 speaking, detection and non-detection), and let p be the probability that a species is de-
 1576 tected in a single survey at a site given that it is present. Thus, $\Pr(y = 1|z = 1) = p$. The
 1577 interpretation of this is that, if the species is present, we will only observe it with probabili-
 1578 ty p . In addition, we assume here that $\Pr(y = 1|z = 0) = 0$. That is, the species cannot
 1579 be detected if it is not present which is a conventional view adopted in most biological
 1580 sampling problems (but see ?). If we survey a site K times but never detect the species,
 1581 then this clearly does not imply that the species is not present ($z = 0$) at this site but that
 1582 we failed to observe it. Rather, our degree of belief in $z = 0$ should be made with a prob-
 1583 abilistic statement, namely the conditional probability $\Pr(z = 1|y_1 = 0, \dots, y_K = 0)$. If
 1584 the K surveys are independent so that we might regard y_k as *iid* Bernoulli trials, then the
 1585 total number of detections, say y , is Binomial with probability p , and we can use Bayes’
 1586 rule to compute the probability that the species is present given that it is not detected in
 1587 K samples, i.e., $\Pr(z = 1|y_1 = 0, \dots, y_K = 0)$. In words, the expression we seek is:

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

1588 Mathematically, this is

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

1589 The denominator here, the probability of not detecting the species, is composed of two
 1590 parts: (1) not observing the species given that it is present (this occurs with probability
 1591 $(1 - p)^K \psi$) and (2) the species is not present (this occurs with probability $1 - \psi$). To apply
 1592 this result, suppose that $K = 2$ surveys are done at a wetland for a species of frog, and the
 1593 species is not detected there. Suppose further that $\psi = 0.8$ and $p = 0.5$ are obtained from
 1594 a prior study. Then the probability that the species is present at this site, even though it
 1595 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
 1596 a 50/50 chance that the site is occupied despite the fact that the species wasn’t observed
 1597 there.

1598 In summary, Bayes’ rule provides a simple linkage between the conditional probabilities
 1599 $[y|z]$ and $[z|y]$, which is useful whenever we need to deduce one from the other.

1600 3.2.2 Principles of Bayesian inference

1601 Bayes’ rule as a basic fact of probability is not disputed. What is controversial to some is
 1602 the scope and manner in which Bayes’ rule is applied by Bayesian analysts. Bayesian ana-
 1603 lysts assert that Bayes’ rule is relevant, in general, to all statistical problems by regarding
 1604 all unknown quantities of a model as realizations of random variables – this includes data,
 1605 latent variables, and also parameters. Classical (non-Bayesian) analysts sometimes object
 1606 to regarding parameters as outcomes of random variables. Classically, parameters are

1607 thought of as “fixed but unknown” (using the terminology of classical statistics). Indeed,
 1608 a common misunderstanding on the distinction between Bayesian and frequentist inference
 1609 goes something like this “in frequentist inference parameters are fixed but unknown
 1610 but in a Bayesian analysis parameters are random.” At best this is a sad caricature of the
 1611 distinction and at worst it is downright wrong. In Bayesian analysis the parameters are
 1612 also unknown and, in fact, there is a single data-generating value of each parameter, and
 1613 so they are also fixed. The difference is that the fixed but unknown values are regarded
 1614 as having been generated from some probability distribution. Specification of that prob-
 1615 ability distribution is necessary to carry out Bayesian analysis, but it is not required in
 1616 classical frequentist inference.

1617 To see the general relevance of Bayes’ rule in the context of statistical inference, let y
 1618 denote observations - i.e., data - and let $[y|\theta]$ be the observation model (often colloquially
 1619 referred to as the “likelihood”). Suppose θ is a parameter of interest having (prior)
 1620 probability distribution $[\theta]$ (also simply referred to as the prior). These are combined to
 1621 obtain the posterior distribution using Bayes’ rule, which is:

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

1622 Asserting the general relevance of Bayes’ rule to all statistical problems, we can conclude
 1623 that the two main features of Bayesian inference are that: (1) parameters, θ , are regarded
 1624 as realizations of a random variable and, as a result, (2) inference is based on the prob-
 1625 ability distribution of the parameters given the data, $[\theta|y]$, which is called the posterior
 1626 distribution. This is the result of using Bayes’ rule to combine the “likelihood” and the
 1627 prior distribution. The key concept is regarding parameters as realizations of a random
 1628 variable because, once you admit this conceptual view, this leads directly to the posterior
 1629 distribution, a very natural quantity upon which to base inference about things we don’t
 1630 know - including parameters of statistical models. In particular, $[\theta|y]$ is a probability
 1631 distribution for θ and therefore we can make direct probability statements to characterize
 1632 uncertainty about θ .

1633 The denominator of our invocation of Bayes’ rule, $[y]$, is the marginal distribution of
 1634 the data y . We note without further remark right now that, in many practical problems,
 1635 this can be an enormous pain to compute. The main reason that the Bayesian paradigm
 1636 has become so popular in the last 20 years or so is because methods have been developed
 1637 for characterizing the posterior distribution that do not require that we possess a math-
 1638 ematical understanding of $[y]$. This means we never have to compute it or know what it
 1639 looks like, or know anything specific about it.

1640 While we can understand the conceptual basis of Bayesian inference merely by under-
 1641 standing Bayes’ rule – that’s really all there is to it – it is not so easy to understand the
 1642 basis of classical frequentist inference. What is mostly coherent in frequentist inference is
 1643 the manner in which procedures are evaluated – the performance of a given procedure is
 1644 evaluated by “averaging over” hypothetical realizations of y , regarding the *estimator* as a
 1645 random variable. For example, if $\hat{\theta}$ is an estimator of θ then the frequentist is interested
 1646 in $E_y(\hat{\theta}|y)$ which is used to characterize bias. If the expected value of $\hat{\theta}$, when averaged
 1647 over realizations of y , is equal to θ , then $\hat{\theta}$ is unbiased.

1648 The view of parameters as being random variables allows Bayesians to use probability
 1649 to make direct probability statements about parameters. Frequentist inference procedures
 1650 do not permit direct probability statements to be made about parameter values. Instead,

1651 the view of parameters as fixed constants and estimators as random variables leads to
1652 interpretations that are not so straightforward. For example confidence intervals having
1653 the interpretation “95% probability that the interval contains the true value” and p-values
1654 being “the probability of observing an outcome of the test statistic as extreme or more
1655 than the one observed.” These are far from intuitive interpretations to most people.
1656 Moreover, this is conceptually problematic to some because we will never get to observe
1657 the hypothetical realizations that characterize the performance of our procedure.

1658 While we do tend to favor Bayesian inference for the conceptual simplicity (parameters
1659 are random, posterior inference), we mostly advocate for a pragmatic non-partisan
1660 approach to inference because, frankly, some of the frequentist methods are actually very
1661 convenient in certain situations, and will generally yield very similar inferences about
1662 parameters, as we will see in later chapters.

1663 3.2.3 Prior distributions

1664 The prior distribution $[\theta]$ is an important feature of Bayesian inference. As a conceptual
1665 matter, the prior distribution characterizes “prior beliefs” or “prior information” about
1666 a parameter. Indeed, an oft-touted benefit of Bayesian analysis is the ease with which
1667 prior information can be included in an analysis. However, more commonly, the prior
1668 is chosen to express a lack of prior information, even if previous studies have been done
1669 and even if the investigator does in fact know quite a bit about a parameter. This is
1670 because the manner in which prior information is embodied in a prior (and the amount
1671 of information) is usually very subjective and thus the result can wind up being very
1672 contentious; e.g., different investigators might report different results based on subjective
1673 assessments of prior information. Thus it is usually better to “let the data speak” and
1674 use priors that reflect absence of information beyond the data set being analyzed. An
1675 example for an uninformative prior is a Uniform(0, 1) for a probability, or a Uniform($-\infty$,
1676 ∞) (also called a “flat” or “improper” prior) for an unbounded continuous parameter.
1677 Alternatively, people use “diffuse priors”; these contain some information, but (ideally)
1678 not enough to exert meaningful influence on the posterior. An example for a diffuse prior
1679 could be a normal distribution with a large standard deviation.

1680 But still the need occasionally arises to embody prior information or beliefs about a
1681 parameter formally into the estimation scheme. In SCR models we often have a parameter
1682 that is closely linked to “home range size” and thus auxiliary information on the home
1683 range size of a species can be used as prior information, which may improve parameter
1684 estimation (e.g., see ?; also Chapt. ??).

1685 At times the situation arises where a prior can inadvertently impose substantial effect
1686 on the posterior of a parameter, and that is not desirable. For example, we use data
1687 augmentation to deal with the fact that the population size N is an unknown parameter
1688 (?) which is equivalent to imposing a Binomial(M, ψ) prior on N for some integer M (see
1689 Sec. ??). One has to take care to make sure that M is sufficiently large so as to not affect
1690 the posterior distribution on N (see Fig. ??, and also ?, Ch. 5). Another situation that
1691 we have to be careful of is that prior distributions are *not* invariant to transformation of
1692 the parameter, and therefore neither are posterior distributions (?; Sec. 6.2.1). Thus, a
1693 prior that is ostensibly non-informative on one scale, may be very informative on another
1694 scale. For example, if we have a flat prior on $\text{logit}(p)$ for some probability parameter p ,

1695 this is very different from having a Uniform(0, 1) prior on p . We show an example where
 1696 this makes a difference in Chapt. ???. Nonetheless, it is always possible to assess the
 1697 influence of prior choice, and it is often the case (with sufficient data and a structurally
 1698 identifiable model) that the influence of priors is negligible.

1699 **3.2.4 Posterior inference**

1700 In Bayesian inference, we are not focusing on estimating a single point or interval but
 1701 rather on characterizing a whole distribution – the posterior distribution – from which
 1702 one can report any summary of interest. A point estimate might be the posterior mean,
 1703 median, mode, etc.. In many applications in this book, we will compute 95% Bayesian
 1704 confidence intervals using the 2.5% and 97.5% quantiles of the posterior distribution. For
 1705 such intervals, it is correct to say $\Pr(L < \theta < U) = 0.95$. That is, “the probability that θ
 1706 lies between L and U is 0.95”.

1707 As an example, suppose we conducted a Bayesian analysis to estimate detection prob-
 1708 ability (p) of some species at a study site, and we obtained a posterior distribution of
 1709 beta(20,10) for the parameter p . The following R commands demonstrate how we make
 1710 inferences based upon summaries of the posterior distribution:

```
1711 > post.median <- qbeta(0.5, 20, 10)
1712 [1] 0.6704151
1713
1714 > post.95ci <- qbeta(c(0.025, 0.975), 20, 10)
1715 [1] 0.4916766 0.8206164
```

1716 Thus, we can state that there is a 95% probability that θ lies between 0.49 and 0.82. Fig.
 1717 ?? shows the posterior along with the summary statistics. It is not a subtle thing that
 1718 such statements cannot be made using frequentist methods, although people tend to say
 1719 it anyway and not really understand why it is wrong or even that it is wrong.

1720 **3.2.5 Small sample inference**

1721 The posterior distribution is an exhaustive summary of the state-of-knowledge about an
 1722 unknown quantity. It is *the* posterior distribution - not an estimate of that thing. It is also
 1723 not, usually, an approximation except to within Monte Carlo error (in cases where we use
 1724 simulation to calculate it, see Sec. ??). One of the great virtues of Bayesian analysis which
 1725 is not widely appreciated is that posterior inference is not “asymptotic”, which is to say,
 1726 valid in a limiting sense as the sample size tends to infinity. Rather, posterior inference is
 1727 valid for *any* sample size and, in particular, *the* sample size on-hand. Conversely, almost
 1728 all frequentist procedures are based on asymptotic approximations to the procedure which
 1729 is being employed.

1730 There seems to be a prevailing view in statistical ecology that classical likelihood-based
 1731 procedures are virtuous because of the availability of simple formulas and procedures for
 1732 carrying out inference, such as calculating standard errors, doing model selection by Akaike
 1733 information criterion (AIC), and assessing goodness-of-fit. In large samples, this may be
 1734 an important practical benefit, but the theoretical validity of these procedures cannot be
 1735 asserted in most situations involving small samples. This is not a minor issue because

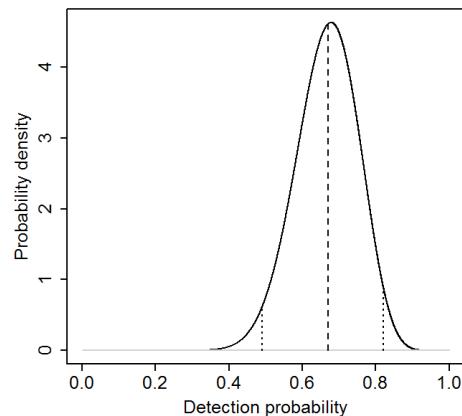


Figure 3.1. Probability density plot of a hypothetical posterior distribution of $\text{beta}(20,10)$; dashed lines indicate mean and upper and lower 95% interval

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 (?). For examples: A recent paper (?) 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. ? estimated density of the endangered southern river otter (*Lontra provocax*) based on 12 individuals captured over 3 years, ? 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, ? reported only 9 individual ocelots captured and ? 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" (?).

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 may be all that is available for species that are extraordinarily important for conservation and management. The Bayesian framework for inference provides a valid, rigorous, and flexible framework that is theoretically justifiable in arbitrary sample sizes. This is not to say that one will obtain precise estimates of density or other parameters, just that your inference is coherent and justifiable from a conceptual and technical statistical point of view. That is, for example when we estimate the density D of some animal population,

²Actual quote from a referee

1758 we report the posterior probability $\Pr(D|data)$ which is easily interpretable and just what
 1759 it is advertised to be and we don't need to do a simulation study to evaluate how well
 1760 the reported $\Pr(D|data)$ deviates from the "true" $\Pr(D|data)$ because they are the same
 1761 quantity.

3.3 CHARACTERIZING POSTERIOR DISTRIBUTIONS BY MCMC SIMULATION

1762 In practice, it is not really feasible to ever compute the marginal probability distribution
 1763 $[y]$, the denominator resulting from application of Bayes' rule (Eq. ??). For decades
 1764 (even centuries!) this impeded the adoption of Bayesian methods by practitioners. Or,
 1765 the few Bayesian analyses done were based on asymptotic normal approximations to the
 1766 posterior distribution. While this was useful from a theoretical and technical standpoint
 1767 and, practically, it allowed people to make the probability statements that they naturally
 1768 would like to make, it was kind of a bad joke around the Bayesian water-cooler to, on
 1769 one hand, criticize classical statistics for being, essentially, completely ad hoc in their
 1770 approach to things but then, on the other hand, have to devise various approximations to
 1771 what they were trying to characterize. The advent of Markov chain Monte Carlo (MCMC)
 1772 methods has made it easier to calculate posterior distributions for just about any problem
 1773 to sufficient levels of precision.

1774 Broadly speaking, MCMC is a class of methods for drawing random samples (i.e.,
 1775 simulating from or just "sampling") from the target posterior distribution. Thus, even
 1776 though we might not recognize the posterior as a named distribution or be able to analyze
 1777 its features analytically, e.g., devise mathematical expressions for the mean and variance,
 1778 we can use these MCMC methods to obtain a large sample from the posterior and then
 1779 use that sample to characterize features of the posterior. What we do with the sample
 1780 depends on our intentions – typically we obtain the mean or median for use as a point
 1781 estimate, and take a confidence interval based on Monte Carlo estimates of the quantiles.

1782 3.3.1 What goes on under the MCMC hood

1783 We will develop and apply MCMC methods in some detail for spatial capture-recapture
 1784 models in Chapt. ???. Here we provide a simple illustration of some basic ideas related to
 1785 the practice of MCMC.

1786 A type of MCMC method relevant to most problems is Gibbs sampling (?) which
 1787 we address in more detail in Chapt. ???. Gibbs sampling involves iterative simulation
 1788 from the "full conditional" distributions (also called conditional posterior distributions).
 1789 The full conditional distribution for an unknown quantity is the conditional distribution
 1790 of that quantity given every other random variable in the model - the data and all other
 1791 parameters (see Sec. ?? for rules of how to construct full conditionals). For example, for
 1792 a normal regression model ³ with $y \sim \text{Normal}(\beta_0 + \beta_1(x - \bar{x}), \sigma^2)$ where lets say σ^2 is
 1793 known, the full conditionals are, using "bracket notation",

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

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

1794 and

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

1795 We might use our knowledge of probability to identify these mathematically. In particular,
1796 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
1797 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
1798 some algebra reveals that

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

1799 where $\tau = 1/\sigma^2$ and $\tau_{\beta_0} = 1/\sigma_{\beta_0}^2$ (the inverse of the variance is sometimes called *precision*),
1800 and $w = \tau n / (\tau n + \tau_{\beta_0})$. We see in this case that the posterior mean is a *precision-weighted*
1801 sum of the sample mean \bar{y} and the prior mean μ_{β_0} , and the posterior *precision* is the
1802 sum of the precision of the likelihood and that of the prior. These results are typical of
1803 many classes of problems. In particular, note that as the prior precision tends to 0, i.e.,
1804 $\tau_{\beta_0} \rightarrow 0$, then the posterior of β_0 tends to $\text{Normal}(\bar{y}, \sigma^2/n)$. We recognize the variance of
1805 this distribution as that of the variance of the sampling distribution of \bar{y} and its mean is
1806 in fact the MLE of β_0 for this model. The conditional posterior of β_1 has a very similar
1807 form:

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

1808 which might look slightly unfamiliar, but note that if $\tau_{\beta_1} = 0$, then the mean of this
1809 distribution is the familiar $\hat{\beta}_1$, and the variance is, in fact, the sampling variance of $\hat{\beta}_1$.
1810 The MCMC algorithm for this model has us simulate in succession, repeatedly, from those
1811 two distributions. See ? for more examples of Gibbs sampling for the normal model, and
1812 we also provide another example in Chapt. ???. A conceptual representation of the MCMC
1813 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. ???
    2. Draw a new value of  $\beta_1$  from Eq. ???
}

```

1815 As we just saw for this simple “normal-normal” model, it is sometimes possible to
1816 specify the full conditional distributions analytically. In general, when certain so-called
1817 conjugate prior distributions are used, which have an analytic form that, in a statistical
1818 sense, “matches” the likelihood, then the form of the full conditional distributions is also
1819 similar to that of the observation model. In this normal-normal case, the normal distribu-
1820 tion for the mean parameters is the conjugate prior for the normal observation model, and
1821 thus the full-conditional distributions are also normal. This is convenient because, in such
1822 cases, we can simulate directly from them using standard methods (or R functions). But,
1823 in practice, we don’t really ever need to know such things because most of the time we

1824 can get by using a simple algorithm, called the Metropolis-Hastings (henceforth “MH”)
 1825 algorithm, to obtain samples from these full conditional distributions without having to
 1826 recognize them as specific, named, distributions. This gives us enormous freedom in devel-
 1827 oping models and analyzing them without having to resolve them mathematically because
 1828 to implement the MH algorithm we need only identify the full conditional distribution up
 1829 to a constant of proportionality, that being the marginal distribution in the denominator
 1830 (e.g., $[y|\beta_1]$ above).

1831 We will talk about the Metropolis-Hastings algorithm shortly, and we will use it ex-
 1832 tensively in the analysis of SCR models (e.g., Chapt. ??).

1833 3.3.2 Rules for constructing full conditional distributions

1834 The basic strategy for constructing full-conditional distributions for devising MCMC al-
 1835 gorithms can be reduced conceptually to a couple of basic steps summarized as follows:

- 1836 (step 1) Identify all stochastic components of the model and collect their probability
 1837 distributions;
- 1838 (step 2) Express the full conditional in question as proportional to the product of all
 1839 probability distributions identified in step 1;
- 1840 (step 3) Remove the ones that don’t have the focal parameter in them.
- 1841 (step 4) Do some algebra on the result in order to identify the resulting probability
 1842 distribution function (pdf) or mass function (pmf).

1843 Of the 4 steps, the last of those is the main step that requires quite a bit of statistical
 1844 experience and intuition because various algebraic tricks can be used to reshape the mess
 1845 into something recognizable – i.e., a standard, named distribution. But step 4 is not
 1846 necessary if we decide instead to use the Metropolis-Hastings algorithm as described below.

1847 In the context of our simple linear regression model that we’ve been working with,
 1848 to characterize $[\beta_0|y, \beta_1]$ we first apply step 1 and identify the model components as:
 1849 $[y|\beta_0, \beta_1]$, with prior distributions $[\beta_0]$ and $[\beta_1]$. Step 2 has us write $[\beta_0|y, \beta_1] \propto [y|\beta_0, \beta_1][\beta_0][\beta_1]$.
 1850 Step 3: We note that $[\beta_1]$ is not a function of β_0 and therefore we remove it to obtain
 1851 $[\beta_0|y, \beta_1] \propto [y|\beta_0, \beta_1][\beta_0]$. Similarly, applying step 2 and 3 for β_1 we obtain $[\beta_1|y, \beta_0] \propto$
 1852 $[y|\beta_0, \beta_1][\beta_1]$. We apply step 4 and manipulate these algebraically to arrive at the result
 1853 (which we provided in Eqs. ?? and ??) or, alternatively, we can sample them indirectly
 1854 using the Metropolis-Hastings algorithm, which we discuss now.

1855 3.3.3 Metropolis-Hastings algorithm

1856 The Metropolis-Hastings (MH) algorithm is a completely generic method for sampling
 1857 from any distribution, say $[\theta]$. In our applications, $[\theta]$ will typically be the full conditional
 1858 distribution of θ . While we sometimes use Gibbs sampling, we seldom use “pure” Gibbs
 1859 sampling because full conditionals do not always take the form of known distributions we
 1860 can sample from directly. In such cases, we use MH to sample from the full conditional
 1861 distributions. When the MH algorithm is used to sample from full conditional distributions
 1862 of a Gibbs sampler the resulting hybrid algorithm is called *Metropolis-within-Gibbs*. In
 1863 Sec. ?? we will construct such an algorithm for a simple class of models. We discuss both
 1864 the Gibbs and the MH algorithm, as well as their hybrid in more depth in Chapt. ??.

1865 The MH algorithm generates candidate values for the parameter(s) we want to estimate
 1866 from some proposal or candidate-generating distribution that may be conditional on the
 1867 current value of the parameter, denoted by $h(\theta^*|\theta^{t-1})$. Here, θ^* is the *candidate* or
 1868 proposed value and θ^{t-1} is the value of θ at the previous time step, i.e., at iteration $t - 1$
 1869 of the MCMC algorithm. The proposed value is accepted with probability

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

1870 which is called the MH acceptance probability. This ratio can sometimes be > 1 in which
 1871 case we set it equal to 1. It is useful to note that $h()$ can be any probability distribution.

1872 In the context of using the MH algorithm to do MCMC (in which case the target
 1873 distribution is a full-conditional or posterior distribution), an important fact is, no matter
 1874 the choice of $h()$, we can compute the MH acceptance probability directly because the
 1875 marginal distribution of y cancels from both the numerator and denominator of r . This
 1876 is the magic of the MH algorithm.

3.4 BAYESIAN ANALYSIS USING THE BUGS LANGUAGE

1877 We won't be too concerned with devising our own MCMC algorithms for every analysis,
 1878 although we will do that a few times for fun. More often, we will rely on the freely
 1879 available software package **WinBUGS** or **JAGS** for doing this. We will always execute
 1880 these **BUGS** engines from within **R** using the **R2WinBUGS** (?) or, for **JAGS**, the **R2jags**
 1881 (?) or **rjags** (?) packages. **WinBUGS** and **JAGS** are MCMC black boxes that take
 1882 a pseudo-code description (i.e., written in the **BUGS** language) of all of the relevant
 1883 stochastic and deterministic elements of a model and generate an MCMC algorithm for
 1884 that model. But you never get to see the algorithm. Instead, **WinBUGS/JAGS** will
 1885 run the algorithm and return the Markov chain output - the posterior samples of model
 1886 parameters.

1887 The great thing about using the **BUGS** language is that it forces you to become
 1888 intimate with your statistical model - you have to write each element of the model down,
 1889 admit (explicitly) all of the various assumptions, understand what the actual probability
 1890 assumptions are and how data relate to latent variables and data and latent variables
 1891 relate to parameters, and how parameters relate to one another.

1892 While we normally use **WinBUGS**, we note that **OpenBUGS** is the current active
 1893 development tree of the **BUGS** project. See ? and ?, especially Appendix 1 for more
 1894 on practical analysis in **WinBUGS**. Those books should be consulted for a more com-
 1895 prehensive introduction to using **WinBUGS**. Recently we have migrated many of our
 1896 analyses to **JAGS** (?), which we adopt later in the book. You can refer to ? for an
 1897 ecological introduction to **JAGS**. Next, we provide an example of a Bayesian analysis
 1898 using **WinBUGS**.

3.4.1 Linear regression in WinBUGS

1900 We provide a brief introductory example of a normal regression model using a small
 1901 simulated data set. The following commands are executed from within your **R** workspace.
 1902 First, simulate a covariate x and observations y having prescribed intercept, slope and
 1903 variance:

```

1904 > x <- rnorm(10)
1905 > mu <- -3.2 + 1.5*x
1906 > y <- rnorm(10, mu, sd=4)
```

1907 The **BUGS** model specification for a normal regression model is written within **R** as
 1908 a character string input to the command **cat()** and then dumped to a text file named
 1909 **normal.txt**:

```

1910 > cat("
1911   model{
1912     for (i in 1:10){
1913       y[i] ~ dnorm(mu[i],tau)      # the likelihood
1914       mu[i] <- beta0 + beta1*x[i]  # the linear predictor
1915     }
1916     beta0 ~ dnorm(0,.01)          # prior distributions
1917     beta1 ~ dnorm(0,.01)
1918     sigma ~ dunif(0,100)
1919     tau <- 1/(sigma*sigma)       # tau is the precision
1920   }                                # and a derived parameter
1921 ",file="normal.txt")
```

1922 Alternatively, you can write the model specifications directly within a text file and save it
 1923 in your current working directory, but we do not usually take that approach in this book.

1924 The **BUGS** dialects⁴ parameterize the normal distribution in terms of the mean and
 1925 inverse-variance, called the precision. Thus, **dnorm(0,.01)** implies a variance of 100.
 1926 We typically use diffuse normal priors for mean parameters, β_0 and β_1 in this case, but
 1927 sometimes we might use uniform priors with suitable bounds $-B$ and $+B$. Also, we
 1928 typically use a Uniform($0, B$) prior on standard deviation parameters (?). But sometimes
 1929 we might use a gamma prior on the precision parameter τ . In a **BUGS** model file, every
 1930 variable referenced in the model description has to be either data, which will be input
 1931 (see below), a random variable which must have a probability distribution associated with
 1932 it using the tilde character “~” (a.k.a. “twiddle”) or it has to be a derived parameter
 1933 connected to variables and data using an assignment arrow: “<-”.

1934 To fit the model, we need to describe various data objects to **WinBUGS**. In particular,
 1935 we create an **R** list object called **data** which are the data objects identified in the **BUGS**
 1936 model file. In the example, the data consist of two objects which exist as **y** and **x** in the
 1937 **R** workspace and also in the **WinBUGS** model definition. We also create an **R** function
 1938 that produces a list of starting values, **inits**, that get sent to **WinBUGS**. In general,
 1939 starting values are optional. We recommend to always provide reasonable starting values
 1940 where possible, both for structural parameters and also random effects⁵. Finally, we
 1941 identify the names of the parameters (labeled correspondingly in the **WinBUGS** model
 1942 specification) that we want **WinBUGS** to save the MCMC output for. In this example,

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

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**

1943 we will “monitor” the parameters β_0 , β_1 , σ and τ . **WinBUGS** is executed using the
 1944 **R** command **bugs()**. We set the option **debug=TRUE** if we want the **WinBUGS** GUI to
 1945 stay open (useful for analyzing MCMC output and looking at the **WinBUGS** error log).
 1946 Also, we set **working.dir=getwd()** so that **WinBUGS** output files and the log file are
 1947 saved in the current **R** working directory (note that sometimes you will need to specify the
 1948 place where you installed **WinBUGS** within the **bugs()** call, using the **bugs.directory**
 1949 argument). All of these activities together look like this:

```
1950 > library(R2WinBUGS)      # "load" the R2WinBUGS package
1951 > data <- list( y=y, x=x)
1952 > inits <- function()
1953 > list ( beta0=rnorm(1),beta1=rnorm(1),sigma=runif(1,0,2) )
1954 > parameters <- c("beta0","beta1","sigma","tau")
1955 > out <- bugs(data, inits, parameters, "normal.txt", n.thin=1, n.chains=2,
1956           n.burnin=2000, n.iter=6000, debug=TRUE,working.dir=getwd())
```

1957 Note that the previously created objects defining data, initial values and parameters to
 1958 monitor are passed to the function **bugs()**. In addition, various other things are declared:
 1959 The number of parallel Markov chains (**n.chains**), the thinning rate (**n.thin**), the number
 1960 of burn-in iterations (**n.burnin**) and the total number of iterations (**n.iter**). To develop
 1961 a detailed understanding of the various parameters and settings used for MCMC, consult
 1962 a basic reference such as ?. We also come back to these issues in the following section
 1963 (??) and in Chapt. ???. A common question is “how should my data be formatted?” That
 1964 depends on how you describe the model in the **BUGS** language, and how your data are
 1965 input into **R**. There is no unique way to describe any particular model and so you have
 1966 some flexibility. We talk about data format further in the context of capture-recapture
 1967 models and SCR models in Chapt. ?? and elsewhere.

1968 You should execute all of the commands given above and then close the **WinBUGS**
 1969 GUI, and the data will be read back into **R** (or specify **debug=FALSE** in the **bugs()** call).
 1970 We don’t want to give instructions on how to navigate and use the GUI – but you can fire
 1971 up **WinBUGS** and read the help files, or see Chapt. 4 from ? for a brief introduction.
 1972 The print command applied to the object **out** prints some basic summary output (this is
 1973 slightly edited):

```
1974 > print(out,digits=2)
1975 Inference for Bugs model at "normal.txt", fit using WinBUGS,
1976 2 chains, each with 6000 iterations (first 2000 discarded)
1977 n.sims = 8000 iterations saved
1978          mean   sd  2.5%   25%   50%   75% 97.5% Rhat n.eff
1979 beta0    -6.62 1.64 -9.77 -7.63 -6.64 -5.63 -3.29    1  4200
1980 beta1     0.81 1.20 -1.63  0.09  0.80  1.54  3.24    1  5100
1981 sigma     4.99 1.56  2.93  3.92  4.66  5.70  8.85    1  8000
1982 tau       0.05 0.03  0.01  0.03  0.05  0.07  0.12    1  8000
1983 deviance 58.72 3.21 55.06 56.35 57.85 60.26 67.15    1  6200
1984
1985 For each parameter, n.eff is a crude measure of effective sample size,
1986 and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
```

1987
 1988 DIC info (using the rule, pD = Dbar-Dhat)
 1989 pD = 2.5 and DIC = 61.3

1990 In the **WinBUGS** output you see a column called “Rhat”, as well as one called
 1991 “n.eff”. These are convergence diagnostics (the \hat{R} or Brooks-Gelman-Rubin statistic
 1992 and the effective sample size) and we will discuss those in the following section, ???. DIC
 1993 is the deviance information criterion (?; see section ??) which some people use in a manner
 1994 similar to AIC although it is recognized to have some problems in hierarchical models (?).
 1995 We consider use of DIC in the context of SCR models in Chapt. ??.

3.5 PRACTICAL BAYESIAN ANALYSIS AND MCMC

1996 The mere execution of a Bayesian analysis using the **BUGS** language, as demonstrated
 1997 with the linear regression example, is fairly straight forward. There are, however, a number
 1998 of really important practical issues to be considered in any Bayesian analysis and we cover
 1999 some of these briefly here before we move on to implementing slightly more complex
 2000 GL(M)s in a Bayesian framework.

2001 **3.5.1 Choice of prior distributions**

2002 Bayesian analysis requires that we choose prior distributions for all of the structural pa-
 2003 rameters of the model (we use the term structural parameter to mean all parameters that
 2004 aren’t customary thought of as latent variables). We will strive to use priors that are
 2005 meant to express little or no prior information - default or customary “non-informative”
 2006 or diffuse priors. This will be Uniform(a, b) priors for parameters that have a natural
 2007 bounded support and, for parameters that live on the real line we use either (1) diffuse
 2008 normal priors, as we did in the linear regression example above; (2) improper uniform
 2009 priors which have unbounded support, e.g., $[\theta] \propto 1$, or (3) sometimes even a bounded
 2010 Uniform(a, b) prior, if that greatly improves the performance of **WinBUGS** or other
 2011 software doing the MCMC for us. In **WinBUGS** a prior with low precision, τ , where
 2012 $\tau = 1/\sigma^2$, such as Normal(0, .01) will typically be used. Of course $\tau = 0.01$ ($\sigma^2 = 100$)
 2013 might be very informative for a regression parameter depending on its magnitude and
 2014 scaling of x . Therefore, we recommend that predictor variables (covariates) *always* be
 2015 standardized to have mean 0 and variance 1.

2016 **Lack of invariance of priors to transformation.** Clearly there are a lot of choices
 2017 for ostensibly non-informative priors, and the degree of non-informativeness depends on
 2018 the parameterization. For example, a natural non-informative prior for the intercept of a
 2019 logistic regression

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

2020 would be a very diffuse normal prior, $[\beta_0] = \text{Normal}(0, \text{Large})$ or even $\beta_0 \sim \text{Uniform}(-\text{Large}, \text{Large})$.
 2021 However, we might also use a prior on the parameter $p_0 = \text{logit}^{-1}(\beta_0)$, which is $\Pr(y=1)$
 2022 for the value $x = 0$. Since p_0 is a probability a natural choice is $p_0 \sim \text{Uniform}(0, 1)$. These
 2023 priors are very different in their implications. For example, if we choose the normal prior
 2024 for β_0 with variance Large = 5^2 and look at the implied prior for p_0 we have the result
 2025 shown in Fig. ?? which looks nothing like a Uniform(0, 1) prior. These two priors can

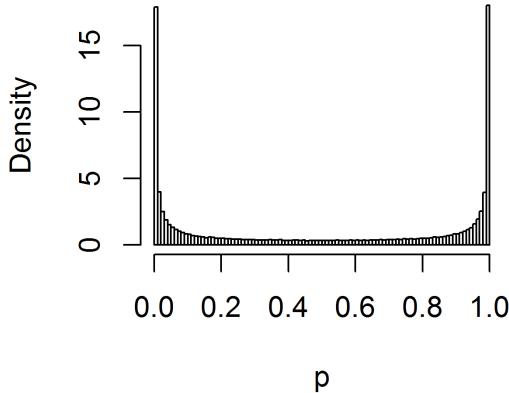


Figure 3.2. Implied prior for $p_0 = \exp(\beta_0)/(1 + \exp(\beta_0))$ if $\beta_0 \sim \text{Normal}(0, 5^2)$.

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

3.5.2 Convergence and so-forth

Once we have carried out an analysis by MCMC, there are many other practical issues that we have to confront. One characteristic of MCMC sampling is that Markov chains take some time to converge to their stationary distribution - in our case the posterior distribution for some parameter given data, $[\theta|y]$. Only when the Markov chain has reached its stationary distribution, the generated samples can be used to characterize the posterior distribution. Thus, one of the most important issues we need to address is “have the chains converged?” Since we do not know what the stationary posterior distribution of our Markov chain should look like (this is the whole point of doing an MCMC analysis), we effectively have no means to assess whether or not it has truly converged to this desired distribution. Most MCMC algorithms only guarantee that, eventually, the samples being

generated will be from the target posterior distribution, but no-one can tell us how long this will take. Also, you only know the part of your posterior distribution that the Markov chain has explored so far – for all you know the chain could be stuck in a local maximum, while other maxima remain completely undiscovered. Acknowledging that there is truly nothing we can do to ever prove convergence of our MCMC chains, there are several things we can do to increase the degree of confidence we have about the convergence of our chains. Some problems are easily detected using simple plots, such as a time-series plot, where parameter values of each MCMC iteration are plotted against the number of iterations. Fig. ?? shows the time series plots for the three parameters – β_0 , β_1 and σ – from our linear regression example, taken from the **WinBUGS** GUI before closing it to return to **R**.

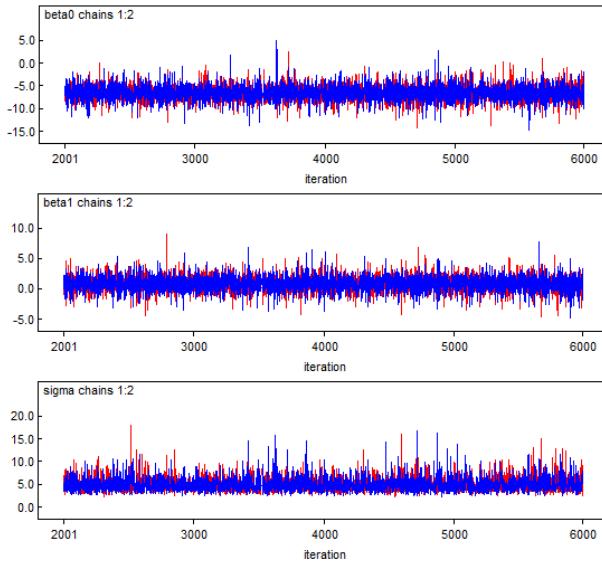


Figure 3.3. Time-series plots for parameters from a linear regression run in **WinBUGS** using two parallel Markov chains.

Typically a period of transience is observed in the early part of the MCMC algorithm, and this is usually discarded as the “burn-in” period. In our linear regression example, within the `bugs()` call we set the burn-in period as 2000 iterations so these are automatically removed by **WinBUGS** and are not part of the output (but Fig. ?? shows a time-series plot that starts at iteration 0 with a clearly visible burn-in period). The quick diagnostic to whether convergence has been achieved is that your Markov chains look “grassy” – this seems a reasonable statement for the plots in Fig. ???. Another way to check convergence is to update the parameters some more and see if the posterior changes. If the chains have converged to the posterior, the posterior mean, confidence intervals, and

other summaries should be relatively static as we continue to run the algorithm. Yet another option, and one generally implemented in **WinBUGS**, is to run several Markov chains and to start them off at different initial values that are over-dispersed relative to the posterior distribution. Such initial values help to explore different areas of the parameter space simultaneously; if, after a while, all chains oscillate around the same average value, chances are good that they indeed converged to the posterior distribution. Gelman and Rubin came up with the so-called “R-hat” statistic (\hat{R}) or Brooks-Gelman-Rubin statistic that essentially compares within-chain and between-chain variance to check for convergence of multiple chains (?). The R-hat statistic should be close to 1 if the Markov chains have converged and sufficient posterior samples have been obtained. For the linear regression example, we ran two parallel chains (also specified in the `bugs()` call) and **WinBUGS** returns the \hat{R} statistic for us as part of the summary model output. If you look back to Sec. ?? you see that $\hat{R} = 1$ for all parameters of the linear model. In practice, $\hat{R} \leq 1.2$ may be good enough for some problems. For some models you can't actually realize a low \hat{R} . E.g., if the posterior is a discrete mixture of distributions 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. ???. 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 (??). Practical considerations might necessitate thinning, even though it is statistically inefficient. For example, in models with many parameters or other unknowns being tabulated, the output files might be enormous and unwieldy to work with. In such cases, thinning is perfectly reasonable. In many cases, how well the Markov chains mix is strongly influenced by parameterization, standardization of covariates, and the prior distributions being used. Some things work better than others, and the investigator should experiment with different settings and remain calm when things don't work out perfectly.

Is the posterior sample large enough? The subsequent samples generated from a Markov chain are not *independent* samples from the posterior distribution, due to the

correlation among samples introduced by the Markov process⁶ and the sample size has to be adjusted to account for the autocorrelation in subsequent samples (see Chapt. 8 in ? for more details). This adjusted sample size is referred to as the 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. **WinBUGS** will automatically return the effective sample size for all monitored parameters, as we saw in our linear regression example (the “n.eff” column of the summary output). If you find that your supposedly long Markov chain has only generated a very short effective sample, you should consider a longer run. What exactly constitutes a reasonable effective sample size is hard to say. A more palpable measure of whether you’ve run your chain for enough iterations is the time-series or Monte Carlo error - the “noise” introduced into your samples by the stochastic MCMC process. The MC error is printed by default in summaries produced in the **WinBUGS** GUI, which can be reproduced in R using `bugs.log('log.txt')$stats` (note that “log.txt” refers to a model log file that **WinBUGS** automatically creates in the working directory; it is overwritten with every new model you run unless you save it under a different name).

```
2126 > bugs.log('log.txt')$stats
2127 $stats
2128      mean      sd   MCerror    2.5%   median   97.5% start sample
2129 beta0   -6.64700 1.60300 0.0179400 -9.7140 -6.70800 -3.2730 2001 8000
2130 beta1    0.82100 1.19000 0.0116800 -1.4900  0.82560  3.1800 2001 8000
2131 deviance 58.66000 3.08800 0.0506800 55.0700 57.93000 66.8400 2001 8000
2132 sigma     4.96800 1.52300 0.0248300  2.9350  4.68100  8.7410 2001 8000
2133 tau       0.05074 0.02677 0.0003651  0.0131  0.04564  0.1162 2001 8000
```

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

3.5.3 Bayesian confidence intervals

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

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, $\theta^{(t-1)}$.

2148 3.5.4 Estimating functions of parameters

2149 A benefit of analysis by MCMC is that we can seamlessly estimate functions of parameters
 2150 by simply tabulating the desired function of the simulated posterior draws. For example,
 2151 if θ is the parameter of interest and let $\theta^{(i)}$ for $i = 1, 2, \dots, M$ be the posterior samples
 2152 of θ . Let $\eta = \exp(\theta)$, then a posterior sample of η can be obtained simply by computing
 2153 $\exp(\theta^{(i)})$ for $i = 1, 2, \dots, M$. Almost all SCR models in this book involve at least 1 derived
 2154 parameter. For example, density D is a derived parameter, being a function of population
 2155 size N and the area A of the underlying state-space of the point process (see Chapt. ??).

2156 **Example: Finding the optimum value of a covariate.** As another example of
 2157 estimating functions of model parameters, suppose that the normal regression model from
 2158 Sec. ?? had a quadratic response function of the form

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

2159 Then the optimum value of x , i.e., that corresponding to the optimal expected response,
 2160 can be found by setting the derivative of this function to 0 and solving for x . We find that

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

2161 yields that $x_{opt} = -\beta_1/(2 * \beta_2)$. We can just take our posterior draws for β_1 and β_2
 2162 and obtain a posterior sample of x_{opt} by this simple calculation applied to the posterior
 2163 output. As an exercise, take the normal model above and simulate a quadratic response
 2164 and then describe the posterior distribution of x_{opt} .

3.6 POISSON GLMS

2165 The Poisson GLM (also known as “Poisson regression”) is probably the most relevant
 2166 and important class of models in all of ecology. The basic model assumes observations
 2167 $y_i; i = 1, 2, \dots, n$ follow a Poisson distribution with mean λ which we write

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

2168 Commonly y_i is a count of animals or plants at some point in space (“site”) i , and λ
 2169 might vary over sites as well. For example, i might index point count locations in a
 2170 forest, survey route centers, or sample quadrats, or similar, and we are interested in how
 2171 λ depends on site characteristics such as habitat. If covariates are available it is typical to
 2172 model them as linear effects on the log mean. If x_i is some measured covariate associated
 2173 with observation i , then,

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

2174 While we only specify the mean of the Poisson model directly, the Poisson model (and
 2175 all GLMs) has a “built-in” variance which is directly related to the mean. In this case,
 2176 $\text{Var}(y) = \mathbb{E}(y) = \lambda$. Thus the model accommodates a linear increase in variance with the
 2177 mean.

2178 **3.6.1 Example: Breeding Bird Survey data**

2179 As an example we consider a classical situation in ecology where counts of an organism
 2180 are made at a collection of spatial locations. In this particular example, we have
 2181 mourning dove (*Zenaida macroura*) counts made along North American Breeding Bird
 2182 Survey (BBS) routes in Pennsylvania, USA. A route consists of 50 stops separated by
 2183 0.5 miles. For the purposes here we are defining y_i = route total count and the sample
 2184 location will be marked by the center point of the BBS route. The survey is run annually
 2185 and the data set we analyze is 1966-1998. BBS data can be obtained online at
 2186 <http://www.pwrc.usgs.gov/bbs/>, but the particular chunk of data we will be using here
 2187 is also included in the **scrbook** package (**data(bbsdata)**). We will make use of the whole
 2188 data set shortly but for now we're going to focus on a specific year of counts (1990) for
 2189 the sake of building a simple model. In 1990 there were 77 active routes; this data set
 2190 contains rows which index the unique route, column 1 is the route ID, columns 2-3 are
 2191 the route coordinates (longitude/latitude), column 4 is a habitat covariate "forest cover"
 2192 (standardized, see below) and the remaining columns are the yearly counts. Years for
 2193 which a survey was not conducted on a route are coded as "NA" in the data matrix. We
 2194 imagine that this will be a typical format for many ecological studies, perhaps with more
 2195 columns representing covariates. To read in the data and display the first few elements of
 2196 the data frame containing the counts, do this:

```
2197 > data(bbsdata)           # loads data frame 'bbs'  

2198 > bbsdata$counts[1:2,1:6]  

2199  

2200      X    lon    lat   habitat X66 X67  

2201 1 72002 -80.445 41.501 -0.3871372 NA 24  

2202 2 72003 -80.347 41.214 -1.0171629 NA NA
```

2203 It is useful to display the spatial pattern in the observed counts. For that we use a
 2204 spatial dot plot – where we plot the coordinates of the observations and mark the color
 2205 of the plotting symbol based on the magnitude of the count. We have a special plotting
 2206 function for that which is called **spatial.plot()** and it is available with the supplemental
 2207 **R** package **scrbook**. Actually, what we want to do here is plot the log-counts (+1 of
 2208 course) which (Fig. ??) display a notable pattern that could be related to something.
 2209 The **R** commands for obtaining this figure are:

```
2210 > library(scrbook)  

2211 > data(bbsdata)  

2212 > library(maps)  

2213  

2214 > y <- bbsdata$counts[, "X90"] # Pick year 1990  

2215 > notna <- !is.na(y)  

2216 > y <- y[notna]  

2217 > locs <- bbsdata$counts[notna,c("lon","lat")]  

2218 > sz <- y/max(y)  

2219  

2220 > par(mar=c(3,3,3,6))  

2221 > plot(locs,pch=" ",axes=FALSE,xlim=range(locs[,1])+c(-.3,+.3),
```

```

2222     ylim=c(range(locs[,2]) + c(-.6,.6)), xlab=" ",ylab=" ")
2223 > map('state', regions='pennsylvania', add=TRUE, lwd=2)
2224 > spatial.plot(bbsdata$counts[notna,2:3], y, cx=1+sz*6, add=TRUE)

```

2225 We can ponder the potential effects that might lead to dove counts being high - corn
 2226 fields, telephone wires, barn roofs along with misidentification of pigeons, these could all
 2227 correlated 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-

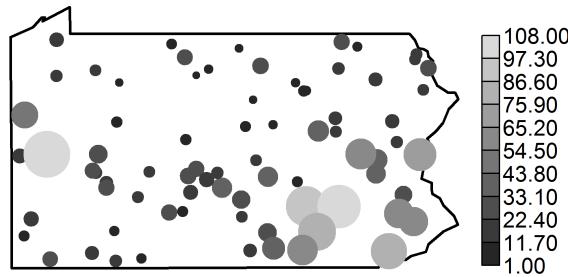


Figure 3.4. Mourning dove counts along North American Breeding Bird Survey routes in Pennsylvania (year = 1990). Plot symbol shading and circle size is proportional to raw count.

```

2228
2229 vided in the data frame bbsdata$habitat) which can be plotted using the spatial.plot
2230 function with the following R commands

```

```

2231 > habdata <- bbsdata$habitat
2232 > map('state',regions="penn",lwd=2)
2233 > I <- matrix(NA, nrow=30, ncol=40)
2234 > I <- matrix(habdata[, "dfor"], ncol=40, byrow=FALSE)
2235 > ux <- unique(habdata[,2])
2236 > uy <- sort(unique(habdata[,3]))
2237
2238 > par(mar=c(3,3,3,6))
2239 > plot(locs,pch=" ", axes=FALSE, xlim=range(locs[,1])+c(-.3,+.3),
2240         ylim=c(range(locs[,2]) + c(-.6,.6)), xlab=" ",ylab=" ")
2241 > image(ux,uy,rot(I), add=TRUE, col=gray(seq(3,17,,10)/20) )

```

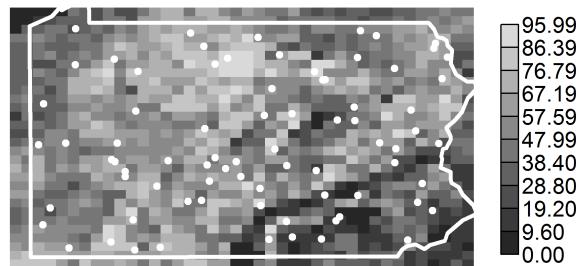


Figure 3.5. Forest cover (percent deciduous) in Pennsylvania. BBS route locations are shown by white dots.

```
2242 > map('state', regions='pennsylvania', add=TRUE, lwd=3, col="white")
2243 > image.scale(I, col=gray(seq(3,17,,10)/20) )
2244 > points(locs,pch=20, col="white")
```

2245 The result appears in Fig. ???. We see a prominent pattern that indicates high forest
 2246 coverage in the central part of the state and low forest cover in the SE. Inspecting the
 2247 previous figure of the raw counts suggests a relationship between counts and forest cover
 2248 which is perhaps not surprising.

2249 **3.6.2 Doing it in WinBUGS**

2250 Here we demonstrate how to fit a Poisson GLM in **WinBUGS** using the covariate $x_i =$
 2251 forest cover along BBS route i . It is advisable that x_i be standardized in most cases as
 2252 this will improve mixing of the Markov chains. We have pre-standardized the forest cover
 2253 covariate for the BBS route locations, and so we don't have to worry about that here. To
 2254 read the BBS data into **R** and get things set up for **WinBUGS** we issue the following
 2255 commands:

```
2256 > library(scrbook)
2257 > data(bbsdata)
2258
2259 > y <- bbsdata$counts[, "X90"] # Pick year 1990
```

```

2260 > notna <- !is.na(y)
2261 > y <- y[notna]
2262
2263 ## Forest cover already standardized here:
2264 > habitat <- bbsdata$counts[notna,"habitat"]
2265 > M <- length(y)
2266
2267 > library(R2WinBUGS)                      # Load R2WinBUGS
2268 > data <- list (y=y, M=M, habitat=habitat) # Bundle data for WinBUGS

```

2269 Now we write out the Poisson model specification in **WinBUGS** pseudo-code, provide
 2270 initial values, identify parameters to be monitored and then execute **WinBUGS**:

```

2271 > cat("
2272 model{
2273   for (i in 1:M){
2274     y[i] ~ dpois(lam[i])
2275     log(lam[i]) <- beta0+beta1*habitat[i]
2276   }
2277   beta0 ~ dunif(-5,5)
2278   beta1 ~ dunif(-5,5)
2279 }
2280 ",file="PoissonGLM.txt")

2281 > inits <- function() list ( beta0=rnorm(1),beta1=rnorm(1) )
2282 > parameters <- c("beta0","beta1")
2283 > out <- bugs(data, inits, parameters, "PoissonGLM.txt", n.thin=2,n.chains=2,
2284           n.burnin=2000,n.iter=6000,debug=TRUE,working.dir=getwd())

```

2285 The **WinBUGS** output can be viewed in **R** using the `print` command:

```

2286 print(out,digits=2)
2287 Inference for Bugs model at "PoissonGLM.txt", fit using WinBUGS,
2288 2 chains, each with 6000 iterations (first 2000 discarded), n.thin = 2
2289 n.sims = 4000 iterations saved
2290          mean    sd   2.5%   25%   50%   75%  97.5% Rhat n.eff
2291 beta0     3.15  0.02   3.10   3.13   3.15   3.17   3.20    1  4000
2292 beta1    -0.50  0.02  -0.54  -0.51  -0.50  -0.48  -0.46    1  4000
2293 deviance 1116.56 1.95 1115.00 1115.00 1116.00 1117.00 1122.00    1  4000

```

2294 3.6.3 Constructing your own MCMC algorithm

2295 At this point it might be helpful to suffer through an example building a custom MCMC
 2296 algorithm. Here, we develop an MCMC algorithm for the Poisson regression model, using
 2297 a Metropolis-within-Gibbs sampling framework. Building MCMC algorithms is covered in
 2298 more detail in Chapt. ?? where you can also find step-by-step instructions for Metropolis-
 2299 within-Gibbs samplers, should the following section move through all this material too
 2300 quickly.

2301 We will assume that the two parameters, β_0 and β_1 , have diffuse normal priors, say
 2302 $[\beta_0] = \text{Normal}(0, 100)$ and $[\beta_1] = \text{Normal}(0, 100)$ where each has *standard deviation* 100
 2303 (recall that **WinBUGS** parameterizes the normal in terms of $1/\sigma^2$). We need to assem-
 2304 ble the relevant elements of the model which are these two prior distributions and the
 2305 likelihood $[\mathbf{y}|\beta_0, \beta_1] = \prod_i [y_i|\beta_0, \beta_1]$ which is, mathematically, the product of the Poisson
 2306 pmf evaluated at each y_i , given particular values of β_0 and β_1 . Next, we need to identify
 2307 the full conditionals $[\beta_0|\beta_1, \mathbf{y}]$ and $[\beta_1|\beta_0, \mathbf{y}]$. We use the all-purpose rule for constructing
 2308 full conditionals (section ??) to discover that:

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

2309 Mathematically, the full conditional is of the form

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

2310 which you can program as an **R** function with arguments β_0 , β_1 and \mathbf{y} without difficulty.
 2311 The full-conditional for β_1 is:

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

2312 which has a similar mathematical representation except the prior is expressed in terms
 2313 of β_1 instead of β_0 . Remember, we could replace the “ \propto ” with “=” if we put $[y|\beta_1]$ or
 2314 $[y|\beta_0]$ in the denominator. But, in general, $[y|\beta_0]$ or $[y|\beta_1]$ will be quite a pain to compute
 2315 and, more importantly, it is a constant as far as the operative parameters (β_0 or β_1 ,
 2316 respectively) are concerned. Therefore, the MH acceptance probability will be the ratio
 2317 of the full-conditional evaluated at a candidate draw to that evaluated at the current
 2318 value, and so the denominator required to change \propto to $=$ winds up canceling from the
 2319 MH acceptance probability.

2320 Here we will use the so-called random walk candidate generator, which is a Normal
 2321 proposal distribution, so that, for example, $\beta_0^* \sim \text{Normal}(\beta_0^t, \delta)$ where δ is the standard-
 2322 deviation of the proposal distribution, which is just a tuning parameter that is set by
 2323 the user and adjusted to achieve efficient mixing of chains (see Sec. ??). We remark
 2324 also that calculations are often done on the log-scale to preserve numerical integrity of
 2325 things when quantities evaluate to small or large numbers, so keep in mind, for example,
 2326 $a * b = \exp(\log(a) + \log(b))$ for two positive numbers a and b . The “Metropolis within
 2327 Gibbs” algorithm for a Poisson regression turns out to be remarkably simple and is given
 2328 in Panel ?? . It is also part of the **scrbook** package and you can run 1000 iterations of it
 2329 by calling `PoisGLMBBS(y=y, habitat=habitat, niter=1000)` (note that y = point count
 2330 data and `habitat` = forest cover have to be defined in your **R** workspace as shown in the
 2331 previous analysis of these data).

2332 The first 300 iterations of the MCMC history of each parameter are shown in Fig. ?? .
 2333 These chains are not very appealing but a couple of things are evident: We see that the
 2334 burn-in takes about 250 iterations and that after that chains seem to mix reasonably well,
 2335 although this is not so clear given the scale of the y-axis, which we have chosen to get

```

> 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.

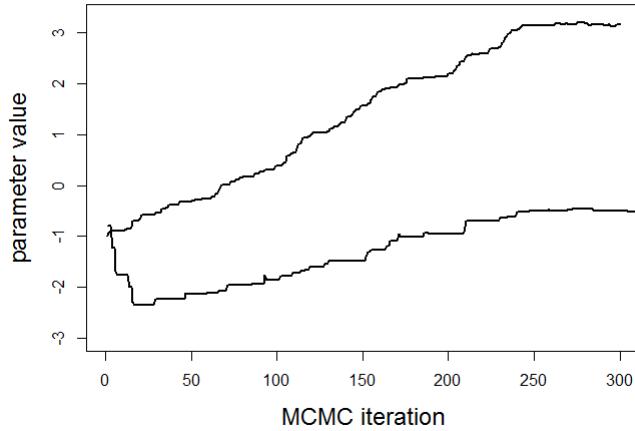


Figure 3.6. First 300 MCMC iterations for the Poisson GLM model parameters β_0 (top) and β_1 (bottom) using a Metropolis-Hastings tuning parameter of $\delta = 0.05$.

both variables on the same graph. We generated 10,000 posterior samples, discarding the first 500 as burn-in, and the result is shown in Fig. ??, this time on separate panels for each parameter. The “grassy” look of the MCMC history is diagnostic of Markov chains that are well-mixing and we would generally be very satisfied with results that look like this.

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

3.7 POISSON GLM WITH RANDOM EFFECTS

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

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

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

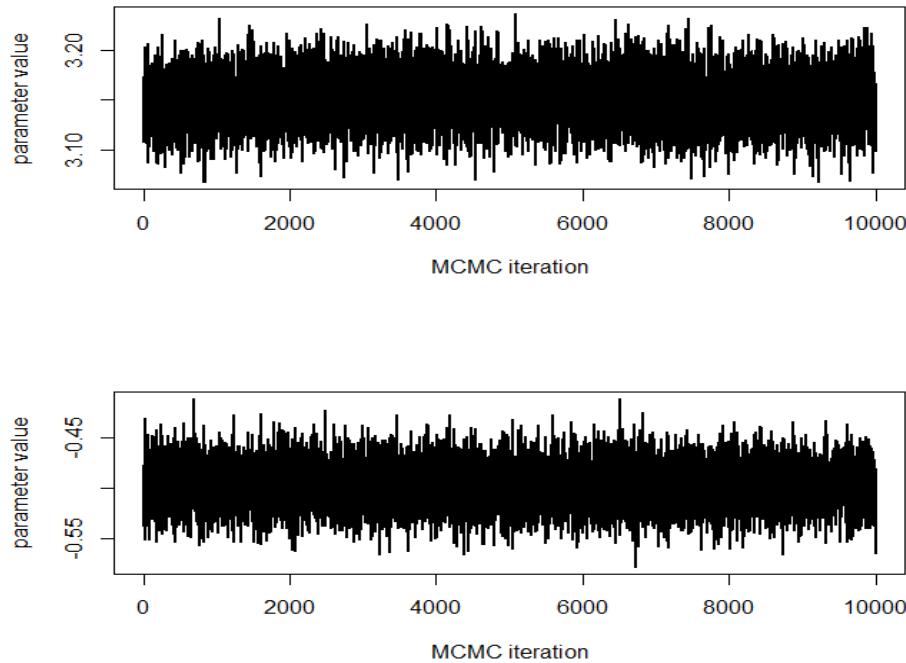


Figure 3.7. Nice grassy plots of 10,000 MCMC iterations for the Poisson GLM model parameters β_0 (top) and β_1 (bottom) using a Metropolis-Hastings tuning parameter of $\delta = 0.05$.

2353 where $\eta_i \sim \text{Normal}(0, \sigma^2)$. In this context, η could represent an error term capturing
 2354 variation in λ_i not accounted for by the covariates, or overdispersion. It is really amazingly
 2355 simple to express this model in the **BUGS** language and have **WinBUGS** (or **JAGS**,
 2356 etc..) draw samples from the posterior distribution. The code for analysis of the BBS
 2357 dove counts is given as follows:

```

2358 > library(scrbook)
2359 ### Grab the BBS Data as before
2360 > data(bbsdata)
2361 ### Set random seed so that results are repeatable
2362 > set.seed(2013)
2363 ### Dump the BUGS model into a file
2364 > cat("
2365 model{

```

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

```

2366   for (i in 1:M){  # Observation model, linear predictor, etc..
2367     y[i] ~ dpois(lam[i])
2368     log(lam[i]) <- beta0+ beta1*habitat[i] + eta[i]
2369     frog[i] <- beta1*habitat[i] + eta[i]
2370     eta[i] ~ dnorm(0,tau)
2371   }
2372   # Prior distributions:
2373   beta0 ~ dunif(-5,5)
2374   beta1 ~ dunif(-5,5)
2375   sigma ~ dunif(0,10)
2376   tau <- 1/(sigma*sigma)
2377 }
2378 ",file="model.txt")

2379 > data <- list ("y","M","habitat") # Define the data
2380 > inits <- function()           #    inits and parameters
2381   list ( beta0=rnorm(1), beta1=rnorm(1), sigma=runif(1,0,4))
2382 > parameters <- c("beta0","beta1","sigma","tau")
2383
2384 > library(R2WinBUGS)           # Load and run R2WinBUGS
2385 > out <- bugs (data, inits, parameters, "model.txt", n.thin=2,n.chains=2,
2386   n.burnin=1000, n.iter=5000, debug=TRUE)

```

2387 This produces the posterior summary statistics given in Table ???. One thing we notice
2388 is that the posterior standard deviations of the regression parameters are much higher,
2389 a result of the extra-Poisson variation allowed for by this model. We would also notice
2390 much less precise predictions of hypothetical new observations.

3.8 BINOMIAL GLMS

2391 Another extremely important class of models in ecology are binomial models. We use
2392 binomial models for count data whenever the observations are counts or frequencies and
2393 it is natural to condition on a “sample size”, say K , the maximum frequency possible in
2394 a sample. The random variable, $y \leq K$, is then the frequency of occurrences out of K

2395 “trials”. The parameter of the binomial models is p , often called “success probability”
 2396 which is related to the expected value of y by $\mathbb{E}(y) = pK$. Usually we are interested
 2397 in modeling covariates that affect the parameter p , and such models are called binomial
 2398 GLMs, binomial regression models or logistic regression, although logistic regression re-
 2399 ally only applies when the logistic link is used to model the relationship between p and
 2400 covariates (see below).

2401 One of the most typical binomial GLMs occurs when the sample size equals 1 and
 2402 the outcome, y , is “presence” ($y = 1$) or “absence” ($y = 0$) of a species. In this case, y
 2403 has a Bernoulli distribution. This is a classical species distribution modeling situation. A
 2404 special situation occurs when presence/absence is observed with error (??). In that case,
 2405 $K > 1$ samples are usually needed for effective estimation of model parameters.

2406 In standard binomial regression problems the sample size is fixed by design but inter-
 2407 esting models also arise when the sample size is itself a random variable. These are the
 2408 N -mixture models (????) and related models (in this case, N being the sample size, which
 2409 we labeled K above)⁷. Another situation in which the binomial sample size is “fixed” is
 2410 closed population capture-recapture models in which a population of individuals is sam-
 2411 pled K times. The number of times each individual is encountered is a binomial outcome
 2412 with parameter (encounter probability) p , based on a sample of size K . In addition, the
 2413 total number of unique individuals observed, n , is also a binomial random variable based
 2414 on population size N . We consider such models in Chapt. ??.

2415 3.8.1 Binomial regression

2416 In binomial models, covariates are modeled on a suitable transformation (the link function)
 2417 of the binomial success probability, p . Let x_i denote some measured covariate for sample
 2418 unit i and let p_i be the success probability for unit or subject i . The standard choice is the
 2419 logit link function (??) but there are many other possible link functions. We sometimes use
 2420 the complementary log-log (= “cloglog”) link function in ecological applications because
 2421 it is natural in some cases when the response should scale in relation to area or effort (?,
 2422 p. 150). As an example, the “probability of observing a count greater than 0” under a
 2423 Poisson model is $\Pr(y > 0) = 1 - \exp(-\lambda)$. In that case, for the i^{th} observation,

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

2424 so that if you have covariates in your linear predictor for $\mathbb{E}(y)$ under a Poisson model then
 2425 they are linear on the complementary log-log link of p . In models of species occurrence
 2426 it seems natural to view occupancy as being derived from local abundance N (??).
 2427 Therefore, models of local abundance in which $N_i \sim \text{Poisson}(A_i \lambda_i)$ for a habitat patch of
 2428 area A_i implies a model for occupancy ψ_i of the form

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

2429 We will use the cloglog link in some analyses of SCR models in Chapt. ?? and elsewhere.

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”

2430 **3.8.2 Example: waterfowl banding data**

2431 The standard binomial modeling problem in ecology is that of modeling species distributions,
2432 where $K = 1$ and the outcome is occurrence ($y = 1$) or not ($y = 0$) of some species.
2433 Such examples abound in books (e.g., ?, ch. 3; ?, ch. 21; ?, ch. 13) and in the literature.
2434 Therefore, instead, we will consider an example involving band returns of waterfowl in the
2435 upper great plains including some Canadian provinces, which were analyzed by ?.

2436 For these data, y_{it} is the number of mallard (*Anas platyrhynchos*) bands recovered out
2437 of B_{it} birds banded at some location s_i in year t . In this case B_{it} is fixed. Thinking about
2438 recovery rate as being proportional to harvest rate, we use these data to explore geographic
2439 gradients in recovery rate resulting from variability in harvest pressure experienced by
2440 different populations. As such, we fit a basic binomial GLM with a linear response to
2441 geographic coordinates (including an interaction term). Here we provide the part of the
2442 script for creating the model and fitting the model in **WinBUGS**. There are few structural
2443 differences between this model and the Poisson GLM fitted previously. The main things
2444 are due to the data structure (we have a matrix here instead of a vector) and otherwise
2445 we change the distributional assumption to binomial (specified with `dbin`) and then use
2446 the `logit` function to relate the parameter p_{it} to the covariates.

2447 **Dummy variables in BUGS:** In the mallard example, we model the band recovery
2448 probability p_{it} not only as a linear function (on the logit scale) of geographic location, but
2449 also allow for variation in p_{it} with year, t ; $t = 1, 2, \dots, T$. In this particular example there
2450 are $T = 5$ years of data and we could describe the full mallard model with a formula in
2451 terms of “dummy variables.” Dummy variables are binary variables, one variable for each
2452 level of the categorical variable they describe, such that variable for level t takes on the
2453 value 1 if the observation belongs with level t and 0 otherwise. So, the mallard model in
2454 terms of dummy variables for “year” looks like this:

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

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

2455 Here, x_2 to x_5 are the dummy variable vectors of length T that take on the value of 1
2456 when t corresponds to the respective year and 0 otherwise; β_0 is the common intercept
2457 term and corresponds to $t = 1$; $\beta_1 - \beta_4$ describe the difference in p_{it} for each t relative to
2458 $t = 1$.

2459 There is a more concise way of implementing such a model with a categorical covariate
2460 in **BUGS**, namely, by using indexing instead of dummy variables⁸. Essentially, instead of
2461 estimating the difference in p relative to category 1, we estimate a separate intercept term
2462 for each category, so that we have 5 different β_0 parameters indexed by t . This reduces
2463 the linear predictor to:

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

2464 The model can be implemented in the **BUGS** language for the mallard banding data
2465 using the following **R** script, provided in the `scrbook` package (see `help(mallard)`):

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 (?)

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

```

2466 > library(scrbook)
2467 > data(mallard)      # Load mallard data
2468
2469 > cat("
2470 model{
2471   for(t in 1:5){
2472     for (i in 1:nobs){
2473       y[i,t] ~ dbin(p[i,t], B[i,t])
2474       pl[i,t] <- beta0[t]+beta1*X[i,1]+beta2*X[i,2]+beta3*X[i,1]*X[i,2]
2475       p[i,t] <- exp(pl[i,t])/(1+exp(pl[i,t]))
2476     }
2477   }
2478   beta1 ~ dnorm(0,.001)
2479   beta2 ~ dnorm(0,.001)
2480   beta3 ~ dnorm(0,.001)
2481   for(t in 1:5){
2482     beta0[t] ~ dnorm(0,.001)
2483   }
2484 }
2485 ",file="BinomialGLM.txt")

2486 > library(R2WinBUGS)
2487 > data <- list(B=mallard$bandings, y=mallard$recoveries,
2488                  X=mallard$locs, nobs=nrow(mallard$locs))
2489 > inits <- function(){ list(beta0=rnorm(5),beta1=0,beta2=0,beta3=0) }
2490 > parms <- c('beta0','beta1','beta2','beta3')
2491 > out <- bugs(data, inits, parms,"BinomialGLM.txt", n.chains=3,
2492                 n.iter=2000, n.burnin=1000, n.thin=2, debug=TRUE)

```

2493 Look at the posterior summaries of model parameters in Table ???. The basic result

2494 suggests a negative east-west gradient and a positive south to north gradient of band
 2495 recovery probabilities, but no interaction. A map of the response surface is shown in Fig.
 2496 ??.

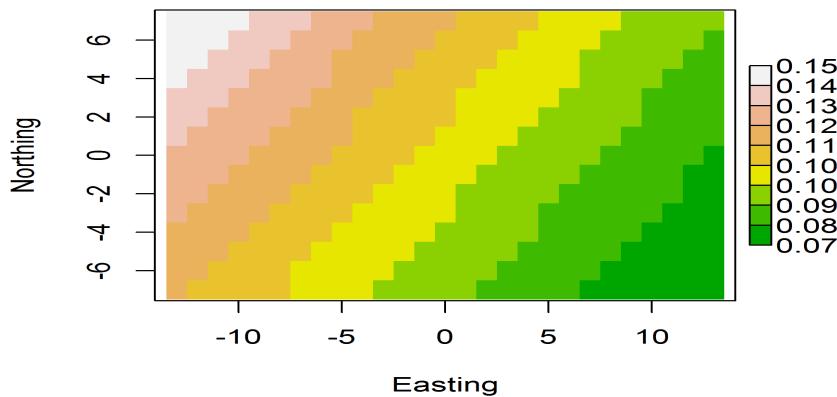


Figure 3.8. Predicted recovery rates of mallard bands in the upper great plains of North America. Note the negative gradient from the NW to the SE.

3.9 BAYESIAN MODEL CHECKING AND SELECTION

2497 In general terms, model checking – or assessing the adequacy of the model – and model
 2498 selection are quite thorny issues and, despite contrary and, sometimes, strongly held belief
 2499 among practitioners, there are not really definitive, general solutions to either problem.
 2500 We're against dogma on these issues and think people need to be open-minded about
 2501 such things and recognize that models can be useful whether or not they pass certain
 2502 statistical tests. Some models are intrinsically better than others because they make more
 2503 biological sense or foster understanding or achieve some objective that some bootstrap or
 2504 other goodness-of-fit test can't decide for you. That said, it gives you some confidence if
 2505 your model seems adequate in a purely statistical sense. We provide a very brief overview
 2506 of concepts here, but provide more detailed coverage in Chapt. ???. See also coverage
 2507 of these topics in ? and ? for specific context related to Bayesian model checking and
 2508 selection.

2509 3.9.1 Goodness-of-fit

2510 Goodness-of-fit testing is an important element of any analysis because our model rep-
 2511 resents a general set of hypotheses about the ecological and observation processes that

generated our data. Thus, if our model “fits” in some statistical or scientific sense, then we believe it to be consistent with the hypotheses that went into the model. More formally, we would conclude that the data are *not inconsistent* with the hypotheses, or that the model appears adequate. If we have enough data, then of course we will reject any set of statistical hypotheses. Conversely, we can always come up with a model that fits by making the model extremely complex. Despite this paradox, it seems to us that simple models that you can understand should usually be preferred even if they don’t fit, for example if they embody essential mechanisms central to our understanding of things, or if we think that some contributing factors to lack-of-fit are minor or irrelevant to the scientific context and intended use of the model. In other words, models can be useful 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. ??.

To evaluate goodness-of-fit in Bayesian analyses, we will most often use the Bayesian p-value (?). 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 MCMC iteration), the equivalent statistic is computed for a “new” data set, say \mathbf{y}^{new} , simulated using the current parameter values. From the new data set, we compute the same fit statistic:

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

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

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

2549 (?), where y_i is the observed value of observation i and $\mathbb{E}(y_i)$ its expected value. In
 2550 contrast to a Chi-square discrepancy, the Freeman-Tukey statistic removes the need to
 2551 pool cells with small expected values. In summary, you can see that the Bayesian p-value
 2552 is easy to compute, and it is widely used as a result.

2553 **3.9.2 Model selection**

2554 In ecology, scientific hypotheses are often manifest as different models or parameters of
 2555 a model, and so evaluating the importance of different models is fundamental to many
 2556 ecological studies. For Bayesian model selection we typically use three different methods:
 2557 First is, let's say, common sense. If a variable should plausibly be relevant to explaining
 2558 the data-generating processes, and it has posterior mass concentrated away from 0, then it
 2559 seems like it should be regarded as important - that is, it is "significant." This approach
 2560 seems to have fallen out of favor in ecology over the last 10 or 15 years but in many
 2561 situations it is a reasonable thing to do.

2562 For regression problems we sometimes use the indicator variable method of ?, in which
 2563 we introduce a set of binary variables w_k for variable k , and express the model as, e.g.,
 2564 for a single covariate model:

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

2565 where w_1 is given a Bernoulli prior distribution with some prescribed probability. E.g.,
 2566 $w_1 \sim \text{Bernoulli}(0.50)$ to provide a prior probability of 0.50 that variable x should be an
 2567 element of the linear predictor. The posterior probability of the event $w_1 = 1$ is a gage of
 2568 the importance of the variable x . i.e., high values of $\Pr(w_1 = 1)$ indicate stronger evidence
 2569 to support that " x is in the model" whereas values of $\Pr(w_1 = 1)$ close to 0 suggest that
 2570 x is less important. Expansion of the model to include the binary variable w_1 defines a
 2571 set of 2 distinct models for which we can directly compute the posterior probabilities for,
 2572 merely by tallying up the posterior frequency of w_1 . See ?, Chapt. 3 for an example in
 2573 the context of logistic regression.

2574 This approach seems to even work sometimes with fairly complex hierarchical models
 2575 of a certain form. E.g., ? applied it to a random effects model to evaluate the importance
 2576 of the random effect component of the model. The main problem, which is really a general
 2577 problem in Bayesian model selection, is that its effectiveness and results will typically be
 2578 highly sensitive to the prior distribution on the structural parameters (e.g., see ?, table
 2579 3.6). The reason for this is obvious: If $w_1 = 0$ for the current iteration of the MCMC
 2580 algorithm, so that β is sampled from the prior distribution, and the prior distribution is
 2581 very diffuse, then extreme values of β are likely. Consequently, when the current value of
 2582 β is far away from the mass of the posterior when $w_1 = 1$, then the Markov chain may
 2583 only jump from $w_1 = 0$ to $w_1 = 1$ infrequently. One seemingly reasonable solution to this
 2584 problem is to fit the full model to obtain posterior distributions for all parameters, and
 2585 then use those as prior distributions in a "model selection" run of the MCMC algorithm
 2586 (?). This seems preferable to more-or-less arbitrary restriction of the prior support to
 2587 improve the performance of the MCMC algorithm.

2588 A third method that we advocate is subject-matter context. It seems that there
 2589 are some situations – some models – where one should not have to do model selection
 2590 because a specific model may be necessitated by the biological context of the problem,
 2591 thus rendering a formal hypothesis test pointless (?). Certain aspects of SCR models are

such an example. In SCR models, we will see that “spatial location” of individuals is an element of the model. The simpler, reduced, model is an ordinary capture-recapture model which is not spatially explicit (i.e., Chapt. ??), but it seems silly and pointless to think about actually using the reduced model even if we could concoct some statistical test to refute the more complex model. The simpler model is manifestly wrong but, more importantly, not even a plausible data-generating model! Other examples are when effort, area or sample rate is used as a covariate. One might prefer to have such things in models regardless of whether or not they pass some statistical litmus test.

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

3.10 SUMMARY AND OUTLOOK

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

We note that GL(M)s are routinely analyzed by likelihood methods but we have focused on Bayesian analysis here in order to develop the tools that are less familiar to most ecologists, and that we will apply in much of the remainder of the book. In particular, Bayesian analysis of models with random effects is relatively straightforward because the models are easy to analyze conditional on the random effect, using MCMC. Thus, we will often analyze SCR models in later chapters by MCMC, explicitly adopting a Bayesian inference framework. In that regard, the various **BUGS** engines (**WinBUGS**, **OpenBUGS**, **JAGS**; see also Appendix 1) are enormously useful because they provide an accessible platform for carrying out analyses by MCMC by just describing the model, and not having to worry about how to actually build MCMC algorithms. That said, the **BUGS** language is more important than just to the extent that it enables one to do MCMC - it is useful as a modeling tool because it fosters understanding, in the sense that it forces you to become intimate with your model. You have to think about and write down all of the probability assumptions, and the relationships between observations and latent variables and parameters in a way that is ecologically sensible and statistically coherent. Because of this, it focuses your thinking on *model construction*, as M. Kéry says in his **WinBUGS** book (?), “**WinBUGS** frees the modeler in you.”

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

2634
2635

4

2636

CLOSED POPULATION MODELS

2637 In this chapter we introduce ordinary *non-spatial* capture-recapture (CR) models for es-
2638 timating population size in closed populations. A closed population is one whose size, N ,
2639 does not change during the study. Two forms of closure are often discussed: demographic
2640 closure, meaning that no births or deaths occur, and geographic closure, which states
2641 that no individuals move onto or off of the sampled area during the study. Although few
2642 populations are actually closed except during very short time intervals, closed population
2643 CR models serve as the basis for the development of the rest of the models presented in
2644 this book, including the models for open populations discussed in Chapt. ??.

2645 We begin with the most basic capture-recapture model, colloquially referred to as
2646 “model M_0 ” (?), in which encounter probability is strictly constant in all respects (across
2647 individuals, and replicates). This allows us to highlight the basic structure of closed
2648 population models as binomial GLMs. We then consider some important extensions of
2649 ordinary closed population models that accommodate various types of “individual effects”
2650 — either in the form of explicit, observed covariates (sex, age, body mass) or unstructured
2651 “heterogeneity” in the form of an individual random effect, which represent unobserved or
2652 unmeasured covariates. A special type of individual covariate models is distance sampling,
2653 which could be thought of as the most primitive spatial capture-recapture model. All of
2654 these different types of closed population models are closely related to binomial (or logistic)
2655 regression-type models. In fact, when N is known, they are precisely logistic regression
2656 models.

2657 We emphasize Bayesian analysis of capture-recapture models and we accomplish this
2658 using a method related to classical “data augmentation” from the statistics literature
2659 (e.g., ?). This is a general concept in statistics but, in the context of capture-recapture
2660 models where N is unknown, it has a consistent implementation across classes of capture-
2661 recapture models and one that is really convenient from the standpoint of doing MCMC
2662 (??). We use data augmentation throughout this book and thus emphasize its conceptual
2663 and technical origins and demonstrate applications to closed population models. We refer
2664 the reader to ?, ch. 6 for an accessible and complementary development of Bayesian
2665 analysis of ordinary, i.e., nonspatial closed population models.

4.1 THE SIMPLEST CLOSED POPULATION MODEL: MODEL M_0

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

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

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

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

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

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

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

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

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

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

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

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 ?? 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
$N - n_0$	6	2	3	1	0	0

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

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

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

2720 to as “cell probabilities” or in the specific context of capture-recapture, encounter history
 2721 probabilities.

2722 To fit the model in which N is *unknown*, we can regard n_0 as a parameter and maximize
 2723 the multinomial likelihood directly. Direct likelihood analysis of the multinomial model is
 2724 straightforward, but that is not always sufficiently useful in practice because we seldom
 2725 are concerned with models for the aggregated encounter history frequencies, which entail
 2726 that capture probabilities are the same for all individuals. In many instances, including
 2727 for spatial capture-recapture (SCR) models, we require a formulation of the model that
 2728 can accommodate individual-level covariates to account for differences in detection among
 2729 individuals, which we address subsequently in this chapter, and also in Chapt. ??.

2730 **4.1.1 The core capture-recapture assumptions**

2731 This basic capture-recapture model – model M_0 – comes with it a host of specific biological
 2732 and statistical assumptions. In addition to the basic assumption of population closure, ?
 2733 list the following:

- 2734 1. animals do not lose their marks during the experiment,
 2735 2. all marks are correctly noted and recorded at each trapping occasion, and
 2736 3. each animal has a constant and equal probability of capture on each trapping oc-
 2737 casion.

2738 The remainder of their classic work is dedicated to relaxing assumption 3. While assump-
 2739 tions 1 and 2 are undoubtedly necessary for inference from basic CR methods to be valid,
 2740 and while they are also assumed by most of the models we present in the following chap-
 2741 ters, we refrain from repeatedly making such statements. Our opinion is that all model
 2742 assumptions are apparent when a model is clearly specified, and it is both redundant and
 2743 impossible to list all the things not allowed by the model. For example, closed population
 2744 models also assume that other sources of error do not occur, but it is not necessary to
 2745 enumerate each possibility. Rather, it is necessary to make clear statements such as

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

2746 This simple model description carries a tremendous amount of information, and it leaves
 2747 very little left to say with respect to assumptions. Although we will not always show
 2748 the *iid* symbol, it will be assumed unless otherwise noted, and this assumption is critical
 2749 for valid inference. It implies that the encounter of one individual does not affect the
 2750 encounter of another individual, and encounter does not affect future encounter. Under
 2751 this assumption, it is easy to write down the likelihood of the parameters and obtain
 2752 parameter estimates; however, whether or not it is true depends upon biological and
 2753 sampling issues. If this assumption is deemed false, the model can be discarded in favor
 2754 of a more realistic alternative. However, once we have settled on our model, statistical
 2755 inference proceeds by assuming the model is truth—not an approximation to truth—but
 2756 actual truth.

2757 In spite of the fact that we assume that all models are truth, but we acknowledge that
 2758 all models are wrong due to their assumptions, assumptions should not be viewed as a
 2759 necessary evil. In fact, one way to view assumptions is as embodiments of our ecological
 2760 hypotheses. If we make these assumptions too complex or too specific, then we will never

2761 be able to study general phenomena that hold true across space and time. Furthermore,
 2762 in practice, we will rarely have enough data to estimate the parameters of highly complex
 2763 models.

2764 **4.1.2 Conditional likelihood**

2765 We saw that the closed population model is a simple logistic regression model if N is known
 2766 and, when N is unknown, the model is multinomial with index or sample size parameter N .
 2767 This multinomial model, being conditional on N , is sometimes referred to as the “joint
 2768 likelihood” the “full likelihood” or the “unconditional likelihood” (sometimes “model”
 2769 in place of “likelihood”) (??). This formulation differs from the so-called “conditional
 2770 likelihood” approach in which the likelihood of the observed encounter histories is devised
 2771 conditional on the event that an individual is captured at least once. To construct this
 2772 likelihood, we have to recognize that individuals appear or not in the sample based on the
 2773 value of the random variable y_i , that is, if and only if $y_i > 0$. The observation model is
 2774 therefore based on $\Pr(y|y > 0)$. For the simple case of model M_0 , the resulting conditional
 2775 distribution is a “zero truncated” binomial distribution which accounts for the fact that
 2776 we cannot observe the value $y = 0$ in the data set. Both the conditional and unconditional
 2777 models are legitimate modes of analysis in all capture-recapture types of studies. They
 2778 provide equally valid descriptions of the data and, for many practical purposes provide
 2779 equivalent inferences, at least in large sample sizes (?).

2780 In this book we emphasize Bayesian analysis of capture-recapture models using data
 2781 augmentation (described in Sec. ?? below), which produces yet a third distinct formu-
 2782 lation of capture-recapture models based on the zero-*inflated* binomial distribution that
 2783 we describe in the next section. Thus, there are 3 distinct formulations of the model – or
 2784 modes of analysis – for analyzing all capture-recapture models based on the (1) binomial
 2785 model for the joint or unconditional specification; (2) zero-truncated binomial that arises
 2786 “conditional on n ”; and (3) the zero-inflated binomial that arises under data augmen-
 2787 tation. Each formulation has distinct model parameters (shown in Table ?? for model
 2788 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

2789 We consider a method of analyzing closed population models using parameter-expanded
 2790 data augmentation (PX-DA), which we abbreviate to “data augmentation” or DA, which

is useful for Bayesian analysis and, in particular, analysis of models using the various **BUGS** engines and other Bayesian model fitting software. Data augmentation is a general statistical concept that is widely used in statistics in many different settings. The classical reference is ?, but see also ?. Data augmentation can be adapted to provide a very generic framework for Bayesian analysis of capture-recapture models with unknown N . This idea was introduced for closed populations by ?, and has subsequently been applied to a number of different contexts including individual covariate models (?), open population models (???), spatial capture-recapture models (???), and many others. ?, Chaps. 6 and 10 provide a good introduction to data augmentation in the context of closed and open population models.

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

4.2.1 DA links occupancy models and closed population models

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

In contrast to occupancy studies, in classical closed population studies, we observe a data set as in the middle panel of Table ?? 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

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

not be obvious. However, the choice of M induces a uniform prior for N on the integers $[0, M]$ (?). 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 ??, 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:

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

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

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

2869 function of these latent variables:

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

2870 Further, we note that the latent z_i parameters *can be* removed from the model by integration, in which case the joint probability of the data is

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

2872 Interpreted as a likelihood, we can directly maximize this expression to obtain the MLEs
2873 of the structural parameters ψ and p or those of other more complex models (e.g., see
2874 ?). We could estimate these parameters and then use them to obtain an estimator of N
2875 using the so-called “Best unbiased predictor” (see ?). Normally, however, we will analyze
2876 the model in its “conditional-on- z ” form using methods of MCMC either in the **BUGS**
2877 engines or using our own MCMC algorithms (see Chapt. ??).

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

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

2878 **4.2.2 Model M_0 in BUGS**

2879 It is helpful to understand data augmentation by seeing what its effect is on implementing
2880 model M_0 . For this model, in which we can aggregate the encounter data to individual-

2881 specific encounter frequencies, the augmented data are given by the vector of frequencies
 2882 $(y_1, \dots, y_n, 0, 0, \dots, 0)$ where the augmented values of $y = 0$ represent the encounter frequency
 2883 for potential individuals y_{n+1}, \dots, y_M . The zero-inflated model of the augmented data combines the model of the latent variables, $z_i \sim \text{Bernoulli}(\psi)$. The **BUGS** model
 2884 description of the closed population model M_0 is shown in Panel ???. The last line of the
 2885 model specification provides the expression for computing N from the data augmentation
 2886 variables z_i . Note that, to improve readability of code snippets (especially of large ones),
 2887 we will sometimes deviate from our standard notation a bit. In this case we use `nind`
 2888 for n (the number of encountered individuals), and $M = nind + nz$ is the total size of the
 2889 augmented data set. In other cases we might also use `nocc` in place of K and `ntraps`
 2890 in place of J . We find that word definitions make code easier to understand, especially
 2891 without having to read surrounding text.

```
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 .

2893 Specification of a more general model in terms of the individual encounter observations
 2894 y_{ik} is not much more difficult than for the individual encounter frequencies. We define
 2895 the observation model by a double loop and change the indexing of quantities accordingly,
 2896 i.e.,

```
2897 for(i in 1:(nind+nz)){
  2898   z[i] ~ dbern(psi)
  2899   for(k in 1:K){
  2900     mu[i,k] <- z[i]*p
  2901     y[i,k] ~ dbin(mu[i,k],1)
  2902 }
```

2903 }

2904 In this manner, it is straightforward to incorporate covariates on p for both individuals
 2905 and sampling occasions (see discussion of this below and also Chapt. ??) as well as to
 2906 devise other extensions of the model, including models for open populations (see Chapt.
 2907 ??).

2908 **4.2.3 Formal development of data augmentation (DA)**

2909 Use of parameter-expanded data augmentation (PX-DA), or DA for short, for solving
 2910 inference problems with unknown N can be justified as originating from the choice of a
 2911 uniform prior on N . The Uniform(0, M) prior for N is innocuous in the sense that the
 2912 posterior associated with this prior is equal to the likelihood for sufficiently large M . One
 2913 way of inducing the Uniform(0, M) prior on N is by assuming the following hierarchical
 2914 prior:

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

2915 The model assumptions, specifically the multinomial model (Eq. ??) and Eq. ??, may be
 2916 combined to yield a reparameterization of the conventional model that is appropriate for
 2917 the augmented data set of known size M :

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

2918 This expression arises by removing N from Eq. ?? by integrating over the binomial
 2919 prior distribution for N . Thus, the models we analyze under data augmentation arise
 2920 formally by removing the parameter N from the ordinary closed-population model, which
 2921 is conditional on N , by integrating over a binomial prior distribution for N .

2922 Note that the $M - n$ unobserved individuals in the augmented data set have probability
 2923 $\psi\pi(0) + (1 - \psi)$, indicating that these unobserved individuals are a mixture of individuals
 2924 that are sampling zeros ($\psi\pi_0$), and belong to the population of size N , and others that
 2925 are “structural zeros” (occurring in the augmented data set with probability $1 - \psi$). In
 2926 Eq. ??, N has been eliminated as a formal parameter of the model by marginalization
 2927 (integration) and replaced with the new parameter ψ , the data augmentation parameter.
 2928 However, the full likelihood containing both N and ψ can also be analyzed (see ?).

2929 **4.2.4 Remarks on data augmentation**

2930 Data augmentation may seem like a strange and mysterious black-box, and likely it is un-
 2931 familiar to most people, even to many of those with substantial experience with capture-
 2932 recapture models. However, it really is just a formal reparameterization of capture-
 2933 recapture models in which N is marginalized out of the ordinary (conditional-on- N) model
 2934 (by summation over a binomial prior). As a result, we could refer to the resulting model
 2935 as the “binomial-integrated likelihood” to reflect that an estimator could be obtained
 2936 from the ordinary likelihood, integrated over a binomial prior. Other such “integrated
 2937 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. ??).

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. ??), 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 (?). 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 (?????). 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 (???) which we consider in Chapt. ?. 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 ?? and ?. The specific data used here are encounter histories on 47 individuals obtained from an array of 38 baited “hair snares” (Fig. ??) 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)`

2982 at the **R** prompt, and the analysis can be set up and run as follows (see `?beardata` for the
 2983 commands to do the analysis). Here, the data were augmented with 128 all-zero encounter
 2984 histories, resulting in a total sample size of $M = 175$.

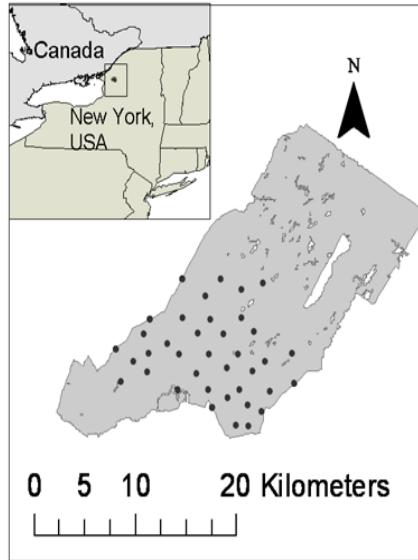


Figure 4.1. Fort Drum Black bear study area and the 38 baited hair snare locations operated for 8 weeks during June and July, 2006.

```

2985 > library(scrbook)
2986 > data(beardata)           # load the bear data and extract components
2987 > trapmat <- beardata$trapmat
2988 > nind <- dim(beardata$bearArray)[1]
2989 > K <- dim(beardata$bearArray)[3]
2990 > ntraps <- dim(beardata$bearArray)[2]
2991
2992 > M <- 175
2993 > nz <- M-nind
2994 > Yaug <- array(0, dim=c(M,ntraps,K))
2995
2996 > Yaug[1:nind,,] <- beardata$bearArray
2997 > y <- apply(Yaug,c(1,3),sum) # summarize by ind x rep
2998 > y[y>1] <- 1             # toss out multiple encounters per occasion
2999   # b/c traditional CR models ignore space
  
```

3000 The raw data object, `beardata$bearArray` is a 3-dimensional array `nind × ntraps × K`
 3001 of individual encounter events (i.e., $y_{ijk} = 1$ if individual i was encountered in trap j during

3002 occasion k , and 0 otherwise). For fitting model M_0 (or M_h , see below), it is sufficient to
 3003 reduce the data to individual encounter frequencies which we have re-labeled “y” above.
 3004 The **BUGS** model file along with commands to fit the model are as follows:

```

3005 > set.seed(2013)                      # to obtain the same results each time
3006 > library(R2WinBUGS)                   # load R2WinBUGS, set-up:
3007 > data0 <- list(y=y, M=M, K=K)        # data ....
3008 > params0 <- c('psi','p','N')          # parameters ....
3009 > zst <- c(rep(1,nind),rbinom(M-nind, 1, .5)) # inits ....
3010 > inits <- function(){ list(z=zst, psi=runif(1), p=runif(1)) }
3011
3012 > cat("
3013 model{
3014
3015   psi ~ dunif(0, 1)
3016   p ~ dunif(0,1)
3017
3018   for (i in 1:M){
3019     z[i] ~ dbern(psi)
3020     for(k in 1:K){
3021       tmp[i,k] <- p*z[i]
3022       y[i,k] ~ dbin(tmp[i,k],1)
3023     }
3024   }
3025   N<-sum(z[1:M])
3026 }
3027 ",file="modelM0.txt")
3028
3029 ## Run the model:
3030 > fit0 <- bugs(data0, inits, params0, model.file="modelM0.txt",n.chains=3,
3031   n.iter=2000, n.burnin=1000, n.thin=1,debug=TRUE,working.directory=getwd())

```

3032 This produces the following posterior summary statistics:

```

3033 > print(fit0,digits=2)
3034 Inference for Bugs model at "modelM0.txt", fit using WinBUGS,
3035 3 chains, each with 2000 iterations (first 1000 discarded)
3036 n.sims = 3000 iterations saved
3037      mean    sd   2.5%   25%   50%   75% 97.5% Rhat n.eff
3038 psi      0.29  0.04   0.22   0.26   0.29   0.31   0.36   1  3000
3039 p        0.30  0.03   0.25   0.28   0.30   0.32   0.35   1  3000
3040 N        49.94 1.99  47.00  48.00  50.00  51.00  54.00   1  3000
3041 deviance 489.05 11.28 471.00 480.45 488.80 495.40 513.70   1  3000
3042
3043 [... some output deleted ...]

```

3044 **WinBUGS** did well in choosing an MCMC algorithm for this model – we have $\hat{R} = 1$
 3045 for each parameter, and an effective sample size of 3000, equal to the total number of

3046 posterior samples³. We see that the posterior mean of N under this model is 49.94 and
 3047 a 95% posterior interval is (48, 54). We revisit these data later in the context of more
 3048 complex models.

3049 In order to obtain an estimate of density, D , we need an area to associate with the
 3050 estimate of N , and in Chapt. 1 we already went through a number of commonly used
 3051 procedures to conjure up such an area, including buffering the trap array by the home
 3052 range radius, often estimated by the mean maximum distance moved (MMDM) (?), $1/2$
 3053 MMDM (?) or directly from telemetry data (?). Typically, the trap array is defined by
 3054 the convex hull around the trap locations, and this is what we applied a buffer to. We
 3055 computed the buffer by using a telemetry-based estimate of the mean female home range
 3056 radius (2.19 km) (?) instead of using an estimate based on our relatively more sparse
 3057 recapture data. For the Fort Drum study, the convex hull has an area of 157.135 km^2 ,
 3058 and the buffered convex hull has an area of 277.011 km^2 . To create this we used functions
 3059 contained in the **R** package **rgeos** and created a utility function **bcharea** which is in our
 3060 **R** package **scrbook**. The commands are as follows:

```
3061 > library(rgeos)
3062
3063 > bcharea <- function(buff,traplocs){
3064   p1 <- Polygon(rbind(traplocs,traplocs[1,]))
3065   p2 <- Polygons(list(p1=p1),ID=1)
3066   p3 <- SpatialPolygons(list(p2=p2))
3067   p1ch <- gConvexHull(p3)
3068   bp1 <- (gBuffer(p1ch, width=buff))
3069   plot(bp1, col='gray')
3070   plot(p1ch, border='black', lwd=2, add=TRUE)
3071   gArea(bp1)
3072 }
3073
3074 > bcharea(2.19,traplocs=trapmat)
```

3075 The resulting buffered convex hull is shown in Fig. ??.

3076 To conjure up a density estimate under model M_0 , we compute the appropriate pos-
 3077 terior summary of the ratio of N and the prescribed area (277.011 km^2):

```
3078 > summary(fit0$sims.list$N/277.011)
3079   Min. 1st Qu. Median Mean 3rd Qu. Max.
3080   0.1697  0.1733  0.1805  0.1803  0.1841  0.2130
3081
3082 > quantile(fit0$sims.list$N/277.011,c(0.025,0.975))
3083   2.5%    97.5%
3084 0.1696684 0.1949381
```

3085 which yields a density estimate of about 0.18 ind/km^2 , and a 95% Bayesian confidence
 3086 interval of (0.170, 0.195). Our estimate of density should be reliable if we have faith in
 3087 our stated value of the “sampled area”. Clearly though this is largely subjective, and not
 3088 something we can formally evaluate (or estimate) from the data based on model M_0 .

³This is even a little suspicious....

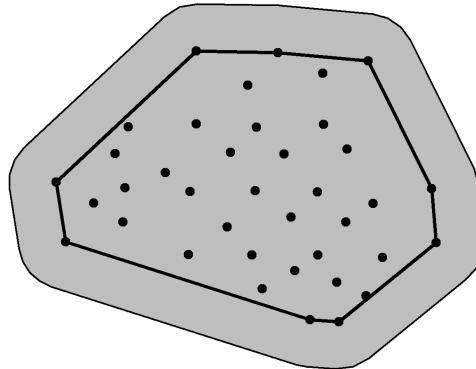


Figure 4.2. Convex hull of the bear hair snare array at Fort Drum, NY, buffered by mean female home range radius (2.19 km).

4.3 TEMPORALLY VARYING AND BEHAVIORAL EFFECTS

3089 The purpose of this chapter is mainly to emphasize the central importance of the binomial
 3090 model in capture-recapture and so we have considered models for individual encounter
 3091 frequencies—the number of times individuals are captured out of K occasions. Sometimes
 3092 we can't aggregate the encounter data for each individual, such as when encounter
 3093 probability varies over time among samples. Time-varying responses that are relevant in
 3094 many capture-recapture studies are “effort” such as amount of search time, number of
 3095 observers, or trap nights, or encounter probability varying over time, as a function of date
 3096 or season (?) due to species behavior. A common situation in many animal studies is
 3097 that in which there exists a “behavioral response” to trapping (even if the animal is not
 3098 physically trapped).

3099 Behavioral response is an important concept in animal studies because individuals
 3100 might learn to come to baited traps or avoid traps due to trauma related to being encoun-
 3101 tered. There are a number of ways to parameterize a behavioral response to encounter.
 3102 The distinction between persistent and ephemeral was made by ? who considered a general
 3103 behavioral response model of the form:

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

3104 where x_{ik} is a covariate indicator variable of previous capture (i.e., $x_{ik} = 1$ if captured
 3105 in any previous period). Therefore, encounter probability changes depending on whether

3106 an individual was captured in the immediate previous period (a Markovian or ephemeral
 3107 behavioral response; (?)), described by the term $\alpha_1 y_{i,k-1}$ or in *any* previous period (persis-
 3108 tent behavioral response), described by the term $\alpha_2 x_{ik}$. Because spatial capture-recapture
 3109 models allow us to include trap-specific covariates, we can describe a 3rd type of behavioral
 3110 response—a local behavioral response that is trap-specific (?). In this local behavioral re-
 3111 sponse, the encounter probability is modified for an individual trap depending on previous
 3112 capture in that trap. Models with temporal effects are easy to describe and analyze in
 3113 the **BUGS** language and we provide a number of examples in Chapt. ?? and elsewhere.

4.4 MODELS WITH INDIVIDUAL HETEROGENEITY

3114 Models in which encounter probability varies by individual have a long history in capture-
 3115 recapture and, indeed, this so-called “model M_h ” is one of the elemental capture-recapture
 3116 models in (?). Conceptually, we imagine that the individual-specific encounter probability
 3117 parameters, p_i , are random variables distributed according to some probability distribu-
 3118 tion, $[\theta]$. We denote this basic model assumption as $p_i \sim [\theta]$. This type of model is similar
 3119 in concept to extending a GLM to a GLMM but in the capture-recapture context N is
 3120 unknown. The basic class of models is often referred to as “model M_h ” (“h” for hetero-
 3121 geneity), but really this is a broad class of models, each being distinguished by the specific
 3122 distribution assumed for p_i . There are many different varieties of model M_h including
 3123 parametric and various non-parametric approaches (???). One important practical mat-
 3124 ter is that estimates of N can be extremely sensitive to the choice of heterogeneity model
 3125 (???). Indeed, ? showed that in some cases it’s possible to find models that yield precisely
 3126 the same expected data, yet produce wildly different estimates of N . In that sense, N for
 3127 most practical purposes is not identifiable across classes of different heterogeneity models,
 3128 and this should be understood before fitting any such model. One solution to this problem
 3129 is to seek to model explicit factors that contribute to heterogeneity, e.g., using individual
 3130 covariate models (See ?? below). Indeed, spatial capture-recapture models do just that, by
 3131 modeling heterogeneity due to the spatial organization of individuals in relation to traps
 3132 or other encounter mechanism. For additional background and applications of model M_h
 3133 see ?, Chapt. 6 and ?, Chapt. 6.

3134 We will work with a specific type of model M_h here which is a natural extension of
 3135 the basic binomial observation model of model M_0 so that

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

3136 where μ is a fixed parameter (the mean) to be estimated, and η_i is an individual random
 3137 effect assumed to be normally distributed:

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

3138 We could as well combine these two steps and write $\text{logit}(p_i) \sim \text{Normal}(\mu, \sigma_p^2)$. This
 3139 “logit-normal mixture” was analyzed by ? and elsewhere. It is a natural extension of the
 3140 basic model with constant p , as a mixed GLMM, and similar models occur throughout
 3141 statistics. It is also natural to consider a beta prior distribution for p_i (?) and so-called
 3142 “finite-mixture” models are also popular (??). In the latter, individuals are assumed to
 3143 belong to a finite number of latent classes, each of which has its own capture probability.

3144 Model M_h has important historical relevance to spatial capture-recapture situations
 3145 (?) because investigators recognized that the juxtaposition of individuals with the array
 3146 of trap locations should yield heterogeneity in encounter probability, and thus it became
 3147 common to use some version of model M_h in spatial trapping arrays to estimate N . While
 3148 this doesn't resolve the problem of not knowing the effective sample area, it does yield
 3149 an estimator that accommodates the heterogeneity in p induced by the spatial aspect of
 3150 capture-recapture studies. To see how this juxtaposition induces heterogeneity, we have
 3151 to understand the relevance of movement in capture-recapture models. Imagine a quadrat
 3152 that can be uniformly searched by a crew of biologists for some species of reptile (see ?).
 3153 Figure ?? shows a sample quadrat searched repeatedly over a period of time. Further,
 3154 suppose that the species exhibits some sense of spatial fidelity in the form of a home range
 3155 or territory, and individuals move about their home range (home range centroids are given
 3156 by the solid dots) in some kind of random fashion. Heuristically, we imagine that each
 3157 individual in the vicinity of the study area is liable to experience variable exposure to
 3158 encounter due to the overlap of its home range with the sampled area - essentially the
 3159 long-run proportion of times the individual is within the sample plot boundaries, say ϕ .
 3160 We might model the exposure or *availability* of an individual to capture by supposing that
 3161 $a_i = 1$ if individual i is available to be captured (i.e., within the survey plot) during any
 3162 sample, and 0 otherwise. Then, $\Pr(a_i = 1) = \phi$. In the context of spatial studies, it is
 3163 natural that ϕ should depend on *where* an individual lives, i.e., it should be individual-
 3164 specific ϕ_i (?). This system describes, precisely, that of "random temporary emigration"
 3165 (?) where ϕ_i is the individual-specific probability of being "available" for capture.

3166 Conceptually, SCR models aim to deal with this problem of variable exposure to sam-
 3167 pling due to movement in the proximity of the trapping array explicitly and formally with
 3168 auxiliary spatial information. If individuals are detected with probability p_0 , *conditional*
 3169 on $a_i = 1$, then the marginal probability of detecting individual i is

$$p_i = p_0 \phi_i$$

3170 so we see clearly that individual heterogeneity in encounter probability is induced as a re-
 3171 sult of the juxtaposition of individuals (i.e., their home ranges) with the sample apparatus
 3172 and the movement of individuals about their home range.

3173 4.4.1 Analysis of model M_h

3174 If N is known, it is worth taking note of the essential simplicity of model M_h as a binomial
 3175 GLMM. This is a type of model that is widely applied throughout statistics using stan-
 3176 dard methods of inference based either on integrated likelihood (??), which we discuss in
 3177 Chapt. ??, or standard Bayesian methods. However, because N is not known, inference is
 3178 somewhat more challenging. We address that here using Bayesian analysis based on data
 3179 augmentation. Although we use data augmentation in the context of Bayesian methods
 3180 here, we note that heterogeneity models formulated under DA are easily analyzed by con-
 3181 ventional likelihood methods as zero-inflated binomial mixtures (?) and more traditional
 3182 analysis of model M_h based on integrated likelihood, without using data augmentation,
 3183 has been considered by ?, ?, and others.

3184 As with model M_0 , we have the Bernoulli model for the zero-inflation variables: $z_i \sim$
 3185 Bernoulli(ψ) and the model of the observations expressed conditional on these latent

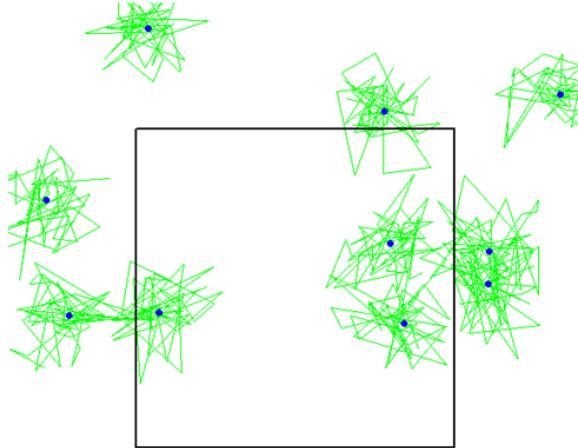


Figure 4.3. A quadrat searched for lizards over some period of time (simulated data). The locations of encounter for each of 10 lizards are connected by lines—the dots are activity centers.

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

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

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

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

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

4.4.2 Analysis of the Fort Drum data with model M_h

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

```

3199 [... get data as before ....]
3200
3201 > set.seed(2013)
3202
3203 > cat("
3204 model{
3205   p0 ~ dunif(0,1)          # prior distributions
3206   mup <- log(p0/(1-p0))
3207   sigmap ~ dunif(0,10)
3208   taup <- 1/(sigmap*sigmap)
3209   psi ~ dunif(0,1)
3210
3211   for(i in 1:(nind+nz)){
3212     z[i] ~ dbern(psi)      # zero inflation variables
3213     lp[i] ~ dnorm(mup,taup) # individual effect
3214     logit(p[i]) <- lp[i]
3215     mu[i] <- z[i]*p[i]
3216     y[i] ~ dbin(mu[i],K)   # observation model
3217   }
3218
3219   N<-sum(z[1:(nind+nz)])
3220 }
3221 ",file="modelMh.txt")
3222
3223 > data1 <- list(y=y, nz=nz, nind=nind, K=K)
3224 > params1 <- c('p0', 'sigmap', 'psi', 'N')
3225 > inits <- function(){ list(z=as.numeric(y>=1), psi=.6, p0=runif(1),
3226   sigmap=runif(1,.7,1.2),lp=rnorm(M,-2)) }
3227 > library(R2jags)
3228 > wabout <- jags(data1, inits, params1, model.file = "modelMh.txt", n.chains = 3,
3229   n.iter = 1010000, n.burnin =10000, working.directory = getwd())

```

3229 We provide an **R** function `modelMhBUGS` in the package `scrbook` which will fit the
 3230 model using either **JAGS** or **WinBUGS** as specified by the user. In addition, for fun,
 3231 we construct our own MCMC algorithm using a Metropolis-within-Gibbs algorithm for
 3232 model M_h in Chapt. ??, where we also develop MCMC algorithms for spatial capture-
 3233 recapture models. Using `modelMhBUGS`, we ran 3 chains of 1 *million* iterations (mixing is
 3234 poor for this model and this data set), which produced the posterior distribution for N
 3235 shown in Fig. ???. Posterior summaries of parameters are given in Table ??.

3236 We used $M = 700$ for this analysis and we note that while the posterior mass of N is
 3237 concentrated away from this upper bound (Fig. ??), the posterior has an extremely long
 3238 right tail, with some MCMC draws at the upper boundary $N = 700$, suggesting that an
 3239 even higher value of M may be called for. To characterize the posterior distribution of
 3240 density we produce the relevant summaries of the posterior distribution of $D = N/277.11$
 3241 (recall the buffered area of the convex hull is 277.11 km^2):

```

3242 > summary(wabout$sims.list$N/277.11)

```

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

```

3243      Min. 1st Qu. Median     Mean 3rd Qu.      Max.
3244 0.1696  0.2959 0.3681  0.4428 0.4944  2.5260
3245
3246 > quantile(wbout$sims.list$N/277.11,c(0.025,0.50,0.975))
3247      2.5%       50%      97.5%
3248 0.2237379 0.3680849 1.1511674

```

3249 Therefore, the point estimate, characterized by the posterior median, is around 0.37 bears
 3250 per square km and a 95% Bayesian credible interval is (0.224, 1.151).

3251 4.4.3 Comparison with MLE

3252 The posterior of N is highly skewed; therefore, we see that the posterior mean ($N = 122.7$)
 3253 is considerably higher than the posterior median ($N = 102$). Further, it may be surprising
 3254 that these posterior summaries do not compare well with the MLE. We used the **R** code
 3255 contained in Panel 6.1 from ? to obtain the MLE of $\log(n_0)$, the logarithm of the number
 3256 of uncaptured individuals, is $\widehat{\log(n_0)} = 3.86$ and therefore $\widehat{N} = \exp(3.86) + 47 = 94.47$,
 3257 which is larger than the mode shown in Fig. ???. To see this, we compute the posterior
 3258 mode, by finding the posterior value of N with the highest mass. Because N is discrete,
 3259 we can use the **table()** function in **R** and find the most frequent value ⁴. If we want
 3260 to smooth out some of the Monte Carlo error a bit, we can use a smoother of some sort
 3261 applied to the tabled posterior frequencies of N . Here we use a smoothing spline (**R**
 3262 function **smooth.spline**) with the degree of smoothing chosen by cross-validation (the
 3263 **cv=TRUE** argument):

```

3264 > N <- table(jout$BUGSoutput$sims.list$N)
3265 > xg <- as.numeric(names(N))
3266
3267 > sp <- smooth.spline(xg,N,cv=TRUE)
3268
3269 > sp
3270
3271 Call:

```

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

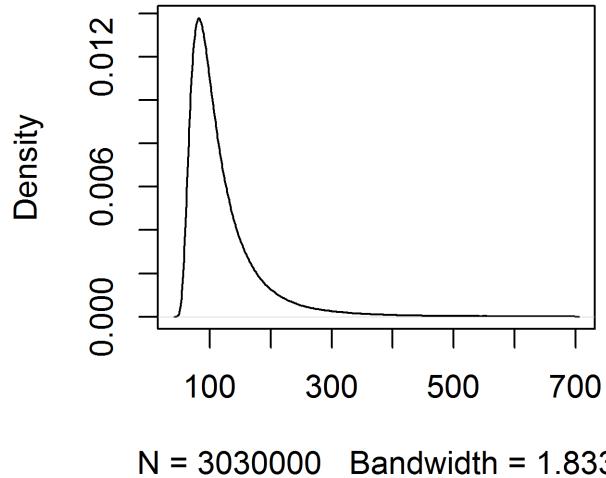


Figure 4.4. Posterior of N for Fort Drum bear study data under the logit-normal version of model M_h .

```

3272 smooth.spline(x = xg, y = N, cv = TRUE)
3273 
3274 Smoothing Parameter  spar= 0.09339815  lambda= 8.201724e-09 (17 iterations)
3275 Equivalent Degrees of Freedom (Df): 121.1825
3276 Penalized Criterion: 2544481
3277 PRESS: 5903.4

```

3278 We obtain the mode of the smoothed frequencies as follows:

```

3279 sp$x[sp$y==max(sp$y)]
3280 [1] 82

```

3281 We don't dwell too much on the difference between the MLE and features of the poste-
 3282 rior, but we do note here that the posterior distribution for the parameters of this model,
 3283 for the Fort Drum data set, are very sensitive to the prior distributions. In the present
 3284 case, the use of a Uniform(0, 1) prior for $p_0 = \logit^{-1}(\mu)$ is somewhat informative—in
 3285 particular, it is not at all “flat” on the scale of μ , and this affects the posterior. We gen-
 3286 erally always recommend use of a Uniform(0, 1) prior for $\logit^{-1}(\mu)$ in such models. That
 3287 said, we were surprised at this result, and we experimented with other prior configurations

3288 including putting a flat prior on μ directly. This kind of small sample instability has been
 3289 widely noted in model M_h (??), as has extreme sensitivity to the specific form of model
 3290 M_h (?). In summary, while the mode is well-defined, the data set is relatively sparse and
 3291 hence inferences are poor and sensitive to model choice.

4.5 INDIVIDUAL COVARIATE MODELS: TOWARD SPATIAL CAPTURE-RECAPTURE

3292 A standard situation in capture-recapture models is when a covariate which is thought
 3293 to influence encounter probability is measured for each individual. These are often called
 3294 “individual covariate models” but, in keeping with the classical nomenclature on closed
 3295 population models, ? referred to this class of models as “model M_x ” (the x here being
 3296 an explicit covariate). As with other closed population models, we begin with the basic
 3297 binomial observation model:

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

3298 To model the covariate, we use a logit model for encounter probability of the form:

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

3299 where x_i is the covariate value for individual i and the parameters $\boldsymbol{\alpha} = (\alpha_0, \alpha_1)$ are the
 3300 regression coefficients. Classical examples of covariates influencing detection probability
 3301 are type of animal (juvenile/adult or male/female), a continuous covariate such as body
 3302 mass, or a discrete covariate such as group or cluster size. For example, in models of aerial
 3303 survey data, it is natural to model the detection probability of a group as a function of
 3304 the observation-level individual covariate, “group size” (??).

3305 Model M_x is similar in structure to model M_h , except that the individual effects are
 3306 *observed* for the n individuals that appear in the sample. These models are important
 3307 here because spatial capture-recapture models can be described precisely as a form of
 3308 model M_x , where the covariate describes *where* the individual is located in relation to the
 3309 trapping array. Specifically, SCR models *are* individual covariate models, but where the
 3310 individual covariate is only observed imperfectly (or partially observed) for each captured
 3311 individual. Unlike model M_h , in SCR models (and model M_x) we do have some direct
 3312 information about the latent variable, which comes from the spatial locations/distribution
 3313 of individual recaptures.

3314 Traditionally, estimation of N in model M_x is achieved using methods based on ideas
 3315 of unequal probability sampling (i.e., Horvitz-Thompson estimation⁵; ?, ? and ?). An
 3316 estimator of N is

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

3317 where \tilde{p}_i is the probability that individual i appeared in the sample. This quantity is
 3318 $\tilde{p}_i = \Pr(y_i > 0)$ and, in closed population capture-recapture models, it can be computed
 3319 as:

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

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

3320 where p_i is a function of parameters α_0 and α_1 according to Eq. ???. In practice, pa-
 3321 rameters are estimated from the conditional-likelihood of the observed encounter histories
 3322 which is, for observation y_i ,

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

3323 This derives from a straightforward application of the law of total probability. Conceptually,
 3324 we partition $\Pr(y)$ according to $\Pr(y) = \Pr(y|y > 0)\Pr(y > 0) + \Pr(y|y = 0)\Pr(y = 0)$. For any positive value of y the 2nd term is necessarily 0, and so we rearrange to obtain
 3325 $\Pr(y|y > 0) = \Pr(y)/\Pr(y > 0)$ which, in the specific case where $\Pr(y)$ is the binomial
 3326 probability mass function (pmf) produces Eq. ??.

3327 Here we take a formal model-based approach to Bayesian analysis of such models based
 3328 on the joint likelihood using data augmentation (?). Classical likelihood analysis of the
 3329 so-called “full likelihood” is covered by ?. For Bayesian analysis of model M_x , because the
 3330 individual covariate is unobserved for the $n_0 = N - n$ uncaptured individuals, we require
 3331 a model to describe variation in x among individuals, essentially allowing the sample to
 3332 be extrapolated to the population. For example, if we have a continuous trait measured
 3333 on each individual, then we might assume that x has a normal distribution:

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

3335 Data augmentation can be applied directly to this class of models. In particular, reformu-
 3336 lation of the model under DA yields a basic zero-inflated binomial model of the following
 3337 form, for each $i = 1, 2, \dots, M$:

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

3338 Fully spatial capture-recapture models use this formulation with a latent covariate that
 3339 is directly related to the individual detection probability (see next section). As with
 3340 the previous models, implementation is trivial in the **BUGS** language. The **BUGS**
 3341 specification is very similar to that for model M_h , but we require the distribution of the
 3342 covariate to be specified, along with priors for the parameters of that distribution.

3343 4.5.1 Example: Location of capture as a covariate

3344 Here we consider a special type of model M_x that is especially relevant to spatial capture-
 3345 recapture. Intuitively, some measure of distance from home range center to traps for an
 3346 individual should be a reasonable covariate to explain heterogeneity in encounter probabil-
 3347 ity, i.e., individuals with more exposure to traps should have higher encounter probabilities
 3348 and vice versa. So we can imagine estimating such a quantity, say average distance from
 3349 home range center to “the trap array”, and then using it as an individual covariate in
 3350 capture-recapture models. A version of this idea was put forth by ? (see also ?), but
 3351 using the Huggins-Alho estimator and with covariate “distance from home range center

3352 to edge” of the trapping array, where the home range center is estimated by the average
 3353 capture location. This is intuitively appealing because we can imagine, in some kind of
 3354 an ideal situation where we have a dense grid of traps over some geographic region, that
 3355 the average location of capture would be a decent estimate (heuristically) of an individ-
 3356 ual’s home range center. We provide an example of this type of approach using a fully
 3357 model-based analysis of the version of model M_x described above, analyzed by data aug-
 3358 mentation. We take a slightly different approach than that adopted by ?. By analyzing
 3359 the full likelihood and placing a prior distribution on the individual covariate, we will
 3360 resolve the problem of having an ill-defined sample area. After you read later chapters of
 3361 this book, it will be apparent that SCR models represent a formalization of this heuristic
 3362 procedure.

3363 For our purposes here, we define the scalar individual covariate x_i to be the distance
 3364 from the average encounter location of individual i , say \mathbf{s}_i , to the centroid of the trap
 3365 array, \mathbf{x}_0 : $x_i = \|\mathbf{s}_i - \mathbf{x}_0\|$. Note that $\|\mathbf{u}\|$ is standard notation for Euclidean norm or
 3366 magnitude of the vector \mathbf{u} , and we use it throughout the book. In practice, people have
 3367 used distance from edge of the trap array but that is less easy to quantify, as “edge” itself
 3368 is not precisely defined. Conceptually, individuals in the middle of the array should have
 3369 a higher probability of encounter and, as x_i increases, p_i should therefore decrease. We
 3370 note that we have defined \mathbf{s}_i in terms of a sample quantity—the observed mean encounter
 3371 location—which, while ad hoc, is consistent with the use of individual covariate models in
 3372 the literature. For an expansive, dense trapping grid we might expect the sample mean
 3373 encounter location to be a good estimate of home range center but, clearly this is biased
 3374 for individuals that live around the edge (or off) the trapping array.

3375 A key point is that \mathbf{s}_i is missing for each individual that is not encountered and so
 3376 x_i is also missing. Therefore, it is a latent variable, and we need to specify a probability
 3377 distribution for it. As a measurement of distance we know it must be positive-valued, and
 3378 it seems sensible that an individual located extremely far from the array of traps would
 3379 not be captured. Therefore, let’s assume that x_i is uniformly distributed from 0 to some
 3380 large number, say B , beyond which it would be difficult to imagine an individual being
 3381 captured by the trap array:

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

3382 where B is a specified constant, which we may choose to be arbitrarily large. For example,
 3383 B should be at least a home range diameter past the furthest trap from the centroid of
 3384 the array.

3385 4.5.2 Fort Drum bear study

3386 We have to do a little bit of data processing to fit this individual covariate model to the
 3387 Fort Drum data. We need to compute the individual covariate \mathbf{x}_i (distance from the
 3388 centroid of the trapping array) using the **R** function **spiderplot** provided in **scrbook**.
 3389 This function also produces the keen plot shown in Fig. ?? which we call a “spider plot”.
 3390 The **R** commands for obtaining the individual covariate “distance from trap centroid”
 3391 (the variable **xcent** returned by **spiderplot**) and making the spider plot are as follows:

```
3392 > library(scrbook)
3393 > data(beardata)
```

```
3394 > toad <- spiderplot(beardata$bearArray,beardata$trapmat)
3395 > xcent <- toad$xcent
```

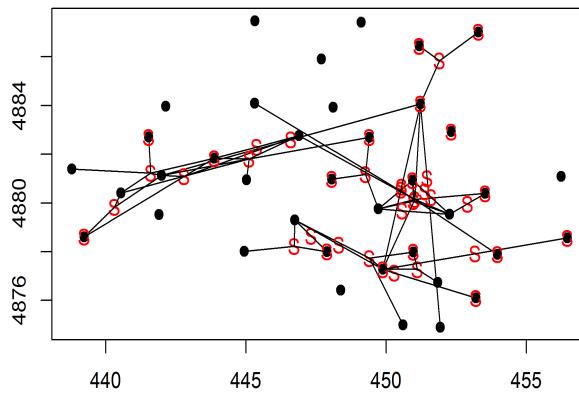


Figure 4.5. Spider plot of the Fort Drum study data. The black dots represent the 47 trap locations with the "S" symbols being the average capture location of each bear. i.e., its estimated home range center. All traps in which a bear was captured are connected to its estimated home range center with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

3396 For the analysis of these data using the individual covariate “distance from centroid”
 3397 we used $x_i \sim \text{Uniform}(0, B)$ with $B = 11.5 \text{ km}^2$, which is about the distance from the
 3398 array center to the furthest trap. Once we choose a value for B , the direct implication is
 3399 that the population size parameter, N , applies to the area within 11.5 units of the trap
 3400 centroid. Therefore, the model associates a precise area within which the population of N
 3401 individuals resides. We will see shortly that N does, in fact, scale with our choice of B to
 3402 reflect the changing area over which the N individuals of the model reside. The **BUGS**
 3403 model specification and **R** commands to package the data and fit the model are as follows:

```
3404 cat("
3405 model{
3406   p0 ~ dunif(0,1)                               # prior distributions
3407   alpha0 <- log(p0/(1-p0))
3408   psi ~ dunif(0,1)
3409   beta ~ dnorm(0,.01)
3410
3411   for(i in 1:(nind+nz)){
```

```

3412   xcent[i] ~ dunif(0,B)
3413   z[i] ~ dbern(psi)           # DA variables
3414   lp[i] <- alpha0 + beta*xcent[i] # individual effect
3415   logit(p[i]) <- lp[i]
3416   mu[i] <- z[i]*p[i]
3417   y[i] ~ dbin(mu[i],K)        # observation model
3418 }
3419
3420 N <- sum(z[1:(nind+nz)])
3421 }
3422 ",file="modelMcov.txt")

3423 data2 <- list(y=y,nz=nz, nind=nind, K=K, xcent=xcent,B=11.5)
3424 params2 <- c('p0','psi','N','beta')
3425 inits <- function() {list(z=zst, psi=psi, p0=rnorm(1), beta=rnorm(1) ) }
3426 fit2 <- bugs(data2, inits, params2, model.file="modelMcov.txt",
3427                 n.chains=3, n.iter=11000, n.burnin=1000, n.thin=1)

```

3428 This produces the posterior summary statistics in Table ??.

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

3429 We note that the estimated N is much lower than obtained by model M_h but there
 3430 is a good explanation for this which we discuss in the next section. That issue notwithstanding,
 3431 it is worth pondering how this model could be an improvement (conceptually or technically)
 3432 over some other model/estimator including M_0 and M_h considered previously.
 3433 Well, for one, we have accounted formally for heterogeneity due to spatial location
 3434 of individuals relative to exposure to the trap array, characterized by the centroid of the
 3435 array. Moreover, we have done so using a model that is based on an explicit mechanism,
 3436 as opposed to a phenomenological one such as model M_h . In addition, and importantly,
 3437 using our new model, *the estimated N applies to an explicit area which is defined by our*
 3438 *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 .

3443 4.5.3 Extension of the model

3444 The model developed in the previous section is not a very good model for one important
 3445 reason: Imposing a uniform prior distribution on x implies that density is *not constant*
 3446 over space. In particular, this model implies that density *decreases* as we move away from
 3447 the centroid of the trap array. That is, $x_i \sim \text{Uniform}(0, B)$ implies constant N in each
 3448 distance band from the centroid but obviously the *area* of each distance band is increasing.
 3449 This is one reason we have a lower estimate of density than that obtained previously from
 3450 model M_h (Sec. ??) and also why, if we were to increase B , we would see density continue
 3451 to decrease.

3452 Fortunately, we are not restricted to use of this specific distribution for the individual
 3453 covariate. Clearly, it is a bad choice and, therefore, we should think about whether we
 3454 can choose a better distribution for B —one that doesn’t imply a decreasing density as
 3455 distance from the centroid increases. Conceptually, what we want to do is impose a
 3456 prior on distance from the centroid, x , such that abundance should be proportional to
 3457 the amount of area in each successive distance band as you move farther away from the
 3458 centroid, so that density is *constant*. In fact, theory exists which tells us we should choose
 3459 $[x] = 2x/B^2$. This can be derived by noting that $F(x) = \Pr(X < x) = (\pi x^2)/(\pi * B^2)$
 3460 . Then, $f(x) = dF/dx = 2 * x/(B^2)$. This is a sort of triangular distribution in density
 3461 induced because the incremental area in each additional distance band increases linearly
 3462 with radius (i.e., distance from centroid). This can be verified empirically as follows:

```
3463 > u <- runif(10000,-1,1)
3464 > v <- runif(10000,-1,1)
3465 > d <- sqrt(u*u+v*v)
3466 > hist(d[d<1])
3467 > hist(d[d<1],100)
3468 > hist(d[d<1],100,probability=TRUE)
3469 > abline(0,2)
```

3470 It would be useful if we could describe this distribution directly in **BUGS** but there
 3471 is not a built-in way to do so. However, we can implement a discrete version of the pdf⁶.
 3472 To do this, we break B into L distance classes of width δ , with probabilities proportional
 3473 to $2 * x$. In particular, if we denote the cut-points by $g_1 = 0, g_2, \dots, g_{L+1} = B$ and the
 3474 interval midpoints are $m_i = g_{i+1} - \delta$. Then the interval probabilities are, approximately⁷,
 3475 $p_i = \delta(2m_i/B^2)$, which we can compute once and then pass them to **BUGS** as data. The
 3476 **R** commands for doing all of this (noting that we have already loaded and processed the
 3477 Fort Drum bear data) are given in the following **R/BUGS** script:

```
3478 > delta <- .2
3479 > xbin <- xcent%/%delta + 1 # Put x in bins
3480 > midpts <- seq(delta,Dmax,delta)
3481 > xprobs <- delta*(2*midpts/(B*B))
3482 > xprobs <- xprobs/sum(xprobs)
3483
```

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.

```

3484 > cat("
3485 model{
3486 p0 ~ dunif(0,1)                                # Prior distributions
3487 alpha0 <- log(p0/(1-p0))
3488 psi ~ dunif(0,1)
3489 beta ~ dnorm(0,.01)
3490
3491 for(i in 1:(nind+nz)){
3492   xbin[i] ~ dcat(xprobs[])
3493   z[i] ~ dbern(psi)                               # DA variables
3494   lp[i] <- alpha0 + beta*xbin[i]*delta          # Individual covariate model
3495   logit(p[i]) <- lp[i]
3496   mu[i] <- z[i]*p[i]
3497   y[i] ~ dbin(mu[i],K)                          # Observation model
3498 }
3499
3500 N <- sum(z[1:(nind+nz)])                      # N is derived
3501 }
3502 ",file="modelMcov.txt")

```

3503 In the model description, the variable x (observed distance from centroid of the trap
 3504 array) has been rounded or binned (placed into a distance bin) so that the discrete version
 3505 of the pdf of x can be used, as described previously. The new variable labeled `xbin` is
 3506 then the *integer category* in units of δ from 0. Thus, to convert back to distance in the
 3507 expression for `lp[i]`, `xbin[i]` has to be multiplied by δ . To fit the model, keeping in
 3508 mind that the data objects required below have been defined in previous analyses of this
 3509 chapter, we do this:

```

3510 > data2 <- list(y=y, nz=nz, nind=nind, K=K, xbin=xbin, xprobs=xprobs,
3511                   delta=delta)
3512 > params2 <- c('p0','psi','N','beta')
3513 > inits <- function() {list(z=z, psi=psi, p0=runif(1),beta=rnorm(1) ) }
3514 > fit <- bugs(data2, inits, params2, model.file="modelMcov.txt",
3515                 working.directory=getwd(), debug=FALSE, n.chains=3,
3516                 n.iter=11000, n.burnin=1000, n.thin=2)

```

3517 By specification of B , this model induces a clear definition of area in which the popu-
 3518 lation of N individuals reside. The parameter N of the model is the population size that
 3519 applies to the particular value of B and, as such, we will see that N scales with our choice
 3520 of B . This might be disconcerting to some—we can get whatever value of N we want
 3521 by changing B ! However, it is intuitively reasonable that, as we increase the area under
 3522 consideration, there should be more individuals in it. Fortunately, we find empirically,
 3523 that while N is highly sensitive to the prescribed value of B , density appears invariant to
 3524 B as long as B is sufficiently large. We fit the model for a set of values of B from $B = 12$
 3525 (restricting values of x to be in close proximity to the trap array) on up to 20. The results
 3526 are given in Table ??.

3527 We see that the posterior mean and SD of density (individuals per square km) appear
 3528 insensitive to choice of B once we reach about $B = 17$ or so. The estimated density of 0.25

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

per km^2 is actually quite a bit lower than we reported using model M_h for which no relevant “area” quantity is explicit in the model (and so we had to make it up). Using MLEs of N in conjunction with buffer strips (see Tab. 1.1) our estimates were in the range of $0.32 - 0.43$ and see Sec. ?? above. On the other hand our estimate of $\hat{D} = 0.25$ here (based on the posterior mean) is higher than that reported from model M_0 using the buffered area ($\hat{D} = 0.18$). There is no basis really for comparing or contrasting these various estimates. In particular, application of models M_0 and M_h are distinctly *not* spatially explicit models—the area within which the population resides is not defined under either model. There is therefore no reason at all to think that the estimates produced under either closed population model, based on a buffered “trap area”, are justifiable by any theory. In fact, we would get exactly the same estimate of N no matter what we declare the area to be. On the other hand, the individual covariate model uses an explicit model for “distance from centroid” that is a reasonable and standard null model—it posits, in the absence of direct information, that individual home range centers are randomly distributed in space and that probability of detection depends on the distance between home range center and the centroid of the trap array. Under this definition of the system, we see that density is invariant to the choice of area, which seems like a desirable feature.

4.5.4 Invariance of density to B

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

4.5.5 Toward fully spatial capture-recapture models

While the use of an individual covariate model resolves two important problems inherent in almost all capture-recapture studies (induced heterogeneity and absence of a precise relationship between N and area), is not ideal for all purposes because it does not make

3559 full use of the spatial information in the data set, i.e., the trap locations and the locations
 3560 of each individual encounter, so that we cannot use this model to model trap-specific
 3561 effects (e.g., trap effort or type). Moreover, we applied this model for “data” being the
 3562 average observed encounter location, and equated that summary to the home range center
 3563 \mathbf{s}_i . Intuitively, taking the average encounter location as an estimate of home range center
 3564 makes sense but more so when the trapping grid is dense and expansive relative to typical
 3565 home range sizes which might not be reasonable in practice. Moreover, this approach
 3566 also ignored the variable precision with which each \mathbf{s}_i is estimated. Finally, it ignores
 3567 that estimates of \mathbf{s}_i around the “edge” (however we define that) are biased because the
 3568 observations are truncated—we can only observe locations interior to the array.

3569 However, there is hope to extend this model in order to resolve these remaining defi-
 3570 ciencies. In the next chapter we provide a further extension of this individual covariate
 3571 model that definitively resolves the *ad hoc* nature of the approach we took here. In that
 3572 chapter we build a model in which \mathbf{s}_i are regarded as latent variables and the observation
 3573 locations (i.e., trap specific encounters) are linked to those latent variables with an explicit
 3574 model. We note that the model fitted previously could be adapted easily to deal with \mathbf{s}_i
 3575 as a latent variable, simply by adding a prior distribution for \mathbf{s}_i . This is actually easier,
 3576 and less ad hoc in a number of respects, and you should try it out.

4.6 DISTANCE SAMPLING: A PRIMITIVE SCR MODEL

3577 Distance sampling is a class of methods for estimating animal density from measurements
 3578 of distance from an observer to individual animals (or groups). The basic assumption
 3579 is that detection probability is a function of distance. Distance sampling is one of the
 3580 most popular methods for estimating animal abundance (???) because, unlike ordinary
 3581 closed population models, distance sampling provides explicit estimates of *density*. In
 3582 terms of methodological context, the distance sampling model is a special case of a closed
 3583 population model with an individual covariate. The covariate in this case, x , is the distance
 3584 between an individual’s location say \mathbf{u} and the observation location or transect. In fact,
 3585 distance sampling is precisely an individual-covariate model, except that observations
 3586 are made at only $K = 1$ sampling occasion. Distance sampling eliminates the need to
 3587 explicitly identify individuals (except they need to be *distinguished* from other individuals)
 3588 repeatedly and so distance sampling can be applied to unmarked populations. This first
 3589 and most basic spatial capture-recapture model has been used routinely for decades and,
 3590 formally, it is a spatially-explicit model in the sense that it describes, explicitly, the spatial
 3591 organization of individual locations (although this is not always stated explicitly) and, as
 3592 a result, somewhat general models of how individuals are distributed in space can be
 3593 specified (?????).

3594 As with other models we’ve encountered in this chapter, the distance sampling model,
 3595 under data augmentation, includes a set of M zero-inflation variables z_i and a binomial
 3596 observation model expressed conditional on z (binomial for $z = 1$, and fixed zeros for
 3597 $z = 0$). In distance sampling we pay for having only a single sample occasion (i.e., $K = 1$)
 3598 by requiring constraints on the model of detection probability, normally imposed as the
 3599 assumption that detection probability is 1.0 when distance equals 0. A standard model
 3600 for detection probability is the “half-normal” model:

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

3601 for $\alpha_1 > 0$, where x_i denotes the distance at which the i th individual is detected relative
 3602 to some reference location where perfect detectability ($p = 1$) is assumed. This encounter
 3603 probability model is more often written with $\alpha_1 = 1/2\sigma^2$. If $K > 1$ then an intercept in
 3604 this model, say α_0 , is identifiable and such models are usually called “capture-recapture
 3605 distance sampling”(??).

3606 As with previous examples, we require a distribution for the individual covariate x_i .
 3607 The customary choice is

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

3608 wherein $B > 0$ is a known constant, being the upper limit of data recording by the observer
 3609 (i.e., the point count radius, or transect half-width). Specification of this distance sampling
 3610 model in the **BUGS** language is shown in Panel ??, taken from ?.

```

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.

3611 As with the individual covariate model in the previous section, the distance sampling
 3612 model can be equivalently specified by putting a prior distribution on individual *location*
 3613 instead of distance between individual and observation point (or transect). Thus we can
 3614 write the general distance sampling model as

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

3615 along with

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

3616 where \mathbf{x}_0 is a fixed point (or line) and \mathbf{u}_i is the individual’s location, which is observed for
 3617 the sample of n individuals. In practice it is easier to record distance instead of location.
 3618 Basic math can be used to argue that if individuals have a uniform distribution in space,
 3619 then the distribution of Euclidean distance is also uniform. In particular, if a transect of
 3620 length L is used and x is distance to the transect then $F(x) = \Pr(X \leq x) = L*x/L*B =$
 3621 x/B and $f(x) = dF/dx = (1/B)$. For measurements of radial distance, we provided the
 3622 analogous argument in the previous section.

3623 The preceding paragraph makes it clear that distance sampling is a special case of
 3624 spatial capture-recapture models, such as those derived from model M_x of the previous
 3625 section, where the encounter probability is related directly to *distance*, which is a reduced
 3626 information summary of *location*, \mathbf{u} . Some intermediate forms of SCR/DS models can
 3627 be described (?). In the context of our general characterization of SCR models (Chapt.
 3628 2.6), we suggested that every SCR model can be described, conceptually, by a hierarchical
 3629 model of the form:

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

3630 Distance sampling ignores the part of the model pertaining to \mathbf{s} , and deals only with the
 3631 model components for the observed data \mathbf{u} ⁸. Thus, we are left with a hierarchical model
 3632 of the form

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

3633 In contrast, as we will see in the next chapters, many SCR models (Chapt. ??) ignore \mathbf{u}
 3634 and condition on \mathbf{s} , which is not observed:

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

3635 Since $[\mathbf{u}]$ and $[\mathbf{s}]$ are both assumed to be uniformly distributed, these are equivalent models!
 3636 The main differences have to do with interpretation of model components and whether or
 3637 not the latent variables are observable (in distance sampling they are).

3638 So why bother with SCR models when distance sampling yields density estimates and
 3639 accounts for spatial heterogeneity in detection? For one, imagine trying to collect distance
 3640 sampling data on species such as jaguars or tigers! Clearly, distance sampling requires
 3641 that one can collect large quantities of distance data, which is not always possible. For
 3642 tigers, it is much easier, efficient, and safer to employ camera traps or track plates and
 3643 then apply SCR models. Furthermore, as we will see in Chapt. ??, SCR models can make
 3644 use of distance data, allowing us to study distribution, movement, and density. Thus,
 3645 SCR models are more general and versatile than distance sampling models (which clearly
 3646 are a special case), and can accommodate data from virtually all animal survey designs.

3647 **4.6.1 Example: Sonoran desert tortoise study**

3648 We illustrate the application of distance sampling models using data on the Sonoran desert
 3649 tortoise (*Gopherus agassizii*), shown in Fig. ??, collected along transects in southern
 3650 Arizona (see ? for details). The data are from 120 square transects having four 250-
 3651 m sides, although we ignore this detail in our analysis here and regard them as 1 km
 3652 transects, and we pooled the detection data from all 120 transects. The histogram of
 3653 encounter distances from the 65 encountered individuals is shown in Fig. ??

3654 Commands for reading in and organizing the data for analysis using **WinBUGS** are
 3655 given in the help file `?tortoise` provided with the `scrbook` package. To compute density,
 3656 the total sampled area of the transects `striparea` is input as data, and computed as: 120
 3657 (transects) multiplied by the length (1000 m) and half-width ($B = 40$ m), then multiplied
 3658 by 2, and divided by 10000 to convert to units of individuals per ha. We also provide
 3659 commands for analyzing the data with `unmarked` (?) using hierarchical distance sampling
 3660 models (?).

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



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

3661 Posterior summaries for the tortoise data are given in Tab. ???. Estimated density
 3662 (posterior mean) is 0.54 individuals per ha and the estimated scale parameter of the
 3663 distance function (posterior mean) is $\sigma = 9.12$ meters. The R-hat statistics of around 1.02
 3664 suggest that slightly longer MCMC simulations might be called for. The posterior mass
 3665 of the data augmentation parameter ψ is located away from the upper bound $\psi = 1$ and
 3666 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

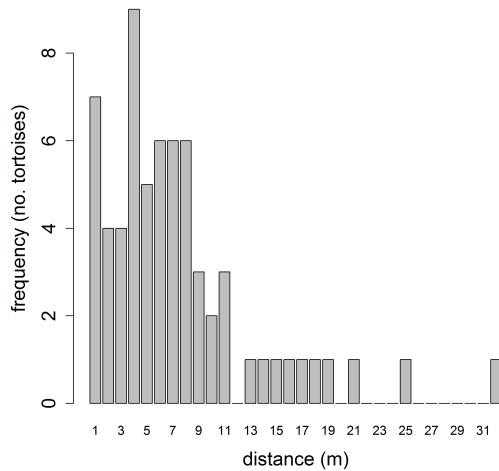


Figure 4.7. Distance histogram of $n = 65$ Sonoran desert tortoise detections from a total of 120 km of survey transect.

4.7 SUMMARY AND OUTLOOK

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

Spatial capture-recapture models that we will consider in the rest of the chapters of this book are closely related to individual covariate models (model M_x). Naturally, spatial capture-recapture models arise by defining individual covariates based on observed locations of individuals—we can think of using some function of mean encounter location as an individual covariate. We did this in a novel way, by using distance to the centroid of the trapping array as a covariate. We analyzed the *full likelihood* using data augmentation, and placed a prior distribution on the individual covariate which was derived from an assumption that individual locations are, a priori, uniformly distributed in space. This

3688 assumption provides for invariance of the density estimator to the choice of population
3689 size area (induced by maximum distance from the centroid of the trap array). The model
3690 addressed some important problems in the use of closed population models: it allows for
3691 heterogeneity in encounter probability due to the spatial juxtaposition of individuals with
3692 the array of traps, and it also provides a direct estimate of density because area is a
3693 feature of the model (via the prior on the individual covariate). The model is still not
3694 completely general, however, because it does not make full use of the spatial encounter
3695 histories, which provide direct information about the locations and density of individuals.

3696 A specific individual covariate model that is in widespread use is classical distance
3697 sampling. The model underlying distance sampling is precisely a special kind of SCR
3698 model—but one without replicate samples. Understanding distance sampling and individ-
3699 ual covariate models more broadly provides a solid basis for understanding and analyzing
3700 spatial capture-recapture models. In fact if, instead of placing an explicit model on *dis-*
3701 *tance* in the classical distance sampling model, we were to place the prior distribution on
3702 *location*, \mathbf{s} , of each individual, then the form of the distance sampling model more closely
3703 resembles the SCR model we introduce in the next chapter.

3704

Part II

3705

3706

Basic SCR Models

3707
3708

5

3709

FULLY SPATIAL CAPTURE-RECAPTURE 3710 MODELS

3711 In the previous chapter, we discussed models that could be viewed as primitive spatial
3712 capture-recapture models. We looked at a basic distance sampling model, and we also
3713 considered a classical individual covariate modeling approach in which we defined a co-
3714 variate to be the distance from the (estimated) home range center to the center of the
3715 trap array. The individual covariate model that we conjured up was “spatial” in the sense
3716 that it included some characterization of where individuals live but, on the other hand,
3717 only a primitive or no characterization of trap location. That said, there is only a small
3718 step from this model to spatial capture-recapture models that we consider in this chapter,
3719 which fully recognize the spatial attribution of both individual animals *and* the locations
3720 of encounter devices.

3721 Capture-recapture models must accommodate the spatial organization of individuals
3722 and the encounter devices because the encounter process occurs at the level of individual
3723 traps. Failure to consider the trap-specific data is one of the key deficiencies with classical
3724 ad-hoc approaches which aggregate encounter information to the resolution of the entire
3725 trap array. We have previously addressed some problems that this causes including induced
3726 heterogeneity in encounter probability, imprecise notation of “sample area” and not being
3727 able to accommodate trap-specific effects or trap-specific missing values. In this chapter
3728 we resolve these issues by developing our first fully spatial capture-recapture model. This
3729 model is not too different from that considered in Sec. ?? but, instead of defining the
3730 individual covariate to be distance to the centroid of the array we define J individual
3731 covariates - the distance to *each* trap. And, instead of using estimates of individual
3732 locations \mathbf{s} , we consider a fully hierarchical model in which we regard \mathbf{s} as a latent variable
3733 and impose a prior distribution on it.

3734 In this chapter we investigate the basic spatial capture-recapture model, which we re-
3735 fer to as “model SCR0”, and address some important considerations related to its analysis
3736 in **BUGS**. We demonstrate how to summarize posterior output for the purposes of pro-
3737 ducing density maps or spatial predictions of density. The key aspect of the SCR models

3738 considered in this chapter is the formulation of a model for encounter probability that is
 3739 a function of distance between individual home range center and trap locations. We also
 3740 discuss how encounter probability models are related to explicit models of space usage
 3741 or “home range area.” Understanding this allows us to compute, for example, the area
 3742 used by an individual during some prescribed time. While it is intuitive that SCR models
 3743 should be related to some model of space usage, this has not been discussed much in the
 3744 literature (but see ? which we address further in Chapt. ??).

5.1 SAMPLING DESIGN AND DATA STRUCTURE

3745 In our development here, we will assume a standard sampling design in which an array
 3746 of J traps is operated for K sample occasions (say, nights) producing encounters of n
 3747 individuals. Because sampling occurs by traps and also over time, the most general data
 3748 structure yields temporally *and* spatially indexed encounter histories for *each individual*.
 3749 Thus a typical data set will include an encounter history *matrix* for each individual indicating
 3750 which trap the individual was captured, during each sample occasion. For example,
 3751 suppose we sample at 4 traps over 3 nights. A plausible data set for a single individual
 3752 captured one time in trap 1 on the first night and one time in trap 3 on the 3rd night is:

```
3753     night1 night2 night3
3754 trap1    1    0    0
3755 trap2    0    0    0
3756 trap3    0    0    1
3757 trap4    0    0    0
```

3758 This data structure would be obtained for *each* of the $i = 1, 2, \dots, n$ captured individuals.

3759 We develop models in this chapter for passive detection devices such as “hair snares”
 3760 or other DNA sampling methods (??) and related types of sampling devices in which
 3761 (i) devices (“traps”) may capture any number of individuals (i.e., they don’t fill up);
 3762 (ii) an individual may be captured in more than one trap during each occasion but (iii)
 3763 individuals can be encountered at most 1 time by each trap during any occasion. Hair
 3764 snares for sampling DNA from bears and other species function according to these rules.
 3765 An individual bear wandering about its territory might come into contact with > 1 devices;
 3766 a device may encounter multiple bears; however, in practice, it will often not be possible
 3767 to attribute multiple visits of the same individual during a single occasion (e.g., night) to
 3768 distinct encounter events. Thus, an individual may be captured at most 1 time in each
 3769 trap during any occasion. While this model, which we refer to as SCR0, is most directly
 3770 relevant to hair snares and other DNA sampling methods for which multiple detections
 3771 of an individual are not distinguishable, we will also make use of the model for data that
 3772 arise from camera-trapping studies. In practice, with camera trapping, individuals might
 3773 be photographed several times in a night but it is common to distill such data into a single
 3774 binary encounter event for reasons discussed later in Chapt. ??.

3775 The statistical assumptions we make to build a model for these data are that individual
 3776 encounters within and among traps are independent, and this allows us to regard
 3777 individual- and trap-specific encounters as *independent* Bernoulli trials (see next section).
 3778 These basic (but admittedly at this point somewhat imprecise) assumptions define the
 3779 basic spatial capture-recapture model, SCR0. We will make things more precise as we
 3780 develop a formal statistical definition of the model shortly.

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

5.2 THE BINOMIAL OBSERVATION MODEL

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

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

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

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

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

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

3808 location” we will refer to it as the “(bivariate) normal” or “Gaussian” model although
 3809 the distance sampling term “half-normal” is widely used. In the context of 2-dimensional
 3810 space, the model is clearly interpretable as a primitive model of movement outcomes or
 3811 space usage (we discuss this in Sec. ??).

3812 There are a large number of standard detection models commonly used (see Chapt.
 3813 ??). All other standard models that relate encounter probability to \mathbf{s} will also have a
 3814 parameter that multiplies distance in some non-linear function. To be consistent with pa-
 3815 rameter naming across models, we will sometimes parameterize any encounter probability
 3816 model so that the coefficient on distance (or distance squared) is α_1 . So, for the Gaussian
 3817 model, $\alpha_1 = 1/(2\sigma^2)$. A characteristic of the common parametric forms is they are mono-
 3818 tone decreasing with distance, but vary in their characteristic behavior as they approach
 3819 distance = 0. We show the standard Gaussian, Gaussian hazard, negative exponential and
 3820 logistic models in Fig. ???. The negative exponential model has $p_{ij} = p_0 \exp(-\alpha_1 \|\mathbf{x}_j - \mathbf{s}_i\|)$
 3821 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 Gaus-
 sian kernel. Whatever model we choose for encounter probability, we should always keep

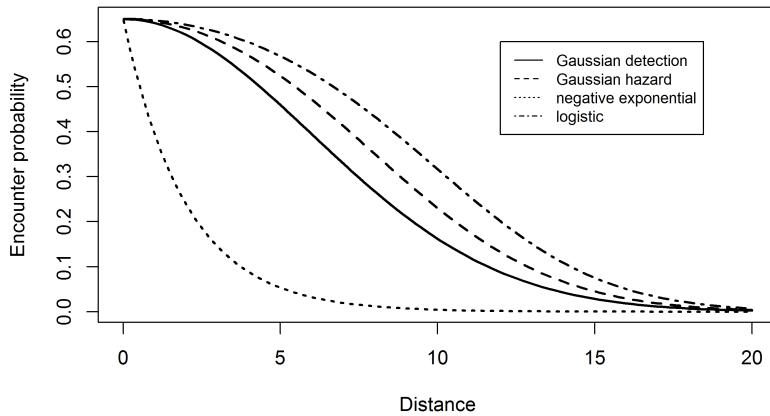


Figure 5.1. Some common encounter probability models showing the characteristic monotone decrease of encounter probability with distance between activity center and trap location.

3822 in mind that the activity center for individual i , \mathbf{s}_i , is an unobserved random variable. To
 3823 be precise about this in the model, we should express the observation model as
 3824

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

3825 but sometimes, for notational simplicity, we abbreviate this by omitting some of the
 3826 arguments to p .

5.2.1 Definition of home range center

3827 We define an individual's home range as *the area used by an organism during some time period* which has a clear meaning for most species regardless of their biology. We therefore
 3828 define the home range center (or activity center) to be the center of the space that individual
 3829 was occupying (or using) during the period in which traps were active. Thinking about
 3830 it in that way, it could even be observable (almost) as the centroid of a very large number
 3831 of radio fixes over the course of a survey period or a season. Thus, this practical version
 3832 of a home range center in terms of space usage is a well-defined construct regardless of
 3833 whether one thinks the home range itself is a meaningful concept. We use the terms home
 3834 range center and activity center interchangeably, and we recognize that this is a transient
 3835 thing which applies only to a well-defined period of study.

3838 **5.2.2 Distance as a latent variable**

3839 If we knew precisely every \mathbf{s}_i in the population (and population size N), then the model
 3840 specified by Eqs. ?? and ?? would be just an ordinary logistic regression-type of a model
 3841 (with covariate d_{ij}) which we learned how to fit using **WinBUGS** previously (Chapt. ??).
 3842 However, the activity centers are unobservable even in the best possible circumstances.
 3843 In that case, d_{ij} is an unobserved variable, analogous to the situation in classical random
 3844 effects models. We need to therefore extend the model to accommodate these random
 3845 variables with an additional model component – the random effects distribution. The
 3846 customary assumption is the so-called “uniformity assumption,” which is to assume that
 3847 the \mathbf{s}_i are uniformly distributed over space (the obvious next question: “which space?”
 3848 is addressed below). This uniformity assumption amounts to a uniform prior distribution
 3849 on \mathbf{s}_i , i.e., the pdf of \mathbf{s}_i is constant, which we may express

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

3850 As it turns out, this assumption is usually not precise enough to fit SCR models in practice
 3851 for reasons we discuss shortly. We will give another way to represent this prior distribution
 3852 that is more concrete, but depends on specifying the “state-space” of the random variable
 3853 \mathbf{s}_i . The term state-space is a technical way of saying “the space of all possible outcomes”
 3854 of the random variable.

5.3 THE BINOMIAL POINT PROCESS MODEL

3855 In the SCR model, the individual activity centers are unobserved and thus we treat them
 3856 as random effects. Specifically, the collection of individual activity centers $\mathbf{s}_1, \dots, \mathbf{s}_N$
 3857 represents a realization of a *binomial point process* (? , p. 61). The binomial point process
 3858 (BPP) is analogous to a Poisson point process in the sense that it represents a “random
 3859 scatter” of points in space – except that the total number of points is *fixed*, whereas, in
 3860 a Poisson point process, it is random (having a Poisson distribution). As an example, we
 3861 show in Fig. ?? locations of 20 individual activity centers (black dots) in relation to a
 3862 grid of 25 traps. For a Poisson point process the number of such points in the prescribed
 3863 state-space would be random whereas often we will simulate fixed numbers of points, e.g.,
 3864 for evaluating the performance of procedures, e.g., how well does our estimator perform
 3865 when $N = 50$?

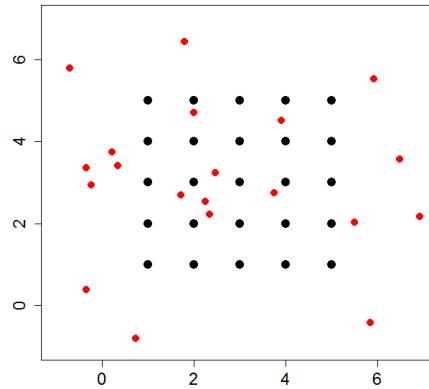


Figure 5.2. Realization (small dots) of a binomial point process with $N = 20$. The large dots represent trap locations.

3866 It is natural to consider a binomial point process in the context of capture-recapture
 3867 models because it preserves N in the model and thus preserves the linkage directly with
 3868 closed population models. In fact, under the binomial point process model, model M_0
 3869 and other closed models are simple limiting cases of SCR models, i.e., they arise as the
 3870 coefficient on distance (α_1 above) tends to 0.

3871 While we often will express SCR models “conditional-on- N ”, it will sometimes be
 3872 convenient to impose specific prior distributions on N . By assuming N has a binomial
 3873 distribution, we can make use of data augmentation, our preferred tool, for Bayesian
 3874 analysis of the models as in Chapt. ??, thus yielding a methodologically coherent approach
 3875 to analyzing the different classes of models. We might also assume that N has a Poisson
 3876 distribution in some cases (see Chapt. ??). Of course, the two assumptions are closely
 3877 related in the usual limiting sense.

3878 One consequence of having fixed N in the BPP model is that the model is not
 3879 strictly a model of “complete spatial randomness”. This is because, if one forms counts
 3880 $n(A_1), \dots, n(A_k)$ in any set of disjoint regions of the state-space, say A_1, \dots, A_k , then
 3881 these counts are *not* independent. In fact, they have a multinomial distribution (see ?,
 3882 p. 61). Thus, the BPP model introduces a slight bit of dependence in the distribution
 3883 of points. However, in most situations this will have no practical effect on any inference
 3884 or analysis and, as a practical matter, we will usually regard the BPP model as one of
 3885 spatial independence among individual activity centers because each activity center is dis-
 3886 tributed independently of each other activity center. Despite this independence we see
 3887 in Fig. ?? that *realizations* of randomly distributed points will typically exhibit distinct
 3888 non-uniformity. Thus, independent, uniformly distributed points will almost never appear
 3889 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. ??.

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 $\mathbf{s}_1, \dots, \mathbf{s}_N$ 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. ??, \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. ??, 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 situations this might be an acceptable approach to defining the state-space, i.e., just a rectangle around the trap array. Although defining the state-space to be a regular polygon has computational advantages (e.g., we can implement this more efficiently in **BUGS** and cannot for irregular polygons), a regular polygon induces an apparent problem of admitting into the state-space regions that are distinctly non-habitat (e.g., oceans, large lakes, ice

fields, etc.). It is difficult to describe complex regions in mathematical terms that can be used in **BUGS**. As an alternative, we can provide a representation of the state-space as a discrete set of points which the **R** package **secr** (?) permits (**secr** uses the term “mask” for what we call the state-space). Defining the state-space by a discrete set of points is handy because it allows specific points to be deleted or not, depending on whether they represent available or suitable habitat (see Sec. ??). We can also define the state-space as an arbitrary collection of polygons stored as a GIS shapefile which can be analyzed easily by MCMC in **R** (see Sec. ??), but not so easily in the **BUGS** engines. In Sec. ??, we provide an analysis of the wolverine camera trapping data, in which we define the state-space to be a regular continuous polygon (a rectangle).

3941 **Invariance to the state-space**

3942 We will assert for all models we consider in this book that density is invariant to the size
 3943 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
 3944 decreasing function of distance. We can prove this easily by drawing an analogy with a 1-d
 3945 case involving distance sampling. Let y_j be the number of individuals captured in some
 3946 interval $[d_{j-1}, d_j)$, and define $d_J = B$ for some large value of B . The observations from a
 3947 survey are y_1, \dots, y_J and the likelihood is a multinomial likelihood, so the log-likelihood
 3948 is of the form

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

3949 where π_j is the probability of detecting an individual in distance class j , which depends on
 3950 parameters of the detection function (the manner of which is not relevant for the present
 3951 discussion). Choosing B sufficiently large guarantees that $\mathbb{E}(y_J) = 0$ and therefore the
 3952 observed frequency in the “last cell” contributes nothing to the likelihood, in regular
 3953 situations in which the detection function decays monotonically with distance and prior
 3954 density is constant. We can think of B as being related to the state-space in an SCR
 3955 model, as the width of a rectangular state-space with area $B \times L$, L being the length
 3956 of the transect. Thus, if we choose B large enough, then we ensure that the expected
 3957 trap-frequencies beyond B will be 0, and thus contribute nothing to the likelihood.

3958 Sometimes our estimate of density can be affected by choosing \mathcal{S} too small. However,
 3959 this might be sensible if \mathcal{S} is naturally well-defined. As we discussed in Chapt. 1, \mathcal{S} is
 3960 part of the model, and thus it is sensible that estimates of density might be sensitive to
 3961 its definition in problems where it is natural to restrict \mathcal{S} . One could imagine, however,
 3962 in specific cases, e.g., a small population with well-defined habitat preferences, that a
 3963 problem could arise because changing the state-space based on differing opinions, and
 3964 GIS layers, might have substantial affects on the density estimate. But this is a real
 3965 biological problem, and a natural consequence of the spatial formalization of capture-
 3966 recapture models – a feature, not a bug or some statistical artifact – and it should be
 3967 resolved with better information, research, and thinking. For situations where there is not
 3968 a natural choice of \mathcal{S} , we should default to choosing \mathcal{S} to be very large in order to achieve
 3969 invariance or, otherwise, evaluate sensitivity of density estimates by trying a couple of
 3970 different choices of \mathcal{S} . This is a standard “sensitivity to prior” argument that Bayesians
 3971 always have to be conscious of. We demonstrate this in our analysis of Sec. ?? below. As
 3972 an additional practical consideration, we note that the area of the state-space \mathcal{S} affects
 3973 data augmentation. If you increase the size of \mathcal{S} , then there are more individuals to

3974 account for and therefore the size of the augmented data set M must increase. This has
 3975 computational implications.

3976 **5.3.2 Connection to model M_h and distance sampling**

3977 SCR models are closely related to “model M_h ” and also distance sampling. In SCR
 3978 models, heterogeneity in encounter probability is induced by both the effect of distance in
 3979 the model for detection probability and also from specification of the state-space. Hence,
 3980 the state-space is an explicit element of the model. To understand this, suppose activity
 3981 centers have the uniform distribution:

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

3982 and encounter probability is a function of \mathbf{s} , denoted by $p(\mathbf{s}) = p(y = 1|\mathbf{s})$. For example,
 3983 under Eq. ?? we have that

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

3984 and we can work out, either analytically or empirically, what is the implied distribution
 3985 of p for a population of individuals. Fig. ?? shows a histogram of p for a hypothetical
 3986 population of 100000 individuals on a state-space enclosing our 5×5 trap array above,
 3987 under the logistic model for distance given by Eq. ?? with buffers of 0.2, 0.5 and 1.0.
 3988 We see the mass shifts to the left as the buffer increases, implying more individuals with
 3989 lower encounter probabilities, as their home range centers increase in distance from the
 3990 trap array.

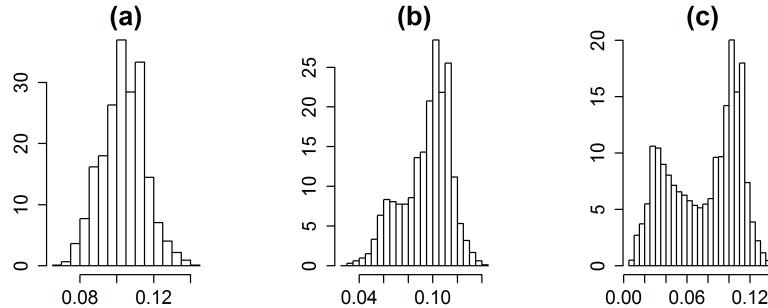


Figure 5.3. Implied distribution of p_i for a population of individuals as a function of the size of the state-space buffer around the trap array. The state-space buffer is 0.2, 0.5 and 1.0 for panels (a), (b), (c), respectively. In each case, the trap array is fixed and centered within a square state-space.

3991 Another way to understand this is by representing \mathcal{S} as a set of discrete points on a
 3992 grid. In the coarsest possible case where \mathcal{S} is a single arbitrary point, then every individual
 3993 has exactly the same p . As we increase the number of points in \mathcal{S} , more distinct values of

3994 p are possible. Indeed, when \mathcal{S} is characterized by discrete points, then SCR models are
 3995 precisely a type of finite-mixture model (??), except, in the case of SCR models, we have
 3996 some information about which group an individual belongs to (i.e., where their activity
 3997 center is), as a result of which traps it is captured in.

3998 It is also worth re-emphasizing that the basic SCR encounter model is a binomial
 3999 encounter model in which distance is a covariate. As such, it is strikingly similar to classical
 4000 distance sampling models (?). Both have distance as a covariate but, in classical distance
 4001 sampling problems, the focus is on the distance between the observer and the animal at
 4002 an instant in time, not the distance between a trap and an animal's home range center.
 4003 As a practical matter, in distance sampling, "distance" is *observed* for those individuals
 4004 that appear in the sample. Conversely, in SCR problems, it is only imperfectly observed
 4005 (we have partial information in the form of trap observations). Clearly, it is preferable
 4006 to observe distance if possible, but distance sampling requires field methods that are not
 4007 practical in many situations, e.g. when studying carnivores such as bears or large cats.
 4008 Furthermore, SCR models allow us to relax many of the assumptions made in classical
 4009 distance sampling, such as perfect detection at distance zero, and SCR models allow for
 4010 estimates of quantities other than density, such as home range size, and space usage (see
 4011 Chaps. ?? and ??).

5.4 THE IMPLIED MODEL OF SPACE USAGE

4012 We developed the basic SCR model in terms of a latent variable, s , the home range center
 4013 or activity center. Surely the encounter probability model, which relates encounter of
 4014 individuals in specific traps to s must somehow imply a certain model for home range
 4015 geometry and size. Here we explore the nature of that relationship and we argue that
 4016 any given detection model implies a model of space usage – i.e., the amount and extent of
 4017 area used some prescribed percentage of the time. So we might say, for example, 95% of
 4018 animal movements are within some distance from an individual's activity center. While we
 4019 have used the term "home range" or similar, what we really mean to imply is something
 4020 that would be more clearly identified as resource selection or space usage (the latter term
 4021 meaning resource selection, when the resource is only homogeneous space).

4022 Intuitively, the detection function of SCR models is related to space usage by indi-
 4023 viduals. Indeed, it is natural to interpret the detection model as the composite of two
 4024 processes: movement of an individual about its home range i.e., how it uses space within
 4025 its home range ("space usage"), and detection *conditional on use* in the vicinity of a
 4026 trapping device. It is natural to decompose encounter probability according to:

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

4027 In practice it might make sense to think about the first component,
 4028 i.e., $\Pr(\text{encounter}|\text{usage of } \mathbf{x}, \mathbf{s})$ as being a constant (e.g., if traps are located within arbit-
 4029 rarily small grid cells) and then, in that case, the encounter probability model is directly
 4030 proportional to this model for individual movements about their home range center deter-
 4031 mining the use frequency of each \mathbf{x} . This is a sensible heuristic model for what ecologists
 4032 would call a central place forager although, as we have stated previously, it may be mean-
 4033 ingful as a description of transient space usage as well (that is, the space usage during the
 4034 period of sampling).

4035 To motivate a specific model for space usage, imagine the area we are interested in
 4036 consists of some large number of small pixels (i.e. we're looking at a discrete representation
 4037 of space), and that we have some kind of perfect observation device (e.g., continuous
 4038 telemetry) so that we observe every time an individual moves into a pixel. After a long
 4039 period of time, we observe an enormous sample size of \mathbf{x} values. We tally those up into
 4040 each pixel, producing the frequency $m(\mathbf{x}, \mathbf{s})$, which is something like the "true" usage of
 4041 pixel \mathbf{x} by individual with activity center \mathbf{s} . So, then, the usage model should be regarded
 4042 as a probability mass function for these counts and, naturally, we regard the counts $m(\mathbf{x}, \mathbf{s})$
 4043 as a multinomial observation with probabilities $\pi(\mathbf{x}|\mathbf{s})$, and prescribe a suitable model for
 4044 $\pi(\mathbf{x}|\mathbf{s})$ that describes how use events should accumulate in space. A natural null model
 4045 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
 4046 more time close to their activity centers than far away. We can regard points used by
 4047 the individual with activity center \mathbf{s} as the realization of a point process with conditional
 4048 intensity:

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

4049 where $k(\mathbf{x}, \mathbf{s})$ is any positive function. In continuous space, the equivalent representation
 4050 would be:

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

4051 Clearly the space used by an individual will be proportional to whatever kernel, $k(\mathbf{x}, \mathbf{s})$,
 4052 we plug-in here. If we use a negative exponential function, then this produces a standard
 4053 resource selection function (RSF) model (e.g., ?, Chapt. 8). But, here we use a Gaussian
 4054 kernel, i.e.,

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

4055 so that contours of the probability of space usage resemble a bivariate normal or Gaussian
 4056 probability distribution function.

4057 To apply this model of space-usage to SCR problems we allow for imperfect detection
 4058 by introducing a non-uniform "thinning rate" of the true counts $m(\mathbf{x}, \mathbf{s})$. This yields,
 4059 precisely, our Gaussian encounter probability model where the thinning rate is our baseline
 4060 encounter probability p_0 for each pixel where we place a trap, and $p = 0$ in each pixel
 4061 where we don't place a trap.

4062 The main take-away point here is that underlying most SCR models is some kind of
 4063 model of space-usage, implied by the specific choice of $k(\mathbf{x}, \mathbf{s})$. Whether or not we have
 4064 perfect sampling devices, the function we use in the encounter probability model equates
 4065 to some conditional distribution of points, a utilization distribution, as in Eq. ??, from
 4066 which we can compute effective home range area, i.e., the area that contains some percent
 4067 of the mass of a probability distribution proportional to $k(\mathbf{x}, \mathbf{s})$; e.g., 95% of all space used
 4068 by an individual with activity center \mathbf{s} .

4069 5.4.1 Bivariate normal case

4070 One encounter model that allows direct analytic computation of home range area is the
 4071 Gaussian encounter probability model

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

4072 For this model, encounter probability is proportional to the kernel of a bivariate normal
 4073 (Gaussian) pdf and so the natural interpretation is that in which movement outcomes (or
 4074 successive locations of an individual) are draws from a bivariate normal distribution with
 4075 standard deviation σ . We say that use of this model implies a bivariate normal model of
 4076 space usage. Under this model we can compute precisely the effective home range area. In
 4077 particular, if use outcomes are bivariate normal, then $\|\mathbf{x} - \mathbf{s}\|^2$ has a chi-square distribution
 4078 with 2 d.f. and the quantity $B(\alpha)$ that encloses $(1 - \alpha)\%$ of all realized distances i.e.,
 4079 $\Pr(d \leq B(\alpha)) = 1 - \alpha$, is $B(\alpha) = \sigma * \sqrt{q(\alpha, 2)}$ where $q(\alpha, 2)$ is the 0.05 chi-square
 4080 critical value on 2 df. For example, to compute $q(.05, 2)$ in R we execute the command
 4081 `qchisq(.95, 2)` which is $q(2, \alpha) = 5.99$. Then, for $\sigma = 1$, $B(\alpha) = 1 * \sqrt{5.99} = 2.447$.
 4082 Therefore 95% of the points used will be within 2.447 (standard deviation) units of the
 4083 home range center. So, in practice, we can estimate σ by fitting the bivariate normal
 4084 encounter probability model to some SCR data, and then use the estimated σ to compute
 4085 the “95% radius”, say $r_{.95} = \sigma\sqrt{5.99}$, and convert this to the 95% use area – the area
 4086 around \mathbf{s} which contains 95% of the movement outcomes – according to $A_{.95} = \pi r_{.95}^2$.
 4087 An alternative bivariate normal model is the bivariate normal hazard rate model:

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

4088 We use λ_0 here because this parameter, the baseline encounter *rate*, can be > 1 . This arises
 4089 by assuming the latent “use frequency” $m(\mathbf{x}, \mathbf{s})$ is a Poisson random variable with intensity
 4090 $\lambda_0 k(\mathbf{x}, \mathbf{s})$. The model is distinct from our Gaussian encounter model $p(\mathbf{x}, \mathbf{s}) = p_0 k(\mathbf{x}, \mathbf{s})$
 4091 used previously, although we find that they produce similar results in terms of estimates
 4092 of density or 95% use area, as long as baseline encounter probability is low. We discuss
 4093 these two formulations of the bivariate normal model further in Chapt. ??.

4094 5.4.2 Empirical analysis

4095 For any encounter model we can compute space usage quantiles empirically by taking a fine
 4096 grid of points and either simulating movement outcomes with probabilities proportional to
 4097 $p(\mathbf{x}, \mathbf{s})$ and accumulating area around \mathbf{s} , or else we can do this precisely by varying $B(\alpha)$
 4098 to find that value within which 95% of all movements are concentrated, i.e., the set of all
 4099 \mathbf{x} such that $\|\mathbf{x} - \mathbf{s}\| \leq B(q)$. Under any detection model, movement outcomes will occur
 4100 in proportion to $p(\mathbf{x}, \mathbf{s})$, as long as the probability of encounter is constant, *conditional on*
 4101 use, and so we can define our space usage distribution according to:

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

4102 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

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

4103 (here, we use \ni to mean “such that”). We have a function called `hra` in the `scrbook`
 4104 package that computes the home range area for any encounter model and prescribed
 4105 parameter values. The help file for `hra` has an example of simulating some data. The

4106 following commands illustrate this calculation for two different bivariate normal models
 4107 of space usage:

```

4108 ##
4109 ## Define encounter probability model as R function
4110 ##
4111 > pGauss2 <- function(parms,Dmat){
4112   a0 <- parms[1]
4113   sigma <- parms[2]
4114   lp <- parms[1] -(1/(2*parms[2]*parms[2]))*Dmat*Dmat
4115   p <- 1-exp(-exp(lp))
4116   p
4117 }
4118
4119 > pGauss1 <- function(parms,Dmat){
4120   a0 <- parms[1]
4121   sigma <- parms[2]
4122   p <- plogis(parms[1])*exp( -(1/(2*parms[2]*parms[2]))*Dmat*Dmat )
4123   p
4124 }
4125
4126 ##
4127 ## Execute hra with sigma = .3993
4128 ##
4129 > hra(pGauss1,parms=c(-2,.3993),plot=FALSE,xlim=c(0,6),ylim=c(0,6),
4130   ng=500,tol=.0005)
4131
4132 [1] 0.9784019
4133 radius to achieve 95% of area: 0.9784019
4134 home range area: 3.007353
4135 [1] 3.007353
4136
4137
4138 ## Analytic solution:
4139 ##      true sigma that produces area of 3
4140 > sqrt(3/pi)/sqrt(5.99)
4141 [1] 0.3992751

```

4142 What this means is that $B(q) = 0.978$ is the radius that encloses about 95% of all
 4143 movements under the standard bivariate normal encounter model. Therefore, the area is
 4144 about $\pi * 0.978^2 = 3.007$ spatial units. You can change the intercept of the model and find
 4145 that it has no effect. The true (analytic) value of σ that produces a home range area of 3.0
 4146 is 0.3993 which is the value we initially plugged in to the `hra` function. We can improve
 4147 on the numerical approximation to home range area (get it closer to 3.0) by increasing the
 4148 resolution of our spatial grid (increase the `ng` argument) along with the `tol` argument.

4149 We can also reverse this process, and find, for any detection model, the parameter
 4150 values that produce a certain $(1 - q)\%$ home range area, which we imagine would be

4151 useful for doing simulation studies. The function `hra` will compute the value of the scale
 4152 parameter that achieves a certain target $(1 - q)\%$ home range area, by simply providing a
 4153 non-null value of the variable `target.area`. Here we use `target.area = 3.00735` (from
 4154 above) to obtain a close approximation to the value σ we started with (the parameter
 4155 argument is meaningless here):

```
4156 > hra(pGauss1,parms=c(-2,.3993),plot=FALSE,xlim,ylim,ng=500,  

4157   target.area=3.00735,tol=.0005)  

4158  

4159 Value of parm[2] to achieve 95% home range area of 3.00735: 0.3993674
```

4160 **5.4.3 Relevance of understanding space usage**

4161 One important reason that we need to be able to deduce “home range area” from a
 4162 detection model is so that we can compare different models with respect to a common
 4163 biological currency. Many encounter probability models have some “scale parameter”,
 4164 which we might call σ no matter the model, but this relates to 95% area in a different
 4165 manner under each model. Therefore, we want to be able to convert different models
 4166 to the same currency. Another reason to understand the relationship between models of
 4167 encounter probability and space usage is that it opens the door to combining traditional
 4168 resource selection data from telemetry with spatial capture-recapture data. In Chapt. ??
 4169 we consider this problem, for the case in which a sample of individuals produces encounter
 4170 history data suitable for SCR models and, in addition, we have telemetry relocations on a
 4171 sample of individuals. This is achieved by regarding the two sources of data as resulting
 4172 from the same underlying process of space usage but telemetry data produce “perfect”
 4173 observations, like always-on camera traps blanketing a landscape. We use this idea to
 4174 model the effect of a measured covariate at each pixel, say $C(\mathbf{x})$, on home range size and
 4175 geometry and, hence, the probability of encounter in traps.

4176 **5.4.4 Contamination due to behavioral response**

4177 Interpretation of encounter probability models as models of animal home range and space
 4178 usage can be complicated by a number of factors, including whether traps are baited
 4179 or not. In the case of baited traps, this might lead to a behavioral response (Sec. ??)
 4180 which could affect animal space usage. For example, if traps attract animals from a long
 4181 distance, it could make typical home ranges appear larger than normal. More likely, in our
 4182 view, it wouldn’t change the typical size of a range but would change how individuals use
 4183 their range e.g., by moving from baited trap to baited trap, so that observed movement
 4184 distances of individuals are typically larger than normal.

4185 In other cases, the reliance on Euclidean distance in models for encounter probability
 4186 might be unrealistic, and can lead to biased estimates of density (?). For example, animals
 4187 might concentrate their movements along trails, roads, or other landscape features. In this
 4188 case, models that accommodate other distance metrics can be considered. We present
 4189 models based on least-cost path in Chapt. ??.

5.5 SIMULATING SCR DATA

4190 It is always useful to simulate data because it allows you to understand the system that
 4191 you're modeling and also calibrate your understanding with specific values of the model
 4192 parameters. That is, you can simulate data using different parameter values until you
 4193 obtain data that "look right" based on your knowledge of the specific situation that
 4194 you're interested in. Here we provide a simple script to illustrate how to simulate spatial
 4195 encounter history data. In this exercise we simulate data for 100 individuals and a 25 trap
 4196 array laid out in a 5×5 grid of unit spacing. The specific encounter model is the Gaussian
 4197 model given above and we used this code to simulate data used in subsequent analyses.
 4198 The 100 activity centers were simulated on a state-space defined by a 8×8 square within
 4199 which the trap array was centered (thus the trap array is buffered by 2 units). Therefore,
 4200 the density of individuals in this system is fixed at $100/64$.

```

4201 > set.seed(2013)
4202 # Create 5 x 5 grid of trap locations with unit spacing
4203 > traplocs <- cbind(sort(rep(1:5,5)),rep(1:5,5))
4204 > ntraps <- nrow(traplocs)
4205 # Compute distance matrix:
4206 > Dmat <- e2dist(traplocs,traplocs)
4207
4208
4209 # Define state-space of point process. (i.e., where animals live).
4210 # "buffer" just adds a fixed buffer to the outer extent of the traps.
4211 #
4212 > buffer <- 2
4213 > xlim <- c(min(traplocs[,1] - buffer),max(traplocs[,1] + buffer))
4214 > ylim <- c(min(traplocs[,2] - buffer),max(traplocs[,2] + buffer))
4215
4216 > N <- 100    # population size
4217 > K <- 20     # number nights of effort
4218
4219 > sx <- runif(N,xlim[1],xlim[2])    # simulate activity centers
4220 > sy <- runif(N,ylim[1],ylim[2])
4221 > S <- cbind(sx,sy)
4222 # Compute distance matrix:
4223 > D <- e2dist(S,traplocs) # distance of each individual from each trap
4224
4225 > alpha0 <- -2.5      # define parameters of encounter probability
4226 > sigma <- 0.5        # scale parameter of half-normal
4227 > alpha1 <- 1/(2*sigma*sigma) # convert to coefficient on distance
4228
4229 # Compute Probability of encounter:
4230 #
4231 > probcap <- plogis(-2.5)*exp( - alpha1*D*D)
4232
4233 # Generate the encounters of every individual in every trap

```

```

4234 > Y <- matrix(NA,nrow=N,ncol=ntraps)
4235 > for(i in 1:nrow(Y)){
4236   Y[i,] <- rbinom(ntraps,K,probcap[i,])
4237 }
```

4238 We remind the reader that, in presenting **R** or other code snippets throughout the
 4239 book, we will deviate from our standard variable expressions for some quantities. In
 4240 particular, we sometimes substitute words for integer variable designations: **nind** (for n),
 4241 **ntraps** (for J), and **nocc** (for K). In our opinion this leaves less to be inferred by the
 4242 reader in trying to understand code snippets.

4243 Subsequently we will generate data using this code packaged in an **R** function called
 4244 **simSCRO** in the package **scrbook** which takes a number of arguments including **discard0**
 4245 which, if TRUE, will return only the encounter histories for captured individuals. A second
 4246 argument is **array3d** which, if TRUE, returns the 3-dimensional encounter history array
 4247 instead of the aggregated **nind** \times **ntraps** encounter frequencies (see below). Finally we
 4248 provide a random number seed, **rnd** = 2013 to ensure repeatability of the analysis here.
 4249 We obtain a data set as above using the following command:

```

4250 > data <- simSCRO(discard0=TRUE, array3d=FALSE, rnd=2013)
```

4251 The **R** object **data** is a list, so let's take a look at what's in the list and then harvest some
 4252 of its elements for further analysis below.

```

4253 > names(data)
4254 [1] "Y"      "traplocs" "xlim"      "ylim"      "N"       "alpha0"    "beta"
4255 [8] "sigma"   "K"
4256
4257 ## Grab encounter histories from simulated data list
4258 > Y <- data$Y
4259 ## Grab the trap locations
4260 > traplocs <- data$traplocs
```

4261 5.5.1 Formatting and manipulating real data sets

4262 Conventional capture-recapture data are easily stored and manipulated as a 2-dimensional
 4263 array, an **nind** \times **K** (individuals by sample occasions) matrix, which is maximally informative
 4264 for any conventional capture-recapture model, but not for spatial capture-recapture
 4265 models. For SCR models we must preserve the spatial information in the encounter history
 4266 information. We will routinely analyze data from 3 standard formats:

- 4267 (1) The basic 2-dimensional data format, which is an **nind** \times **ntraps** encounter frequency
 4268 matrix such as that simulated previously. These are the total number of encounters in
 4269 each trap, summed over the K sample occasions.
- 4270 (2) The maximally informative 3-dimensional array, for which we establish here the convention
 4271 that it has dimensions **nind** \times **ntraps** \times **K**.
- 4272 (3) We use a compact format – the “encounter data file” – which we describe below in
 4273 Sec. ??.

4274 To simulate data in the most informative format - the “3-d array” - we can use the **R**
 4275 commands given previously but replace the last 4 lines with the following:

```
4276 > Y <- array(NA,dim=c(N,ntraps,K))
4277
4278 > for(i in 1:nrow(Y)){
4279   for(j in 1:ntraps){
4280     Y[i,j,1:K] <- rbinom(K,1,probcap[i,j])
4281   }
4282 }
```

4283 We see that a collection of K binary encounter events are generated for *each* individual
 4284 and for *each* trap. The probabilities of those Bernoulli trials are computed based on the
 4285 distance from each individual’s home range center and the trap (see calculation above),
 4286 and those are housed in the matrix `probcap`. Our data simulator function `simSRC0` will
 4287 return the full 3-d array if `array3d=TRUE` is specified in the function call. To recover the
 4288 2-d matrix from the 3-d array, and subset the 3-d array to individuals that were captured,
 4289 we do this:

```
4290 # Sum over the ‘‘sample occasions’’ dimension (3rd margin of the array)
4291 > Y2d <- apply(Y,c(1,2),sum)
4292
4293 # Compute how many times each individual was captured
4294 > ncaps <- apply(Y2d,1,sum)
4295
4296 # Keep those individuals that were captured
4297 > Y <- Y[ncaps>0,,]
```

5.6 FITTING MODEL SCR0 IN BUGS

4298 Clearly if we somehow knew the value of N then we could fit this model directly because,
 4299 in that case, it is a special kind of logistic regression model, one with a random effect (`s`)
 4300 that enters into the model in a peculiar fashion, and also with a distribution (uniform)
 4301 which we don’t usually think of as standard for random effects models. So our aim here is
 4302 to analyze the known- N problem, using our simulated data, as an incremental step in our
 4303 progress toward fitting more generally useful models. To begin, we use our simulator to
 4304 grab a data set and then harvest the elements of the resulting object for further analysis.

```
4305 > data <- simSRC0(discard0=FALSE,rnd=2013)
4306 > y <- data$Y
4307 > traplocs <- data$traplocs
4308
4309 # In this case nind=N because we’re doing the known-N problem
4310 #
4311 > nind <- nrow(y)
4312 > X <- data$traplocs
4313 > J <- nrow(X)    # number of traps
4314 > K <- data$K
```

```
4315 > xlim <- data$xlim
4316 > ylim <- data$ylim
```

4317 Note that we specify `discard0 = FALSE` so that we have a “complete” data set, i.e.,
 4318 one with the all-zero encounter histories corresponding to uncaptured individuals. Now,
 4319 within an **R** session, we can create the **BUGS** model file and fit the model using the
 4320 following commands.

```
4321 cat("
4322   model{
4323     alpha0 ~ dnorm(0,.1)
4324     logit(p0) <- alpha0
4325     alpha1 ~ dnorm(0,.1)
4326     sigma <- sqrt(1/(2*alpha1))
4327     for(i in 1:N){ # note N here -- N is KNOWN in this example
4328       s[i,1] ~ dunif(xlim[1],xlim[2])
4329       s[i,2] ~ dunif(ylim[1],ylim[2])
4330       for(j in 1:J){
4331         d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
4332         y[i,j] ~ dbin(p[i,j],K)
4333         p[i,j] <- p0*exp(- alpha1*d[i,j]*d[i,j])
4334       }
4335     }
4336   }
4337 ",file = "SCR0a.txt")
```

4338 This model describes the Gaussian encounter probability model, but it would be trivial
 4339 to modify that to various others including the logistic described above. One consequence
 4340 of using the half-normal is that we have to constrain the encounter probability to be in
 4341 $[0, 1]$ which we do here by defining `alpha0` to be the logit of the intercept parameter `p0`.
 4342 Note that the distance covariate is computed within the **BUGS** model specification given
 4343 the matrix of trap locations, `X`, which is provided to **WinBUGS** as data.

4344 Next we do a number of organizational activities including bundling the data for **Win-**
 4345 **BUGS**, defining some initial values, the parameters to monitor and some basic MCMC
 4346 settings. We choose initial values for the activity centers `s` by generating uniform random
 4347 numbers in the state-space but, for the observed individuals, we replace those values by
 4348 each individual’s mean trap coordinate for all encounters

```
4349 ### Starting values for activity centers, s
4350 > sst <- cbind(runif(nind,xlim[1],xlim[2]),runif(nind,ylim[1],ylim[2]))
4351 > for(i in 1:nind){
4352   if(sum(y[i,])==0) next
4353   sst[i,1] <- mean( X[y[i,>0,1] )
4354   sst[i,2] <- mean( X[y[i,>0,2] )
4355 }
4356 > data <- list (y=y, X=X, K=K, N=nind, J=J, xlim=xlim, ylim=ylim)
4357 > inits <- function(){
```

```

4359     list (alpha0=rnorm(1,-4,.4), alpha1=runif(1,1,2), s=sst)
4360   }
4361
4362 > library(R2WinBUGS)
4363 > parameters <- c("alpha0","alpha1","sigma")
4364 > out <- bugs (data, inits, parameters, "SCR0a.txt", n.thin=1, n.chains=3,
4365           n.burnin=1000,n.iter=2000,debug=TRUE,working.dir=getwd())

```

4366 There is little to say about the preceding operations other than to suggest that you might
 4367 explore the output and investigate additional analyses by running the `simSCR0` script
 4368 provided in the **R** package `scrbook`.

4369 For purposes here, we ran 1000 burn-in and 1000 post-burn-in iterations, and 3 chains,
 4370 to obtain 3000 posterior samples. Because we know N for this particular data set we only
 4371 have 2 parameters of the detection model to summarize (`alpha0` and `alpha1`), along with
 4372 the derived parameter σ , the scale parameter of the Gaussian kernel, i.e., $\sigma = \sqrt{1/(2\alpha_1)}$.
 4373 When the object `out` is produced we print a summary of the results as follows:

```

4374 > print(out,digits=2)
4375 Inference for Bugs model at "SCR0a.txt", fit using WinBUGS,
4376   3 chains, each with 2000 iterations (first 1000 discarded)
4377   n.sims = 3000 iterations saved
4378     mean    sd   2.5%   25%   50%   75% 97.5% Rhat n.eff
4379 alpha0   -2.50  0.22 -2.95 -2.65 -2.48 -2.34 -2.09 1.01  190
4380 alpha1    2.44  0.42  1.64  2.15  2.44  2.72  3.30 1.00  530
4381 sigma     0.46  0.04  0.39  0.43  0.45  0.48  0.55 1.00  530
4382 deviance 292.80 21.16 255.60 277.50 291.90 306.00 339.30 1.01  380
4383
4384
4385 [...some output deleted...]
4386

```

4387 We know the data were generated with `alpha0 = -2.5` and `alpha1 = 2`. The estimates
 4388 look reasonably close to those data-generating values and we probably feel pretty good
 4389 about the performance of the Bayesian analysis and MCMC algorithm that **WinBUGS**
 4390 cooked-up based on our sample size of 1 data set. It is worth noting that the `Rhat`
 4391 statistics indicate reasonable convergence but, as a practical matter, we might choose to
 4392 run the MCMC algorithm for additional time to bring these closer to 1.0 and to increase
 4393 the effective posterior sample size (`n.eff`). Other summary output includes “deviance”
 4394 and related things including the deviance information criterion (DIC). We discuss general
 4395 issues of convergence and other MCMC considerations in Chapt. ??, and DIC and model
 4396 selection in Chapt. ??.

5.7 UNKNOWN N

4397 In all real applications N is unknown. We handled this important issue in Chapt. ??
 4398 using the method of data augmentation (DA) which we apply here to achieve a realistic
 4399 analysis of model SCR0. As with the basic closed population models considered previously,

4400 we formulate the problem by augmenting our observed data set with a number of “all-
 4401 zero” encounter histories - what we referred to in Chapt. ?? as potential individuals.
 4402 If n is the number of observed individuals, then let $M - n$ be the number of potential
 4403 individuals in the data set. For the 2-dimensional y_{ij} data structure (n individual $\times J$
 4404 traps encounter frequencies) we simply add additional rows of all-zero observations to
 4405 that data set. Because such “individuals” are unobserved, they therefore necessarily have
 4406 $y_{ij} = 0$ for all j . A data set, say with 4 traps and 6 individuals, augmented with 4
 4407 pseudo-individuals therefore might look like this:

```
4408      trap1 trap2 trap3 trap4
4409  [1,]    1    0    0    0
4410  [2,]    0    2    0    0
4411  [3,]    0    0    0    1
4412  [4,]    0    1    0    0
4413  [5,]    0    0    1    1
4414  [6,]    1    0    1    0
4415  [7,]    0    0    0    0
4416  [8,]    0    0    0    0
4417  [9,]    0    0    0    0
4418  [10,]   0    0    0    0
```

4419 We typically have more than 4 traps and, if we’re fortunate, many more individuals in
 4420 our data set.

4421 For the augmented data set, we introduce a set of binary latent variables (the data
 4422 augmentation variables), z_i , and the model is extended to describe $\Pr(z_i = 1)$ which is, in
 4423 the context of this problem, the probability that an individual in the augmented data set
 4424 is a member of the population of size N that was exposed to sampling. In other words,
 4425 if $z_i = 1$ for one of the all-zero encounter histories, this is implied to be a sampling zero
 4426 whereas observations for which $z_i = 0$ are “structural zeros” under the model. Under DA,
 4427 we also express the binomial observation model *conditional on z_i* as follows:

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

4428 where we see that the binomial probability evaluates to 0 if $z_i = 0$ (so y_{ij} is a fixed 0 in
 4429 that case) and evaluates to p_{ij} if $z_i = 1$.

4430 How big does the augmented data set have to be? We discussed this issue in Chapt.
 4431 ?? where we noted that the size of the data set is equivalent to the upper limit of a uniform
 4432 prior distribution on N . Practically speaking, it should be sufficiently large so that the
 4433 posterior distribution for N is not truncated. On the other hand, if it is too large then
 4434 unnecessary calculations are being done. An approach to choosing M by trial-and-error is
 4435 indicated. Do a short MCMC run and then consider whether you need to increase M . See
 4436 Chapt. ?? for an example of this. ?, Chapt. 6 provide an assessment of choosing M in
 4437 closed population models. The useful thing about DA is that it removes N as an explicit
 4438 parameter of the model. Instead, N is a derived parameter, computed by $N = \sum_{i=1}^M z_i$.
 4439 Similarly, *density*, D , is also a derived parameter computed as $D = N/\text{area}(\mathcal{S})$.

4440 5.7.1 Analysis using data augmentation in WinBUGS

4441 We provide a complete **R** script for simulating and organizing a data set, and analyzing
 4442 the data in **WinBUGS**. As before we begin by obtaining a data set using our `simSCR0`
 4443 function and then harvesting the required data objects from the resulting data list. Note
 4444 that we use the `discard0=TRUE` option this time so that we get a “real looking” data set
 4445 with no all-zero encounter histories:

```
4446 ##  
4447 ## Simulate the data and extract the required objects  
4448 ##  
4449 > data <- simSCR0(discard0=TRUE,rnd=2013)  
4450 > y <- data$Y  
4451 > nind <- nrow(y)  
4452 > X <- data$traplocs  
4453 > K <- data$K  
4454 > J <- nrow(X)  
4455 > xlim <- data$xlim  
4456 > ylim <- data$ylim
```

4457 After harvesting the data we augment the data matrix y with $M - n$ all-zero encounter
 4458 histories, and create starting values for the variables z_i and also the activity centers s_i
 4459 of which, for each, we require M values. One thing to take care of in using the **BUGS**
 4460 engines is the starting values for the activity centers. It is usually helpful to start the s_i
 4461 for each observed individual at or near the trap(s) it was captured. All of this happens as
 4462 follows:

```
4463 ## Data augmentation  
4464 > M <- 200  
4465 > y <- rbind(y,matrix(0,nrow=M-nind,ncol=ncol(y)))  
4466 > z <- c(rep(1,nind),rep(0,M-nind))  
4467  
4468 ## Starting values for s  
4469 > sst <- cbind(runif(M,xlim[1],xlim[2]),runif(M,ylim[1],ylim[2]))  
4470 > for(i in 1:nind){  
4471   sst[i,1] <- mean( X[y[i,]>0,1] )  
4472   sst[i,2] <- mean( X[y[i,]>0,2] )  
4473 }
```

4474 Next, we write out the **BUGS** model specification and save it to an external file
 4475 called `SCR0b.txt`. The model specification now includes M encounter histories including
 4476 the augmented potential individuals, the data augmentation parameters z_i , and the data
 4477 augmentation parameter ψ :

```
4478 > cat("model{  
4479   alpha0 ~ dnorm(0,.1)  
4480   logit(p0) <- alpha0
```

```

4482 alpha1 ~ dnorm(0,.1)
4483 sigma <- sqrt(1/(2*alpha1))
4484 psi ~ dunif(0,1)
4485
4486 for(i in 1:M){
4487   z[i] ~ dbern(psi)
4488   s[i,1] ~ dunif(xlim[1],xlim[2])
4489   s[i,2] ~ dunif(ylim[1],ylim[2])
4490   for(j in 1:J){
4491     d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
4492     y[i,j] ~ dbin(p[i,j],K)
4493     p[i,j] <- z[i]*p0*exp(- alpha1*d[i,j]*d[i,j])
4494   }
4495 }
4496 N <- sum(z[])
4497 D <- N/64
4498 }
4499 ",file = "SCR0b.txt")

```

4500 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:

```

4503 > data <- list (y=y, X=X, K=K, M=M, J=J, xlim=xlim, ylim=ylim)
4504 > inits <- function(){
4505   list (alpha0=rnorm(1,-4,.4), alpha1=runif(1,1,2), s=sst, z=z)
4506 }
4507
4508 > library(R2WinBUGS)
4509 > parameters <- c("alpha0","alpha1","sigma","N","D")
4510 > out <- bugs (data, inits, parameters, "SCR0b.txt", n.thin=1,n.chains=3,
4511   n.burnin=1000,n.iter=2000,debug=TRUE,working.dir=getwd())

```

4512 Note the differences in this new **WinBUGS** model with that appearing in the known-
4513 N version – there are not many! The loop over individuals goes up to M now, and there is a
4514 model component for the DA variables z . We are also computing some derived parameters:
4515 population size $N(\mathcal{S})$ is computed by summing up all of the data augmentation variables
4516 z_i (as we've done previously in Chapt. ??) and density, D , is also a derived parameter,
4517 being a function of N . The input data has changed slightly too, as the augmented data
4518 set has more rows to include excess all-zero encounter histories. Previously we knew that
4519 $N = 100$ but in this analysis we pretend not to know N , but think that $N = 200$ is a
4520 good upper bound. This analysis can be run directly using the **SCR0bayes** function once
4521 the **scrbook** package is loaded, by issuing the following commands:

```

4522 > library(scrbook)
4523 > data <- simSCR0(discard0=TRUE,rnd=2013)
4524 > out1 <- SCR0bayes(data,M=200,engine="winbugs",ni=2000,nb=1000)

```

4525 Summarizing the output from **WinBUGS** produces:

```

4526 > print(out1,digits=2)
4527 Inference for Bugs model at "SCR0b.txt", fit using WinBUGS,
4528   3 chains, each with 2000 iterations (first 1000 discarded)
4529   n.sims = 3000 iterations saved
4530     mean    sd   2.5%   25%   50%   75% 97.5% Rhat n.eff
4531 alpha0   -2.57  0.23  -3.04  -2.72  -2.56  -2.41  -2.15 1.01   320
4532 alpha1    2.46  0.42   1.63   2.16   2.46   2.73   3.33 1.02   120
4533 sigma     0.46  0.04   0.39   0.43   0.45   0.48   0.55 1.02   120
4534 N        113.62 15.73  86.00 102.00 113.00 124.00 147.00 1.01   260
4535 D         1.78  0.25   1.34   1.59   1.77   1.94   2.30 1.01   260
4536 deviance 302.60 23.67 261.19 285.47 301.50 317.90 354.91 1.00 1400
4537
4538 [...some output deleted...]
4539

```

4540 The **Rhat** statistic (discussed in Secs. ?? and ??) for this analysis indicates satisfactory
 4541 convergence. We see that the estimated parameters (α_0 and α_1) are comparable to the
 4542 previous results obtained for the known- N case, and also not too different from the data-
 4543 generating values. The posterior of N overlaps the data-generating value substantially.

4544 **Use of other BUGS engines: JAGS**

4545 There are two other popular **BUGS** engines in widespread use: **OpenBUGS** (?) and
 4546 **JAGS** (?). Both of these are easily called from **R**. **OpenBUGS** can be used instead of
 4547 **WinBUGS** by changing the package option in the **bugs** call to **package='OpenBUGS'**.
 4548 **JAGS** can be called using the function **jags()** in package **R2jags** which has nearly the
 4549 same arguments as **bugs()**. Or, it can be executed from the **R** package **rjags** (?) which
 4550 has a slightly different implementation that we demonstrate here as we reanalyze the
 4551 simulated data set in the previous section (note: the same **R** commands are used to
 4552 generate the data and package the data, inits and parameters to monitor). The function
 4553 **jags.model** is used to initialize the model and run the MCMC algorithm for an adaptive
 4554 period during which tuning of the MCMC algorithm might take place. These samples
 4555 cannot be used for inference. Then the Markov chains are updated using **coda.samples()**
 4556 to obtain posterior samples for analysis, as follows:

```

4557 > jinit <- jags.model("SCR0b.txt", data=data, inits=inits,
4558   n.chains=3, n.adapt=1000)
4559 > jout <- coda.samples(jinit, parameters, n.iter=1000, thin=1)

```

4560 These commands can be executed using the function **SCR0bayes** provided with the
 4561 **R** package **scrbook**. ? provides a good introduction to ecological modeling with **JAGS**
 4562 which we recommend.

4563 **5.7.2 Implied home range area**

4564 Here we apply the method described in Sec. ?? to compute the effective home range
 4565 area under different encounter probability models fit to simulated data. We simulated a
 4566 data set from the Gaussian kernel model as in Sec. ?? and then we fitted 4 models to it:
 4567 (1) the true data-generating Gaussian encounter probability model; (2) the “hazard” or

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

4568 complementary log-log link model (Eq. ??); (3) the negative exponential model and (4)
 4569 the logit model (Eq. ??). We modified the function `SCR0bayes` for this purpose which
 4570 you should be able to do with little difficulty. We fit each model to the same simulated
 4571 data set using **WinBUGS**, based only on 1000 post-burn-in samples and 3 chains, which
 4572 produced the posterior summaries given in Table ???. The main thing we see is that, while
 4573 the implied home range area can vary substantially, there are smaller differences in the
 4574 estimated N and hence D .

4575 5.7.3 Realized and expected density

4576 In Bayesian analysis of the SCR model, we estimate a parameter N which is the size
 4577 of the population for the prescribed state-space (presumably the state-space is defined
 4578 so as to be relevant to where our traps were located, so N can be thought of as the
 4579 size of the sampled population). In the context of ? this is the *realized* population
 4580 size. Conversely, sometimes we see estimates of *expected* population size reported, which
 4581 are estimates of $\mathbb{E}(N)$, the expected size of some hypothetical, unspecified population.
 4582 Usually the distinction between realized and expected population size is not made in SCR
 4583 models, because almost everyone only cares about actual populations – and their realized
 4584 population size.

4585 If you do likelihood analysis of SCR models, then the distinction between realized and
 4586 expected is often discussed by whether the estimator is “conditional on N ” (realized) or
 4587 not (expected). The naming arises because in obtaining the MLE of N , its properties are
 4588 evaluated *conditional* on N – in particular, if the estimator is unbiased then $\mathbb{E}(\hat{N}|N) = N$
 4589 and $\text{Var}(\hat{N}|N) = \tilde{\sigma}_{\hat{N}}^2$ is the sampling variance. This does not conform to any concept or
 4590 quantity that is relevant to Bayesian inference. If we care about N for the population that
 4591 we sampled it is understood to be a realization of a random variable, but the relevance of
 4592 “conditional on N ” is hard to see. Bayesian analysis will provide a prediction of N that
 4593 is based on the posterior $[N|y, \theta]$ – which is certainly *not* conditional on N .

4594 There is a third type of inference objective that is relevant in practice and that is
 4595 prediction of N for a population that was not sampled – i.e., a “new” population. To
 4596 elaborate on this, consider a situation in which we are concerned about the tiger population
 4597 in 2 distinct reserves in India. We do a camera trapping study on one of the reserves to
 4598 estimate N_1 and we think the reserves are similar and homogeneous so we’re willing to
 4599 apply a density estimate based on N_1 to the 2nd reserve. For the 2nd reserve, do we want
 4600 a prediction of the realized population size, N_2 , or do we want an estimates of its expected
 4601 value? We believe the former is the proper quantity for inference about the population

size in the 2nd reserve. An estimate of N_2 should include the uncertainty with which the mean is estimated (from reserve 1) and it should also include “process variation” for making the prediction of the latent variable N_2 .

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

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

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

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

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

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

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

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

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

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

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

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

- 4641 (2) We need to estimate N for some population that we didn't sample but it is "similar"
4642 to the population that we have information on. In this case, we have to account for
4643 both variation in having to estimate parameters of the distribution of N and we have
4644 to account for process variation in N (i.e., due to the stochastic model of N).
4645 (3) In some extremely limited cases we might care about estimating the expected value of
4646 N , $\mathbb{E}(N)$. This is only useful as a hypothetical statement that we might use, e.g., if we
4647 were to establish a new million ha refuge somewhere, then we might say its expected
4648 population size is 200 tigers.

5.8 THE CORE SCR ASSUMPTIONS

4649 It's always a good idea to sit down and reflect on the meaning of any particular model,
4650 its various assumptions, and what they mean in a specific context. From the statistician's
4651 point of view, the basic assumption, the omnibus assumption, as in all of statistics, and
4652 for every statistical model, is that "the model is correctly specified". So, naturally, that
4653 precludes everything that isn't explicitly addressed by the model. To point this out to
4654 someone seems to cause a lot of anxiety, so we enumerate here what we think are the most
4655 important statistical assumptions of the basic SCR0 model:

- 4656 • **Demographic closure.** The model does not allow for demographic processes. There
4657 is no recruitment or entry into the sampled population. There is no mortality or exit
4658 from the sampled population.
- 4659 • **Geographic closure.** We assume no permanent emigration or immigration from the
4660 state-space. However, we allow for "temporary" movements around the state-space
4661 and variable exposure to encounter as a result. The whole point of SCR models is to
4662 accommodate this dynamic. In ordinary capture-recapture models we have to assume
4663 geographic closure to interpret N in a meaningful way.
- 4664 • **Activity centers are randomly distributed.** That is, uniformity and independence
4665 of the underlying point process s_1, \dots, s_N (see next section).
- 4666 • **Detection is a function of distance.** A detection model that describes how encounter
4667 probability declines as a function of distance from an individual's home range center.
- 4668 • **Independence of encounters** among individuals. Encounter of any individual is
4669 independent of encounter of each other individual.
- 4670 • **Independence of encounters** of the same individual. Encounter of an individual
4671 in any trap is independent of its encounter in any other trap, and subsequent sample
4672 occasion.

4673 It's easy to get worried and question the whole SCR enterprise just on the grounds that
4674 these assumptions combine to form such a simplistic model, one that surely can't describe
4675 the complexity of real populations. On this sentiment, a few points are worth making.
4676 First, you don't have inherently fewer assumptions by using an ordinary capture-recapture
4677 model but, rather, the SCR model relaxes a number of important assumptions compared
4678 to the non-spatial counterpart. For one, here, we're not assuming that p is constant for all
4679 individuals but rather that p varies substantially as a matter of the spatial juxtaposition of
4680 individuals with traps. So maybe the manner in which p varies isn't quite right, but that's
4681 not an argument that supports doing less modeling. Fundamentally a distance-based
4682 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. ??).

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* (??). The study took place in SE Alaska (Fig. ??) 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. ?? 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.

4716 Encounter Data File (EDF) – We store the encounter data in the an efficient file format
4717 which is easily manipulated in **R** and easy to create in Excel and other spreadsheets which
4718 are widely used for data management. The file structure is a simple matrix with 4 columns,
4719 those being: (1) **session ID**: the trap *session* which usually corresponds to a year or a
4720 primary period in the context of a Robust Design situation, but it could also correspond to
4721 a distinct spatial unit (see Sec. ?? and Chapt. ??). For a single-year study (as considered
4722 here) this should be an integer that is the same for all records; (2) **individual ID**: the



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

4723 individual identity, being an integer from 1 to n (repeated for multiple captures of the
 4724 same individual) indicating which individual the record (row) of the matrix belongs to; (3)
 4725 occasion ID: The integer sample occasion which generated the record, and (4) trap ID:
 4726 the trap identity, an integer from 1 to J , the number of traps. The structure of the EDF
 4727 is the same as used in the **secr** package (?) and similar to that used in the **SPACECAP**
 4728 (?), and **SCRbayes** (?) packages, both of which have a 3-column format (**trapID**, **indID**,
 4729 **sampID**). We note that the naming of the columns is irrelevant as far as anything we do in
 4730 this book, although **secr** and other software may have requirements on variable naming.

4731 To illustrate this format, the wolverine data are available in the package **scrbook** by
 4732 typing:

4733 > data(wolverine)

4734 which contains a list having elements `wcaps` (the EDF) and `wtraps` (the TDF). We see
 4735 that `wcaps` has 115 rows, each representing a unique encounter event including the trap
 4736 identity, the individual identity and the sample occasion index (`sample`). The first 5 rows
 4737 of `wcaps` are:

```
4738 > wolverine$wcaps[1:5,]
4739   year individual day trap
4740 [1,]    1        2 127    1
4741 [2,]    1        2 128    1
4742 [3,]    1        2 129    1
4743 [4,]    1       18 130    1
4744 [5,]    1        3 106    2
```

4745 The 1st column here, labeled `year`, is an integer indicating the year or session of the
 4746 encounter. All these data come from a single year (2008) and so `year` is set to 1. Variable
 4747 `individual` is an integer identity of each individual captured, `day` is the sample occasion of
 4748 capture (in this case, the sample occasions correspond to days), and `trap` is the integer trap
 4749 identity. The variable `trapid` will have to correspond to the row of a matrix containing
 4750 the trap coordinates - in this case the TDF file `wtraps` which we describe further below.

4751 Note that the information provided in this encounter data file `wcaps` does not represent
 4752 a completely informative summary of the data. For example, if no individuals were
 4753 captured in a certain trap or during a certain period, then this compact data format will
 4754 have no record. Thus we will need to know J , the number of traps, and K , the number of
 4755 sample occasions when reformatting this SCR data format into a 2-d encounter frequency
 4756 matrix or 3-d array. In addition, the encounter data file does not provide information
 4757 about which periods each trap was operated. This additional information is also necessary
 4758 as the trap-specific sample sizes must be passed to **BUGS** as data. We provide this
 4759 information along with trap coordinates, in the “trap deployment file” (TDF) which is
 4760 described below.

4761 For our purposes, we need to convert the `wcaps` file into the $n \times J$ array of binomial
 4762 encounter frequencies, although more general models might require an encounter-history
 4763 formulation of the model which requires a full 3-d array. To obtain our encounter frequency
 4764 matrix, we do this the hard way by first converting the encounter data file into a 3-d array
 4765 and then summarize to trap totals. We have a handy function `SCR23darray` which takes
 4766 the compact encounter data file, and converts it to a 3-d array, and then we use the **R**
 4767 function `apply` to summarize over the sample occasion dimension (by convention here,
 4768 this is the 2nd dimension). To apply this to the wolverine data in order to compute the
 4769 3-d array we do this:

```
4770 > y3d <- SCR23darray(wolverine$wcaps,wolverine$wtraps)
4771 > y <- apply(y3d,c(1,2),sum)
```

4772 See the help file for more information on `SCR23darray`. The 3-d array is necessary to
 4773 fit certain types of models (e.g., behavioral response) and this is why we sometimes will
 4774 require this maximally informative 3-d data format but, here, we analyze the summarized
 4775 data.

4776 **Trap Deployment File (TDF)** – The other important information needed to fit SCR
 4777 models is the “trap deployment file” (TDF) which provides additional information not

4778 contained in the encounter data file. The traps file has $K + 3$ columns. The first column is
 4779 assumed to be a trap identifier, columns 2 and 3 are the easting and northing coordinates
 4780 (assumed to be in a Euclidean coordinate system), and columns 4 to $K + 3$ are binary
 4781 indicators of whether each trap was operational during each sample occasion. The first 10
 4782 rows (out of 37) and 10 columns (out of 167) of the trap deployment file for the wolverine
 4783 data are shown as follows:

```
4784 > wolverine$wtraps[1:10,1:10]
4785
4786   Easting Northing 1 2 3 4 5 6 7 8
4787 1 632538 6316012 0 0 0 0 0 0 0 0
4788 2 634822 6316568 1 1 1 1 1 1 1 1
4789 3 638455 6309781 0 0 0 0 0 0 0 0
4790 4 634649 6320016 0 0 0 0 0 0 0 0
4791 5 637738 6313994 0 0 0 0 0 0 0 0
4792 6 625278 6318386 0 0 0 0 0 0 0 0
4793 7 631690 6325157 0 0 0 0 0 0 0 0
4794 8 632631 6316609 0 0 0 0 0 0 0 0
4795 9 631374 6331273 0 0 0 0 0 0 0 0
4796 10 634068 6328575 0 0 0 0 0 0 0 0
```

4797 This tells us that trap 2 was operated during occasions (days) 1-7 but the other traps
 4798 were not operational during those periods. It is extremely important to recognize that
 4799 each trap was operated for a variable period of time and thus the binomial “sample size”
 4800 is different for each, and this needs to be accounted for in the **BUGS** model specification.
 4801 To compute the vector of sample sizes K , and extract the trap locations, we do this:

```
4802 > traps <- wolverine$wtraps
4803 > traplocs <- traps[,1:2]
4804 > K <- apply(traps[,3:ncol(traps)],1,sum)
```

4805 This results in a matrix `traplocs` which contains the coordinates of each trap and a vector
 4806 `K` containing the number of days that each trap was operational. We now have all the
 4807 information required to fit a basic SCR model in **BUGS**.

4808 Summarizing the data for the wolverine study, we see that 21 unique individuals were
 4809 captured a total of 115 times. Most individuals were captured 1-6 times, with 4, 1, 4, 3, 1,
 4810 and 2 individuals captured 1-6 times, respectively. In addition, 1 individual was captured
 4811 each 8 and 14 times and 2 individuals each were captured 10 and 13 times. The number
 4812 of unique traps that captured a particular individual ranged from 1-6, with 5, 10, 3, 1, 1,
 4813 and 1 individual captured in each of 1 to 6 different traps, respectively, for a total of 50
 4814 unique wolverine-trap encounters. These numbers might be hard to get your mind around
 4815 whereas some tabular summary is often more convenient. For that it seems natural to
 4816 tabulate individuals by trap and total encounter frequencies. The spatial information in
 4817 SCR data is based on multi-trap captures, and so, it is informative to understand how
 4818 many unique traps each individual is captured in, and the total number of encounters.
 4819 For the wolverine data, we reproduce Table 1 from ? as Table ??.

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

4820 5.9.2 Fitting the model in WinBUGS

4821 Here we fit the simplest SCR model with the Gaussian encounter probability model,
 4822 although we revisit these data and fit additional models in later chapters. Model SCR0 is
 4823 summarized by the following 4 elements:

- 4824 (1) $y_{ij} | \mathbf{s}_i \sim \text{Binomial}(K, z_i p_{ij})$
 4825 (2) $p_{ij} = p_0 \exp(-\alpha_1 ||\mathbf{x}_j - \mathbf{s}_i||^2)$
 4826 (3) $\mathbf{s}_i \sim \text{Uniform}(\mathcal{S})$
 4827 (4) $z_i \sim \text{Bernoulli}(\psi)$

4828 We assume customary flat priors on the structural (hyper-) parameters of the model,
 4829 $\alpha_0 = \text{logit}(p_0)$, α_1 and ψ .

4830 It remains to define the state-space \mathcal{S} . For this, we nested the trap array (Fig. ??)
 4831 in a rectangular state-space extending 20 km beyond the traps in each cardinal direction.
 4832 We scaled the coordinate system so that a unit distance was equal to 10 km, producing a
 4833 rectangular state-space of dimension 9.88×10.5 units ($\text{area} = 10374 \text{ km}^2$) within which
 4834 the trap array was nested. As a general rule, we recommend scaling the state-space so
 4835 that it is defined near the origin $(x, y) = (0, 0)$. While the scaling of the coordinate system
 4836 is theoretically irrelevant, a poorly scaled coordinate system can produce Markov chains
 4837 that mix poorly. The buffer of the state space should be large enough so that individuals
 4838 beyond the state-space boundary are not likely to be encountered (Sec. ??). To evaluate
 4839 this, we fit models for various choices of a rectangular state-space based on buffers from
 4840 1.0 to 5.0 units (10 km to 50 km). In the **R** package **scrbook** we provide a function
 4841 **wolvSCR0** which will fit model SCR0. For example, to fit the model in **WinBUGS** using
 4842 data augmentation with $M = 300$ potential individuals, using 3 Markov chains each of
 4843 12000 total iterations, discarding the first 2000 as burn-in, we execute the following **R**
 4844 commands:

```
4845 > library(scrbook)
4846 > data(wolverine)
4847 > traps <- wolverine$wtraps
4848 > y3d <- SCR23darray(wolverine$wcaps,wolverine$wtraps)
4849 > 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

4850 The argument `buffer` determines the buffer size of the state-space in the scaled units
 4851 (i.e., 10 km). Note that this analysis takes between 1-2 hours on many machines (in 2013)
 4852 so we recommend testing it with lower values of M and fewer iterations. The posterior
 4853 summaries are shown in Table ??.

4854 5.9.3 Summary of the wolverine analysis

4855 We see that the estimated density is roughly consistent as we increase the state-space
 4856 buffer from 15 to 55 km. We do note that the data augmentation parameter ψ (and,
 4857 correspondingly, N) increase with the size of the state space in accordance with the deter-
 4858 ministic relationship $N = D * A$. However, density is more or less constant as we increase
 4859 the size of the state-space beyond a certain point. For the 10 km state-space buffer, we see
 4860 a slight effect on the posterior distribution of D because the state-space is not sufficiently
 4861 large. The full results from the analysis based on 20 km state-space buffer are given in
 4862 Table ??.

4863 Our point estimate of wolverine density from this study, using the posterior mean from
 4864 the state-space based on the 20 km buffer, is approximately 5.77 individuals/1000 km²

4865 with a 95% posterior interval of [3.86, 8.29]. Density is estimated imprecisely which might
 4866 not be surprising given the low sample size ($n = 21$ individuals!). This seems to be a
 4867 basic feature of carnivore studies although it should not (in our view) preclude the study
 4868 of their populations by capture-recapture nor attempts to estimate density or vital rates.

4869 It is worth thinking about this model, and these estimates, computed under a rect-
 4870 angular state space roughly centered over the trapping array (Fig. ??). Does it make
 4871 sense to define the state-space to include, for example, ocean? What are the possible
 4872 consequences of this? What can we do about it? There's no reason at all that the state
 4873 space has to be a regular polygon – we defined it as such here strictly for convenience and
 4874 for ease of implementation in **WinBUGS** where it enables us to specify the prior for the
 4875 activity centers as uniform priors for each coordinate. While it would be possible to define
 4876 a more realistic state-space using some general polygon GIS coverage, it might take some
 4877 effort to implement that in the **BUGS** language but it is not difficult to devise custom
 4878 MCMC algorithms to do that (see Chapt. ??). Alternatively, we recommend using a
 4879 discrete representation of the state-space – i.e., approximate \mathcal{S} by a grid of G points. We
 4880 discuss this in Sec. ??.

4881 5.9.4 Wolverine space usage

4882 The parameter α_1 is related to the home range radius (Sec. ??). For the Gaussian model
 4883 we interpret the scale parameter σ , related to α_1 by $\alpha_1 = 1/(2\sigma^2)$, as the radius of a
 4884 bivariate normal model of space usage. In this case $\sigma = 0.64$ standardized units (10 km),
 4885 which corresponds to $0.64 \times 10 = 6.4$ km. It can be argued then that 95% of space used
 4886 by an individual is within $6.4 \times \sqrt{5.99} = 15.66$ km of the home range center. The effective
 4887 “home range area” is then the area of this circle, which is $\pi \times 15.66^2 = 770.4$ km². Using
 4888 our handy function **hra** we do this:

```
4889 hra(pGauss1,parms=c(-2,1/(2*.64*.64)),xlim=c(-1,7),ylim=c(-1,7))
4890
4891 [1] 7.731408
```

4892 which is in units of 100 km², so 773.1. The difference in this case is due to numerical
 4893 approximation of our all-purpose tool **hra**. This home range size is relatively huge for
 4894 measured home ranges, which range between 100 and 535 km² (?).

4895 ? reported estimates for σ in the range 6.3 – 9.8 km depending on the model, which
 4896 isn't too different than here¹. However, these estimates are larger than the typical home
 4897 range sizes suggested in the literature. One possible explanation is that if a wolverine
 4898 is using traps as a way to get yummy chicken, so it's moving from trap to trap instead
 4899 of adhering to “normal” space usage patterns, then the implied home range size might
 4900 not be worth much biologically. Thus, interpretation of detection models in terms of
 4901 home range area depends on some additional context or assumptions, such as that traps
 4902 don't effect individual space usage patterns. As such, we caution against direct biological

¹ ? 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.

4903 interpretations of home range area based on σ , although SCR models can be extended to
 4904 handle more general, non-Euclidean, patterns of space usage. See Chaps. ?? and ??.

4905 We can calibrate the desired size of the state-space by looking at the estimated home
 4906 range radius of the species. We should target a buffer of width 2 to $3 \times \sigma$ in order that
 4907 the probability of encountering an individual is very close to 0 beyond the prescribed
 4908 state-space. Essentially, by specifying a state-space, we're setting $p = 0$ for individuals
 4909 beyond the prescribed state-space. For the wolverine data, with σ in the range of 6-9 km,
 4910 a state-space buffer of 20 km is sufficiently large.

5.10 USING A DISCRETE HABITAT MASK

4911 The SCR model developed previously in this chapter assumes that individual activity
 4912 centers are distributed uniformly over the prescribed state-space. Clearly this will not
 4913 always be a reasonable assumption. In Chapt. ??, we develop models that allow explicitly
 4914 for non-uniformity of the activity centers by modeling covariate effects on density. A
 4915 simplistic method of affecting the distribution of activity centers, which we address here,
 4916 is to modify the shape and organization of the state-space explicitly. For example, we
 4917 might be able to classify the state-space into distinct blocks of habitat and non-habitat.
 4918 In that case we can remove the non-habitat from the state-space and assume uniformity of
 4919 the activity centers over the remaining portions judged to be suitable habitat. There are
 4920 several ways to approach this: We can use a grid of points to represent the state-space, i.e.,
 4921 by the set of coordinates s_1, \dots, s_G , and assign equal probabilities to each possible value.
 4922 Alternatively, we can retain the continuous formulation of the state-space but attempt
 4923 to describe constraints analytically, or we can use polygon clipping methods to enforce
 4924 constraints on the state-space in the MCMC analysis. We focus here on the formulation of
 4925 the basic SCR model in terms of a discrete state-space but in Chapt. ?? we demonstrate
 4926 the latter approach based on using polygon operations to define an irregular state-space.
 4927 Use of a discrete state-space can be computationally expensive in **WinBUGS**. That said,
 4928 it isn't too difficult to perform the MCMC calculations in **R** (discussed in Chapt. ??).
 4929 The **R** package **SPACECAP** (?) arose from the **R** implementation of the SCR model in ?.

4930 While clipping out non-habitat seems like a good idea, we think investigators should
 4931 go about this very cautiously. We might prefer to do it when non-habitat represents a
 4932 clear-cut restriction on the state-space such as a reserve boundary or a lake, ocean or
 4933 river. But, having the capability to do this also causes people to start defining "habitat"
 4934 vs. "non-habitat" based on their understanding of the system whereas it can't be known
 4935 whether the animal being studied has the same understanding. Moreover, differentiating
 4936 the landscape by habitat or habitat quality must affect the geometry and morphology
 4937 of home ranges (see Chapt. ??) much more so than the plausible locations of activity
 4938 centers. That is, a home range centroid could, in actual fact, occur in a shopping mall
 4939 parking lot if there is pretty good habitat around the shopping mall, so there is probably
 4940 no sense preclude it as the location for an activity center. It would generally be better to
 4941 include some definition of habitat quality in the model for the detection probability (?)
 4942 which we address in Chaps. ?? and ??.

4943 5.10.1 Evaluation of coarseness of habitat mask

4944 The coarseness of the state-space should not really have much of an effect on estimates
 4945 if the grain is sufficiently fine relative to typical animal home range sizes. Why is this?
 4946 We have two analogies that can help us understand. First is the relationship to model
 4947 M_h . As noted in Sec. ?? above, we can think about SCR models as a type of finite
 4948 mixture (??) where we are fortunate to be able to obtain direct information about which
 4949 group individuals belong to (group being location of activity center). In the standard
 4950 finite mixture models we typically find that a small number of groups (e.g., 2 or 3 at the
 4951 most) can explain high levels of heterogeneity and are adequate for most data sets of small
 4952 to moderate sample sizes. We therefore expect a similar effect in SCR models when we
 4953 discretize the state-space. We can also think about discretizing the state-space as being
 4954 related to numerical integration where we find (see Chapt. ??) that we don't need a
 4955 very fine grid of support points to evaluate the integral to a reasonable level of accuracy.
 4956 We demonstrate this here by reanalyzing simulated data using a state-space defined by a
 4957 different number of support points. We provide an **R** script called `SCR0bayesDss` in the
 4958 **R** package `scrbook`. We note that for this comparison we generated the actual activity
 4959 centers as a continuous random variable and thus the discrete state-space is, strictly
 4960 speaking, an approximation to truth. That said, we regard all state-space specifications
 4961 as approximations to truth in the sense that they represent a component of the SCR
 4962 model.

4963 As with our **R** function `SCR0bayes`, the modification `SCR0bayesDss` will use either
 4964 **WinBUGS** or **JAGS**. In addition, it requires a grid resolution argument (`ng`) which
 4965 is the dimension of 1 side of a square state-space. To execute this function we do, for
 4966 example:

```
4967 > library(scrbook)
4968 > data <- simSCR0(discard0=TRUE,rnd=2013)    # Generate data set
4969
4970 # Run with JAGS
4971 > out1 <- SCR0bayesDss(data,ng=8,M=200,engine="jags",ni=2000,nb=1000)
4972
4973 # Run with WinBUGS
4974 > out2 <- SCR0bayesDss(data,ng=8,M=200,engine="winbugs",ni=2000,nb=1000)
```

4975 We fit this model to the same simulated data set for 6×6 , 9×9 , 12×12 , 15×15
 4976 state-space grids. For **WinBUGS**, we used 3 chains of 5000 total length with 1000 burn-
 4977 in, which yields 12000 total posterior samples. Summary results are shown in Table ??.
 4978 The results are broadly consistent except for the 6×6 case. We see that the run time
 4979 increases with the size of the state-space grid (not unexpected), such that we imagine it
 4980 would be impractical to run models with more than a few hundred state-space grid points.
 4981 We found (not shown here) that the runtime of **JAGS** is much faster and, furthermore,
 4982 relatively *constant* as we increase the grid size. We suspect that **WinBUGS** is evaluating
 4983 the full-conditional for each activity center at all G possible values whereas it may be
 4984 that **JAGS** is evaluating the full-conditional only at a subset of values or perhaps using
 4985 previous calculations more effectively. While this might suggest that one should always
 4986 use **JAGS** for this analysis, we found in our analysis of the wolverine (next section) that
 4987 **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

4988 often simply do not work for some problems, so be careful when using **JAGS**. To improve
 4989 its performance, always start the latent activity centers at values near where individuals
 4990 were captured. The performance of either should improve if we compute the full distance
 4991 matrix outside of **BUGS** and pass it as data, although we haven't fully evaluated this
 4992 approach.

4993 5.10.2 Analysis of the wolverine camera trapping data

4994 We reanalyzed the wolverine data using discrete state-space grids with points spaced by
 4995 2, 4 and 8 km (see Fig. ??). These were constructed from a 40 km buffered state-space,
 4996 and deleting the points over water (see ?). Our interest in doing this was to evaluate the
 4997 relative influence of grid resolution on estimated density because the coarser grids will
 4998 be more efficient from a computational stand-point and so we would prefer to use them,
 4999 but only if there is no strong influence on estimated density. The posterior summaries
 5000 for the 3 habitat grids are given in Table ???. We see that the density estimates are quite
 5001 a bit larger than obtained in our analysis (Table ??) based on a rectangular, continuous
 5002 state-space. We also see that there are slight differences depending on the resolution of the
 5003 state-space grid. Interestingly, the effectiveness of the MCMC algorithms, as measured by
 5004 effective sample size (**n.eff**) is pretty remarkably different. Furthermore, the finest grid
 5005 resolution (2 km spacing) took about 6 days to run and thus it would not be practical for
 5006 large problems or with many models.

5.11 SUMMARIZING DENSITY AND ACTIVITY CENTER LOCATIONS

5007 One of the most useful aspects of SCR models is that they are parameterized in terms of
 5008 individual locations – i.e., *where* each individual lives – and, thus, we can compute many
 5009 useful and interesting summaries of the activity centers using output from an MCMC sim-
 5010 ulation, including maps of density (the number of activity centers per unit area), estimates
 5011 of N for any well-defined polygon, or estimates of where the activity centers for specific
 5012 individuals reside. In Bayesian analysis by MCMC, obtaining such summaries entails no
 5013 added calculations, because we need only post-process the output for the individual ac-
 5014 tivity centers to obtain the desired summaries. We demonstrate that in this section. Note
 5015 that you have to be sure to retain the MCMC history for the **s** variables and also the data
 5016 augmentation variables z in order to do the following analyses.

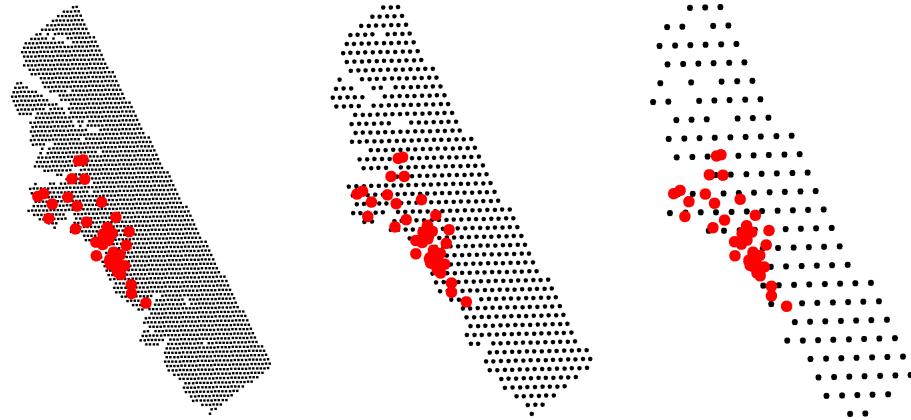


Figure 5.5. Three habitat mask grids used in the comparison of the effect of pixel size on the estimated density surface of wolverines. The 3 cases are 2 (left) (center) and 8 (right) km spacing of state-space points, extending 40 km from the vicinity of the trap array.

5.11.1 Constructing density maps

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

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

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

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

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

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

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

5032 Thus, $N(\mathbf{x}, 1), N(\mathbf{x}, 2), \dots$, is the Markov chain for parameter $N(\mathbf{x})$. In what follows we
 5033 will provide a set of **R** commands for doing this calculation and making a basic image
 5034 plot from the MCMC output.

5035 **Step 1:** Define the center points of each pixel $B(\mathbf{x})$, or point at which local density will
 5036 be estimated:

```
5037 > xg <- seq(xlim[1], xlim[2], , 50)
  5038 > yg <- seq(ylim[1], ylim[2], , 50)
```

5039 **Step 2:** Extract the MCMC histories for the activity centers and the data augmentation
 5040 variables. Note that these are each $N \times \text{niter}$ matrices. Here we do this assuming that
 5041 **WinBUGS** was run producing the **R** object named **out**:

```
5042 > Sxout <- out$sims.list$s[, , 1]
  5043 > Syout <- out$sims.list$s[, , 2]
  5044 > z <- out$sims.list$z
```

5045 **Step 3:** We associate each coordinate with the proper pixel using the **R** command `cut()`.
 5046 Note that we keep only the activity centers for which $z = 1$ (i.e., individuals that belong
 5047 to the population of size N):

```
5048 > Sxout <- cut(Sxout[z==1], breaks=xg, include.lowest=TRUE)
5049 > Syout <- cut(Syout[z==1], breaks=yg, include.lowest=TRUE)
```

5050 **Step 4:** Use the `table()` command to tally up how many activity centers are in each
 5051 $B(\mathbf{x})$:

```
5052 > Dn <- table(Sxout, Syout)
```

5053 **Step 5:** Use the `image()` command to display the resulting matrix.

```
5054 > image(xg, yg, Dn/nrow(z), col=topo.colors(10))
```

5055 It is worth emphasizing here that density maps will not usually appear uniform despite
 5056 that we have assumed that activity centers are uniformly distributed. This is because
 5057 the observed encounters of individuals provide direct information about the location of
 5058 the $i = 1, 2, \dots, n$ activity centers and thus their “estimated” locations will be affected
 5059 by the observations. In a limiting sense, were we to sample space intensely enough,
 5060 every individual would be captured a number of times and we would have considerable
 5061 information about all N point locations. Consequently, the uniform prior would have
 5062 almost no influence at all on the estimated density surface in this limiting situation.
 5063 Thus, in practice, the influence of the uniformity assumption decreases as the fraction of
 5064 the population encountered, and the total number of encounters per individual, increases.

5065 **On the non-intuitiveness of `image()`** – the **R** function `image()`, invoked for a
 5066 matrix M by `image(M)`, might not be very intuitive to some – it plots $M[1, 1]$ in the lower
 5067 left corner. If you want $M[]$ to be plotted “as you look at it” then $M[1, 1]$ should be in the
 5068 upper left corner. We have a function `rot()` which does that. If you do `image(rot(M))`
 5069 then it puts it on the monitor as if it was a map you were looking at. You can always
 5070 specify the x - and y -labels explicitly as we did above.

5071 **Spatial dot plots** – A cruder version of the density map can be made using our
 5072 “spatial dot map” function `spatial.plot` (in `scrbook`). This function requires, as input,
 5073 point locations and the value to be displayed. A simplified version of this function is as
 5074 follows:

```
5075 > spatial.plot <- function(x,y){
5076   nc <- as.numeric(cut(y,20))
5077   plot(x,pch=" ")
5078   points(x,pch=20,col=topo.colors(20)[nc],cex=2)
5079   image.scale(y,col=topo.colors(20))
5080 }
5081 #
5082 # To execute the function do this:
5083 #
5084 > spatial.plot(cbind(xg,yg), Dn/nrow(z))
```

5085 **5.11.2 Example: Wolverine density map**

5086 We return to the wolverine study which took place in 2008 in SE Alaska (Fig. ??) and
 5087 we produce a density map of wolverines from that analysis. We include the function
 5088 **SCRdensity** which requires a specific data structure as shown below. In particular, we
 5089 have to package up the MCMC history for the activity centers and the data augmentation
 5090 variables z into a list. This also requires that we add those variables to the parameters-
 5091 to-be-monitored list when we pass things to **BUGS**.

5092 We used the posterior output from the wolverine model fitted previously to compute
 5093 a relatively coarse version of a density map, using 100 pixels in a 10×10 grid (Fig. ??
 5094 top panel) and using 900 pixels arranged in a 30×30 grid (Fig. ?? lower panel) for a
 5095 fine-scale map. The **R** commands for producing such a plot (for a short MCMC run) are
 5096 as follows:

```
5097 > library(scrbook)
5098 > data(wolverine)
5099 > traps <- wolverine$wtraps
5100 > y3d <- SCR23darray(wolverine$wcaps,wolverine$wtraps)
5101
5102 # This takes 341 seconds on a standard CPU circa 2011
5103 > out <- wolvSCRO(y3d,traps,nb=1000,ni=2000,buffer=1,M=100,keepz=TRUE)
5104
5105 > Sx <- out$sims.list$s[,,1]
5106 > Sy <- out$sims.list$s[,,2]
5107 > z <- out$sims.list$z
5108 > obj <- list(Sx=Sx,Sy=Sy,z=z)
5109 > tmp <- SCRdensity(obj,nx=10,ny=10,scalein=100,scaleout=100)
```

5110 In these figures density is expressed in units of individuals per 100 km^2 , while the area of
 5111 the pixels is about 103.7 km^2 and 11.5 km^2 , respectively. That calculation is based on:

```
5112 > total.area <- (ylim[2]-ylim[1])*(xlim[2]-xlim[1])*100
5113 > total.area/(10*10)
5114 [1] 103.7427
5115 > total.area/(30*30)
5116 [1] 11.52697
```

5117 A couple of things are worth noting: First is that as we move away from “where the
 5118 data live” – away from the trap array – we see that the density approaches the mean
 5119 density. This is a property of the estimator as long as the detection function decreases
 5120 sufficiently rapidly as a function of distance. Relatedly, it is also a property of statistical
 5121 smoothers such as splines, kernel smoothers, and regression smoothers – predictions tend
 5122 toward the global mean as the influence of data diminishes. Another way to think of it is
 5123 that it is a consequence of the prior, which imposes uniformity, and as you get far away
 5124 from the data, the predictions tend to the expected constant density under the prior.
 5125 Another thing to note about this map is that density is not 0 over water (although the
 5126 coastline is not shown). This might be perplexing to some who are fairly certain that
 5127 wolverines do not like water. However, there is nothing about the model that recognizes

5128 water from non-water and so the model predicts over water *as if* it were habitat similar to
 5129 that within which the array is nested. But, all of this is OK as far as estimating density
 5130 goes and, furthermore, we can compute valid estimates of N over any well-defined region
 5131 which presumably wouldn't include water if we so wished. Alternatively, areas covered by
 5132 water could be masked out, which we discuss in the next section.

5133 5.11.3 Predicting where an individual lives

5134 The density maps in the previous section show the expected number of individuals per
 5135 unit area. A closely related problem is that of producing a map of the probable location
 5136 of a specific individual's activity center. For any observed encounter history, we can easily
 5137 generate a posterior distribution of \mathbf{s}_i for individual i . In addition, for an individual that
 5138 is *not* captured, we can use the MCMC output to produce a corresponding plot of where
 5139 such an individual might live, say \mathbf{s}_{n+1} . Obviously, all such uncaptured individuals (for
 5140 $i = n + 1, \dots, N$) should have the same posterior distribution. To illustrate, we show the
 5141 posterior distribution of \mathbf{s}_1 , the activity center for the individual labeled 1 in the data
 5142 set, in Fig. ???. This individual was captured a single time at trap 30 which is circled
 5143 in Fig. ???. We see that the posterior distribution is affected by traps of capture *and*
 5144 traps of non-capture in fairly intuitive ways. In particular, because there are other traps
 5145 in close proximity to trap 30, in which individual 1 was *not* captured, the model pushes
 5146 its activity center away from the trap array. The help file for **SCRdensity** shows how to
 5147 calculate Fig. ???.

5.12 EFFECTIVE SAMPLE AREA

5148 One of the key issues in using ordinary capture recapture models which we've brought up
 5149 over and over again is this issue that the area which is sampled by a trapping array is
 5150 unknown – in other words, the N that is estimated by capture-recapture models does not
 5151 have an explicit region of space associated with it. Classically this has been addressed in
 5152 the ad hoc way of prescribing an area that contains the trap array, usually by adding a
 5153 buffer of some width, which is not estimated as part of the capture-recapture model. In
 5154 SCR models we avoid the problem of not having an explicit linkage between N and “area”,
 5155 by prescribing explicitly the area within which the underlying point process is defined – the
 5156 state-space of the point process. This state-space is *not* the effective sample (or sampled)
 5157 area (ESA) – it is desirable that it be somewhat larger than the ESA, whatever that may
 5158 be, in the sense that individuals at the edge of the state-space have no probability of being
 5159 captured, but as part of the SCR model we don't need to try to estimate or otherwise
 5160 characterize the ESA explicitly.

5161 However, it is possible to provide a characterization of effective sampled area under
 5162 any SCR model. This is directly analogous to the calculation of “effective strip width” in
 5163 distance sampling (??). The conceptual definition of ESA follows from equating density
 5164 to “apparent density” – ESA is the magic number that satisfies that equivalence:

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

5165 In other words, the ratio of N to the area of the state-space should be equal to the ratio
 5166 of the observed sample size n to this number ESA. Both of these should equal density.

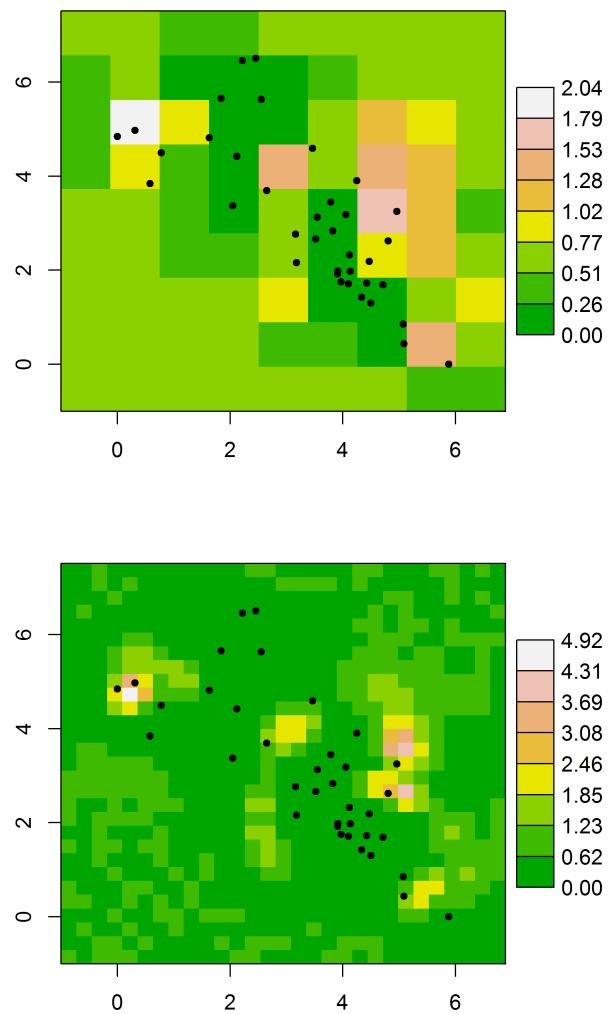


Figure 5.6. Density of wolverines (individuals per 100 km²) in SE Alaska in 2007 based on model SCR0. Map grid cells are about 103.7 km² (top panel) and 11.5 km² (bottom panel) in area. Dots are the trap locations.

5167 So, to compute ESA for a model, we substitute $\mathbb{E}(n)$ for n into the above equation, and
 5168 solve for ESA , to get:

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

5169 Our following development assumes that D is constant, but these calculations can be
 5170 generalized to allow for D to vary spatially. Imagine our habitat mask for the wolverine
 5171 data, or the bins we just used to produce a density map, then we can write $\mathbb{E}(n)$ according
 5172 to

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

5173 where if we prefer to think of this more conceptually we could replace the summation with
 5174 an integration (which, in practice, we would just replace with a summation, and so we
 5175 just begin there). In this expression note that $\mathbb{E}(N(\mathbf{s}))$ is the expected population size at
 5176 pixel \mathbf{s} which is the density times the area of the pixel, i.e., $\mathbb{E}(N(\mathbf{s})) = D \times a$. Therefore

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

5177 and (plugging this into the expression above for ESA)

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

5178 We see that D cancels and we have $ESA = a \times \sum_s \Pr(\text{encounter}|\mathbf{s})$ So what you have to
 5179 do here is substitute in $\Pr(\text{encounter}|\mathbf{s})$ and just sum them up over all pixels. For the
 5180 Bernoulli model of model SCR0

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

5181 with slight modifications when encounter probability depends on covariates. Thus,

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

5182 Clearly the calculation of ESA is affected by the use of a habitat mask, because the
 5183 summation in Eq. ?? only occurs over pixels that define the state-space.

5184 For the wolverine camera trapping data, we used the 2×2 km habitat mask and the
 5185 posterior means of p_0 and σ (see Sec. ??) to compute the probability of encounter for
 5186 each \mathbf{s} of the mask points. The result is shown graphically in Fig. ???. The ESA is the
 5187 sum of the values plotted in that figure multiplied by 4, the area of each pixel. For the
 5188 wolverine study, the result is 2507.152 km 2 . We note that the probability of encounter
 5189 declines rapidly to 0 as we move away from the periphery of the camera traps, indicating
 5190 the state-space constructed from a 40 km buffered trap array was indeed sufficient for the
 5191 analysis of these data. An **R** script for producing this figure is in the **wolvESA** function of
 5192 the **scrbook** package.

5.13 SUMMARY AND OUTLOOK

5193 In this chapter, we introduced the simplest SCR model – “model SCR0” – which is an ordi-
 5194 nary capture-recapture model like model M_0 , but augmented with a set of latent individual

5195 effects, s_i , which relate encounter probability to some sense of individual location using a
5196 covariate, “distance”, from s_i to each trap location. Thus, individuals in close proximity
5197 to a trap will have a higher probability of encounter, and *vice versa*. The explicit modeling
5198 of individual locations and distance in this fashion resolves classical problems related to
5199 estimating density: unknown sample area, and heterogeneous encounter probability due
5200 to variable exposure to traps.

5201 SCR models are closely related to classical individual covariate models (“model M_x ”,
5202 as introduced in Chapt. ??), but with imperfect information about the individual covari-
5203 ate. Therefore, they are also not too dissimilar from standard GLMMs used throughout
5204 statistics and, as a result, we find that they are easy to analyze using standard MCMC
5205 methods encased in black boxes such as **WinBUGS** or **JAGS**. We will also see that they
5206 are easy to analyze using likelihood methods, which we address in Chapt. ??.

5207 Formal consideration of the collection of individual locations (s_1, \dots, s_N) is funda-
5208 mental to all models considered in this book. In statistical terminology, we think of the
5209 collection of points $\{s_i\}$ as a realization of a point process. Because SCR models formally
5210 link individual encounter history data to an underlying point process, we can obtain for-
5211 mal inferences about the point process. For example, we showed how to produce a density
5212 map (Fig. ??), or even a probability map for an individual’s home range center (Fig.
5213 ??). We can also use SCR models as the basis for doing more traditional point process
5214 analyses, such as testing for “complete spatial randomness” (CSR) (see Chapt. ??), and
5215 computing other point process summaries (?).

5216 Part of the promise, and ongoing challenge, of SCR models is to develop models that
5217 reflect interesting biological processes, for example interactions among points or temporal
5218 dynamics in point locations. In this chapter we considered the simplest possible point
5219 process model in which points are independent and uniformly (“randomly”) distributed
5220 over space. Despite the simplicity of this model, it should suffice in many applications of
5221 SCR models, although we do address generalizations in later chapters. Moreover, even
5222 though the *prior* distribution on the point locations is uniform, the realized pattern may
5223 deviate markedly from uniformity as the observed encounter data provide information to
5224 impart deviations from uniformity. Thus, estimated density maps will typically appear
5225 distinctly non-uniform (as we saw in the wolverine example). In applications of the basic
5226 SCR model, we find that this simple *a priori* model can effectively reflect or adapt to
5227 complex realizations of the underlying point process. For example, if individuals are
5228 highly territorial then the data should indicate this in the form of individuals not being
5229 encountered in the same trap – the resulting posterior distribution of point locations should
5230 therefore reflect non-independence. Obviously the complexity of posterior estimates of the
5231 point pattern will depend on the quantity of data, both number of individuals and captures
5232 per individual. Because the point process is such an integral component of SCR models,
5233 the state-space of the point process plays an important role in developing SCR models.
5234 As we emphasized in this chapter, the state-space is part of the model. It can have an
5235 influence on parameter estimates and other inferences, such as model selection (see chapter
5236 ??).

5237 One concept we introduced in this chapter, which has not been discussed much in
5238 the literature on SCR models, is the manner in which the encounter probability model
5239 relates to a model of space usage by individuals. The standard SCR models of encounter
5240 probability can all be motivated as simplistic models of space usage and movement, in

5241 which individuals make random use decisions from a probability distribution proportional
5242 to the encounter probability model. This both clarifies the simplicity of the underlying
5243 model of space usage and also suggests a direct extension to produce more realistic models,
5244 which we discuss in Chapt. ???. We consider some other important extensions of the basic
5245 SCR model in later chapters. For example, we consider models that include covariates
5246 that vary by individual, trap, or over time (Chapt. ??), spatial covariates on density
5247 (Chapt. ??), open populations (Chapt. ??), and methods for model assessment and
5248 selection (Chapt. ??) among other topics. We also consider technical details of maximum
5249 likelihood (Chapt. ??) and Bayesian (Chapt. ??) estimation, so that the interested reader
5250 can develop or extend methods to suit their own needs.

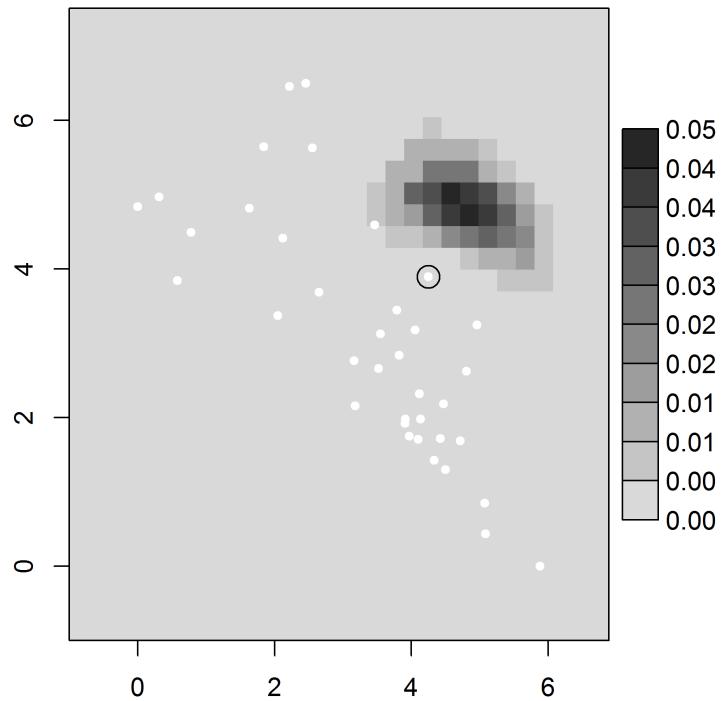


Figure 5.7. Posterior probability distribution of s_1 , the activity center for individual 1 in the wolverine data set. This individual was captured a single time in one trap (trap 30) which is circled. White dots are trap locations.

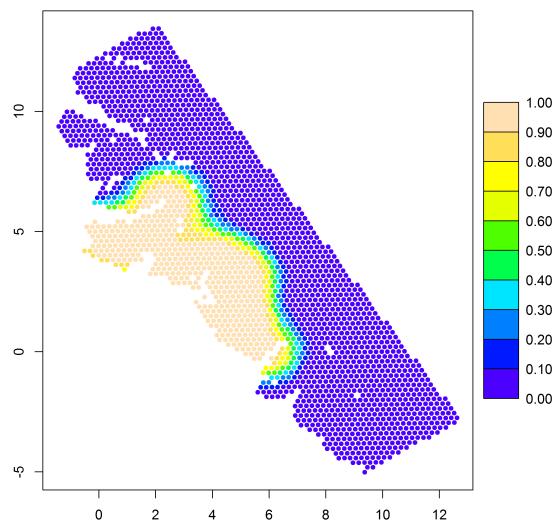


Figure 5.8. Probability of encounter used in computing effective sampled area for the wolverine camera trapping array, using the parameter estimates (posterior means) for the 2×2 km habitat mask.

5251

6

5252

5253

5254

LIKELIHOOD ANALYSIS OF SPATIAL CAPTURE-RECAPTURE MODELS

5255 We have so far mainly focused on Bayesian analysis of spatial capture-recapture models.
5256 And, in the previous chapters we learned how to fit some basic spatial capture-recapture
5257 models using a Bayesian formulation of the models analyzed in **BUGS** engines including
5258 **WinBUGS** and **JAGS**. Despite our focus on Bayesian analysis, it is instructive to de-
5259 velop the basic concepts and ideas behind classical analysis based on likelihood methods
5260 and frequentist inference for SCR models. We recognized earlier (Chapt. ??) that SCR
5261 models are versions of binomial (or other) GLMs, but with random effects (i.e., GLMMs).
5262 Throughout statistics, such models are routinely analyzed by likelihood methods. In par-
5263 ticular, likelihood analysis is based on the integrated or marginal likelihood in which the
5264 random effects are removed, by integration, from the conditional-on-s likelihood (s being
5265 the individual activity center). This has been the approach taken by ?? and related
5266 papers. Therefore, in this chapter, we provide some conceptual and technical foundation
5267 for likelihood-based analysis of spatial capture-recapture models.

5268 We will show here that it is straightforward to compute the maximum likelihood esti-
5269 mates (MLE) for SCR models by integrated likelihood. We develop the MLE framework
5270 using **R**, and we also provide a basic introduction to the **R** package **secr** (?) which does
5271 likelihood analysis of SCR models (see also the stand-alone program **DENSITY** (?)).
5272 To set the context for likelihood analysis of SCR models, we first analyze the SCR model
5273 when N is known because, in that case, analysis is no different at all than a standard
5274 GLMM. We generalize the model to allow for unknown N using both conventional ideas
5275 based on the “full likelihood” (e.g., ?) and also using a formulation based on data aug-
5276mentation. We obtain the MLEs for the SCR model from the wolverine camera trapping
5277 study (?) analyzed in previous chapters to compare/contrast the results.

6.1 MLE WITH KNOWN N

We noted in Chapt. ?? 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. ??).

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. ?? 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:

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

5307 We emphasize that two independence assumptions are explicit in this development: independence of trap-specific encounters within individuals and also independence among
 5308 individuals. In particular, this would only be valid when individuals are not physically
 5309 restrained or removed upon capture, and when traps do not “fill up.”

5310 The key operation for computing the likelihood is solving a 2-dimensional integration
 5311 problem. There are some general purpose **R** packages that implement a number of multi-
 5312 dimensional integration routines including `adapt` (?) and `R2cuba` (?). In practice, we
 5313 won’t rely on these extraneous **R** packages (except see Chapt. ?? for an application of
 5314 `R2cuba`) but instead will use perhaps less efficient methods in which we replace the integral
 5315 with a summation over an equal area mesh of points on the state-space \mathcal{S} and explicitly
 5316 evaluate the integrand at each point. We invoke the rectangular rule for integration here¹
 5317 in which we evaluate the integrand on a regular grid of points of equal area and compute
 5318 the average of the integrand over that grid of points. Let $u = 1, 2, \dots, nG$ index a grid of
 5319 nG points, \mathbf{s}_u , where the area of grid cells is constant, say A . In this case, the integrand,
 5320 i.e., the marginal pmf of \mathbf{y}_i , is approximated by

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

5322 This is a specific case of the general expression that could be used for approximating
 5323 the integral for any arbitrary distribution $[\mathbf{s}]$. The general case is

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

5324 Under the uniformity assumption, $[\mathbf{s}] = 1/A(\mathcal{S})$ and thus the grid-cell area cancels in the
 5325 above expression to yield Eq. ???. The rectangular rule for integration can be seen as an
 5326 application of the Law of Total Probability for a discrete random variable \mathbf{s} , having nG
 5327 unique values with equal probabilities $1/nG$.

5328 6.1.1 Implementation (simulated data)

5329 Here we will illustrate how to carry out this integration and optimization based on the
 5330 integrated likelihood using simulated data (i.e., see Sec. ??). Using `simSCR0` we simulate
 5331 data for 100 individuals and an array of 25 traps laid out in a 5×5 grid of traps having unit
 5332 spacing. The specific encounter model is the Gaussian model. The 100 activity centers
 5333 were simulated on a state-space defined by an 8×8 square within which the trap array was
 5334 centered (thus the trap array is buffered by 2 units). Therefore, the density of individuals
 5335 in this system is fixed at $100/64$. In the following set of **R** commands we generate the
 5336 data and then harvest the required data objects:

```
5337 ## simulate a complete data set (perfect detection)
5338 > data <- simSCR0(discard0=FALSE, rnd=2013)
5339   ## extract the objects that we need for analysis
5340 > y <- data$Y
```

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

```

5341 > traplocs <- data$traplocs
5342 > nind <- nrow(y) ## in this case nind=N
5343 > J <- nrow(traplocs)
5344 > K <- data$K
5345 > xlim <- data$xlim
5346 > ylim <- data$ylim

```

5347 Now, we need to define the integration grid, say **G**, which we do with the following set of
 5348 **R** commands (here, **delta** is the grid spacing):

```

5349 > delta <- .2
5350 > xg <- seq(xlim[1]+delta/2,xlim[2]-delta/2,by=delta)
5351 > yg <- seq(ylim[1]+delta/2,ylim[2]-delta/2,by=delta)
5352 > npix <- length(xg)           # valid for square state-space only
5353 > G <- cbind(rep(xg,npix),sort(rep(yg,npix)))
5354 > nG <- nrow(G)

```

5355 In this case, the integration grid is set up as a grid with spacing $\delta = 0.2$ which produces,
 5356 for our example, a 40×40 grid of points for evaluating the integrand if the state-space
 5357 buffer is set at 2. We note that the integration grid is set-up here to correspond exactly
 5358 to the state-space used in simulating the data. However, in practice, we wouldn't know
 5359 this, and our estimate of N (for the unknown case, see below) would be sensitive to choice
 5360 of the extent of the integration grid. As we've discussed previously, density, which is N
 5361 standardized by the area of the state-space, will not be so sensitive in most cases.

5362 We are now ready to compute the conditional-on-s likelihood and carry out the
 5363 marginalization described by Eq. ???. We need to do this by defining an **R** function
 5364 that computes the likelihood for the integration grid, as a function of the data objects
 5365 **y** and **traplocs** which were created above. However, it is a bit untidy to store the grid
 5366 information in your workspace, and define the likelihood function in a way that depends
 5367 on these things that exist in your workspace. Therefore, we build the **R** function so that
 5368 it computes the integration grid *within* the function, thereby avoiding potential problems
 5369 if our trapping grid locations change, or if we want to modify the state-space buffer easily.
 5370 We therefore define the function, called **intlik1**, to which we pass the data objects and
 5371 other information necessary to compute the marginal likelihood. This function is available
 5372 in the **scrbook** package (use **?intlik1** at the **R** prompt). The code is reproduced here:

```

5373 intlik1 <- function(parm,y=y,X=traplocs, delta=.2, ssbuffer=2){
5374
5375   Xl <- min(X[,1]) - ssbuffer ## These lines of code are setting up the
5376   Xu <- max(X[,1]) + ssbuffer ## support for the integration which is
5377   Yu <- max(X[,2]) + ssbuffer ## the same as the state-space of "s"
5378   Yl <- min(X[,2]) - ssbuffer
5379   xg <- seq(Xl+delta/2,Xu-delta/2,,length=npix)
5380   yg <- seq(Yl+delta/2,Yu-delta/2,,length=npix)
5381   npix<- length(xg)
5382
5383   G <- cbind(rep(xg,npix),sort(rep(yg,npix)))

```

```

5384   nG <- nrow(G)
5385   D <- e2dist(X,G)
5386
5387   alpha0 <- parm[1]
5388   alpha1 <- exp(parm[2]) # alpha1 restricted to be positive here
5389
5390   probcap <- plogis(alpha0)*exp(-alpha1*D*D)
5391   Pm <- matrix(NA,nrow=nrow(probcap),ncol=ncol(probcap))
5392           # Frequency of all-zero encounter histories
5393   n0 <- sum(apply(y,1,sum)==0)
5394           # Encounter histories with at least 1 detection
5395   ymat <- y[apply(y,1,sum)>0,]
5396   ymat <- rbind(ymat,rep(0,ncol(ymat)))
5397   lik.marg <- rep(NA,nrow(ymat))
5398
5399   for(i in 1:nrow(ymat)){
5400     ## Next line: log conditional likelihood for ALL possible values of s
5401     Pm[1:length(Pm)] <- dbinom(rep(ymat[i,],nG),K,probcap[1:length(Pm)],
5402                                     log=TRUE)
5403     ## Next line: sum the log conditional likelihoods, exp() result
5404     ## same as taking the product
5405     lik.cond <- exp(colSums(Pm))
5406     ## Take the average value == computing marginal
5407     lik.marg[i] <- sum(lik.cond*(1/nG))
5408   }
5409   ## n0 = number of all-0 encounter histories
5410   nv <- c(rep(1,length(lik.marg)-1),n0)
5411   return( -1*(sum(nv*log(lik.marg)) ) )
5412 }

```

5413 We emphasize that this function (and subsequent) are not meant to be general-purpose
 5414 routines for solving all of your SCR problems but, rather, they are meant for illustrative
 5415 purposes – so you can see how the integrated likelihood is constructed and how we connect
 5416 it to data and other information that is needed.

5417 The function `intlik1` accepts as input the encounter history matrix, `y`, the trap locations,
 5418 `X`, and the state-space buffer. This allows us to vary the state-space buffer and easily
 5419 evaluate the sensitivity of the MLE to the size of the state-space. Note that we have a
 5420 peculiar handling of the encounter history matrix `y`. In particular, we remove the all-zero
 5421 encounter histories from the matrix and tack-on a single all-zero encounter history as the
 5422 last row which then gets weighted by the number of such encounter histories (`n0`). This is
 5423 a bit long-winded and strictly unnecessary when N is known, but we did it this way be-
 5424 cause the extension to the unknown- N case is now transparent (as we demonstrate in the
 5425 following section). The matrix `Pm` holds the log-likelihood contributions of each encounter
 5426 frequency for each possible state-space location of the individual. The log contribu-
 5427 tions are summed up and the result exponentiated on the next line, producing `lik.cond`, the
 5428 conditional-on-s likelihood (Eq. ?? above). The marginal likelihood (`lik.marg`) sums up
 5429 the conditional elements weighted by the probabilities [s] (Eq. ?? above).

5430 This is a fairly primitive function which doesn't allow much flexibility in the data
 5431 structure. For example, it assumes that K , the number of replicates, is constant for each
 5432 trap. Further, it assumes that the state-space is a square. We generalize this to some
 5433 extent later in this chapter.

5434 Here is the **R** command for maximizing the likelihood using **nlm** (the function **optim**
 5435 could also be used) and saving the results into an object called **frog**. The output is a list
 5436 of the following structure and these specific estimates are produced using the simulated
 5437 data set:

```
5438 # should take 15-30 seconds
5439
5440 > starts <- c(-2,2)
5441 > frog <- nlm(intlik1,starts,y=y,X=traplocs,delta=.1,ssbuffer=2,hessian=TRUE)
5442 > frog
5443
5444 $minimum
5445 [1] 297.1896
5446
5447 $estimate
5448 [1] -2.504824 2.373343
5449
5450 $gradient
5451 [1] -2.069654e-05 1.968754e-05
5452
5453 $hessian
5454 [,1]      [,2]
5455 [1,] 48.67898 -19.25750
5456 [2,] -19.25750 13.34114
5457
5458 $code
5459 [1] 1
5460
5461 $iterations
5462 [1] 11
```

5463 Details about this output can be found on the help page for **nlm**. We note briefly that
 5464 **frog\$minimum** is the negative log-likelihood value at the MLEs, which are stored in the
 5465 **frog\$estimate** component of the list. The order of the parameters is as they are defined
 5466 in the likelihood function so, in this case, the first element (value = -2.504824) is the
 5467 logit transform of p_0 and the second element (value = 2.373343) is the value of α_1 the
 5468 “coefficient” on distance-squared. The Hessian is the observed Fisher information matrix,
 5469 which can be inverted to obtain the variance-covariance matrix using the command:

```
5470 > solve(frog$hessian)
```

5471 It is worth drawing attention to the fact that the estimates are slightly different than
 5472 the Bayesian estimates reported previously in Sec. ???. There are several reasons for this.
 5473 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,

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

5511 (2) We compute the probability of an all-0 encounter history:

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

5512 (3) The combinatorial term: $\binom{N}{n}$. Then, the marginal likelihood has this form:

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

5513 This is discussed in ?, p. 379 as the conditional-on- N form of the likelihood – we also call
5514 it the “binomial form” of the likelihood because of its appearance.

5515 Operationally, things proceed much as before: We compute the marginal probability
5516 of each observed \mathbf{y}_i , i.e., by removing the latent \mathbf{s}_i by integration. In addition, we com-
5517 pute the marginal probability of the “all-zero” encounter history \mathbf{y}_{n+1} , and make sure to
5518 weight it n_0 times. We accomplish this by “padding” the data set with a single encounter
5519 history having $y_{n+1,j} = 0$ for all traps $j = 1, 2, \dots, J$. Then we be sure to include the
5520 combinatorial term in the likelihood or log-likelihood computation. We demonstrate this
5521 shortly. To analyze a specific case, we’ll simulate our fake data set (simulated using the
5522 parameters given above). To set some things up in our workspace we do this:

```
5523 ## Obtain a simulated data set
5524 > data <- simSCRO(discard0=TRUE, rnd=2013)
5525
5526 ## Extract the items we need for analysis
5527 > y <- data$Y
5528 > nind <- nrow(y)
5529 > traplocs <- data$traplocs
5530 > J <- nrow(traplocs)
5531 > K <- data$K
```

5532 Recall that these data are simulated by default with $N = 100$, on an 8×8 unit state-
5533 space representing the trap locations buffered by 2 units, although you can modify the
5534 simulation script easily.

5535 As before, the likelihood is defined in the **R** workspace as an **R** function, **intlik2**,
5536 which takes an argument being the unknown parameters of the model and additional
5537 arguments as prescribed. In particular, we provide the encounter history matrix **y**, the
5538 trap locations **traplocs**, the spacing of the integration grid (argument **delta**) and the
5539 state-space buffer. Here is the new likelihood function:

```
5540 intlik2 <- function(parm,y=y,X=traplocs,delta=.3,ssbuffer=2){
5541
5542   Xl <- min(X[,1]) - ssbuffer
5543   Xu <- max(X[,1]) + ssbuffer
5544   Yu <- max(X[,2]) + ssbuffer
5545   Yl <- min(X[,2]) - ssbuffer
5546
5547   xg <- seq(Xl+delta/2,Xu-delta/2,delta)
```

```

5548   yg <- seq(Yl+delta/2,Yu-delta/2,delta)
5549   npix.x <- length(xg)
5550   npix.y <- plength(yg)
5551   area <- (Xu-Xl)*(Yu-Yl)/((npix.x)*(npix.y))
5552   G <- cbind(rep(xg,npix.y),sort(rep(yg,npix.x)))
5553   nG <- nrow(G)
5554   D <- e2dist(X,G)
5555   # extract the parameters from the input vector
5556   alpha0 <- parm[1]
5557   alpha1 <- exp(parm[2])
5558   n0 <- exp(parm[3]) # note parm[3] lives on the real line
5559   probcap <- plogis(alpha0)*exp(-alpha1*D*D)
5560   Pm <- matrix(NA,nrow=nrow(probcap),ncol=ncol(probcap))
5561   ymat <- rbind(y,rep(0,ncol(y)))
5562
5563   lik.marg <- rep(NA,nrow(ymat))
5564   for(i in 1:nrow(ymat)){
5565     Pm[1:length(Pm)] <- (dbinom(rep(ymat[i,],nG),K,probcap[1:length(Pm)],
5566       log=TRUE))
5567     lik.cond <- exp(colSums(Pm))
5568     lik.marg[i] <- sum(lik.cond*(1/nG) )
5569   }
5570   nv <- c(rep(1,length(lik.marg)-1),n0)
5571   ## part1 here is the combinatorial term.
5572   ## math: log(factorial(N)) = lgamma(N+1)
5573   part1 <- lgamma(nrow(y)+n0+1) - lgamma(n0+1)
5574   part2 <- sum(nv*log(lik.marg))
5575   return( -1*(part1+ part2) )
5576 }

```

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

```

5577 > starts <- c(-2.5,0,4)
5578 > frog <- nlm(intlik2,starts,hessian=TRUE,y=y,X=traplocs,delta=.2,ssbuffer=2)
5579
5580 Warning message:
5581 In nlm(intlik2, starts, hessian = TRUE, y = y, X = traplocs, delta = 0.2, :
5582   NA/Inf replaced by maximum positive value
5583
5584 > frog
5585 $minimum
5586 [1] 113.5004
5587
5588 $estimate

```

```
5593 [1] -2.538333 0.902807 4.232810
5594
5595 [... additional output deleted ...]
```

5596 Executing `nls` here usually produces one or more **R** warnings due to numerical calculations
 5597 happening on extremely small or large numbers (calculation of p near the edge of the
 5598 state-space), and they also happen if a poor parameterization is used which produces
 5599 evaluations of the objective function beyond the boundary of the parameter space (e.g.,
 5600 $n_0 < 0$). Such numerical warnings can often be minimized or avoided altogether by picking
 5601 judicious starting values of parameters or properly transforming or scaling the parameters
 5602 but, in general, they can be ignored. You will see from the `nls` output that the algorithm
 5603 performed satisfactory in minimizing the objective function. The estimate of population
 5604 size, \hat{N} , for the state-space (using the default state-space buffer) is

```
5605 > Nhat <- nrow(y) + exp(4.2328) #### This is n + MLE of n0
5606 > Nhat
5607 [1] 110.9099
```

5608 Which differs from the data-generating value ($N = 100$), as we might expect for a single
 5609 realization. We usually will present an estimate of uncertainty associated with this MLE
 5610 which we can obtain by inverting the Hessian. Note that $\text{Var}(\hat{N}) = n + \text{Var}(\hat{n}_0)$. Since
 5611 we have parameterized the model in terms of $\log(n_0)$ we use the delta method² described
 5612 in ?, Appendix F4 (see also ?) to obtain the variance on the scale of n_0 as follows:

```
5613 > (exp(4.2328)^2)*solve(frog$hessian)[3,3]
5614 [1] 260.2033
5615
5616 > sqrt(260)
5617 [1] 16.12452
```

5618 Therefore, the asymptotic “Wald-type” confidence interval for N is $110.91 \pm 1.96 \times 16.125 =$
 5619 $(79.305, 142.515)$. To report this in terms of density, we scale appropriately by the area
 5620 of the prescribed state-space which is 64 units of area (i.e., an 8×8 square). Our MLE
 5621 of D is $\hat{D} = 110.91/64 = 1.733$ individuals per square unit. To get the standard error
 5622 for \hat{D} we need to divide the SE for \hat{N} by the area of the state-space, and so $\text{SE}(\hat{D}) =$
 5623 $(1/64) * 16.12452 = 0.252$.

5624 6.2.1 Integrated likelihood under data augmentation

5625 The likelihood analysis developed in the previous sections is based on the likelihood in
 5626 which N (or n_0) is an explicit parameter. This is usually called the “full likelihood” or
 5627 sometimes “unconditional likelihood” (?) because it is the likelihood for all individuals in
 5628 the population, not just those which have been captured, i.e., not that which is *conditional*
 5629 *on capture*. It is also possible to express an alternative unconditional likelihood using data
 5630 augmentation, replacing the parameter N with ψ (e.g., see Sec. 7.1.6 ?, for an example).

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>

5631 We don't go into detail here, but we note that the likelihood under data augmentation is a
 5632 zero-inflated binomial mixture – precisely an occupancy type model (?). Thus, while it is
 5633 possible to carry out likelihood analysis of models under data augmentation, we primarily
 5634 advocate data augmentation for Bayesian analysis.

5635 6.2.2 Extensions

5636 We have only considered basic SCR models with no additional covariates. However,
 5637 in practice, we are interested in covariate effects including “behavioral response”, sex-
 5638 specificity of parameters, and potentially others. Some of these can be added directly to
 5639 the likelihood if the covariate is fixed and known for all individuals captured or not. An
 5640 example is a behavioral response, which amounts to having a covariate $x_{ik} = 1$ if individual
 5641 i was captured prior to occasion k and $x_{ik} = 0$ otherwise. For uncaptured individuals,
 5642 $x_{ik} = 0$ for all k . ? called this a global behavioral response because the covariate is
 5643 defined for all traps, no matter the trap in which an individual was captured. We could
 5644 also define a *local* behavioral response which occurs at the level of the trap, i.e., $x_{ijk} = 1$ if
 5645 individual i was captured in trap j prior to occasion k , etc... Trap-specific covariates such
 5646 as trap type or status, or time-specific covariates such as date, are easily accommodated
 5647 as well. As an example, ? develop a model for the European wildcat *Felis silvestris* in
 5648 which traps are either baited or not (a trap-specific covariate with only 2 values), and also
 5649 encounter probability varies over time in the form of a quadratic seasonal response. We
 5650 consider models with behavioral response or fixed covariates in Chapt. ???. The integrated
 5651 likelihood routines we provided above can be modified directly for such cases, which we
 5652 leave to the interested reader to investigate.

5653 Sex-specificity is more difficult to deal with since sex is not known for uncaptured
 5654 individuals (and sometimes not even for all captured individuals). To analyze such models,
 5655 we do Bayesian analysis of the joint likelihood using data augmentation (??), discussed
 5656 further in Chapt. ???. For such covariates (i.e., that are not fixed and known for all
 5657 individuals), it is somewhat more challenging to do MLE based on the joint likelihood
 5658 as we have developed above. Instead it is more conventional to use what is colloquially
 5659 referred to as the “Huggins-Alho” type model which is one of the approaches taken in the
 5660 software package **secr** (?). We introduce the **secr** package in Sec. ?? below.

6.3 CLASSICAL MODEL SELECTION AND ASSESSMENT

5661 In most analyses, one is interested in choosing from among various potential models,
 5662 or ranking models, or something else to do with assessing the relative merits of a set
 5663 of models. A good thing about classical analysis based on likelihood is we can apply
 5664 Akaike Information Criterion (AIC) methods (?) without difficulty. AIC is convenient
 5665 for assessing the relative merits of these different models although if there are only a few
 5666 models it is not objectionable to use hypothesis tests or confidence intervals to determine
 5667 importance of effects. A second model selection context has to do with choosing among
 5668 various detection models, although, as a general rule, we don't recommend this application
 5669 of model selection. This is because there is hardly ever (if at all) a rational subject-matter
 5670 based reason motivating specific distance functions. As a result, we believe that doing
 5671 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. ??.

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, (? , 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. ?? . 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)`):

```

5699 > library(scrbook)
5700 > data(wolverine)
5701
5702 > traps <- wolverine$traps
5703 > traplocs <- traps[,2:3]/10000
5704 > K.wolv <- apply(traps[,4:ncol(traps)],1,sum)
5705
5706 > y3d <- SCR23darray(wolverine$wcaps,traps)
5707 > y2d <- apply(y3d,c(1,2),sum)
5708
5709 > starts <- c(-1.5,0,3)
5710
5711 > wolv <- nlm(intlik3,starts,hessian=TRUE,y=y2d,K=K.wolv,X=traplocs,
5712           delta=.2,ssbuffer=2)
5713

```

```

5714 > wolv
5715 $minimum
5716 [1] 220.4313
5717
5718 $estimate
5719 [1] -2.8176120 0.2269395 3.5836875
5720
5721 [.... output deleted ....]

```

5722 Of course we're interested in obtaining an estimate of population size for the prescribed
 5723 state-space, or density, and associated measures of uncertainty which we do using the delta
 5724 method (? , Appendix F4). To do all of that we need to manipulate the output of `nlm`
 5725 since we have our estimate in terms of $\log(n_0)$. We execute the following commands:

```

5726 > wolv <- nlm(intlik3,starts,hessian=TRUE,y=y2d,K=K.wolv,X=traplocs,delta=.2,
5727           ssbuffer=2)
5728 > Nhat <- nrow(y2d)+exp(wolv$estimate[3])
5729 > area <- attr(intlik3(starts,y=y2d,K=K.wolv,X=traplocs,delta=.2,ssbuffer=2),
5730           "SSarea")
5731 > Dhat <- Nhat/area
5732
5733 > Dhat
5734 [1] 0.5494947
5735
5736 > SE <- (1/area)*exp(wolv$estimate[3])*sqrt(solve(wolv$hessian)[3,3])
5737
5738 > SE
5739 [1] 0.1087073

```

5740 Our estimate of density is 0.55 individuals per “standardized unit” which is 100 km^2 ,
 5741 because we divided UTM coordinates by 10000. So this is about 5.5 individuals per 1000
 5742 km^2 , with a SE of around 1.09 individuals. This compares closely with 5.77 reported in
 5743 Sec. ?? based on Bayesian analysis of the model.

5744 6.4.1 Sensitivity to integration grid and state-space buffer

5745 The effect of approximating the integral by a discrete mesh of points is that it induces
 5746 some numerical error in evaluation of the integral and, further, that error increases as the
 5747 coarseness of the mesh increases. To evaluate the effect (or sensitivity) of the integration
 5748 grid spacing, we obtained the MLEs for a state-space buffer of 2 (standardized units) and
 5749 for integration grid with spacing $\delta = .3, .2, .1, .05$. The MLEs for these 4 cases including
 5750 the relative runtime are given in Table ???. We see the results change only slightly as the
 5751 integration grid changes. Conversely, the runtime on the platform of the day for the 4 cases
 5752 increases rapidly. These runtimes could be regarded in relative terms, across platforms,
 5753 for gaging the decrease in speed as the fineness of the integration grid increases.

5754 We studied the effect of the state-space buffer on the MLEs, using a fixed $\delta = .2$ for
 5755 all analyses. We used state-space buffers of 1 to 4 units stepped by .5. As we can see

Table 6.1. Runtime and MLEs for different integration grid resolutions for the wolverine camera trapping data.

δ	Estimates			
	runtime (s)	$\hat{\alpha}_0$	$\hat{\alpha}_1$	$\log(\hat{n}_0)$
0.30	9.9	-2.819786	1.258468	3.569731
0.20	32.3	-2.817610	1.254757	3.583690
0.10	115.1	-2.817570	1.255112	3.599040
0.05	407.3	-2.817559	1.255281	3.607158

5756 (Table ??), the estimates of D stabilize rapidly and the incremental difference is within
 5757 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

5758 6.4.2 Using a habitat mask (Restricted state-space)

5759 In Sec. ?? we used a discrete representation of the state-space in order to have control over
 5760 its extent and shape. This makes it easy to do things like clip out non-habitat, or create
 5761 a *habitat mask* which defines suitable habitat. Clearly that formulation of the model is
 5762 relevant to the calculation of the marginal likelihood in the sense that the discrete state-
 5763 space is equivalent to the integration grid. Thus, for example, we could easily compute
 5764 the MLE of parameters under some model with a restricted state-space merely by creating
 5765 the required state-space at whatever grid resolution is desired, and then inputting that
 5766 state-space into the likelihood function above, instead of computing it within the function.
 5767 We can easily create an explicit state-space grid for integration from arbitrary polygons or
 5768 GIS shapefiles which we demonstrate here. Our approach is to create the integration grid
 5769 (or state-space grid) outside of the likelihood evaluation, and then determine which points
 5770 of the grid lie in the polygon defined by the shapefile using functions in the **R** packages **sp**
 5771 and **maptools**. For each point in the state-space grid (object **G** in the code below which is
 5772 assumed to exist), we determine whether it is inside the polygon³, identifying such points

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

5773 with a value of `mask=1` and `mask=0` for points that are *not* in the polygon. We load the
 5774 shapefile which originates by an application of the `readShapeSpatial` function. We have
 5775 saved the result into an **R** data object called `SSp` which is in the `scrbook` package. Here
 5776 are the **R** commands for doing this (see the helpfile `?intlik4`):

```
5777 > library(mapproj)
5778 > library(sp)
5779 > library(scrbook)
5780
5781 ##### If we have the .shp file in place, we would use this command:
5782 ##### SSp <- readShapeSpatial('Sim_Polygon.shp')
5783 ##### The object SSp is in data(fakeshapefile)
5784 > data(fakeshapefile)
5785 > Pcoord <- SpatialPoints(G)
5786 > PinPoly <- over(Pcoord,SSp) #### determine if each point is in polygon
5787 > mask <- as.numeric(!is.na(PinPoly[,1])) ## convert to binary 0/1
5788 > G <- G[mask==1,]
```

5789 We created the function `intlik4` which accepts the integration grid as an explicit argument,
 5790 and this function is also available in the package `scrbook`.

5791 We apply this modification to the wolverine camera trapping study. `?intlik4` created 2, 4
 5792 and 8 km state-space grids so as to remove “non-habitat” (mostly ocean, bays, and large
 5793 lakes). We previously analyzed the model using **JAGS** and **WinBUGS** in Chapt. ???.
 5794 To set up the wolverine data and fit the model using maximum likelihood we execute the
 5795 following commands:

```
5796 > library(scrbook)
5797 > data(wolverine)
5798
5799 > traps <- wolverine$wtraps
5800 > traplocs <- traps[,2:3]/10000
5801 > K.wolv <- apply(traps[,4:ncol(traps)],1,sum)
5802
5803 > y3d <- SCR23darray(wolverine$wcaps,traps)
5804 > y2d <- apply(y3d,c(1,2),sum)
5805 > G <- wolverine$grid2/10000
5806
5807 > starts <- c(-1.5,0,3)
5808 > wolv <- nlm(intlik4, starts, y=y2d, K=K.wolv, X=traplocs, G=G)
5809
5810 > wolv
```

can hold additional information for each polygon, and the output value of the function differs slightly for these two classes: if using a “`SpatialPolygons`” object, the function returns a vector of length equal to the number of points (e.g., in the example above), but if using a “`SpatialPolygonsDataFrame`” it returns a data frame (e.g., see Sec. ?? in Chapt. ??). 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

```

5811
5812 $minimum
5813 [1] 225.8355
5814
5815 $estimate
5816 [1] -2.9955424 0.2350885 4.1104757
5817
5818 [... some output deleted ...]

```

5819 Next we convert the parameter estimates to estimates of total population size for the
 5820 prescribed state-space, and then obtain an estimate of density (per 1000 km²) using the
 5821 area computed as the number of pixels in the state-space grid, G, multiplied by the area
 5822 per grid cell. In the present case (the calculation above) we used a state-space grid with 2
 5823 km × 2 km pixels. Finally, we compute a standard errors using the delta approximation:

```

5824 > area <- nrow(G)*4
5825 # Nhat = n (observed) + MLE of n0 (not observed)
5826 > Nhat <- 21 + exp(wolv$estimate[3])
5827 > SE <- exp(wolv$estimate[3])*sqrt(solve(wolv$hessian)[3,3])
5828 > D <- (Nhat/(nrow(G)*area))*1000
5829 > SE.D <- (SE/(nrow(G)*area))*1000

```

5830 We did this for each the 2 km, 4 km and 8 km state-space grids which produced the
 5831 estimates summarized in Table ???. These estimates compare with the 8.6 (2 km grid) and
 5832 8.2 (8 km grid) reported in ? based on a clipped state-space as described in Sec. ??.

6.5 DENSITY AND THE R PACKAGE SECR

5833 **DENSITY** is a software program developed by ? for fitting spatial capture-recapture
 5834 models based mostly on classical maximum likelihood estimation and related inference
 5835 methods. ? has also released an **R** package called **secr**, that contains much of the
 5836 functionality of **DENSITY** but also incorporates new models and features. Here, we
 5837 briefly introduce the **secr** package which we prefer to use over **DENSITY**, because it
 5838 allows us to remain in the **R** environment for data processing and summarization. We
 5839 provide a brief introduction to **secr** and some of its capabilities here, and we also use it
 5840 for doing some analysis in other parts of this book. We believe that **secr** will be sufficient
 5841 for many (if not most) of the SCR problems that one might encounter. It provides a
 5842 flexible analysis platform, with a large number of summary features, and “publication

5843 ready” output. Its user-interface is clean and intuitive to **R** users, and it has been stable,
 5844 efficient and reliable in the (fairly extensive) evaluations that we have done.

5845 To install and run models in **secr**, you must download the package and load it in **R**.

```
5846 > install.packages("secr")
5847 > library(secr)
```

5848 **secr** allows the user to simulate data and fit a suite of models with various detection func-
 5849 tions and covariate responses. It also contains a number of helpful constructor functions
 5850 for creating objects of the proper class that are recognized by other **secr** functions. We
 5851 provide a brief overview of the capabilities here, but the **secr** help manual can be accessed
 5852 with the command:

```
5853 > RShowDoc("secr-manual", package = "secr")
```

5854 We note that **secr** has many capabilities that we will not cover or do so only sparingly.
 5855 We encourage you to read through the manual, the extensive documentation, and the
 5856 vignettes, in order to get a better understanding of what the package is capable of. We
 5857 also cover certain capabilities of **secr** in other chapters.

5858 The main model-fitting function in **secr** is called **secr.fit**, which makes use of the
 5859 standard **R** model specification framework with tildes. As an example, the equivalent of
 5860 the basic model SCR0 is fitted as follows:

```
5861 > secr.fit(capturedata, model = list(D ~ 1, g0 ~ 1, sigma ~ 1),
5862       buffer = 20000)
```

5863 where **capturedata** is the object created by **secr** containing the encounter history data
 5864 and the trap information, and the model expression $g0^1$ indicates the intercept-only (i.e.,
 5865 constant) model. Note that we use p_0 for the baseline encounter probability parameter,
 5866 which is g_0 in **secr** notation. A number of possible models for encounter probability can
 5867 be fitted including both pre-defined variables (e.g., **t** and **b** corresponding to “time” and
 5868 “behavior”), and user-defined covariates of several kinds. For example, to include a global
 5869 behavioral response, this would be written as $g0^1b$. The discussion of this (global versus
 5870 local trap-specific behavioral response) and other covariates is developed more in Chapt.
 5871 **??**. We can also model covariates on density in **secr**, which we discuss in Chapt. **??**. It
 5872 is important to note that **secr** requires the buffer distance to be defined in meters and
 5873 density will be returned as number of animals per hectare. Thus to make comparisons
 5874 between **secr** and output from other programs, we will often have to convert the density
 5875 to the same units.

5876 Before we can fit the models, the data must first be packaged properly for **secr**.
 5877 We require data files that contain two types of information: trap layout (location and
 5878 identification information for each trap), which is equivalent to the trap deployment file
 5879 (TDF) described in Sec. **??** and the capture data file containing sampling *session*, animal
 5880 identification, trap occasion, and trap location, equivalent in information content to the
 5881 encounter data file (EDF). Sample session can be thought of as primary period identifier
 5882 in a robust design like framework – it could represent a yearly sample or multiple sample
 5883 periods within a year, each of them producing data on a closed population. We discuss
 5884 “multi-session” models in more detail below, in Sec. **??** and Chapt. **??**.

5885 There are three important constructor functions that help package-up your data for
 5886 use in **secr**: **read.traps**, **make.capthist** and **read.mask**. We provide a brief description
 5887 of each here, but apply them to our wolverine camera trapping data in the next section:

5888 (1) **read.traps**: This function points to an external file or **R** data object containing the
 5889 trap coordinates, and other information, and also requires specification of the type of
 5890 encounter devices (described in the next section). A typical application of this function
 5891 looks like the following, invoking the **data=** option when there is an existing **R** object
 5892 containing the trap information:

5893 > trapfile <- **read.traps**(**data=traps**, **detector="proximity"**)

5894 (2) **make.capthist**: This function takes the EDF and combines it with trap information,
 5895 and the number of sampling occasions. A typical application looks like this:

5896 > capturedata <- **make.capthist**(**enc.data**, **trapfile**, **fmt="trapID"**,
 5897 **noccasions=165**)

5898 See **?make.capthist** for definition of distinct file formats. Specifying **fmt = trapID** is
 5899 equivalent to our EDF format.

5900 (3) **read.mask**: If there is a habitat mask available (as described in sec. ??), then this
 5901 function will organize it so that **secr.fit** knows what to do with it. The function
 5902 accepts either an external file name (see **?read.mask** for details of the structure) or a
 5903 $NG \times 2$ **R** object, say **mask.coords**, containing the coordinates of the mask. A typical
 5904 application looks like the following:

5905 > grid <- **read.mask**(**data=mask.coords**)

5906 These constructor functions produce output that can then be used in the fitting of models
 5907 using **secr.fit**.

5908 6.5.1 Encounter device types and detection models

5909 The **secr** package requires that you specify the type of encounter device. Instead of
 5910 describing models by their statistical distribution (Bernoulli, Poisson, etc..), **secr** uses
 5911 certain operational classifications of detector types including ‘proximity’, ‘multi’, ‘single’,
 5912 ‘polygon’ and ‘signal’. For camera trapping/hair snares we might consider ‘proximity’
 5913 detectors or ‘count’ detectors. The ‘proximity’ detector type allows, at most, one detection
 5914 of each individual at a particular detector on any occasion (i.e., it is equivalent to what
 5915 we call the Bernoulli or binomial encounter process model, or model SCR0). The ‘count’
 5916 detector designation allows repeat encounters of each individual at a particular detector
 5917 on any occasion. There are other detector types that one can select such as: ‘polygon’
 5918 detector type which allows for a trap to be a sampled polygon (?) which we discuss further
 5919 in Chapt. ??, and ‘signal’ detector which allows for traps that have a strength indicator,
 5920 e.g., acoustic arrays (?). The detector types ‘single’ and ‘multi’ refer to traps that retain
 5921 individuals, thus precluding the ability for animals to be captured in other traps during
 5922 the sampling occasion. The ‘single’ type indicates trap that can only catch one animal
 5923 at a time (single-catch traps), while ‘multi’ indicates traps that may catch more than one
 5924 animal at a time (multi-catch). These are both variations of the multinomial encounter
 5925 models described in Chapt. ??.

5926 As with all SCR models, **secr** fits an encounter probability model (“detection function”
 5927 in **secr** terminology relating the probability of encounter to the distance of a detector from
 5928 an individual activity center. **secr** allows the user to specify one of a variety of detection
 5929 functions including the commonly used half-normal (“Gaussian”), hazard rate (“Gaussian
 5930 hazard”), and (negative) exponential models. There are 12 different functions as of version
 5931 2.3.1 (see Table ?? in Chapt. ??), but some are only available for simulating data. The
 5932 different detection functions are defined in the **secr** manual and can be found by calling
 5933 the help function for the detection function:

5934 > ?detectfn

5935 Most of the detection functions available in **secr** contain some kind of a scale parameter
 5936 which is usually labeled σ . The units of this parameter default to meters in the **secr**
 5937 output. We caution that the meaning of this parameter depends on the specific detection
 5938 model being used, and it should not be directly compared as a measure of home-range size
 5939 across models. Instead, as we noted in Sec. ?? most encounter probability models imply
 5940 a model of space-usage and fitted encounter models should be converted to a common
 5941 currency such as “area used.”

5942 6.5.2 Analysis using the **secr** package

5943 To demonstrate the use of the **secr** package, we will show how to do the same analysis on
 5944 the wolverine study as shown in Sec. ?? To use the **secr** package, the data need to be
 5945 formatted in a similar but slightly different manner than we use in **WinBUGS**.

5946 For example, in Sec. ?? we introduced a standard data format for the encounter data
 5947 file (EDF) and trap deployment file (TDF). The EDF shares the same format as that used
 5948 by the **secr** package with 1 row for every encounter observation and 4 columns representing
 5949 trap session (‘Session’), individual identity (‘ID’), sample occasion (‘Occasion’), and trap
 5950 identity (‘trapID’). For a standard closed population study that takes place during a single
 5951 season, the ‘Session’ column in our case is all 1’s, to indicate a single primary sampling
 5952 occasion. In addition to providing the encounter data file (EDF), we must tell **secr** infor-
 5953 mation about the traps, which is formed as a matrix with column labels ‘trapID’, ‘x’ and
 5954 ‘y’, the last two being the coordinates of each trap, with additional columns representing
 5955 the operational state of each trap during each occasion (1=operational, 0=not).

5956 We demonstrate these differences now by walking through an analysis of the wolverine
 5957 camera trapping data using **secr**. To read in the trap locations and other related infor-
 5958 mation, we make use of the constructor function **read.traps** which also requires that we
 5959 specify the detector type. The detector type is important because it will determine the
 5960 likelihood that **secr** will use to fit the model. Here, we have selected “proximity” which
 5961 corresponds to the Bernoulli encounter model in which individuals are captured at most
 5962 once in each trap during each sampling occasion:

```
5963 > library(secr)
5964 > library(scrbook)
5965 > data(wolverine)
5966 > traps <- as.matrix(wolverine$wtraps)
```

```

5968 > dimnames(traps) <- list(NULL,c("trapID","x","y",paste("day",1:165,sep="")))
5969 > traps1 <- as.data.frame(traps[,1:3])
5970 > trapfile1 <- read.traps(data=traps1,detector="proximity")

```

5971 Here we note that trap coordinates are extracted from the wolverine data but we do
5972 *not* scale them. This is because **secr** defaults to coordinate scaling of meters which is
5973 the extant scaling of the wolverine trap coordinates. Note that we add a 'trapID' column
5974 to the trap coordinates and provide appropriate column labels to the 'traps' matrix. An
5975 important aspect of the wolverine study is that while the camera traps were operated over
5976 a 165 day period, each trap was operational during only a portion of that period. We need
5977 to provide the trap operation information which is contained in the columns to the right
5978 of the trap coordinates in our standard trap deployment file (TDF). Unfortunately, this is
5979 less easy to do in **secr**⁴, which requires an external file with a single long string of 1's and
5980 0's indicating the days in which each trap was operational (1) or not (0). The **read.traps**
5981 function will not allow for this information on trap operation if the data exists as an **R**
5982 object – instead, we can create this external file and then read it back in with **read.traps**
5983 using these commands:

```

5984 > hold <- rep(NA,nrow(traps))
5985 > for(i in 1:nrow(traps)){
5986 >   hold[i] <- paste(traps[i,4:ncol(traps)],collapse="")
5987 > }
5988 > traps1 <- cbind(traps[,1:3],"usage"=hold)
5989
5990 > write.table(traps1, "traps.txt", row.names=FALSE, col.names=FALSE)
5991 > trapfile2 <- read.traps("traps.txt",detector="proximity")

```

5992 These operations can be accomplished using the function **scr2secr** which is provided in
5993 the **R** package **scrbook**.

5994 After reading in the trap data, we now need to create the encounter matrix or array
5995 using the **make.capthist** command, where we provide the capture histories in EDF format,
5996 which is the existing format of the data input file **wcaps**. In creating the capture history,
5997 we provide also the trapfile created previously, the format (e.g., here EDF format is
5998 **fmt= "trapID"**), and finally, we provide the number of occasions.

```

5999 #
6000 # Grab the encounter data file and format it:
6001 #
6002 wolv.dat <- wolverine$wcaps
6003 dimnames(wolv.dat) <- list(NULL,c("Session","ID","Occasion","trapID"))
6004 wolv.dat <- as.data.frame(wolv.dat)
6005 wolvcapt2 <- make.capthist(wolv.dat,trapfile2,fmt="trapID",noccasions=165)

```

6006 We also set up a habitat mask using the 2×2 km grid which we used previously in the
6007 analysis of the wolverine data and then pass the relevant objects to **secr.fit** as follows:

⁴as of v. 2.3.1

```

6008 #
6009 # Grab the habitat mask (2 x 2 km) and format it:
6010 #
6011 gr2 <- (as.matrix(wolverine$grid2))
6012 dimnames(gr2) <- list(NULL,c("x","y"))
6013 gr2 <- read.mask(data=gr2)
6014 #
6015 # To fit the model we use secr.fit:
6016 #
6017 wolv.secr2 <- secr.fit(wolvcapt2,model=list(D ~ 1, g0 ~ 1, sigma ~ 1),
6018                         buffer=20000,mask=gr2)

```

6019 We are using the “proximity detector” model (SCR0), so we do not need to make any
6020 specifications in the command line because we have specified the detector type using the
6021 constructor function **read.traps**, except to provide the buffer size (in meters). To specify
6022 different models, you can change the default model $D \sim 1$, $g_0 \sim 1$, $\sigma \sim 1$. We provide all
6023 of these commands and additional analyses in the **scrbook** package with the function called
6024 **secr_wolverine**. Printing the output object produces the following (slightly edited):

```

6025 > wolv.secr2
6026
6027 secr 2.3.1, 15:52:45 29 Aug 2012
6028
6029 Detector type      proximity
6030 Detector number     37
6031 Average spacing    4415.693 m
6032 x-range             593498 652294 m
6033 y-range             6296796 6361803 m
6034 N animals           : 21
6035 N detections        : 115
6036 N occasions         : 165
6037 Mask area           : 987828.1 ha
6038
6039 Model               : D ~ 1 g0 ~ 1 sigma ~ 1
6040 Fixed (real)         : none
6041 Detection fn        : halfnormal
6042 Distribution          : poisson
6043 N parameters         : 3
6044 Log likelihood       : -602.9207
6045 AIC                  : 1211.841
6046 AICc                 : 1213.253
6047
6048 Beta parameters (coefficients)
6049          beta   SE.beta      lcl      ucl
6050 D      -9.390124 0.22636698 -9.833795 -8.946452
6051 g0     -2.995611 0.16891982 -3.326688 -2.664535
6052 sigma   8.745547 0.07664648  8.595323  8.895772

```

```

6053
6054 Variance-covariance matrix of beta parameters
6055          D      g0      sigma
6056 D     0.0512420110 -0.0004113326 -0.003945371
6057 g0    -0.0004113326  0.0285339045 -0.006269477
6058 sigma -0.0039453711 -0.0062694767  0.005874683
6059
6060 Fitted (real) parameters evaluated at base levels of covariates
6061      link   estimate   SE.estimate      lcl      ucl
6062 D     log 8.354513e-05 1.915674e-05 5.360894e-05 1.301982e-04
6063 g0    logit 4.762453e-02 7.661601e-03 3.466689e-02 6.509881e-02
6064 sigma log 6.282651e+03 4.822512e+02 5.406315e+03 7.301037e+03

```

6065 The object returned by `secr.fit` provides extensive default output when printed.
6066 Much of this is basic descriptive information about the model, the traps, or the encounter
6067 data. We focus here on the parameter estimates. Under the fitted (real) parameters, we
6068 find D , the density, given in units of individuals/hectare (1 hectare = 10000 m^2). To
6069 convert this into individuals/1000 km 2 , we multiply by 100000, thus our density estimate
6070 is 8.35 individuals/1000 km 2 . The parameter σ is given in units of meters, and so this
6071 corresponds to 6.283 km. Both of these estimates are very similar to those obtained in
6072 our likelihood analysis summarized in Table ?? which, for the 2×2 km grid, we obtained
6073 $\hat{D} = 8.31$ with a SE of $100000 \times 1.915674e - 05 = 1.9156$ and, accounting for the scale
6074 difference (1 unit = 10000 m in the previous analysis), $\hat{\sigma} = \sqrt{1/(2\hat{\alpha}_1)} * 10000 = 6.289$ km.
6075 The difference in the MLE between Table ?? and those produced by `secr` could be due to
6076 subtle differences in internal tuning of optimization algorithms, starting values or other
6077 numerical settings. In addition, the likelihood is based on a Poisson prior for N (see the
6078 next section). On the other hand, the SE is slightly larger based on `secr` which is due to
6079 a subtle difference in the interpretation of D under the `secr` model (See below).

6080 6.5.3 Likelihood analysis in the `secr` package

6081 The `secr` package does likelihood analysis of SCR models for most classes of models as
6082 developed by ?. Their formulation deviates slightly from the binomial form we presented
6083 in Sec. ?? above (though ? also mention the binomial form). Specifically, the likelihood
6084 that `secr` implements is that based on removing N from the likelihood by integrating the
6085 binomial likelihood (Eq. ?? above) over a Poisson prior for N – what we will call the
6086 *Poisson-integrated likelihood* as opposed to the conditional-on- N (*binomial-form*) consid-
6087 ered previously.

6088 To develop the Poisson-integrated likelihood we compute the marginal probability of
6089 each \mathbf{y}_i and the probability of an all-0 encounter history, π_0 , as before, to arrive at the
6090 marginal likelihood in the binomial-form:

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

6091 Now, what ? do is assume that $N \sim \text{Poisson}(\Lambda)$ and they do a further level of marginal-

6092 ization over this prior distribution:

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

6093 In Chapt. ?? we write $\Lambda = \mu ||\mathcal{S}||$ where $||\mathcal{S}||$ is the area of the state-space, and μ is the
 6094 density (“intensity”) of the point process. Carrying out the summation above produces
 6095 exactly this marginal likelihood:

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

6096 which is Eq. 2 of ? except for notational differences. It also resembles the binomial-form
 6097 of the likelihood in Eq. ?? with $\Lambda^n \exp(-\Lambda\pi_0)$ replacing the combinatorial term and the
 6098 $\pi_0^{n_0}$ term. We emphasize there are two marginalizations going on here: (1) the integration
 6099 to remove the latent variables \mathbf{s} ; and, (2) summation to remove the parameter N . We
 6100 provide a function for computing this in the **scrbook** package called **intlik3Poisson**. The
 6101 help file for that function shows how to conduct a small simulation study to compare the
 6102 MLE under the Poisson-integrated likelihood with that from the binomial form.

6103 The essential distinction between our MLE and Borchers and Efford as implemented in
 6104 **secr** is whether you keep N in the model or remove it by integration over a Poisson prior.
 6105 If you have prescribed a state-space explicitly with a sufficiently large buffer, then we
 6106 imagine there should be hardly any difference at all between the MLEs obtained by either
 6107 the Poisson-integrated likelihood or the binomial-form of the likelihood which retains N .
 6108 There is a subtle distinction in the sense that under the binomial form, we estimate the
 6109 realized population size N for the state-space whereas, for the Poisson-integrated form we
 6110 estimate the *prior* expected value which would apply to a hypothetical new study of a
 6111 similar population (see Sec. ??).

6112 Both models (likelihoods) assume \mathbf{s} is uniformly distributed over space, but for the
 6113 binomial model we make no additional assumption about N whereas we assume N is
 6114 Poisson using the formulation in **secr** from (?). Using data augmentation we could do a
 6115 similar kind of integration but integrate N over a binomial (M, ψ) prior – which we referred
 6116 to as the binomial-integrated likelihood in Sec. ???. So obviously the two approaches (data
 6117 augmentation and Poisson-integrated likelihood) are approximately the same as M gets
 6118 large. However, doing a Bayesian analysis by MCMC, we obtain an estimate of both N ,
 6119 the *realized population size*, and the parameter controlling its expected value ψ which are,
 6120 in fact, both identifiable from the data even using likelihood analysis (?). That said we
 6121 can integrate N out completely and just estimate ψ as we noted in Sec. ?? above.

6122 6.5.4 Multi-session models in **secr**

6123 In practice we will often deal with SCR data that have some meaningful stratification or
 6124 group structure. For example, we might conduct mist-netting of birds on K consecutive
 6125 days, repeated, say, T times during a year, or perhaps over T years. Or we might collect
 6126 data from R distinct trapping grids. In these cases, we have T or R groups which we might
 6127 reasonably regard as being samples of independent populations. While the groups might
 6128 be distinct sites, year, or periods within years, they could also be other biological groups

such as sex or age. Conveniently, **secr** fits a specific model for stratified populations – referred to as *multi-session* models. These models build on the Poisson assumption which underlies the integrated likelihood used in **secr** (as described in the previous section). To understand the technical framework, let N_g be the population size of group g and *assume*

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

Naturally, we model group-specific covariates on Λ_g :

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

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

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

```
6147 > out <- secr.fit(capdata, model=list(D ~ session, g0 ~ session, sigma~ 1),
6148           buffer=20000)
```

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

We can also easily implement stratified population models in the various **BUGS** engines using data augmentation (??) which we discuss, with examples, in Chapt. ??.

6.5.5 Some additional capabilities of **secr**

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

Alternative observation models

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

6164 Summary statistics

6165 `secr` provides a useful default summary of the data, but it also has summary statistics
 6166 about animal movement including mean-maximum distance moved (the function `MMDM`).
 6167 For example, see the help page `?MMDM` which lists a number of other summary functions
 6168 which take a `capthist` object:

```
6169 > moves(capthist)
6170 > dbar(capthist)
6171 > RPSV(capthist)
6172 > MMDM(capthist, min.recapt = 1, full = FALSE)
6173 > ARL(capthist, min.recapt = 1, plt = FALSE, full = FALSE)
```

6174 The function `moves` returns the observed distances moved, `dbar` returns the average distance
 6175 moved, `RPSV` produces a measure of dispersion about the home-range center, and
 6176 `ARL` gives the *Asymptotic Range Length* which is the asymptote of an exponential model
 6177 fit to the observed range length vs. the number of detections of each individual (?).

6178 State-space buffer

6179 `secr` will produce a warning if the state-space buffer is chosen too small. For example, in
 6180 fitting the wolverine data as in Sec. ?? but with a 1000 m buffer, and we see the following
 6181 warning message:

```
6182 Warning message:
6183 In secr.fit(wolvcapt2, model=list(D ~ 1, g0 ~ 1, sigma ~ 1), buffer=1000):
6184   predicted relative bias exceeds 0.01 with buffer = 1000
```

6185 This should cause you to contemplate modifying the state-space buffer if that is a reasonable
 6186 thing to do in the specific application.

6187 Model selection and averaging

6188 `secr` does likelihood ratio tests to compare nested models using the function `LR.test`.
 6189 You can create model selection tables based on AIC or AICc, using the function `AIC`,
 6190 and obtain model-averaged parameter estimates using the function `model.average` (See
 6191 Chapt. ?? for examples).

6192 Population closure test

6193 `secr` has a population closure test with the function `closure.test` which implements the
 6194 tests of ? or ?. The function is used like this:
 6195 `closure.test(object, SB = FALSE)`. Here `object` is a `capthist` object and `SB` is a logical
 6196 variable that, if TRUE, produces the ? test.

6197 Density mapping and effective sample area

6198 `secr` produces likelihood versions of the various summaries of posterior density and effec-
 6199 tive sample area that we discussed in Chapt. ???. For example, while `secr` reports
 6200 estimates of the expected value of N or density directly in the summary output from
 6201 fitting a model, you can use the function `region.N` to produce estimates of N for any
 6202 given region. In addition, `secr` has functions for creating maps of detection contours for
 6203 individuals traps, or for the entire trap array. See the function `pdot.contour`, and also

6204 **fxi.contour** for computing the 2-dimensional pdf of the locations of one or more individual
 6205 activity centers (as in Sec. ??). In the context of likelihood analysis, estimation of a
 6206 random effect **s** is based on a plug-in application of Bayes' Rule. When **s** has a uniform
 6207 distribution, and we use a discrete evaluation of the integral, it can be computed simply
 6208 by renormalizing the likelihood:

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

6209 Any of the **intlik** functions given previously in this chapter can be easily modified to
 6210 return the posterior distribution of **s** for any, or all, individuals, or an individual that is
 6211 not encountered.

6212 Effective sample area (see Sec. ??) can be calculated in **secr** using the functions **esa**
 6213 and **esa.plot**).

6214 Covariate models

6215 **secr** has many capabilities for modeling covariates. It has a number of built-in models
 6216 that allow certain covariates on encounter probability, which we cover to a large extent in
 6217 Chapt. ??, and also see Chapt. ?? for more examples. **secr** also allows covariates to be
 6218 built into the density model (see Chapt. ??). It has some built in response surface models,
 6219 allowing for the fitting of linear or quadratic response surfaces. This is done by modifying
 6220 the density model in **secr.fit**. For example, $D \sim 1$ is a constant density surface, and
 6221 $D \sim x + y$ fits a linear response surface, etc.. See the manual **secr-densitysurfaces.pdf**
 6222 for the details.

6223 There are a number of ways to model your own “custom” covariates (as opposed to
 6224 pre-specified models). One way is to use the **addCovariates** function and supply it a
 6225 **mask** or **traps** object along with some “spatialdata.” Or, if you have covariates at each
 6226 trap location then it will extrapolate to all points on the habitat mask. There’s also a
 6227 method by which the user can create a function of geographic coordinates, **userDfn**, which
 6228 seems to provide additional flexibility, although we haven’t used this method. There is a
 6229 handy function **predictDsurface** for producing density maps under the specified model
 6230 for density.

6.6 SUMMARY AND OUTLOOK

6231 In this chapter, we discussed basic concepts related to classical analysis of SCR models
 6232 based on likelihood methods. Analysis is based on the so-called integrated or marginal
 6233 likelihood in which the individual activity centers (random effects) are removed from the
 6234 conditional-on-**s** likelihood by integration. We showed how to construct the integrated
 6235 likelihood and fit some simple models in the **R** programming language. In addition,
 6236 likelihood analysis for some broad classes of SCR models can be accomplished using the
 6237 **R** library **secr** (?) which we provided a brief introduction to. In later chapters we provide
 6238 more detailed analyses of SCR data using likelihood methods and the **secr** package.

6239 Why or why not use likelihood inference exclusively? For certain specific models, it
 6240 is may be more computationally efficient to produce MLEs (for an example see Chapt.
 6241 ??). And, likelihood analysis makes it easy to do model-selection by AIC and compute
 6242 standard errors or confidence intervals. However, **BUGS** is extremely flexible in terms

6243 of describing models and we can devise models in the **BUGS** language easily that we
6244 cannot fit in **secr**. For example, in Chapt ?? we consider open population models which
6245 are straightforward to develop in **BUGS** but, so far, there is no available platform for
6246 doing MLE of such models. We can also fit models in **BUGS** that accommodate missing
6247 covariates in complete generality (e.g., unobserved sex of individuals), and we can adopt
6248 SCR models to include auxiliary data types. For example, we might have camera trapping
6249 and genetic data and we can describe the models directly in **BUGS** and fit a joint model
6250 (?). To do maximum likelihood estimation, we have to write a custom new piece of code
6251 for each model⁵ or hope someone has done it for us. You should have some capability to
6252 develop your own MLE routines with the tools we provided in this chapter.

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

6253
6254

7

6255
6256

MODELING VARIATION IN ENCOUNTER PROBABILITY

6257 In previous chapters we showed how to fit basic spatial capture-recapture models using
6258 Bayesian analysis (in **WinBUGS** or **JAGS**; Chapt. ??) or by classical likelihood methods
6259 (Chapt. ?? or using **secr**). We mostly focused on a specific observation model, the
6260 Bernoulli or binomial model for devices such as “proximity detectors” (although we extend
6261 this model to Poisson and multinomial type observation models in Chapt. ??). We have
6262 not, however, described a general framework for modeling covariates that might influence
6263 encounter probability of individuals, traps or over time. In practice, investigators are
6264 invariably concerned with explicit factors or covariates that might influence variation in
6265 parameters. Such covariates include time (e.g., day of year, or season), behavior (e.g.,
6266 is there an effect of trapping on subsequent capture probabilities), sex of the individual,
6267 and trap type (e.g., various camera types, or different constructions for hair snares).
6268 Traditionally, in the non-spatial capture recapture literature, such models were called
6269 “model M_t ”, “model M_h ”, or “model M_b ”, identifying models that account for variation
6270 in detection probability as a function of time, “individual heterogeneity” or “behavior”,
6271 where behavior describes whether or not an individual had been previously captured. In
6272 SCR models, more complex covariate models are possible because we might also have trap-
6273 specific covariates, or covariates that vary spatially over the landscape, and because we
6274 generally have more than one parameter describing the detection function: Most encounter
6275 probability functions include a baseline encounter rate (λ_0) or probability (p_0) parameter,
6276 and a scale parameter (σ), which takes on different interpretations depending on the
6277 specific encounter probability function under consideration.

6278 In this chapter, we generalize the basic SCR model to accommodate both alternative
6279 detection functions as well as many different kinds of covariates. We focus on the binomial
6280 observation model used throughout Chaps. ?? and ?? and the Gaussian encounter model
6281 (also called the “half-normal” model in the distance sampling literature), but the extension
6282 to other observation models is straightforward (and other encounter probability models
6283 with different functions of distance are considered in Sec. ??). Specifically, we consider

6284 three distinct types of covariates – those which are fixed, partially observed or completely
 6285 unobserved (latent). Fixed covariates are those that are fully observed; for example, the
 6286 date of all sampling occasions. Partially observed covariates are those which are not known
 6287 for all observations; for example, the sex of an individual cannot always be determined
 6288 from photos taken during camera trapping. Even if we are able to observe the sex of all
 6289 individuals sampled, we cannot know it for those individuals never observed during the
 6290 study. And finally, unobserved covariates are those which we cannot observe at all, for
 6291 example, the home range size of individuals, or unstructured random “individual effects”.

6292 We will see that models containing these different types of covariates are relatively easy
 6293 to describe in **WinBUGS** or **JAGS**, and therefore to analyze using Bayesian analysis
 6294 of the joint likelihood based on data augmentation thus providing a coherent and flexible
 6295 framework for inference for all classes of SCR models. Throughout the chapter, we will
 6296 continue to develop the analysis of the black bear study introduced in Chapt. ??, using the
 6297 software **JAGS**. We also consider the likelihood analysis of many of these models; to do so,
 6298 we continue to use the **R** package **secr**, and we introduce some ideas of model comparison
 6299 using AIC (Sec. ?? at the end of the chapter). There are other types of covariates that
 6300 we do *not* cover in this chapter; for example, covariates that vary across the landscape
 6301 might affect density, and we consider these covariates in Chapt. ?? . Alternatively, these
 6302 landscape covariates might affect the way individuals use space. There are probably very
 6303 few circumstances under which animals use all space uniformly and we develop more
 6304 realistic models of encounter probability in which covariates affect space usage in Chapt.
 6305 ?? .

7.1 ENCOUNTER PROBABILITY MODELS

6306 In Chapt. ??, we developed a basic spatial capture recapture model using a standard
 6307 encounter probability function based on the kernel of a normal (Gaussian) probability
 6308 distribution:

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

6309 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
 6310 Sec. ??) that one can view this model as corresponding to an explicit model of space
 6311 usage – namely, that individual locations are draws from a bivariate normal distribution.
 6312 We also mentioned that other detection models are possible, including a logit model of
 6313 the form:

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

6314 However, there's nothing preventing us from constructing a myriad of other models for
 6315 encounter probability as a function of distance. The most commonly used detection prob-
 6316 ability models are also those used in the distance sampling literature: the half-normal
 6317 (Gaussian), the hazard, and the negative exponential. The negative exponential model is:

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

6318 where we define $\alpha_1 = 1/\sigma$. We could use the general power model (?):

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

6319 of which the Gaussian and exponential models are special cases. Another model that could
 6320 be considered is the Gaussian hazard rate model (?):

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

which was previously discussed in Sec. .

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

Table 7.1. Basic encounter probability models (“distance functions”) available in **secr**. (Table taken from the **secr** help files). Notation deviates from that used in the text. In this table g_0 is the baseline encounter rate or probability parameter used in **secr** 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	$g(d) = 1 - F[\{c - (\beta_0 + \beta_1(d-1) - 10 * \log_{10}(d^2))\}/S]$

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

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

7.1.1 Bayesian analysis with bear.JAGS

6339 To demonstrate how to incorporate various types of covariates into models for encounter
 6340 probability using **JAGS**, we return to the data collected during the Fort Drum bear study.
 6341 This data set was first introduced in Chapt. ??, but, to refresh your memory, there were
 6342 38 baited hair snares that were operated between June and July 2006. The snares were
 6343 checked each week for a total for $K = 8$ sample occasions and $n = 47$ individual bears
 6344 were encountered at least once. The data are provided in the **R** package **scrbook** and an
 6345 **R** function called **bear.JAGS** allows the user to easily pick which model to analyze. The
 6346 function **bear.JAGS** will set up the data, write the model, define the MCMC specifications
 6347 (e.g., initial values, etc.) and, finally, run the selected model in **JAGS**. In addition to
 6348 choosing which model to run, the user can also specify the number of chains, iterations and
 6349 length of the burn-in phase. Calling the function will provide all the code to implement
 6350 the models independently as well. In the following sections we will present the model code
 6351 and output for the most commonly employed models; for all analyses we ran 3 chains with
 6352 a burn-in of 500 iterations and 20000 saved iterations.

7.1.2 Bayesian analysis of encounter probability models

6354 In Panel ??, we present the basic SCR model and show how to specify the negative exponential
 6355 encounter probability model. To call each of these from the function **bear.JAGS** set
 6356 **model='SCRO'** or **model='SCRexp'** in the function call, respectively. To reduce repetition
 6357 of the R coding, we include the basic code here and then only show modifications when
 6358 necessary throughout the chapter. All of the R coding can be found within the **bear.JAGS**
 6359 function as well. The function begins by loading the required **R** libraries as well as the
 6360 Ft. Drum bear data set. This data set includes a 3-d data array (called **bearArray** in our
 6361 code), with dimensions **nind** \times **ntraps** \times **nreps** representing the capture histories of **nind**
 6362 captured individuals at **ntraps** trap locations. In the Bayesian analysis, data augmentation
 6363 is used to estimate N and therefore the **bearArray** data must be augmented with
 6364 $M - nind$ all zero encounter histories. In models without time dependence, the augmented
 6365 **bearArray** (called **Yaug** in the code) will be reduced to a 2 dimensional array (denoted **y**
 6366 in the code) that has dimensions **M** \times **ntraps**.

```
6368 > library(rjags) # Load the necessary libraries
6369 > library(scrbook)
6370
6371 > data(beardata) # Attach the bear data for Ft. Drum
6372 > ymat <- beardata$bearArray
6373 > trapmat <- beardata$trapmat
6374 > nind <- dim(beardata$bearArray)[1]
6375 > K <- dim(beardata$bearArray)[3]
6376 > ntraps <- dim(beardata$bearArray)[2]
6377 > M <- 650
6378 > nz <- M-nind
6379
6380 # Create augmented array
6381 > Yaug <- array(0, dim=c(M,ntraps,K))
```

```

6382 > Yaug[1:nind,,] <- ymat
6383 > y <- apply(Yaug,1:2, sum)

```

6384 The function `bear.JAGS` also establishes the upper and lower limits on the state space
 6385 by centering the trap array coordinates (which are imported with the `beardata` and saved
 6386 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.

6387 Applying the SCR model with Gaussian encounter probability model provides an
 6388 estimate (posterior mean) of $D = 0.167$ bears per km^2 and with the negative exponential
 6389 encounter probability model the posterior mean is virtually the same $D = 0.167$. In
 6390 distance sampling, the use of different encounter probability models often results in very
 6391 different estimates of density (especially when using the negative exponential model).
 6392 There are two main reasons why the different models may have less of an impact on the
 6393 density estimates under the SCR models. First, we can estimate the baseline encounter
 6394 probability parameter (p_0). In most distance sampling models, detection at distance 0 is
 6395 set to 1. In Table ??, the posterior mean of p_0 is 0.11 under the Gaussian model and
 6396 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 ??), 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. ??.

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 ?? 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.

6423 Broadly speaking, we recognize (here) 3 types of covariates. Fixed covariates are fully
 6424 observable and might vary by trap alone (e.g., type of trap, baited or not, disturbance
 6425 regime, even habitat), sample occasion (e.g., day of season or weather conditions), or both
 6426 (e.g., behavior, weather - if over a large region). Another class of covariates are those
 6427 which vary at the level of the individual (and possibly also over time). As a technical
 6428 matter, and as noted before, these are different from fixed covariates because we cannot
 6429 see all of the individuals and the covariates are almost always incompletely observed (if
 6430 at all). The lone exception is the effect of previous capture, used to model a behavioral
 6431 response to capture, which is known for all individuals, captured or not (an animal never
 6432 captured/observed has never been captured before). We noted many times before that
 6433 space itself (i.e., the activity centers) is a type of individual covariate and this notion
 6434 actually helped us derive the fully spatial capture-recapture model from the traditional,
 6435 non-spatial model (Chapt. ??). We do not get to observe the activity center for any
 6436 individuals, but for individuals that are encountered we get to observe some information
 6437 about it in the form of which traps the individual was encountered in. And finally, we have
 6438 completely unobserved covariates such as heterogeneity in home range size. We consider
 6439 heterogeneity in a separate section below since there are a suite of models for describing
 6440 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 ??)
time	season, shedding, weather
individual x time	global behavioral response
trap x time	trap failures
individual x trap x time	local behavioral response

6441 To develop covariate models, we assume a standard sampling design in which an array
 6442 of J traps is operated for K sample occasions, which produces encounter histories for n
 6443 individuals. For the null model, there are no time-varying covariates that influence en-
 6444 counter, there are no explicit individual-specific covariates, and there are no covariates
 6445 that influence density. For fixed effects, those which we observe fully, we can easily incor-
 6446 porate these into the encounter probability model, just as we would do in any standard
 6447 GLM or GLMM, on some suitable scale for the encounter probability, p_{ijk} . For example,

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * C_{ijk}$$

$$p_{ijk} = p_{0,ijk} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

6448 where C_{ijk} is some covariate that varies (potentially) by individual (i), trap (j) and
 6449 occasions (k), and α_2 is the coefficient to be estimated. How we define specific covariates
 6450 (e.g., trap specific versus individual specific) will influence exactly how we include them
 6451 in the model. Table ?? shows examples of covariates by type – trap, individual, and time
 6452 – and also gives examples of some combined types. These are the types of covariates we
 6453 will specifically address in this chapter, demonstrating how to analyze the different types
 6454 in the following sections.

6455 **7.2.1 Date and time**

6456 Often, researchers are interested in modeling the effect of date or chronological time on
 6457 encounter probability. For example, in a long term hair snare study, we may expect
 6458 that seasonal shedding (?) will influence encounter probabilities directly. Or, we may
 6459 expect behaviors such as denning, mating, etc., to influence the encounter of certain
 6460 species at certain times of year (?). There are two common ways to incorporate date or
 6461 time information into a model for encounter probability. For cases with a small number
 6462 of sampling occasions we can fit a time-specific intercept (analogous to “model M_t ” in
 6463 classical capture-recapture (?)). In this model, there are K sampling occasion-specific
 6464 parameters to reflect potential variation in sampling effort or other factors that might
 6465 vary across samples. Alternatively, we can model parametric functions of date or time
 6466 such as polynomial or sinusoidal functions.

6467 In the first case, we allow each sampling occasion, k , to have its own baseline encounter
 6468 probability, e.g.,

$$\text{logit}(p_{0,k}) = \alpha_{0,k}$$

6469 so that

$$p_{ijk} = p_{0,k} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2).$$

6470 This description of the model includes k occasion-specific baseline encounter probabilities.
 6471 Thus, if there are 4 sampling occasions, then there are 4 different baseline encounter
 6472 probabilities. We imagine that complete time-specificity of p_0 (i.e., one distinct value
 6473 for each sample occasion) would be most useful in situations where there are just a few
 6474 sampling occasions (if there are many, this formulation will dramatically increase the
 6475 number of parameters to be estimated) or we do not expect systematic patterns over time
 6476 (e.g., explainable by a polynomial function or time-varying covariates).

6477 To implement this in **JAGS**, α_0 has to be estimated for each time period k either
 6478 using an index vector or dummy variables (as described in Chapt. 2 and Sec. ??) and this
 6479 can be done by only changing only a few lines in Panel ??:

```
6480 alpha0[k] ~ dnorm(0,.1)
6481 logit(p0[k]) <- alpha0[k]
6482 .....
6483 .....
6484 y[i,j,k] ~ dbin(p[i,j,k],K)
6485 p[i,j,k] <- z[i]*p0[k]*exp(- alpha1*d[i,j]*d[i,j])
```

6486 Since the model contains a parameter for each time period, the encounter histories
 6487 must be time-dependent. Thus, a 3-d data array (called **bearArray** in our code), with
 6488 dimensions **nind** × **ntraps** × **nreps** is required (recall that we use the 3-d augmented array
 6489 called **Yaug** with dimensions **M** × **ntraps** × **nreps** for the Bayesian analysis). In addition
 6490 to using the 3-d data array, the initial values must be updated so that there are K values
 6491 generated for α_0 . And finally, this means that another nested *for loop* is needed in the
 6492 code to account for the K sample occasions. A side note: the computation time will
 6493 increase quite a bit (this model for the bear data may take up to 15 hours or more on
 6494 your machine to obtain a sufficient posterior sample).

6495 Running this model with the function **bear.JAGS** by setting **model=SCRt**, returns es-
 6496 timates of density similar to those from the model without covariates (see Table ??), 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 (?). 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. ?? 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 ? 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:

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * \text{Date}_k$$

$$p_{ijk} = p_{0,ijk} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

6521 or a quadratic effect of day-of-year:

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * \text{Date}_k + \alpha_3 * \text{Date}_k^2 \\ p_{ijk} = p_{0,ijk} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

6522 where the variable `Date` is an integer coding of day-of-year, indexed to some arbitrary
6523 start point in time.

6524 7.2.2 Trap-specific covariates

6525 In some studies it makes sense to model encounter probability as a function of local or trap-
6526 specific covariates. These can be one of two types: genuine trap covariates that describe
6527 the trap or encounter site, such as whether a trap is baited or not, or how many traps
6528 were set at a sampling location, or what kind of bait was used, etc., or local covariates
6529 that describe the likelihood that an animal would use the habitat in the vicinity of the
6530 trap (see Chapt. ?? for more on this situation). We imagine that these covariates, of
6531 either type, should affect baseline encounter probability. For example, ? found a large
6532 difference in the encounter probability of jaguars due to traps being located on roads,
6533 which the animals were using to travel along, as opposed to traps placed off of roads. In
6534 this case, the trap type is a binary variable – on/off road, (another binary variable could
6535 be baited/non-baited). We can write this as:

$$\text{logit}(p_{0,j}) = \alpha_{0,type_j} \\ p_{ijk} = p_{0,j} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2).$$

6536 Here, we use an index variable, “type”, an integer value for the trap-specific covariate.
6537 Thus for our example of on/off road, we would have $type_j = 1$ if trap j is on a road and
6538 $type_j = 2$ otherwise, and we would estimate two separate α_0 parameters – one for on-road
6539 and one for off-road cameras. An alternative way to express the 2-category model, using
6540 dummy variables, requires that we specify our “type” vector as $Type_j = 0$ if trap j is on
6541 a road and $Type_j = 1$ otherwise, and write the model as

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * Type_j.$$

6542 Now, α_0 is the baseline encounter probability (on the logit scale) for traps on a road
6543 ($Type_j = 0$) and α_2 is the effect on baseline encounter probability of a trap being of
6544 $Type = 1$. This general set up also allows for more than 2 categories, say if 4 different
6545 camera models were used in a study, we would use a set of 3 binary dummy variables
6546 to allow for estimation of the different encounter rates (i.e., the intercept). While these
6547 models are equivalent, and should yield identical results, sometimes one parameterization
6548 might work better than the other in **WinBUGS** or **JAGS** (?).

6549 7.2.3 Behavior or trap response by individual

6550 One of the most basic of encounter models is that which accommodates a change in
6551 encounter probability as a result of initial encounter. This is colloquially referred to as
6552 “trap happiness” or “trap shyness”, or in other words, a behavioral response of individuals

6553 to being captured (?). If a trap is baited with a food source, an individual might come back
 6554 for more. On the other hand, if being captured is traumatic then an individual might learn
 6555 to avoid traps. Both of these types of responses can occur in most species depending on
 6556 the type of encounter mechanisms being employed. Moreover, behavioral response can be
 6557 either global (?) or local (?). The local response is a trap-specific response while a global
 6558 response suggests that initial capture provides a net increase or decrease in subsequent
 6559 probabilities of capture (across all traps). A behavioral response does not need to be
 6560 enduring (i.e., persist for the entire study after the individual has been captured/observed
 6561 for the first time) but can also be ephemeral, if, for example, an animal only avoids a trap
 6562 on the occasion immediately after it was captured (??). While we will focus the examples
 6563 in this chapter on enduring behavioral effects, extending such a model to the case of an
 6564 ephemeral response should not pose any difficulties.

6565 To describe these behavioral models we need to create a binary matrix that indicates
 6566 if an individual has been captured previously. For the global behavioral response, define
 6567 the $n \times K$ matrix, \mathbf{C} , where $C_{ik} = 1$ if individual i was captured at least once prior to
 6568 session k , otherwise $C_{ik} = 0$.

$$\text{logit}(p_{0,ik}) = \alpha_0 + \alpha_2 * C_{ik}$$

$$p_{ijk} = p_{0,ik} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2).$$

6569 For the local behavioral response, which is trap specific, we create an array, C_{ijk} , that
 6570 indicates if an individual i has been previously captured in trap j at time k . (For the
 6571 augmented individuals, the entries are all 0 since the animals were never captured.) We
 6572 then include this in the model in the exact same form as above (with the sole difference
 6573 that both C and p are now also indexed by k):

$$\text{logit}(p_{0,ijk}) = \alpha_0 + \alpha_2 * C_{i,j,k}$$

$$p_{ijk} = p_{0,ijk} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2).$$

6574 Since the behavioral response is occasion specific, to implement either the local or
 6575 global response model in **JAGS**, we will have to use the 3-d array of the augmented
 6576 capture histories ($M \times ntraps \times nreps$) as we did for the time-varying encounter probability
 6577 model above. The code must loop over each sampling occasion, but otherwise, the model
 6578 varies only a little from the basic SCR model shown in Panel ???. Here is the specification
 6579 of the the occasion specific (k) loop:

```
6580 for(k in 1:K){  

  6581   logit(p0[i,j,k]) <- alpha0 + alpha2*C[i,j,k]  

  6582   y[i,j,k] ~ dbin(p[i,j,k],1)  

  6583   p[i,j,k] <- z[i]*p0[i,j,k]*exp(- alpha1*d[i,j]*d[i,j]).  

  6584 }
```

6585 Despite only minor changes to the **BUGS** code, this model can require quite a bit
 6586 of time and computational effort. Implementing the behavioral models with the function
 6587 **bear.JAGS** by setting **model=SCRb** or **model=SCRb** for the local or global model respec-
 6588 tively, returns the results shown in Table ???. There is a strong global behavioral response
 6589 suggested by the posterior mean of $\alpha_2 = 0.90$. The estimate of N and subsequently D are
 6590 larger than under the model without a behavioral response; here we estimate the posterior

mean of $N = 577.56$, whereas in the SCR0 model, we estimated the posterior mean as $N = 500$. This makes sense given the large estimate of α_2 , which suggests that bears are trap happy. In situations where animals are trap happy, the null model tends to over estimate encounter probability (i.e., the bears that are never observed have a lower encounter probability than those that have been captured in the study) and thereby reduce the estimate of N . We do not include the results here, but the estimates were similar 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

6598 7.2.4 Individual covariates

6599 Individual covariates are those which are measured (or measurable) on individuals, so we
6600 get to observe them only for the captured individuals. Sex is a simple example of an
6601 individual covariate, but one of the most commonly used in capture-recapture studies.
6602 The sex of an individual can influence many aspects of its ecology and behavior, including
6603 for example, the frequency of movement, seasonal behavior, and its home range size. This
6604 is common in studies of carnivores where females often have smaller home ranges than
6605 males (??). Additionally, we may find differences in the baseline encounter probability
6606 between males and females because females may move around less frequently, or possibly
6607 because they are less likely to use landscape structures that researchers may target with
6608 sampling devices in order to increase sample size, such as roads (e.g. ?). Therefore, we can
6609 imagine that sex may impact both the baseline encounter probability α_0 and the typical
6610 home range size, so that α_1 might also be sex-specific also. The fully sex-specific model
6611 is:

$$\text{logit}(p_{0,i}) = \alpha_{0,sex_i}$$

$$p_{ijk} = p_{0,i} \exp(-\alpha_{1,sex_i} * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

6612 where sex_i is a vector indicating the sex of each individual (1 = male, 2 = female).
6613 While we might know the sex of all individuals observed in the study, we will never
6614 know the sex of individuals that are not observed (?). It is also possible that we may
6615 not be able to determine the sex of individuals that are observed during the study. For
6616 example photographic captures do not necessarily result in pictures that allow the sex to
6617 be absolutely determined, thus sometimes resulting in missing values of this covariate for
6618 animals captured in the study. We deal with this slightly differently depending on the
6619 inference framework that we adopt (Bayesian or likelihood). Here we demonstrate the

6620 Bayesian implementation and we discuss the likelihood approach using `secr` in detail
 6621 below in Sec. ???. Before proceeding with that, we note that it would be possible also to
 6622 model covariates directly on the parameter σ (or its logarithm), e.g., $\log(\sigma_i) = \theta_1 + \theta_2 \text{sex}_i$
 6623 (see Sec. ??). One or the other (or perhaps *some* other) parameterization may yield a
 6624 better performing MCMC algorithm or provide a more natural or preferred interpretation.
 6625 In the context of Bayesian analysis, given that priors are not invariant to transformation of
 6626 the parameters, this may be a consideration in choosing the particular parameterization.

6627 Specifying a fully sex-specific model for **JAGS** is similar to the time-specific model
 6628 shown above. We need to use an index or dummy variable to let α_0 and/or α_1 be defined
 6629 separately for males and females. The main difference in this specification is that we do
 6630 not observe sex for the augmented individuals. Therefore, we have missing observations
 6631 of the covariate for those individuals. As a result, sex is regarded as a random variable
 6632 and so the missing values can be estimated along with the other structural parameters of
 6633 the model.

6634 Because we are regarding sex as a random variable, we have to specify a distribution for
 6635 it. With only two possible outcomes, it is natural to suppose that $\text{Sex}_i \sim \text{Bernoulli}(\psi_{\text{sex}})$
 6636 where the parameter ψ_{sex} is the sex ratio of the population. We assume our default non-
 6637 informative prior for this parameter: $\psi_{\text{sex}} \sim \text{Uniform}(0, 1)$. The model specification in
 6638 Panel ?? demonstrates how to incorporate a partially observed covariate (i.e., “sex”). It
 6639 is important to note that in the previous equation, sex_i is a vector with two categories
 6640 indicating the sex of each individual (e.g., 1 = male, 2 = female). This corresponds
 6641 directly to having a binary indicator of sex (e.g., $\text{Sex}_i = 1$ if individual i is female, and 0
 6642 otherwise). In the Bayesian formulation of the model, we use both the binary indicator
 6643 (`Sex`) and a categorical indicator (`Sex2 = Sex + 1`). The former (termed `Sex` in Panel
 6644 ??) allows us to specify the Bernoulli distribution for the random variable, and the latter
 6645 (termed `Sex2`) allows us to use the dummy or indicator variable specification in the model.

6646 In both **JAGS** or **BUGS** missing data are indicated by `NA` in the data objects passed
 6647 to the program through the `bugs` or `jags` functions in **R**. To set up the data, we need to
 6648 create a vector of length M with the first n elements being 0 if individual i is a female, or
 6649 1 if i is a male (for the Fort Drum black bear data the function `bear.JAGS` extracts this
 6650 information automatically from the `beardata` object), and the subsequent $M - n$ elements
 6651 being `NA`. It is generally a good idea to provide starting values for the missing data, but we
 6652 cannot provide starting values for observed data; in this case where one vector (or other
 6653 object) contains both observed and missing data, initial values for the observed data have
 6654 to be specified as `NA`. The code snippet below shows you how to set up the data including
 6655 the `Sex` vector and the initial values function (the remainder of the code is identical to
 6656 what we’ve shown before).

```
6657 > sex <- beardata$sex #the sex data for captured individual
6658 > Sex <- c(sex-1, rep(NA, nz)) #sex enters as 1/2, this recodes it to 0/1
6659 #so we can use Bernoulli distribution
6660
6661 > data <- list(y=y,Sex=Sex, M=M,K=K, J=ntraps, xlim=xlim, ylim=ylim,area=areaX)
6662 > params <- c('psi','p0','N', 'D', 'sigma', 'psi.sex')
6663 > inits <- function() { list(z=c(rep(1,nind), rbinom(nz,1,0.5)),psi=runif(1),
6664   s=cbind(runif(M, xlim[1],xlim[2]), runif(M,ylim[1],ylim[2])), 
6665   psi.sex=runif(1),Sex=c(rep(NA, nind), rbinom(nz,1,0.5)),
```

6666 `sigma=runif(2,2,3),alpha0=runif(2)) }`

6667 The **BUGS** model specification is shown in Panel ??.

```
model{

psi ~ dunif(0,1)                                # Prior distributions
psi.sex ~ dunif(0,1)
for(t in 1:2){
  alpha0[t] ~ dnorm(0,.1)
  logit(p0[t]) <- alpha0[t]
  alpha1[t] <- 1/(2*sigma[t]*sigma[t])
  sigma[t] ~ dunif(0, 15)
}

for(i in 1:M){
  z[i] ~ dbern(psi)
  Sex[i] ~ dbern(psi.sex)                      # Sex is binary
  Sex2[i] <- Sex[i] + 1                         # Convert to categorical
  s[i,1] ~ dunif(xlim[1],xlim[2])
  s[i,2] ~ dunif(ylim[1],ylim[2])

  for(j in 1:J){
    d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
    y[i,j] ~ dbin(p[i,j],K)
    p[i,j] <- z[i]*p0[Sex2[i]]*exp(-alpha1[Sex2[i]]*d[i,j]*d[i,j])
  }
}
N <- sum(z[])
D <- N/area
}
```

Panel 7.2: **JAGS** model specification for an SCR model with sex-specific encounter probability parameters.

6668 Our estimate of density under the fully sex-specific model is still very similar to the
 6669 previous models (Table ??), and while the baseline detection was not very different be-
 6670 tween males and females, we can see that they had very different σ estimates (note that
 6671 the BCIs do not overlap). As usual, you can reproduce this analysis by calling the function
 6672 `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

6673 Here we consider SCR models with individual heterogeneity. Capture-recapture models
 6674 with individual heterogeneity in detection probability, so-called model M_h , have a long
 6675 history in classical capture recapture models and they have special relevance to SCR (Sec.
 6676 ??). While the advent of SCR models may appear to have rendered the use of classical
 6677 model M_h obsolete (because one major source of heterogeneity, namely exposure to the
 6678 trap array is being accounted for explicitly) we may still wish to consider heterogeneity
 6679 models for other biological reasons. It is reasonable to expect in real populations that there
 6680 exists heterogeneity in home range size and so we think that α_1 could exhibit heterogeneity
 6681 among individuals. As we noted previously, it may be advantageous or desirable in some
 6682 cases to model heterogeneity directly in terms of the scale parameter of the encounter
 6683 probability function, σ , or some other transformation of the “distance coefficient”, perhaps
 6684 even 95% home range area.

6685 In this section, we describe a class of spatial capture-recapture models to allow for
 6686 individual heterogeneity in encounter probability. In particular, one class of models we
 6687 propose explicitly admits individual heterogeneity in home range *size*. In addition, we con-
 6688 sider a standard representation for heterogeneity in which an additive individual-specific
 6689 random effect is included in the linear predictor for baseline encounter probability.

6690 7.3.1 Models of heterogeneity

6691 An obvious extension to the SCR model is to include an additive individual effect, analo-
 6692 gous to classical “model M_h ”. We’ll call this model “SCR+Mh”:

$$\text{logit}(p_{0,i}) = \alpha_0 + \eta_i$$

$$p_{ijk} = p_{0,i} \exp(-\alpha_1 * ||\mathbf{x}_j - \mathbf{s}_i||^2)$$

6693 where η_i is an individual random effect having distribution $[\eta|\sigma_p]$. A popular class of
 6694 models arises by assuming $\eta_i \sim \text{Normal}(0, \sigma_p^2)$ (??). We show how to implement this
 6695 specific SCR + Mh model in Panel ??, and this model can be used to analyze the Ft.
 6696 Drum bear data by calling the function `bear.JAGS` and setting `model='SCRh'`. While
 6697 we show one possible implementation here, many other random effects distributions are
 6698 possible. A popular one is the finite-mixture of point masses (??) which we demonstrate
 6699 how to fit using `secr` in Sec. ??.

```

model{

alpha0 ~ dnorm(0,.1)                                # Prior distributions
alpha1 <- 1/(2*sigma*sigma)
sigma ~ dunif(0, 15)
psi ~ dunif(0,1)
tau_p ~ dgamma(.001,.001)

for(i in 1:M){
  eta[i] ~ dnorm(0, tau_p)                         # Individual level variables
  z[i] ~ dbern(psi)
  s[i,1] ~ dunif(xlim[1],xlim[2])
  s[i,2] ~ dunif(ylim[1],ylim[2])

  for(j in 1:J){                                    # The "likelihood" etc..
    d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
    y[i,j] ~ dbin(p[i,j],K)
    logit(p0[i,j]) <- alpha0 + eta[i]
    p[i,j] <- z[i]*p0[i,j]*exp(- alpha1*d[i,j]*d[i,j])
  }
}
N <- sum(z[])
D <- N/area                                         # N, D are derived
}

```

Panel 7.3: **JAGS** model specification for the SCR + Mh model with Gaussian encounter probability model and additive normal random effect.

6700 7.3.2 Heterogeneity induced by variation in home range size

6701 An alternative heterogeneity model, one that has more of a direct biological motivation and
 6702 interpretation, describes heterogeneity in home range size among individuals. To model
 6703 heterogeneity in home range area, we can assume a distribution for a transformation of
 6704 the scale parameter of the encounter probability model such as σ^2 , or $\log(\sigma^2)$, etc.. We
 6705 call this “model SCR + Ah” (Ah here for area-induced heterogeneity).

6706 Consider the following log-normal model for the individual scale parameter of the
 6707 Gaussian encounter probability model, σ_i^2 :

$$\log(\sigma_i^2) \sim \text{Normal}(\mu_{hra}, \tau_{hra}^2)$$

6708 then the 95% home range area has a scaled log-normal distribution with mean

$$6\pi \exp(\mu_{hra} + \tau_{hra}^2/2).$$

6709 The variance is slightly more complicated, but you can look up the variance of a log-normal
 6710 distribution and combine it with the 95% home range area calculation in Sec. ?? to work
 6711 out the implied variance of home range area under this model. We show two examples of
 6712 the implied *population* distribution of home range area under this log-normal model that
 6713 indicates a mean home range area of about 6.9 area units (Figure ??). The left panel
 6714 shows a standard deviation in home range area of 2.88 units and the right panel shows
 6715 a standard deviation in home range area of 0.70 units. The two cases were generated by
 6716 tweaking the μ_{hra} and τ_{hra}^2 parameters of the log-normal distribution to achieve a constant
 6717 expected value of home range area, but modify the standard deviation.

7.4 LIKELIHOOD ANALYSIS IN SECR

6718 Previously, in Chapt. ??, we introduced the **R** package **secr** and described the likelihood
 6719 based inference approach taken by that package (see Sec. ??). Here we discuss how to
 6720 implement some standard covariate models in **secr** and provide an example of model
 6721 selection using AIC. As we saw in Chapt. ??, **secr** uses the standard **R** model specifi-
 6722 cation syntax, defining the dependent and independent variable relationship using tildes
 6723 (e.g., $y \sim x$). Thus, in **secr** we might have $g0 \sim \text{behavior}$ or $\text{sigma} \sim \text{time}$; when left
 6724 unspecified or set to 1 (e.g., $g0 \sim 1$), this will default to a model with no covariates (i.e.,
 6725 constant parameter values). A number of default model formulas for the baseline and
 6726 scale parameter of the encounter probability model are available in **secr**. Additionally,
 6727 **secr** allows us to specify covariates on density (we cover this in Chapt. ??), which are
 6728 set for example as $D \sim \text{habitat}$.

6729 To demonstrate models with various types of covariates using **secr**, we continue using
 6730 the Fort Drum black bear data. We include in the **scrbook** package a function called
 6731 **secr.bear** that will format the data (see Chapt. ?? for the **secr** data format) and then
 6732 fit and compare 8 models (details shown in Panel ??). We have described all of these
 6733 models in the previous sections, so we only briefly comment here on how to fit certain
 6734 models in **secr** and compare them using AIC, and give a few helpful notes.

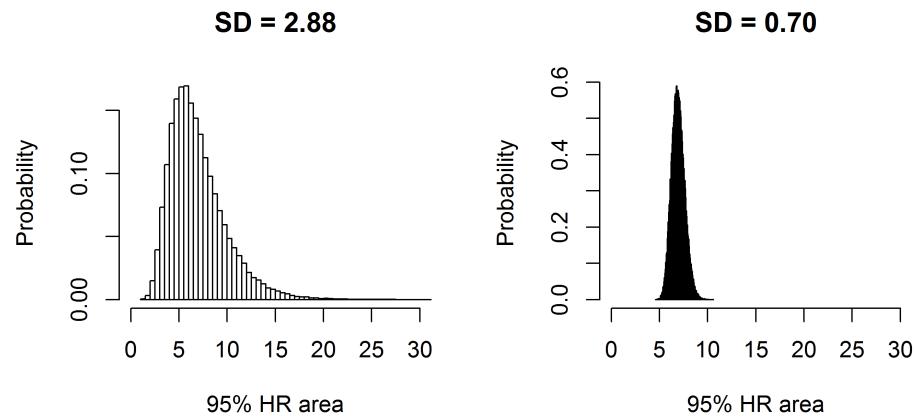


Figure 7.1. Population distribution of home range area for a model in which $\log(\sigma^2)$ has a normal distribution with mean μ_{hra} and variance τ_{hra}^2 . The parameters were chosen to yield a constant expected value of about 6.9 units of area, but to produce two different levels of heterogeneity: A population standard deviation of 2.88 units (left panel) and 0.70 units (right panel).

6735 7.4.1 Notes for fitting standard models

6736 In the **secr** package, the encounter probability model is called the “detection function”
 6737 and it is specified by changing the “**detectfn**” option (an integer code) within the
 6738 **secr.fit** command. Table ?? shows the possible encounter probability models that **secr**
 6739 allows; the default is that based on the kernel of a bivariate normal probability distribution
 6740 function (hence we call this the Gaussian model, but it is referred to as “half-normal” in
 6741 **secr**) and the (negative) exponential is **detectfn** = 2. See model 2 in Panel ?? for how
 6742 to fit the exponential model to the Fort Drum bear data set.

6743 The **secr** package easily fits a range of SCR equivalents of standard capture-recapture
 6744 models. The package has pre-defined versions of the classic model M_t where each occa-
 6745 sion has its own encounter probability, as well as a linear trend in baseline encounter
 6746 probability over occasions (in a spatial modeling framework σ could also be an occasion
 6747 specific parameter, but having encounter probability change with time seems like the more
 6748 common case). For the classical time-effects type of model with K distinct parameters
 6749 **secr** uses ‘t’ to denote this in the model specification formula (see model 3 in panel ??);
 6750 whereas, for a linear trend over occasions **secr** uses ‘T’.

6751 The global trap response model (what we called model M_B), or a local trap-specific
 6752 behavioral response (model M_b) can be fitted in **secr** using formulae with “b” for the
 6753 global response model and “bk” for the local trap response model (see models 4 and 5 in
 6754 Panel ??; note that to fit the trap specific behavioral response model you need version

6755 2.3.1 or newer of `secr`).

```

1. null model with a bivariate normal encounter probability model
bear_0=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ 1, sigma ~ 1))

2. null model with an exponential encounter probability model
bear_0exp=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ 1, sigma ~ 1),
                    detectfn=2)

3. model with fixed time effects
bear_t=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ t, sigma ~ 1))

4. global behavioral model
bear_B=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ b, sigma ~ 1))

5. trap specific behavioral response
bear_b=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ bk, sigma ~ 1))

6. global behavior model with fixed time effects
bear_bt=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ b+t, sigma ~ 1))

7. sex-specific model
bear_sex=secr.fit(bear.cap, model=list(D ~ session, g0 ~ session,
                                         sigma ~ session))

8. heterogeneity model
bear_h2=secr.fit(bear.cap, model=list(D ~ 1, g0 ~ h2, sigma ~ h2))

```

Panel 7.4: Models called from `secr.bear` function. All models use `buffer = 20000`

6756 **7.4.2 Sex effects**

6757 Incorporating sex effects into models with `secr` can be done a few different ways, but
 6758 there are not pre-defined models for this. A limitation of fitting models with sex effects
 6759 in `secr` is that it does not accommodate missing values of the sex variable. Thus, in all
 6760 cases, individuals that are of unknown sex must be removed from the data set (recall that
 6761 in a Bayesian framework we can keep these individuals in the data set by specifying a
 6762 distribution for the individual covariate “sex”). In `secr`, the easiest way to include sex
 6763 effects is to code sex as a “session” variable using the multi-session models (see Sec. ?? for
 6764 a description of the multi-session models), providing two sessions, one representing males
 6765 and one for females (see model 7 in Panel ??). This method provides two separate density
 6766 estimates, which can then be combined into a total density.

7.4.3 Individual heterogeneity

To incorporate heterogeneity, `secr` fits a set of finite mixture models (??). These are expensive in terms of parameters but they have been widely adopted because they are easy to analyze using likelihood methods, as the marginal distribution of the data is just a sum of a small number of components. Using `secr`, individual heterogeneity can be incorporated into the encounter probability model using default models for either a 2- or 3-component finite mixture model using the “`h2`” or “`h3`” model terms. The 2-part mixture is shown in model 8 of panel ?? and the 3-part mixture can easily be fit by substituting `h3` for `h2`. We only showed the SCR + Mh logit-normal mixture in the version above (see Sec. ??), but finite-mixture models can also be fit in **JAGS** or **BUGS**.

7.4.4 Model selection in `secr` using AIC

One practical advantage to using the `secr` package, or likelihood inference in general, is the convenience of automatic model selection using AIC (?). The `secr` package has a number of convenient functions for computing AIC and producing model selection tables, or doing model-averaging (as described in Chapt. ??). Running the function `secr.bear`, which calls all of the models we have described, will return, in addition to all model results, an AIC table with all of the summarized results including the AIC values, delta AIC, and model weights (see Table ?? or reproduce results in R using `out<- secr.bear(); out$AIC.tab`).

It is important to note that AIC is not comparable between a multi-session model and a model that is not a multi-session model. Therefore, to compare the sex-specific model (which uses “sessions”) with all the other models including the null, time, and behavioral models, we coded the data set as a multi-session design when first loading it to `secr`. This results in all the model outputs listing separate parameter estimates for each session, even the null model with no covariates; however, the estimates are the same for both “sessions” in all but the sex-specific model (in other words, we don’t specify any effect of session on parameters, except in the sex specific model).

Table 7.7. Log-likelihood, AIC, deltaAIC and AIC weight for several models run in `secr` for the Fort Drum black bear data set.

model	logLik	AIC	AICc	dAICc	AICwt
bear.b	-641.7215	1291.443	1292.395	0.000	1
bear.h2	-653.8382	1319.676	1321.776	29.381	0
bear.0exp	-663.9152	1333.830	1334.389	41.994	0
bear.B	-677.6175	1363.235	1364.187	71.792	0
bear.bt	-668.3044	1358.609	1366.152	73.757	0
bear.sex	-677.7151	1367.430	1369.530	77.135	0
bear.t	-674.4134	1368.827	1374.938	82.543	0
bear.0	-686.2455	1378.491	1379.049	86.654	0

The results from this AIC analysis are straightforward to interpret; the model with a local trap response of encounter probability, “`bk`”, has a model weight of 1 and thus, according to AIC, 100% support compared with the other models in this model set. The 2-part finite mixture model for g_0 and σ has the second lowest AIC, but considering the

6798 large dAICc compared to the local trap response model we would probably not consider
 6799 it any further.

7.5 SUMMARY AND OUTLOOK

6800 There are endless covariates and encounter probability models that can be defined and our
 6801 goal in this chapter was to introduce basic types of covariate models and demonstrate how
 6802 to implement them in **BUGS** and **secr**. Essentially, SCR's are GLMMs and therefore
 6803 we develop covariate models in much the same way, using a suitable transformation (link
 6804 function) of the parameter(s). In SCR models, we typically have 2 parameters of the
 6805 encounter probability model for which we might specify covariate models – the **baseline**
 6806 encounter probability (or rate) parameter, and a scale parameter that is related in many
 6807 cases to the home range size of the species. A few examples of different covariate models
 6808 are given in Table ???. We can also consider covariates by their classification as fixed,
 6809 partially observed, or unobserved (see Table ??). This classification of covariate types can
 6810 be important because the MLE and Bayesian approaches to dealing with partially and
 6811 unobserved covariates is often different. This was seen above in how the covariate **Sex** was
 6812 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

6813 While the move to spatially explicit models in capture-recapture studies has largely
 6814 rendered the basic CR models (?) obsolete, we continue to find this classification useful
 6815 for categorizing the *spatial* extensions of these standard CR models. The extended models
 6816 include the standard M_0 , M_t , M_b , and M_h , but also new models that allow for trap-specific
 6817 information such as "baited/not-baited" or "on/off road". In addition, in Chaps. ??, ??
 6818 and ??, we explore models for explaining variation in encounter probability and density
 6819 based on spatial covariates that describe variation in landscape or habitat conditions.

6820
6821

8

MODEL SELECTION AND ASSESSMENT

6823 Our purpose in life is to analyze models. By that, we mean one or more of the following
6824 basic 4 tasks: (1) estimate parameters, (2) make predictions of unobserved random variables,
6825 (3) evaluate the relative merits of different models or choosing a best model (model
6826 selection), and (4) checking whether a specific model appears to provide a reasonable
6827 description of the data or not (model checking, assessment, or “goodness-of-fit”). In previous
6828 chapters we addressed the problems of estimation of model parameters, and also making
6829 predictions of latent variables, s or z , or functions of these variables such as density or
6830 population size. In this chapter, we focus on the last two of these basic inference tasks:
6831 model selection (which model or models should be favored), and model assessment (do
6832 the data appear to be consistent with a particular model).

6833 In this chapter we review basic strategies of model selection using both likelihood
6834 methods (as implemented in the `secr` package) and Bayesian analysis. Specifically, we
6835 review a number of standard methods of model selection that apply to “variable selec-
6836 tion” problems, when our set of models consists of distinct covariate effects and they
6837 represent constraints of some larger model. For classical analysis based on likelihood,
6838 model selection by Akaike Information Criterion (AIC) is the standard approach (?). For
6839 Bayesian analysis we rely on a number of different methods. We demonstrate the use of
6840 the deviance information criterion (DIC) (?) for variable selection problems although it
6841 has deficiencies when applied to hierarchical models in some cases (?). We use the Kuo
6842 and Mallick indicator variable selection approach (?) which produces direct statements
6843 of posterior model probabilities which we think are the most useful, and leads directly to
6844 model-averaged estimates of density. There is a good review paper recently by ? that
6845 discusses these and many other related ideas for variable selection. In addition to ? we
6846 also recommend ?, Chapt. 7 for general information on model selection and assessment.

6847 To check model adequacy in a Bayesian framework, or whether a specific model pro-
6848 vides a satisfactory description of our data set, we rely exclusively on the Bayesian p-value
6849 framework (?). For assessing fit of SCR models, part of the challenge is coming up with
6850 good measures of model fit, and there does not appear much definitive guidance in the
6851 literature on this point. Following ?, we break the problem up into 2 components which

6852 we attack separately: (1) Conditional on the underlying point process, does the encounter
 6853 model fit? (2) Do the uniformity and independence assumptions appear adequate for the
 6854 point process model of activity centers? The latter component of model fit has a consider-
 6855 able precedence in the ecological literature as it is analogous to the classical problem of
 6856 testing “complete spatial randomness” (??).

6857 We apply some of these methods to the wolverine camera trapping data first introduced
 6858 in Chapt. ?? to investigate sex specificity of model parameters and whether there is a
 6859 behavioral response to encounter. We note that individuals are drawn to the camera
 6860 trap devices by bait and therefore it stands to reason that once an individual discovers a
 6861 trap, it might be more likely to return subsequently, a response termed “trap happiness”.
 6862 We evaluate whether certain models for encounter probability appear to be adequate
 6863 descriptions of the data, and we evaluate the uniformity assumption for the underlying
 6864 point process.

8.1 MODEL SELECTION BY AIC

6865 Using classical analysis based on likelihood, model selection is easily accomplished using
 6866 AIC (?) which we demonstrate below. The AIC of a model is simply twice the negative
 6867 log-likelihood evaluated at the MLE, penalized by the number of parameters (np) in the
 6868 model:

$$AIC = -2\log L(\hat{\theta}|\mathbf{y}) + 2np$$

6869 Models with small values of AIC are preferred. It is common to use a modified (“cor-
 6870 rected”) AIC referred to as AIC_c for small sample sizes which is

$$AIC_c = -2\log L(\hat{\theta}|\mathbf{y}) + \frac{2np(np+1)}{n-np-1}$$

6871 where n is the sample size. Two important problems with the use of AIC and AIC_c are
 6872 that they don’t apply directly to hierarchical models that contain random effects, unless
 6873 they are computed directly from the marginal likelihood (for SCR models we can do this,
 6874 see Chapt. ??). Moreover, it is not clear what should be the effective sample size n
 6875 in calculation of AIC_c , as there can be covariates that affect individuals, that vary over
 6876 time, or space. We do not offer strict guidelines as to when to use a small sample size
 6877 adjustment.

6878 The R package **secr** computes and outputs AIC automatically for each model fitted
 6879 and it provides some capabilities for producing a model selection table (function **AIC**) and
 6880 also doing model-averaging (function **model.average**), which we recommend for obtaining
 6881 estimates of density from multiple models.

6882 8.1.1 AIC analysis of the wolverine data

6883 We provide an example of model selection for the wolverine camera trapping data using
 6884 **secr**. We consider a model set with distinct models to accommodate various types of sex
 6885 specificity of model parameters:

6886 Model 0: model SCR0 with constant density and constant encounter model parameters;

6887 Model 1: model SCR0 with constant parameter values for both male and female wolverines but with sex-specific density only;
 6888
 6889 Model 2: Sex-specific density, sex-specific p_0 but constant σ ;
 6890 Model 3: Sex-specific density, sex-specific σ but constant p_0 ;
 6891 Model 4: Sex-specific density, sex-specific p_0 and sex-specific σ .

6892 To model sex-specific abundance (density), we use the multi-session models provided
 6893 by **secr** (introduced in Sec. ??), which allow one to model session-specific effects on den-
 6894 sity, baseline encounter probability, p_0 (labeled g_0 in **secr**), and also the scale parameter
 6895 σ of the encounter probability model. Using this formulation, we define the “Session”
 6896 variable to be a *categorical* sex code having value 1 or 2 (demonstrated below) and thus
 6897 *session*-specific parameters represent *sex*-specific parameters. For example, if we model
 6898 session-specific density, D , then this corresponds to Model 1 in our list above. We note
 6899 that “Model 0” in our list corresponds to a model where all of the encounter histories
 6900 have the same session ID. This model is one of constant density, which implies that the
 6901 population sex ratio is fixed at 0.5, i.e., $\psi_{\text{sex}} = 0.5$.

6902 Although **secr** also uses the logit/log linear predictors as the default for modeling
 6903 covariates on baseline encounter probability and the scale parameter, respectively, **secr**
 6904 does something different with the multi-session models. It reports estimates in a *session*
 6905 *mean* parameterization (equivalent to, in **BUGS**, using an index variable instead of a set
 6906 of dummy variables), and not the *session effect* (i.e., deviation from the intercept) which
 6907 arises from the use of dummy variables. We show this **BUGS** model description in Sec.
 6908 ??.

6909 To fit these models using **secr**, we load the wolverine data and do a slight bit of
 6910 formatting to prepare the data objects for analysis by **secr**. The key difference from our
 6911 analysis in Chapt. ?? is, here, we use the wolverine sex information (**wolverine\$wsex**)
 6912 which is a binary 0/1 variable (1=male) and we add 1 so that we can define a categorical
 6913 “Session” variable (having values 1 or 2). We also have a function **scr2secr** which converts
 6914 a standard trap-deployment file (TDF) matrix into a **secr** object of class “traps.” The
 6915 **R** commands are as follows (contained in the help file **?secr_wolverine**):
 6916

```

6916
6917 > library(secr)
6918 > library(scrbook)
6919 > data(wolverine)
6920 > traps <- as.matrix(wolverine$wtraps)
6921
6922 ## Name variables as required by secr
6923 > dimnames(traps) <- list(NULL,c("trapID","x","y",paste("day",1:165,sep="")))
6924 ## Convert trap information to a secr "traps" object
6925 > trapfile <- scr2secr(scrtraps=traps,type="proximity")
6926
6927 ## Grab the wolverine state-space grid (2km here)
6928 > gr <- as.matrix(wolverine$grid2)
6929 > dimnames(gr) <- list(NULL,c("x","y"))
6930 > gr2 <- read.mask(data=gr)
6931
  
```

```

6932 ## Grab the encounter data, and re-name variables
6933 > wolv.dat <- wolverine$wcaps
6934 > dimnames(wolv.dat) <- list(NULL,c("Session","ID","Occasion","trapID"))
6935
6936 ## Convert binary 0/1 sex variable to categorical 1/2 for "session"
6937 > wolv.dat[,1] <- wolverine$wsex[wolv.dat[,2]]+1
6938 > wolv.dat <- as.data.frame(wolv.dat)
6939
6940 ## Convert to capthist object
6941 > wolvcapt <- make.capthist(wolv.dat,trapfile,fmt="trapID",noccasions=165)

```

6942 Once the data have been prepared in this way, we use the `secr` model fitting function
 6943 `secr.fit` to fit the different models, and then the function `AIC` to package the models
 6944 together and summarize them in the form of an AIC table, with rows of the table ordered
 6945 from best to worst. The function `model.average` performs AIC-based model-averaging of
 6946 the parameters specified by the `realnames` variable (below this is demonstrated for the
 6947 parameter density, D). Because this function defaults to averaging by AIC_c , we slightly
 6948 modified this function (called `model.average2`) to do model averaging by either AIC or
 6949 AIC_c as specified by the user. The model fitting commands look like this (for Model 0
 6950 and Model 1):

```

6951 > model0 <- secr.fit(wolvcapt, model=list(D~1, g0~1, sigma~1),
6952                      buffer=20000)
6953 > model1 <- secr.fit(wolvcapt, model=list(D~session, g0~1, sigma~1),
6954                      buffer=20000)

```

6955 Next we use the function `AIC`, passing the fit objects from all 5 models, and that
 6956 produces the following output (abbreviated horizontally to fit on the page):

```

6957 > AIC (model0,model1,model2,model3,model4)
6958           model      ... npar logLik   AIC    AICc dAICc AICwt
6959 model0  D~1 g0~1 sigma~1  ...  3 -627.2603 1260.521 1261.932 0.000 0.5831
6960 model2      ..      ...  5 -624.9051 1259.810 1263.810 1.878 0.2280
6961 model1      ..      ...  4 -627.2365 1262.473 1264.973 3.041 0.1275
6962 model4      ..      ...  6 -624.6632 1261.326 1267.326 5.394 0.0393
6963 model3      ..      ...  5 -627.2358 1264.472 1268.472 6.540 0.0222

```

6964 Model averaging the results is done as follows:

```

6965 > model.average (model0,model1,model2,model3,model4,realnames="D")
6966           estimate SE.estimate      lcl      ucl
6967 session=1 2.707190e-05 7.913577e-06 1.544474e-05 4.745224e-05
6968 session=2 2.927423e-05 8.270402e-06 1.700631e-05 5.039193e-05

```

6969 As usual, estimates and standard errors of the individual model parameters can be
 6970 obtained from the `secr.fit` summary output of any of the `modelX` objects shown above.
 6971 The default output of estimated density is in individuals per ha, so we have to scale this
 6972 up to something more reasonable. To get into units of per 1000 km², we need to first
 6973 multiply by 100 to get to units of km² and then multiply by 1000. This produces an

estimated density of about 2.71 for `session=1` (females) and 2.93 for `session=2` (males). We can use the generic **R** function `predict` applied to the `secr.fit` output to obtain specific information about the MLEs on the natural scale.

We don't necessarily agree with the use of AIC_c here and think its better to use AIC, in general. This is because, as noted previously, it is not clear what the effective sample size is for most capture-recapture problems. While we have 21 individuals in the data set, most of the model structure has to do with encounter probability samples and for that there are hundreds of observations. We do note that the AIC and AIC_c results are not entirely consistent. By looking at the best model by AIC (Table ??), we find that the model with sex specific density and sex-specific baseline encounter probability, p_0 , is preferred (Model 2). This is just slightly better than the null model (Model 0) with no sex effects at all and hence an implied fixed sex ratio of $\psi_{sex} = 0.50$.

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 km². Model abbreviations indicate which parameters are sex-specific in order $D/p_0/\sigma$.

NO HABITAT MASK									
model	npar	AIC	AICc	D	Female		Male		
					p_0	σ	D	p_0	σ
2: sex/sex/1	5	1259.8	1263.8	2.45	0.08	6435.51	3.16	0.04	6435.51
0: 1/1/1	3	1260.5	1261.9	2.83	0.06	6298.66	2.83	0.06	6298.66
4: sex/sex/sex	6	1261.3	1267.3	2.59	0.08	6080.70	2.99	0.04	6833.16
1: sex/1/1	4	1262.5	1265.0	2.69	0.06	6298.69	2.96	0.06	6298.69
3: sex/1/sex	5	1264.5	1268.5	2.70	0.06	6280.49	2.95	0.06	6319.03
WITH HABITAT MASK									
model	npar	AIC	AICc	D	Female		Male		
					p_0	σ	D	p_0	σ
2: sex/sex/1	5	1268.1	1272.1	3.64	0.07	6382.88	4.73	0.03	6382.88
4: sex/sex/sex	6	1268.7	1274.7	3.87	0.07	5859.40	4.41	0.03	7039.09
0: 1/1/1	3	1271.2	1272.6	4.18	0.05	6282.62	4.18	0.05	6282.62
1: sex/1/1	4	1273.1	1275.6	3.98	0.05	6282.65	4.38	0.05	6282.65
3: sex/1/sex	5	1275.1	1279.1	3.93	0.05	6357.26	4.41	0.05	6220.22

We fit the same models but now using a modified state-space which excludes the ocean (this is a habitat mask in `secr`). Results are shown in Table ?? along with the previous models without a mask. We see AIC values are smaller for the model without the mask. It is probably acceptable to compare these different fits (with and without habitat mask) by AIC because we recognize the mask as having the effect of modifying the random effects distribution (i.e., of the activity centers, *s*) and the results should be sensitive to choice of the distribution for *s*. That said, we tend to prefer the mask model because it makes sense to exclude the areas of open water from the state-space of *s*. For females the model-averaged density is 3.88 individuals per 1000 km² and for males the model-averaged density estimate is 4.46 individuals per 1000 km² as we see here:

```
> model.average (model0b,model1b,model2b,model3b,model4b,realnames="D")
6997
```

```

6998      estimate   SE.estimate      lcl      ucl
6999 session=1 3.876615e-05 1.189102e-05 2.153795e-05 6.977518e-05
7000 session=2 4.459658e-05 1.323696e-05 2.523280e-05 7.882022e-05

```

7001 This is quite a bit higher than that based on the rectangular state-space (i.e., not
 7002 specifying a habitat mask). This is not surprising given that **the state-space is part**
 7003 **of the model** and the specific state-space modification we made here, which reduces the
 7004 area from the rectangular state-space, should be extremely important from a biological
 7005 standpoint (i.e., wolverines are not actively using open ocean).

8.2 BAYESIAN MODEL SELECTION

7006 Model selection is somewhat less straightforward as a Bayesian, and there is no canned
 7007 all-purpose method like AIC. As such we recommend a pragmatic approach, in general,
 7008 for all problems, based on a number of basic considerations:

- 7009 (1) For a small number of fixed effects we think it is reasonable to adopt a conventional
 7010 “hypothesis testing” approach – i.e., if the posterior for a parameter overlaps zero
 7011 substantially, then it is probably reasonable to discard that effect from the model.
- 7012 (2) Calculation of posterior model probabilities: In some cases we can implement methods
 7013 which allow calculation of posterior model probabilities. One such idea is the indicator
 7014 variable selection method from ?. For this, we introduce a latent variable $w \sim \text{Bern}(.5)$
 7015 and expand the model to include the variable w as follows:

$$\text{logit}(p_{ijk}) = \alpha_0 + w * \alpha_1 * C_{ijk}.$$

7016 The importance of the covariate C is then measured by the posterior probability that
 7017 $w = 1$.

- 7018 (3) The Deviance Information Criterion (DIC): Bayesian model selection is now routinely
 7019 carried out using DIC (?), although its effectiveness in hierarchical models depends
 7020 very much on the manner in which it is constructed (?). We recommend using it if it
 7021 leads to sensible results, but we think it should be calibrated to the extent possible for
 7022 specific classes of models. This has not yet been done in the literature for SCR models,
 7023 to our knowledge.
- 7024 (4) Logical argument: For something like sex specificity of certain parameters, it seems
 7025 to make sense to leave an extra parameter in the model no matter what because, bio-
 7026 logically, we might expect a difference (e.g., home range size). In some cases failure to
 7027 apply logical argument leads to meaningless tests of gratuitous hypotheses (?).

7028 In all modeling activities, as in life itself, the use of logical argument should not be under-
 7029 utilized.

8.2.1 Model selection by DIC

7031 The availability of AIC makes the use of likelihood methods convenient for problems where
 7032 likelihood estimation is achievable. For Bayesian analysis, DIC seemed like a general-
 7033 purpose equivalent, at least for a brief period of time after its invention. However, there
 7034 seem to be many variations of DIC, and a consistent version is not always reported across

computing platforms. Even statisticians don't have general agreement on practical issues related to the use of DIC (?). Despite this, it is still widely reported. We think DIC is probably reasonable for certain classes of models that contain only fixed effects, or for which the latent variable structure is the same across models so that only the fixed effects are varied (this covers many SCR model selection problems). However, it would be useful to see some calibration of DIC for some standardized model selection problems.

Model deviance is defined as negative twice the log-likelihood; i.e., for a given model with parameters θ : $\text{Dev}(\theta) = -2 * \log L(\theta|\mathbf{y})$. The DIC is defined as the posterior mean of the deviance, $\overline{\text{Dev}}(\theta)$, plus a measure of model complexity, p_D :

$$\text{DIC} = \overline{\text{Dev}}(\theta) + p_D$$

The standard definition of p_D is

$$p_D = \overline{\text{Dev}}(\theta) - \text{Dev}(\bar{\theta})$$

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. ? suggest a different version of p_D based on one-half the posterior variance of the deviance:

$$p_V = \text{Var}(\text{Dev}(\theta)|\mathbf{y})/2.$$

This is what is produced from **WinBUGS** and **JAGS** if they are run from **R2WinBUGS** or **R2jags**, respectively. It is less easy to get DIC summaries from **rjags**, so we used **R2jags** in our analyses below.

8.2.2 DIC analysis of the wolverine data

We repeated the analysis of the wolverine models with sex specificity, but this time doing a Bayesian analysis paralleling the likelihood analysis we did above in **secr**, using the logit/log parameterization of the model parameters. To do so in **BUGS**, we used dummy variables. Thus, we can express models allowing for sex specificity using a dummy variable **Sex** and new parameters (α_{sex} , β_{sex}) which represent the effect of **Sex** at level 1:

$$\text{logit}(p_{0,i}) = \alpha_0 + \alpha_{sex} \mathbf{Sex}_i$$

and

$$\log(\sigma_i) = \log(\sigma_0) + \beta_{sex} \mathbf{Sex}_i.$$

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.

Unlike the multi-session model in **secr**, we carry out the analysis of the sex-specific model here by putting all of the data into a single data set, and explicitly accounting for the covariate 'sex' in the model by assigning it a Bernoulli prior distribution with ψ_{sex} being the proportion of males in the population. In this case, we produce "Model 0" above, the model with no sex effect on density, by setting the population proportion of males at one-half: $\psi_{sex} = 0.5$ (see also Sec. ??). As usual, handling of missing values of the 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).

7069 The **BUGS** model specification for the most complex model, Model 4, is shown in
 7070 Panel ???. This model has sex-specific intercept, scale parameter, σ , and density. We
 7071 provide an **R** script named `wolvSCR0ms` in the `scrbook` package which will fit each model.
 7072 The function uses **JAGS** by default for the fitting, using the `R2jags` package. The kernel
 7073 of this function is the model specification in Panel ???, which gets modified depending on
 7074 the model we wish to fit using a command line option `model`. For example, `model = 1`
 7075 fits the model with constant parameter values for males and females, but sex-specific
 7076 population sizes (`model = 0` constrains the male probability parameter, ψ_{sex} , to be 0.5).
 7077 The **R** function fits each of the 5 models using a binary indicator variable to turn ‘on’ or
 7078 ‘off’ each effect. Here is how we obtain the MCMC output for each of the 5 models:

```
7079 > wolv0 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=0)
7080 > wolv1 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=1)
7081 > wolv2 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=2)
7082 > wolv3 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=3)
7083 > wolv4 <- wolvSCR0ms(nb=1000,ni=21000,buffer=2,M=200,model=4)
```

7084 We fitted the 5 models to the wolverine data and summarize the DIC computation
 7085 results in Table ???. The model rank has model 0, model 2, model 1, model 4, model 3.
 7086 Interestingly, this is the same order as the models based on AIC_c which we found above
 7087 (see Table ???). The posterior mean and SD of model parameters under the 5 models are
 7088 given in Table ???.

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

7089 8.2.3 Bayesian model averaging with indicator variables

7090 A convenient way to deal with model selection and averaging problems in Bayesian analysis
 7091 by MCMC is to use the method of model indicator variables (?). Using this approach,
 7092 we expand the model to include a set of prescribed models as specific reductions of a
 7093 larger model. This has been demonstrated in some specific capture-recapture models in
 7094 ?, Sec. 3.4.3, and ? and in the context of SCR by ?. A useful aspect of this method is
 7095 that model-averaged parameters are produced by default. We emphasize the need to be
 7096 careful of reporting model-averaged parameters that don’t have a common interpretation
 7097 in the different models because they are meaningless (averaging apples and oranges....).
 7098 For example, if a regression parameter is in a specific model then the posterior is informed
 7099 by the data and a specific MCMC draw is from the appropriate posterior distribution.

```

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
	$pD = 77.1$		$pD = 77.5$		$pD = 78.4$		$pD = 79.5$		$pD = 80.1$	
	$DIC = 518.1$		$DIC = 519.3$		$DIC = 518.6$		$DIC = 522.8$		$DIC = 521.3$	

7100 On the other hand, if the regression parameter is not in the model then the MCMC draw
 7101 is obtained directly from the prior distribution, and so we need to think carefully about
 7102 whether it makes sense to report an average of such a thing (in the vast majority of
 7103 cases the answer is no). But some parameters like N or density, D , do have a consistent
 7104 interpretation and we support producing model-averaged results of those parameters.

7105 To implement the Kuo and Mallick approach, we expand the model to include the
 7106 latent indicator variables, say w_m , for variable m in the model, such that

$$w_m = \begin{cases} 1 & \text{linear predictor includes covariate } m \\ 0 & \text{linear predictor does not include covariate } m \end{cases}$$

7107 We assume that the indicator variables w_m are mutually independent with

$$w_m \sim \text{Bernoulli}(0.5)$$

7108 for each variable $m = 1, 2, \dots$, in the model. For example, with 2 variables, the expanded
 7109 model has the linear predictor:

$$\text{logit}(p_{ijk}) = \alpha_0 + \alpha_1 w_1 C_{1,i} + \alpha_2 w_2 C_{2,ijk}$$

7110 where, let's suppose, $C_{1,i}$ is an individual covariate such as sex, and $C_{2,ijk}$ is a behavioral
 7111 response covariate which is individual-, trap-, and occasion-specific. We can assume a
 7112 parallel model specification on the parameter σ which is liable to vary by individual level
 7113 covariates such as sex:

$$\log(\sigma_i) = \beta_0 + \beta_1 w_3 C_{1,i}.$$

7114 Using this indicator variable formulation of the model selection problem we can char-
 7115 acterize unique models by the sequence of w variables. In this case, each unique sequence
 7116 (w_1, w_2, w_3) represents a model, and we can tabulate the posterior frequencies of each
 7117 model by post-processing the MCMC histories of (w_1, w_2, w_3) , as we demonstrate shortly.
 7118 This method then evaluates all possible combinations of covariates or 2^m models.

7119 Conceptually, analysis of this expanded model within the data augmentation frame-
 7120 work does not pose any additional difficulty. One broader, technical consideration is that
 7121 posterior model probabilities are well known to be sensitive to priors on parameters (??).

See also ?, Sec. 3.4.3 and ?, 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 (?). 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 (?). 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:

```
7147 > mod <- wolv5$BUGSoutput$sims.list$mod
7148 > mod <- paste(mod[,1],mod[,2],mod[,3],sep="")
7149 >
7150 > table(mod)
7151 mod
7152   000   001   010   011   100   101   110   111
7153 17181  4935  1057   296 25211   8337  2275    708
7154
7155 > round( table(mod)/length(mod) , 3)
7156 mod
7157   000   001   010   011   100   101   110   111
7158  0.286  0.082  0.018  0.005  0.420  0.139  0.038  0.012
```

This results in a comparison of all 8 possible models (based on $m = 3$ covariates) instead of just the 5 models we originally proposed. We see that the best model is that labeled 100 which, according to our construction above, has `mod[1]=1`, `mod[2]=0` and `mod[3]=0`. This is the model having sex-specific baseline encounter probability p_0 , and $\psi_{\text{sex}} = 0.5$.

7163 This model has posterior model probability 0.420. The model with no sex specificity at
 7164 all (the model with label 000) has posterior probability 0.286 and the remaining posterior
 7165 mass is distributed over the other six models. We could arrive at a qualitatively similar
 7166 conclusion using a more ad hoc approach based on looking at the posterior mass for each
 7167 parameter under the full model (model 4; see Table ??, in part). Considering the sex-
 7168 specific intercept, it appears to be very important as its posterior mass is mostly away
 7169 from 0. On the other hand, the coefficient on log-sigma is concentrated around 0, and
 7170 the estimated ψ_{sex} (probability that an individual is a male) is 0.54 with a large posterior
 7171 standard deviation. We might therefore be inclined to discard the sex effect on $\log(\sigma)$
 7172 based on classical thinking-like-a-hypothesis-testing-person and settle for the model with
 7173 a sex-specific intercept in the encounter probability model. This is consistent with our
 7174 indicator variable approach which found that model (1,0,0) has posterior probability of
 7175 0.420. Looking at the posteriors for each parameter to thin the model down is consistent
 7176 with these results. We can obtain model-averaged estimates from the indicator variable
 7177 approach, which produces direct model-averaged estimates of N and D :

```
7178   mu.vect sd.vect  2.5%    25%    50%    75%  97.5% Rhat n.eff
7179 D      5.695   1.133   3.759   4.916   5.591   6.362   8.193 1.002 3600
7180 N     59.077  11.758  39.000  51.000  58.000  66.000  85.000 1.002 3600
```

7181 We obtain a model-averaged estimate (posterior mean) for density of $D = 5.695$ which
 7182 is hardly any different from our model specific estimates (Table ??) and, in particular,
 7183 from model 2 which has only a sex-specific intercept.

7184 8.2.4 Choosing among detection functions

7185 Another approach to implementing model indicator variables is to introduce a categorical
 7186 “model identity” variable which is itself a parameter of the model. Using this approach,
 7187 then each distinct model is associated with a unique set of covariates or other set of model
 7188 features. This is convenient especially when we cannot specify the linear predictor as
 7189 some general model that reduces to various alternative sub-models simply by switching
 7190 binary variables on or off. In the context of SCR models, choosing among different en-
 7191 counter probability models would be an example. For this case we do something like this
 7192 `mod ~ dcat(probs[])` where `probs` is a vector with elements $1/(\#models)$, and the en-
 7193 counter probability matrix is filled in depending on the value of `mod`. In particular, instead
 7194 of a 2-dimensional array `p[i,j]`, we build `p[i,j,m]` for each of $m = 1, 2, \dots, M$ models.
 7195 An example with 3 distinct models is:

```
7196 mod ~ dcat(probs[])
7197 ##
7198 ## Using a double loop construction fill-in p[,] for each model:
7199 ##
7200 p[i,j,1] <- p0[1]*exp( - alpha1[1]*dist2[i,j] )
7201 p[i,j,2] <- 1-exp(-p0[2]*exp( - alpha1[2]*dist2[i,j] ) )
7202 logit(p[i,j,3]) <- p0[3] - alpha1[3]*dist2[i,j]
7203
7204 mu[i,j] <- z[i]*p[i,j,mod]
7205 y[i,j] ~ dbin(mu[i,j],K[j])
```

As before the posterior probabilities can be highly sensitive to priors on the different model parameters and sometimes mixing is really poor and, in general, we've experienced mixed success trying to carry out model selection using this construction. We do provide a template **R/JAGS** script (`wolvSCR0ms2`) in the `scrbook` package which has an example of choosing among 3 different encounter probability models: The Gaussian encounter probability, Gaussian hazard, and logistic model with the square of distance (defined in Sec. ??). The key things to note are that there are 3 intercepts and 3 different '`alpha1`' parameters (the coefficient on distance). The parameters should not be regarded as equivalent across the models, so it is important to have them separately defined (and estimated) for each model. In our analysis we used a vague normal prior (precision = 0.1) for the intercept parameter (either log or logit-scale of baseline encounter probability p_0) and a `Uniform(0,5)` prior for one-half the inverse of the coefficient on distance-squared. In the **BUGS** model specification the priors look like this:

```
7219 for(i in 1:3){
7220   alpha0[i] ~ dnorm(0,.1)
7221   sigma[i] ~ dunif(0,5)
7222   alpha1[i] <- 1/(2*sigma[i]*sigma[i])
7223 }
```

Then, we create a probability of encounter for each individual, trap *and* model so that the holder object "p" in the model description is a 3-dimensional array (sometimes this would have to be a 4 or 5-d array in more complex models with time effects, etc..), so that construction of the encounter probability models look like this:

```
7228 p[i,j,1] <- p0[1]*exp( - alpha1[1]*dist2[i,j] )
7229 p[i,j,2] <- 1-exp(-p0[2]*exp( - alpha1[2]*dist2[i,j] ) )
7230 logit(p[i,j,3]) <- p0[3] - alpha1[3]*dist2[i,j]
```

where

```
7232 logit(p0[1]) <- alpha0[1]
7233 log(p0[2]) <- alpha0[2]
7234 p0[3] <- alpha0[3]
```

You can experiment with the `wolvSCR0ms2` script to investigate the importance of different models of encounter probability and whether they have an affect on the inferences.

8.3 EVALUATING GOODNESS-OF-FIT

In practical settings, we estimate parameters of a desirable model, or maybe fit a bunch of models and report estimates from all of them or a model-averaged summary of density. An important question is: Is our model worth anything? In other words, does the model appear to be an adequate description of our data? Formal assessment of model adequacy or goodness-of-fit is a challenging problem and there are no all-purpose algorithms for doing this in either frequentist or Bayesian paradigms. Moreover, there are some philosophical challenges to evaluating model fit, such as, if we do model averaging then should all of the models have to fit? Or should the averaged model have to fit? What if none of the

models fit? We don't know the answers to these questions and we won't try to answer them. Instead, we will provide what guidance we can on taking the first steps to evaluating fit, of a single model, as if it were a cherished family heirloom of great importance. We suggest that if you have a model that you really like, a single model, then it is a sensible thing to check that the model is a good fit to your data. If it is not, we do not imagine that the model is useless but just that some thought should be put into why the model doesn't fit so that, perhaps, some remediation might happen as future data are collected. After all, you may have spent 2, 3 or many more years of your life collecting that data set, perhaps thousands of hours, and therefore it seems a reasonable proposition to expect to do some estimation and analysis of the model regardless of model fit. You can still learn something from a model that does not pass some technical litmus test of model fit.

Conceptually, we can think of evaluation of model fit as follows: if we simulate data under the model in question, do the simulated realizations resemble the data set that we actually have? For either Bayesian or classical inference, the basic strategy to assessing model fit is to come up with a fit statistic that depends on the parameters and the data set, which we denote by $T(\mathbf{y}, \theta)$, and then we compute this for the observed data set, and compare its value to that computed for perfect data sets simulated under the correct model. In the case of classical inference, we will often rely on the standard practice of parametric bootstrapping (?), where we simulate data sets conditional on the MLE $\hat{\theta}$ and compare realizations with what we've observed. The R package `unmarked` (?) contains generic bootstrapping methods for certain hierarchical models, including distance sampling (e.g., see ?, for an application). In simple cases, using classical inference methods, it is sometimes possible to identify a test statistic of theoretical merit, perhaps with a known asymptotic distribution. For examples from capture-recapture see ?, ?, and Chapt. 5 of ?. For Bayesian analysis we use the Bayesian p-value method (?) (we introduced the Bayesian p-value in sec. ??). Using this approach, data sets are simulated based on a posterior sample of the model parameters θ and some fit statistic for the simulated data sets, usually based on the discrepancy of the observed data from its expected values, is compared to that for the actual data. In most cases, whether Bayesian or frequentist, the main idea for assessing model fit is the same: We compare data sets from the model we're interested in with the data set we have in hand. If they appear to be consistent with one another, then our faith in the model increases, at least to some extent, and we say "the model fits."

To date, we are unaware of any goodness-of-fit applications based on likelihood analysis of SCR models. For Bayesian analysis of SCR models, there has not been a definitive or general proposal for a fit statistic or even a class of fit statistics, although a few specialized implementations of Bayesian p-values have been provided (??????). While 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

7290 For most SCR models, there are at least two distinct components of model fit, and we
 7291 propose to evaluate these two distinct components individually. First, we can ask, are the
 7292 data consistent with the *observation* model, conditional on the underlying point process?
 7293 We can evaluate this based on the encounter frequencies of individuals *conditional* on
 7294 (posterior samples of) the underlying point process $\mathbf{s}_1, \dots, \mathbf{s}_N$. We discuss some potential
 7295 fit statistics for addressing this in the next section. Second, we can evaluate whether the
 7296 data appear consistent with the *state* process model (i.e., the “uniformity” assumption of
 7297 the point process). For the simple model of independence and uniformity, this is similar to
 7298 the assumption of *complete spatial randomness* (CSR) which we consider in Sec. ?? below.
 7299 Actually, this is not strictly the assumption of CSR because of the binomial assumption
 7300 on N under data augmentation, so we instead use the term *spatial randomness*.

7301 8.4.1 Testing uniformity or spatial randomness

7302 Historically, especially in ecology, there has been an extraordinary amount of interest in
 7303 whether a realization of a point process indicates “complete spatial randomness,” i.e., that
 7304 the points are distributed uniformly and independently in space. Two good references for
 7305 such things are ?, Ch. 8 and ?¹. In the context of animal capture-recapture studies, the
 7306 spatial randomness hypothesis is manifestly false, purely on biological grounds. Typically
 7307 individuals will be clustered, or more regular (for territorial species), than expected under
 7308 spatial randomness and heterogeneous habitat will generate the appearance of clustering
 7309 even if individuals are distributed independently of one another. While we recommend
 7310 modeling spatial structure explicitly when possible (Chaps. ??, ??, ??), the uniformity
 7311 assumption may be an adequate description of data sets in some situations. Further, we
 7312 find that it is generally flexible enough to reflect non-uniform patterns in the data, because
 7313 we do observe some direct information about some of the point locations.

7314 The basic technical framework for evaluating the spatial randomness hypothesis is
 7315 based on counts of activity centers in cells or bins. For that we use any standard goodness-
 7316 of-fit test statistic, based on gridding (i.e., binning) the state-space of the point process into
 7317 $g = 1, 2, \dots, G$ cells or bins, and we tabulate $N_g \equiv N(\mathbf{x}_g)$ the number of activity centers
 7318 in bin g , centered at coordinate \mathbf{x}_g . Specifically, let $B(\mathbf{x})$ indicate a bin centered at
 7319 coordinate \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.
 7320 ??, we used the summaries $N(\mathbf{x})$ for producing density maps from MCMC output. Here,
 7321 we use them for constructing a fit statistic. We have used the Freeman-Tukey statistic of
 7322 this form:

$$T(\mathbf{N}, \theta) = \sum_g (\sqrt{N_g} - \sqrt{\mathbb{E}(N_g)})^2$$

7323 where $\mathbb{E}(N_g)$ is estimated by the mean bin count. An alternative conventional assessment
 7324 of fit is based on the following statistic: Conditional on N , the total number of activity
 7325 centers in the state-space \mathcal{S} , the bin counts N_g should have a binomial distribution. It

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.

² $I(arg)$ is the indicator function which evaluates to 1 if arg is true, otherwise 0

7326 will usually suffice to approximate the binomial cell counts by Poisson cell counts, in
 7327 which case we can use the classical “index-of-dispersion” test (? , p. 87), based on the
 7328 variance-to-mean ratio:

$$ID = (G - 1) * s^2 / \bar{N}$$

7329 where s^2 is the sample variance of the bin counts and \bar{N} is the sample mean. When the
 7330 point process realization is *observed*, as in classical point pattern modeling (but not in
 7331 SCR), this statistic has approximately a Chi-square distribution on $(G - 1)$ degrees-of-
 7332 freedom under the spatial randomness hypothesis. If $s^2/\bar{N} > 1$, clustering is suggested
 7333 whereas, $s^2/\bar{N} < 1$ suggests the point process is too regular.

7334 Whatever statistic we choose as our basis for assessing spatial randomness, *the im-*
 7335 *portant technical issue is that we don’t observe the point process and so the standard*
 7336 *statistics for evaluating spatial randomness cannot be computed directly. However, using*
 7337 *Bayesian analysis, we do have a posterior sample of the underlying point process and*
 7338 *so we suggest computing the posterior distribution of any statistic in a Bayesian p-value*
 7339 *framework. For a given posterior draw of all model parameters, N is known, based on the*
 7340 *value of the data augmentation variables z_i , and so we can obtain a posterior sample of*
 7341 *$N(\mathbf{x})$ by taking all of the output for MCMC iterations $m = 1, 2, \dots$, and doing this:*

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

7342 Thus, $N(\mathbf{x})^{(1)}, N(\mathbf{x})^{(2)}, \dots$, is the Markov chain for the derived parameter $N(\mathbf{x})$.

7343 In addition to computing the bin counts for each iteration of the MCMC algorithm,
 7344 at the same time we generate a realization of the activity centers \mathbf{s}_i under the spatial
 7345 randomness model, and we obtain bin counts for these “new” data, $\tilde{N}(\mathbf{x})$. For each of
 7346 the posterior samples – that of the real data, and that of the posterior simulated data, we
 7347 compute the fit-statistic. The fit statistic based on the actual data is:

$$T(\mathbf{N}, \theta) = \sum_x (\sqrt{N(\mathbf{x})} - \sqrt{\tilde{N}(\mathbf{x})})^2$$

7348 whereas the fit statistic based on a simulated realization of points under the spatial ran-
 7349 domness hypothesis is:

$$T(\tilde{\mathbf{N}}, \theta) = \sum_x (\sqrt{\tilde{N}(\mathbf{x})} - \sqrt{\tilde{N}(\mathbf{x})})^2$$

7350 And we compute the Bayesian p-value by tallying up the proportion of times that $T(\tilde{\mathbf{N}}, \theta)$
 7351 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
 7352 **SCRgof** in our package **scrbook** will do this, given the output from **JAGS** (see below).

7353 Sensitivity to bin size

7354 Evaluating fit based on bin counts in point process models are sensitive to the number
 7355 of bins (? , p. 87-88). This is related to the classical problem of fit testing for binary
 7356 regression because in a point process model, as the number of grid cells gets small, the
 7357 grid cell counts go to 0 or 1 and standard fit statistics (e.g., based on deviance or Pearson
 7358 residuals) are known not to be very useful. There is some good discussion of this in ?,

7359 Sec. 4.4.5. What it boils down to is, using the example of the Pearson residual statistic
 7360 considered by ?, the fit statistic is exactly a deterministic function of the sample size only,
 7361 which clearly should not be regarded as useful for model fit. This is why, in order to do a
 7362 check of model fit when you have a binary response, one must always aggregate the data
 7363 in some fashion. In the context of testing spatial randomness, computing the test statistic
 7364 we described above has us chop up the region \mathcal{S} into bins, and tally up N_g , the frequency
 7365 of activity centers in each bin g . Suppose that we choose the bin size to be extremely
 7366 small such that $\mathbb{E}(N_g)$ tends to N/G (N being the number of activity centers). Further,
 7367 N_g tends to a binary outcome. Therefore the fit statistic has N components that have
 7368 value $N_g = 1$, and it has $G - N$ components that have value $N_g = 0$. Therefore, the fit
 7369 statistic resembles:

$$T(\mathbf{N}, \theta) = \sum_{g \ni N_g=1}^N (1 - \sqrt{N/G})^2 + \sum_{g \ni N_g=0}^{G-N} (N/G)^2 = N(1 + (G - N)/G)$$

7370 (here \ni means “such that”). If G is huge relative to N , then we see that this tends to
 7371 about $2 * N$, which does not provide any meaningful assessment of model fit. So if you
 7372 look at this in the limit in which the bin counts become binary, the fit statistic loses all
 7373 its variability to the specific model used and is just a deterministic function of N . As a
 7374 practical matter, it probably makes sense to restrict the number of bins to *fewer* than the
 7375 number of observed individuals in the sample size. In typical SCR applications this will
 7376 therefore result, usually, in very large (and few) bins, and presumably not much power.

7377 There are some extensions that help resolve the issue of sensitivity to bin size. We can
 7378 construct fit statistics based not just on quadrat counts but also the neighboring quadrat
 7379 counts – this is the Greig-Smith method (?). In addition, there are a myriad of “distance
 7380 methods” for evaluating point process models, and we believe that many of these can
 7381 (and will) be adapted to SCR models. Again the main feature is that the point process
 7382 on which inference is focused is completely latent in SCR models – so this makes the fit
 7383 assessment slightly different than in classical point processes. That said, the methods
 7384 should be adaptable, e.g., in a Bayesian p-value kind of way.

7385 Sensitivity to state-space extent

7386 An issue that we have not investigated is that any model assessment that applies to a *latent*
 7387 point process is probably sensitive to the size of the state-space. As the size of the state-
 7388 space increases then the cell counts (far away from the data) *are* independent binomial
 7389 counts with constant density, and so we can overwhelm the fit statistic with extraneous
 7390 “data” simulated from the posterior, which is equal to the prior as we move away from the
 7391 data, and therefore uninformed by the observed data that live in the vicinity of the trap
 7392 array. Therefore we recommend computing these goodness-of-fit statistics in the vicinity
 7393 of the trap array only. Perhaps, as an ad hoc rule-of-thumb, less than the average trap
 7394 spacing from the rectangle enclosing the trap array. For example, if the average trap
 7395 spacing is, say, 10 km, then the bins used to obtain the observed and predicted activity
 7396 centers should not extend any further from the traps than 5 km. This should be a matter
 7397 of future research.

8.4.2 Assessing fit of the observation model

In evaluating the spatial randomness hypothesis, we could draw on well-established ideas from point process modeling. On the other hand, it is less clear how to approach goodness-of-fit evaluation of the observation model. For most SCR problems, we have a 3-dimensional data array of *binary* observations, y_{ijk} for individual i , trap j and sample occasion k . As discussed in the previous section, we need to construct fit statistics based on observed and expected frequencies that are aggregated in some fashion. In practice, the data will be too sparse to have much power, unless the data are highly aggregated. We recommend focusing on summary statistics that represent aggregated versions of y_{ijk} over 1 or 2 of the dimensions. We describe 3 such fit statistics below. We recognize that, depending on the model, some information about model fit will be lost by summarizing the data in this way. For example if there is a behavioral response and we aggregate over time to focus on the individual and trap level summaries then some information about lack of fit due to temporal structure in the data is lost.

Fit statistic 1: individual x trap frequencies We summarize the data by individual and trap-specific counts y_{ijk} aggregated over all sample occasions. Using standard “dot notation” to represent summed quantities, we express that as: $y_{ij\cdot} = \sum_{k=1}^K y_{ijk}$. Conditional on \mathbf{s}_i , the expected value under any encounter model is:

$$\mathbb{E}(y_{ij\cdot}) = p_{ij} K$$

(or K_j if the traps are operational for variable periods). If there is time-varying structure to the model, then expected values would have to be computed according to $\mathbb{E}(y_{ij\cdot}) = \sum_k p_{ijk}$. Then we can define a fit statistic from the Freeman-Tukey residuals according to:

$$T_1(\mathbf{y}, \theta) = \sum_i \sum_j (\sqrt{y_{ij\cdot}} - \sqrt{\mathbb{E}(y_{ij\cdot})})^2$$

where we use θ here to represent the collection of all parameters in the model. This is conditional on \mathbf{s} as well as on the data augmentation variables \mathbf{z} . We compute this statistic for *each* iteration of the MCMC algorithm for the observed data set and also for a new data set simulated from the posterior distribution, say $\hat{\mathbf{y}}$.

We could also use a similar fit statistic derived from summarizing over traps to obtain an $n_{\text{ind}} \times K$ matrix of count statistics. We imagine that either summary of the data will probably be too disaggregated (have mostly values of 0) in most practical settings to have much power.

Fit statistic 2: Individual encounter frequencies. SCR models represent a type of model for heterogeneous encounter probability, like model M_h , but with an explicit factor (space) that explains part of the heterogeneity. For model M_h , the individual encounter frequencies are the sufficient statistic for model parameters, and so it makes intuitive sense to provide some kind of omnibus fit assessment of the core heuristic that SCR model is adequately explaining the heterogeneity using a model M_h -like statistic based on individual encounter frequencies. So, we build a fit statistic based on the individual total encounters (?), $y_{i..} = \sum_j \sum_k y_{ijk}$. In addition, the expected value is a similar summary over traps and occasions: $\mathbb{E}(y_{i..}) = \sum_j \sum_k p_{ijk}$. Then, we define statistic T_2 according to:

$$T_2(\mathbf{y}, \theta) = \sum_i (\sqrt{y_{i..}} - \sqrt{\mathbb{E}(y_{i..})})^2$$

7437 We imagine this test statistic should provide an omnibus test of extra-binomial variation
 7438 and should therefore capture some effect of variable exposure to encounter of individuals,
 7439 although we have not carried out any evaluations of power under specific alternatives.
 7440 Obviously, in using this statistic, we lose information on departures from the model that
 7441 might only be trap- or time-specific.

7442 **Fit Statistic 3: Trap frequencies.** We construct an analogous statistic based
 7443 on aggregating over individuals and replicates to form trap encounter frequencies: $y_{.j} =$
 7444 $\sum_i \sum_k y_{ijk}$ (?) and the expected value is a similar summary over individuals and occa-
 7445 sions: $\mathbb{E}(y_{.j}) = \sum_i \sum_k p_{ijk}$. Then statistic T_3 is:

$$T_3(\mathbf{y}, \theta) = \sum_j (\sqrt{y_{.j}} - \sqrt{\mathbb{E}(y_{.j})})^2$$

7446 This seems like a sensible fit statistic because we can think of SCR models as spatial
 7447 models for counts (?). Therefore, we should seek models that provide good predictions of
 7448 the observable spatial data, which are the trap totals. In this context, it might even make
 7449 sense to pursue cross-validation based methods for model selection. Cross-validation is a
 7450 standard method of evaluating models such as in kriging or spline smoothing, so we could
 7451 as well develop such ideas based on the trap-specific frequencies.

7452 8.4.3 Does the SCR model fit the wolverine data?

7453 We use the ideas described in the previous section to evaluate goodness-of-fit of the SCR
 7454 model to the wolverine camera trapping data.

7455 We consider first whether the simple model of spatial randomness of the activity
 7456 centers is adequate. We think that the encounter model shouldn't have a large effect
 7457 on whether the spatial randomness assumption is adequate or not, so we fit "Model 0"
 7458 (in which parameters are *not* sex-specific) using an **R** script provided in the function
 7459 **wolvSCR0gof** which will default to fitting the model in **JAGS**. This is the same script as
 7460 **wolvSCR0ms** except that it saves the MCMC output for the activity centers **s** and the data
 7461 augmentation variables **z**, which are required in order to compute the Bayesian p-value
 7462 test of spatial randomness.

7463 The MCMC output is processed with the **R** function **SCRgof** which computes the test
 7464 of spatial randomness based on bin counts, using the Bayesian p-value calculation. The
 7465 function **SCRgof** requires a few things as inputs: (1) the output from a **BUGS** run (in
 7466 particular, the activity center coordinates and the data augmentation variables); (2) the
 7467 number of bins to create for computing spatial frequencies of activity centers; (3) the trap
 7468 locations and, (4) the buffer around the trap array to use in computing the bin counts.
 7469 This buffer could be that used in defining the state-space for the model fitting, but we
 7470 think it should be relatively tighter to the trap array than the state-space used in model-
 7471 fitting. For the wolverine analysis, where we're using 10-km grid cells (1 unit = 10 km)
 7472 and a 20 km buffer for model fitting, we'll use a state-space buffer of 0.4 units (4 km) for
 7473 computing the fit statistic. The **R** code to fit the model and obtain the goodness-of-fit
 7474 result is as follows:

```
7475 > wolv1 <- wolvSCR0gof(nb=1000,ni=6000,buffer=2,M=200,model=0)
7476
```

```

7477 > bugsout <- wolv1$BUGSoutput$sims.list
7478
7479 > traplocs <- wolverine$wtraps[,2:3]
7480 > traplocs[,1] <- traplocs[,1] - min(traplocs[,1])
7481 > traplocs[,2] <- traplocs[,2] - min(traplocs[,2])
7482 > traplocs <- traplocs/10000
7483
7484 > set.seed(2013) # set seed so Bayesian p-value is the same each time
7485
7486 > SCRgof(bugsout,5,5,traplocs=traplocs,buffer=.4)
7487
7488 Cluster index observed: 1.099822
7489 Cluster index simulated: 1.000453
7490 P-value index of dispersion: 0.408
7491 P-value2 freeman-tukey: 0.6842667

```

7492 The output produced by `SCRgof` is the index of dispersion based on the ratio of the
 7493 variance to the mean (see above), which is computed as the posterior mean index of
 7494 dispersion for the latent point process, and also the average value for simulated data. If
 7495 this value is > 1 then clustering is suggested, which we see a (very) minor amount of
 7496 evidence for here. Two Bayesian p-values are produced: the first is based on the cluster
 7497 index, and the 2nd is based on the Freeman-Tukey statistic calculated as described in
 7498 Sec. ???. Because our p-values aren't close to 0 or 1, we judge that the model of spatial
 7499 randomness provides an adequate fit to the data. You can verify that a similar result is
 7500 obtained if we use the model with fully sex-specific parameters (Model 4).

7501 Next, we did a Bayesian p-value analysis of the observation component of the model,
 7502 using the 3 fit statistics described in Sec. ???. These statistics can be calculated as part of
 7503 the **BUGS** model specification or by post-processing the MCMC output returned from a
 7504 **BUGS** run. The **R** script `wolvSCR0gof` contains the relevant calculations. For example, to
 7505 compute fit statistic 1, we have to add some commands to the **BUGS** model specification
 7506 such as this (note: this is only a fraction of the model specification):

```

7507 .....
7508 for(j in 1:ntraps){
7509   mu[i,j] <- w[i]*p[i,j]
7510
7511   y[i,j] ~ dbin(mu[i,j],K[j])
7512   ynew[i,j] ~ dbin(mu[i,j],K[j])
7513
7514   err[i,j] <- pow(pow(y[i,j],.5) - pow(K[j]*mu[i,j],.5),2)
7515   errnew[i,j] <- pow(pow(ynew[i,j],.5) - pow(K[j]*mu[i,j],.5),2)
7516 }
7517
7518 T1obs <- sum(err[,])
7519 T1new <- sum(errnew[,])
7520 .....

```

7521 Similar calculations are carried out to obtain the posterior samples of test statistics 2

7522 (individual totals) and 3 (trap totals). For the wolverine data, the Bayesian p-value
 7523 calculations produce:

```
7524 > mean(wolv1$BUGSoutput$sims.list$T1new>wolv1$BUGSoutput$sims.list$T1obs)
7525 [1] 0
7526
7527 > mean(wolv1$BUGSoutput$sims.list$T2new>wolv1$BUGSoutput$sims.list$T2obs)
7528 [1] 0.17
7529
7530 > mean(wolv1$BUGSoutput$sims.list$T3new>wolv1$BUGSoutput$sims.list$T3obs)
7531 [1] 0.02066667
```

7532 Based on statistic T_2 , we might conclude that the model is adequate for explaining
 7533 individual heterogeneity although the other two statistics suggest a general lack of fit of
 7534 the observation model. A similar result is obtained using the fully sex-specific model. We
 7535 note that one individual was captured 8 times in one trap, which is pretty extreme under
 7536 a model which assumes independent Bernoulli trials. We summarize that the trap-counts
 7537 simply are not well-explained by this model.

7538 In attempt to resolve this problem, we extended the model to include a local (trap-
 7539 specific) behavioral response (following ?) which can be fitted using the sample **R** script
 7540 **wolvSCRMb**. To fit a model using **WinBUGS**, and then compute the Bayesian p-values
 7541 we do this:

```
7542 > wolv.Mb <- wolvSCRMb(nb=1000,ni=6000,buffer=2,M=200)
7543
7544 > mean(wolv.Mb$sims.list$T1new>wolv.Mb$sims.list$T1obs)
7545 [1] 0.9666667
7546
7547 > mean(wolv.Mb$sims.list$T2new>wolv.Mb$sims.list$T2obs)
7548 [1] 0.3644667
7549
7550 > mean(wolv.Mb$sims.list$T3new>wolv.Mb$sims.list$T3obs)
7551 [1] 0.4990667
```

7552 Given that this model seems to fit better, we might prefer reporting estimates under
 7553 this model, which we do in Table ???. (the behavioral response parameter is labeled α_2
 7554 in the table). Estimated density is about 1 individual higher per 1000 km² compared
 7555 with the various models that lack a behavioral response. It might be useful to try these
 7556 fit assessment exercises using the habitat mask as described in Sec. ???. That takes an
 7557 extremely long time to run in **BUGS** though, especially for the behavioral response model.

8.5 QUANTIFYING LACK-OF-FIT AND REMEDIATION

7558 ? used a strategy for assessing model fit in dynamic occupancy models (?) similar to
 7559 that which we suggested above. They constructed a fit statistic based on aggregating the
 7560 data over replicate samples (k), to obtain the total detections per site i and year j . They
 7561 used a Bayesian p-value analysis based on a Chi-squared test statistic (also see ?, Chapt.

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

7562 12). Their analysis suggested a model that didn't fit, and, so they computed the "lack-
 7563 of-fit ratio" (see ?, Sec. 12.3) – the ratio of the fit statistic computed for the actual data
 7564 to that of the replicate data sets. They interpret this analogous to the over-dispersion
 7565 coefficient in generalized linear models (?), usually called the c-hat statistic in capture-
 7566 recapture literature (see ?, Chapt. 5). ? reported the lack-of-fit ratio for their model to
 7567 be 1.14 which suggests a minor lack-of-fit, compared to perfect data having a value of 1,
 7568 because the posterior standard deviations will be too small by a factor of $\sqrt{1.14} = 1.07$.
 7569 In classical capture-recapture applications of goodness-of-fit assessment, inference for non-
 7570 fitting models is dealt with by inflating the resulting SEs (of the non-fitting model), by
 7571 the square-root of c-hat. We believe that these ideas related to quantifying lack-of-fit and
 7572 understanding its effect could also be applied to SCR models, although we have not yet
 7573 explored this.

8.6 SUMMARY AND OUTLOOK

7574 In this chapter, we offered some general strategies for model selection and model checking,
 7575 or assessment of model fit. We think the strategies we outlined for model selection are fairly
 7576 standard and can be effectively applied to many SCR modeling problems. Some technical
 7577 issues of Bayesian analysis need to be addressed (in general) before Bayesian methods are
 7578 more generally useful and accessible. For one thing, Bayesian model selection based on
 7579 the indicator variable approach of ? can be tediously slow even for small data sets, and so
 7580 improved computation will improve our ability to do Bayesian model selection in practical
 7581 situations. Also, and most importantly, sensitivity to prior distributions is an important
 7582 issue. Further research and practice might identify preferred prior configurations for SCR

7583 that provide a good calibration in relevant model selection problems. Finally, we believe
7584 that cross-validation should prove to be a useful method in model assessment and selection,
7585 as SCR models are a form of spatial model of counts, and so it is natural to pick models
7586 that predict the observable spatial counts (i.e., at trap locations) well.

7587 For Bayesian model assessment, or goodness-of-fit checking, we suggested a framework
7588 based on independent testing of the spatial model of independence and uniformity, and
7589 testing fit of the observation model conditional on the underlying point process. These
7590 ideas are based on mostly *ad hoc* attempts in a number of published applications (????,
7591 e.g.,). While we think this general strategy should be fruitful, we know of no studies on
7592 the power to detect various model departures, and so the ideas should be viewed as exper-
7593 imental. We have not discussed assessment of model fit for SCR models using likelihood
7594 methods, although we imagine that standard bootstrapping ideas should be effective, per-
7595 haps based on the fit statistics (or similar ones) we suggested here for computing Bayesian
7596 p-values.

7597 Clearly there is much research to be done on assessment of model fit in SCR models.
7598 For testing the spatial randomness hypothesis, we used a classical approach based on
7599 count frequencies, in which point locations are put into spatial bins. Other approaches
7600 from spatial point process modeling should be pursued including nearest-neighbor methods
7601 or distance-based methods. In addition, studies to evaluate the power to detect relevant
7602 departures from the standard assumptions, and the robustness of inferences about N or
7603 density, need to be conducted. If the spatial randomness model appears inadequate, it
7604 is possible to fit models that allow for a non-uniform distribution of points (see Chapt.
7605 ??) and even point process models that allow for interactions among points (?). On the
7606 other hand, we expect that most of these Bayesian p-value tests will have low power
7607 in typical data sets consisting of a few to a few dozen individuals. As such, failure to
7608 detect a lack of fit may not be that meaningful. But, on the other hand, it may not
7609 make a difference in terms of density estimates either. We think inference about density
7610 should be relatively insensitive to departures from spatial randomness, because we get to
7611 observe direct information on some component of the population, component of density is
7612 *observed*. For those activity centers, the assumed model of the point process should exert
7613 little influence on the placement of the activity centers. Conversely, as is the case with
7614 classical closed population models (???), inferences may be somewhat more sensitive to
7615 bad-fitting models for the observation process.

9

7616

7617

7618 ALTERNATIVE OBSERVATION MODELS

7619 In previous chapters we considered various models of *encounter probability*, both in terms
7620 of parametric functions of distance and also a myriad of covariate models (Chapt. ?? and
7621 elsewhere). However, we have so far only considered a specific probability model for the
7622 observations (we'll call this the "observation model") – the Bernoulli encounter process
7623 model which, in **secr**, is the *proximity detector* model. This assumes that individual and
7624 trap-specific encounters are independent Bernoulli trials.

7625 In this chapter, we focus on developing additional observation models. The observa-
7626 tion model could be thought of as being determined by the type of device – or the type
7627 of "detector" using the terminology of **secr** (?). We consider models that apply when
7628 observations are not binary and, in some cases, that do not require independence of the
7629 observations. We present models when the data are encounter *frequencies*, based on the
7630 Poisson distribution, and observation models based on the multinomial distribution. For
7631 example, if sampling devices can detect an individual some arbitrary number of times dur-
7632 ing an interval, then it is natural to consider observation models for encounter frequencies,
7633 such as the Poisson model. Another type of encounter device is the "multi-catch" device
7634 (?) which is a physical device that can capture and hold an arbitrary number of individ-
7635 uals. A typical example is a mist-net for birds (?). It is natural to regard observations
7636 from these kinds of studies as independent multinomial observations. A related type of
7637 device that produces *dependent* multinomial observations are the so-called *single-catch*
7638 traps (??). The canonical example are small-mammal live traps which catch and hold a
7639 single individual. Competition among individuals for traps induces a complex dependence
7640 structure among individual encounters. To date, no formal inference framework has been
7641 devised for this method although it stands to reason that the independent multinomial
7642 model should be a good approximation in some situations (?). We analyze a number of
7643 examples of these different observation models using **JAGS** and also the **R** package **secr**
7644 (?).

9.1 POISSON OBSERVATION MODEL

The models we analyze in Chapt. ?? 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 (?).

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. ??, we now seek to model the expected value of the observation (which was p_{ij} in Chapt. ??) 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. ???. This Poisson SCR model is therefore a type of Poisson generalized linear mixed model (GLMM).

We can accommodate covariates at the level of individual-, trap- or sample occasion by including them on the baseline encounter rate parameter λ_0 . For example, if C_j is some covariate that depends on trap only, then we express the relationship between λ_0 and C_j as:

$$\log(\lambda_{0,ijk}) = \alpha_0 + \alpha_2 C_j$$

and therefore covariates on the logarithm of baseline encounter probability appear also as linear effects on λ_{ij} . In general, covariates might also affect the coefficient on the distance

term (α_1) (e.g., sex of individual). We don't get into too much discussion of general covariate models here, but we covered them in some detail in both Chaps. ?? and ??.

For models in which we do not have covariates that vary across the sample occasions k , we can aggregate the observed data by the property of compound additivity of the Poisson distribution (if x and y are *iid* Poisson with mean λ then $x + y$ is Poisson with mean 2λ). Therefore,

$$y_{ij} = \left(\sum_{k=1}^K y_{ijk} \right) = \text{Poisson}(K\lambda_0 k(\mathbf{x}_j, \mathbf{s}_i))$$

We see that K and λ_0 serve the same role as affecting the base encounter rate. Since the observation model is the same, probabilistically speaking, for all values of K , evidently we need only $K = 1$ "survey" from which to estimate model parameters (?). We know this intuitively, as sampling by multiple traps serves as replication in SCR models. This has great practical relevance to the conduct of capture-recapture studies and the use of SCR models. For example, if individuality is obtained by genetic information from scat sampling, one should only have to carry out a single spatial sampling of the study area. However, one must be certain that sufficient spatial recaptures will be obtained so that effective estimation is possible.

9.1.1 Poisson model of space usage

It is natural to interpret the Poisson encounter model as a model of space usage resulting from movement of individuals about their home range (Sec. ??). Imagine we have perfect samplers in every pixel of the landscape so that whenever an individual moves from one pixel to another, we can record it. Let m_{ij} be the number of times individual i was recorded in pixel j (i.e., it selected or used pixel j). Then, we might think of the Poisson model for the observed *use* frequencies:

$$m_{ij} \sim \text{Poisson}(\lambda_0 k(\mathbf{x}_j, \mathbf{s}_i))$$

where λ_0 is related to the baseline movement rate of the animal (how often it moves). This model of space usage gives rise to the standard resource selection function (RSF) models (see Chapt. ??). But now suppose our samplers are not perfect but, rather, record only a fraction of the resulting visits. A sensible model is

$$y_{ij}|m_{ij} \sim \text{Binomial}(m_{ij}, p).$$

The marginal distribution of y_{ij} is:

$$y_{ij} \sim \text{Poisson}(p_0 k(\mathbf{x}_j, \mathbf{s}_i)).$$

where p_0 is a composite of the movement rate and conditional detection probability p . Therefore, we see that encounters accumulate in proportion to the frequency of outcomes of an individual using space (or "selecting resources").

We introduced an interpretation of SCR models in terms of movement and space usage in Sec. ??, and it is one of the main underlying concepts of SCR models that is not present in ordinary capture-recapture models. As we noted there, the underlying model of space usage is only as complex as the encounter probability model which has been, so far in this book, only symmetric and stationary (does not vary in space). We generalize this model of space usage substantially in Chapt. ??.

9.1.2 Poisson relationship to the Bernoulli model

7720 There is a sense in which the Poisson and Bernoulli models can be viewed as consistent with
 7721 one another. Note that under the Poisson model, the relationship between the expected
 7722 count and the probability of counting “at least 1”, is given by

$$\Pr(y > 0) = 1 - \exp(-\lambda) \quad (9.1.2)$$

7723 where $\mathbb{E}(y) = \lambda$. Therefore, if we equate the event “encountered” with the event that the
 7724 individual was captured at least 1 time under the Poisson model, i.e., $y > 0$, then it would
 7725 be natural to set $p_{ij} = \Pr(y > 0)$ according to Eq. ???. That is, we can use Eq. ?? as the
 7726 model for encounter probability for binary observations. This is the “hazard rate” model
 7727 in distance sampling.

7728 In fact, as λ gets small, the Poisson model is a close approximation to the Bernoulli
 7729 model in the sense that outcomes concentrate on $\{0, 1\}$, i.e., $\Pr(y \in \{0, 1\}) \rightarrow 1$ as $\lambda \rightarrow 0$.
 7730 Indeed, under the Poisson model, $\Pr(y > 0) \rightarrow \lambda$ for small values of λ . This phenomenon
 7731 is shown in Fig. ?? where the left panel shows a plot of $\lambda_{ij} = \lambda_0 k(\mathbf{x}_j, \mathbf{s}_i)$ vs. distance and
 7732 superimposed on that is a plot of $p_{ij} = 1 - \exp(-\lambda_{ij})$ vs. distance, for values $\lambda_0 = 0.1$
 7733 and $\sigma = 1$, and the right panel shows a plot of $\Pr(y > 0)$ vs. $\mathbb{E}(y)$. We see that the two
 7734 quantities are practically indistinguishable. This is convenient in some cases because the
 7735 Poisson model might be more tractable to fit (or even vice versa). For an example, see
 7736 the models described in Chapt. ??, and we also consider another case in Sec. ?? below.
 7737 To evaluate the closeness of the approximation, you can use the following R commands
 7738 which we used to produce Fig. ??:

```
7739 > x <- seq(0.001, 5, , 200)
7740 > lam0 <- .1
7741 > sigma <- 1
7742 > lam <- lam0*exp(-x**/(2*sigma*sigma))
7743
7744 > par(mfrow=c(1,2))
7745 > p1 <- 1-exp(-lam)
7746 > plot(x, lam, ylab="E[y] or Pr(y>0)", xlab="distance", type="l", lwd=2)
7747 > lines(x,p1,lwd=2,col="red")
7748 > plot(lam, p1, xlab="E[y]", ylab="Pr(y>0)", type="l", lwd=2)
7749 > abline(0,1,col="red")
```

7750 To summarize, if y is Poisson then, as λ gets small,

$$\begin{aligned} \Pr(y > 0) &\approx \mathbb{E}(y) \\ 1 - \exp(-\lambda_0 k(\mathbf{x}, \mathbf{s})) &\approx \lambda_0 k(\mathbf{x}, \mathbf{s}) \end{aligned} \quad (9.1.3)$$

7751 What all of this suggests it that if we have very few observations > 1 in our SCR data
 7752 set, then we won’t lose much information by using the Bernoulli model. On the other
 7753 hand, the Poisson model may have some advantages in terms of analytic or numerical
 7754 tractability in some cases. Further, this approximation explains the close correspondence
 7755 we have found between these two versions of the Gaussian encounter probability model
 7756 (Sec. ??). Namely, the Gaussian hazard model and the Gaussian encounter probability
 7757 model are close approximations because $1 - \exp(-\lambda) \approx \lambda$ if λ is small.

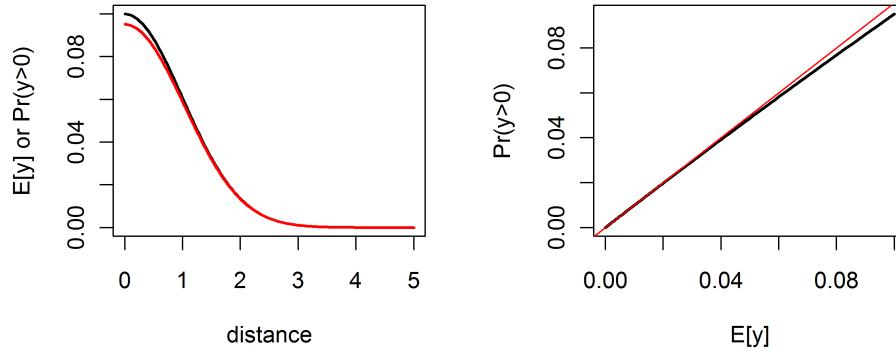


Figure 9.1. Poisson approximation to the binomial. As the Poisson mean approaches 0, then $\Pr(y > 0)$ under the Poisson model approaches λ and therefore $y \sim \text{Poisson}(\lambda)$ is well-approximated by a Bernoulli model with parameter λ .

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. ???. This is equivalent to the complementary log-log link model, or the “Gaussian hazard” as we called it in Chapt. ???:

$$\text{cloglog}(p_{ij}) = \log(\lambda_0) + \log(k(\mathbf{x}, \mathbf{s}))$$

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 (???) but other models for excess-Poisson variation are possible. For example, we might add a normally distributed random effect to the linear predictor (?).

As a general rule we favor the Bernoulli observation model even if our sampling scheme produces encounter frequencies. The main reason is that, with frequency data, we are

7777 forced to confront a model choice problem (i.e., Poisson, negative binomial, log-normal
 7778 mixture) that is wholly unrelated to the fundamental space usage process that underlies
 7779 the genesis of many types of SCR data. Repeated encounters over short time intervals are
 7780 not likely to be the result of independent encounter events. E.g., an individual moving
 7781 back and forth in front of a camera yields a cluster of observations that is not informative
 7782 about the underlying spatial structure of the population. Similarly in scat surveys dogs
 7783 are used to locate scats which are processed in the lab for individuality (???). The process
 7784 of local scat deposition is not strictly the outcome of movement or space usage but rather
 7785 the outcome of complex behavioral considerations as well as dependence in detection of
 7786 scat by dogs. For example, dogs find (or smell) one scat and then are more likely to find
 7787 one or more nearby ones, if present, or they get into a den or latrine area and find many
 7788 scats. The additional assumption required to model variation in observed frequencies (i.e.,
 7789 conditional on location) provides relatively no information about space usage and density,
 7790 and we feel that the model selection issue should therefore be avoided.

7791 To elaborate on this, we suppose that an individual with activity center \mathbf{s} visits
 7792 a particular pixel \mathbf{x} with some probability $p(\mathbf{x}, \mathbf{s})$, and then, once there, deposits a
 7793 number of scat, or visits a camera some number of times with frequency $y(\mathbf{x}, \mathbf{s}) \geq 0$.
 7794 We describe the outcome of this movement/usage process with a two-level hierarchi-
 7795 cal model of the form: $[y|w][w|p(\mathbf{x}, \mathbf{s})]$ where $w(\mathbf{x}, \mathbf{s})$ is a binary variable that indicates
 7796 whether the individual with activity center \mathbf{s} used pixel \mathbf{x} during some interval, and let
 7797 $w(\mathbf{x}, \mathbf{s}) \sim \text{Bernoulli}(p(\mathbf{x}, \mathbf{s}))$. If we suppose encounter frequency y is independent of \mathbf{x} and
 7798 \mathbf{s} conditional on the use variable w , then we see that the model for y (amount of use) does
 7799 not depend on \mathbf{s} .

7800 9.1.4 Analysis of the Poisson SCR model in BUGS

7801 We consider the simplest possible model here in which we have no covariates that vary
 7802 over sample occasions $k = 1, 2, \dots, K$ so that we work with the aggregated individual-
 7803 and trap-specific encounters:

$$y_{ij} = (\sum_{k=1}^K y_{ijk}) = \text{Poisson}(K\lambda_0 k(\mathbf{x}_j, \mathbf{s}_i))$$

7804 and we consider the bivariate normal form of $k(\mathbf{x}, \mathbf{s})$:

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

7805 so that

$$\log(\lambda_{ij}) = \alpha_0 - \alpha_1 d(\mathbf{x}_j, \mathbf{s}_i)^2$$

7806 where $\alpha_0 = \log(\lambda_0)$ and $\alpha_1 = 1/(2\sigma^2)$.

7807 As usual, we approach Bayesian analysis of these models using data augmentation
 7808 (Sec. ??). Under data augmentation, we introduce a collection of all-zero encounter
 7809 histories to bring the total size of the data set up to M , and a corresponding set of data
 7810 augmentation variables $z_i \sim \text{Bern}(\psi)$. Then the observation model is specified conditional
 7811 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. ?? 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. ?? (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`:

```
7821 ##  
7822 ## S =activity centers and traplocs defined as in simSCR0()  
7823 ##  
7824 ## Compute distance between activity centers and traps:  
7825 > D <- e2dist(S,traplocs)  
7826  
7827 ## Define parameter values:  
7828 > alpha0 <- -2.5  
7829 > sigma <- 0.5  
7830 > alpha1 <- 1/(2*sigma*sigma)  
7831  
7832 ## Encounter probability model:  
7833 > muy <- exp(alpha0)*exp(-alpha1*D*D)  
7834  
7835 ## Now generate the encounters of every individual in every trap  
7836 > Y <-matrix(NA,nrow=N,ncol=ntraps)  
7837 > for(i in 1:nrow(Y)){  
7838   Y[i,] <- rpois(ntraps,K*muy[i,])  
7839 }
```

We modified our simulation code from Chapt. ?? 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:

```
7846 ## Simulate data and extract data elements  
7847 ##  
7848 > data <- simPoissonSCR(discard0=TRUE,rnd=2013)  
7849 > y <- data$Y  
7850 > nind <- nrow(y)  
7851 > X <- data$traplocs  
7852 > K <- data$K  
7853 > J <- nrow(X)
```

```

7854 > xlim <- data$xlim
7855 > ylim <- data$ylim
7856
7857 ## Data augmentation
7858 > M <- 200
7859 > y <- rbind(y,matrix(0,nrow=M-nind,ncol=ncol(y)))
7860 > z <- c(rep(1,nind),rep(0,M-nind))

```

7861 The process for fitting the model in **WinBUGS** or **JAGS** is identical to what we've
 7862 done previously in Chapt. ???. In particular, we set up some starting values, package
 7863 the data and inits, identify the parameters to be monitored, and then send everything
 7864 off to our MCMC engine. Here it all is for fitting the Poisson observation model (these
 7865 commands are shown in the help file for `simPoissonSCR`):

```

7866 ## Starting values for activity centers
7867 ##
7868 > sst <- X[sample(1:J,M,replace=TRUE),]
7869 > for(i in 1:nind){
7870   if(sum(y[i,])==0) next
7871   sst[i,1] <- mean( X[y[i,>0,1] )
7872   sst[i,2] <- mean( X[y[i,>0,2] )
7873 }
7874 ## Dithered a little bit from trap locations
7875 > sst <- sst + runif(nrow(sst)*2,0,1)/8
7876 > data <- list (y=y,X=X,K=K,M=M,J=J,xlim=xlim,ylim=ylim)
7877 > inits <- function(){
7878   list (alpha0=rnorm(1,-2,.4),alpha1=runif(1,1,2),s=sst,z=z,psi=.5)
7879 }
7880 > parameters <- c("alpha0","alpha1","N","D")

```

7881 Next, we write the **BUGS** model to an external file:

```

7882 > cat("
7883 model{
7884   alpha0 ~ dnorm(0,.1)
7885   alpha1 ~ dnorm(0,.1)
7886   psi ~ dunif(0,1)
7887
7888   for(i in 1:M){
7889     z[i] ~ dbern(psi)
7890     s[i,1] ~ dunif(xlim[1],xlim[2])
7891     s[i,2] ~ dunif(ylim[1],ylim[2])
7892     for(j in 1:J){
7893       d[i,j] <- pow(pow(s[i,1]-X[j,1],2) + pow(s[i,2]-X[j,2],2),0.5)
7894       y[i,j] ~ dpois(lam[i,j])
7895       lam[i,j] <- z[i]*K*exp(alpha0)*exp(- alpha1*d[i,j]*d[i,j])
7896     }
7897   }

```

```

7898 N <- sum(z[])
7899 D <- N/64
7900 }
7901 ",file = "SCR-Poisson.txt")

```

7902 To fit the model we execute **bugs** in the usual way:

```

7903 > library(R2WinBUGS)
7904 > out1 <- bugs (data, inits, parameters, "SCR-Poisson.txt", n.thin=1,
7905   n.chains=3,n.burnin=1000,n.iter=2000,working.dir=getwd(),
7906   debug=TRUE)

```

7907 Or, using **JAGS** via **rjags** we would do something like this:

```

7908 > library(rjags)
7909 > jm <- jags.model("SCR-Poisson.txt", data=data, inits=inits,
7910   n.chains=3, n.adapt=1000)
7911 > out2 <- coda.samples(jm, parameters, n.iter=1000, thin=1)

```

7912 Summarizing the output from the **WinBUGS** run produces the following:

```

7913 > print(out1,digits=2)
7914 Inference for Bugs model at "SCR-Poisson.txt", fit using WinBUGS,
7915 3 chains, each with 2000 iterations (first 1000 discarded)
7916 n.sims = 3000 iterations saved
7917      mean    sd   2.5%   25%   50%   75% 97.5% Rhat n.eff
7918 alpha0   -2.57  0.19  -2.95  -2.69  -2.57  -2.44  -2.19 1.00  2600
7919 alpha1    2.34  0.36   1.69   2.08   2.32   2.57   3.12 1.00  3000
7920 N        114.13 15.25  87.97 103.00 113.00 124.00 147.00 1.01  370
7921 D        1.78  0.24   1.37   1.61   1.77   1.94   2.30 1.01  370
7922 deviance 329.95 21.92 290.00 314.20 329.50 344.40 375.80 1.00  1700
7923 ...
7924 [..some output deleted..]
7925 ...

```

7926 9.1.6 Analysis of the wolverine study data

7927 We reanalyzed the data from the wolverine camera trapping study that were first introduced in Sec. ???. 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 ???. 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. ??, except that we specify the `detector="count"` option when the traps object is created. The set-up proceeds as follows:

```

7939 > library(secr)
7940 > library(scrbook)
7941 > data(wolverine)
7942
7943 > traps <- as.matrix(wolverine$wtraps)
7944 > dimnames(traps) <- list(NULL,c("trapID","x","y",paste("day",1:165,sep="")))
7945 > traps1 <- as.data.frame(traps[,1:3])
7946 > 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. ??.

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 (?). 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 type of model in which

the 2nd assumption is violated are the “single catch” trap systems which we address in Sec. ?? below.

In this case we assume the observation \mathbf{y}_{ik} for individual i during sample occasion k is a multinomial observation which consists of a sequence of 0’s and a single 1 indicating the trap of capture, or “not captured”. For the “not captured” event we define an additional outcome, by convention element $J + 1$ of the vector. As an example, if we capture an individual in trap 2 during some occasion of a study involving $J = 6$ traps. Then, the multinomial observation has length $J+1 = 7$, and the observation is $\mathbf{y}_i = (0, 1, 0, 0, 0, 0, 0)$. An individual not captured at all would have the observation vector $(0, 0, 0, 0, 0, 0, 1)$. If we sample for 5 occasions in all and the individual is also caught in trap 4 during occasion 3, but otherwise uncaptured, then the 5 encounter observations for that individual are as follows:

occassion	trap						"not captured"
	1	2	3	4	5	6	
1	0	1	0	0	0	0	0
2	0	0	0	0	0	0	1
3	0	0	0	1	0	0	0
4	0	0	0	0	0	0	1
5	0	0	0	0	0	0	1

Statistically we regard the *rows* of this data matrix as *independent* multinomial trials.

Analogous to our previous Bernoulli and Poisson models, we seek to construct the multinomial cell probabilities for each individual, as a function of *where* that individual lives, through its center of activity \mathbf{s} . Thus we suppose that

$$\mathbf{y}_{ik} | \mathbf{s}_i \sim \text{Multinomial}(1, \boldsymbol{\pi}(\mathbf{s}_i)) \quad (9.2.1)$$

where $\boldsymbol{\pi}(\mathbf{s}_i)$ is a vector of length $J + 1$, where $\pi_{i,J+1}$, the last cell, corresponds to the probability of the event “not captured”. Now we have to construct these cell probabilities in some meaningful way that depends on each individual’s \mathbf{s} . We use the standard multinomial logit with distance as a covariate:

$$\pi_{ij} = \frac{\exp(\alpha_0 - \alpha_1 d_{ij})}{1 + \sum_j \exp(\alpha_0 - \alpha_1 d_{ij})}$$

for $j = 1, 2, \dots, J$ and, for $J + 1$, i.e., “not captured”,

$$\pi_{i,(J+1)} = \frac{\exp(0)}{1 + \sum_j \exp(\alpha_0 - \alpha_1 d_{ij})}$$

or, more commonly, we use d_{ij}^2 to correspond to our Gaussian kernel model for encounter probability. Whatever function of distance we use in the construction of multinomial probabilities will have a direct correspondence to the standard encounter probability models we used in the Bernoulli or Poisson models as well (see Sec. ??).

It is convenient to express these multinomial models short-hand as follows, e.g., for the Gaussian encounter probability model:

$$\text{mlogit}(\pi_{ij}) = \alpha_0 - \alpha_1 d_{ij}^2$$

7997 In this way we can refer to models with covariates in a more concise way. For example, a
 7998 model with a trap-specific covariate, say C_j , is:

$$\text{mlogit}(\pi_{ij}) = \alpha_0 - \alpha_1 d_{ij}^2 + \alpha_2 C_j$$

7999 or we could include occasion-specific covariates too, such as behavioral response.

8000 A statistically equivalent distribution to the multinomial is the *categorical* distribution.

8001 If \mathbf{y} is a multinomial trial with probabilities $\boldsymbol{\pi}$ than the *position* of the non-zero element of
 8002 \mathbf{y} is a categorical random variable with probabilities $\boldsymbol{\pi}$. We express this for SCR models
 8003 as

$$\mathbf{y}|\mathbf{s} \sim \text{Categorical}(\boldsymbol{\pi}(\mathbf{s}))$$

8004 In the SCR context, the categorical version of the multinomial trial corresponds to the
 8005 *trap of capture*. Using our example above with 6 traps then we could as well say y_{ik} is a
 8006 categorical random variable with possible outcomes (1, 2, 3, 4, 5, 6, 7) where outcome $y = 7$
 8007 corresponds to “not captured.” Obviously, how this is organized or labeled is completely
 8008 irrelevant, although it is convenient to use the integers 1 to $(J + 1)$ where $J + 1$ is the
 8009 event not captured. Therefore, for our illustration in the previous table, $y_{i1} = 2$, $y_{i2} = 7$,
 8010 $y_{i3} = 4$ and so on.

8011 For simulating and fitting data in the **BUGS** engines we will typically use the categori-
 8012 cal representation of the model because it is somewhat more convenient. We have found
 8013 that fitting multinomial models in **WinBUGS** is less efficient than **JAGS** (?), which we
 8014 use in the subsequent examples involving multinomial observation models.

8015 9.2.1 Multinomial resource selection models

8016 The multinomial probabilities in Eq. ?? look similar to the multinomial resource selec-
 8017 tion function (RSF) model for telemetry data (??). This suggests how we might model
 8018 landscape or habitat covariates using such methods – i.e., by including them as explicit
 8019 covariates in a larger multinomial model for “use” – which, if we take the product of use
 8020 with encounter, produces a model for the observable encounter data. This leads naturally
 8021 to the development of models that integrate RSF data from telemetry studies with SCR
 8022 data (?), which is the topic of Chapt. ??.

8023 9.2.2 Simulating data and analysis using JAGS

8024 We’re going to show the nugget of a simulation function which is used in the function
 8025 **simMnSCR** found in the **R** package **scrbook**. The first lines of the following **R** code make
 8026 use of some things that you need to define, but we omit them here (e.g., **xlim**, **ylim** are
 8027 the boundaries of the state-space, **N** is the population size, etc.):

```
8028 ##
8029 ## Simulate random activity centers:
8030 ##      (first define N, xlim, ylim, etc..)
8031 ##
8032 > S <- cbind(runif(N,xlim[1],xlim[2]),runif(N,ylim[1],ylim[2]))
8033
8034 ## Distance from each individual to each trap
```

```

8035 > D <- e2dist(S,traplocs)
8036
8037 ## Set parameter values
8038 > sigma <- 0.5
8039 > alpha0 <- -1
8040 > alpha1 <- -1/(2*sigma*sigma)
8041
8042 ## make an empty data matrix and fill it up with data
8043 > Ycat <- matrix(NA,nrow=N,ncol=K)
8044 > for(i in 1:N){
8045   for(k in 1:K){
8046     lp <- alpha0 + alpha1*D[i,]*D[i,]
8047     cp <- exp(c(lp,0))
8048     cp <- cp/sum(cp)
8049     Ycat[i,k] <- sample(1:(ntraps+1),1,prob=cp)
8050   }
8051 }
```

8052 We save the data in the matrix `Ycat` to clarify that it is the categorical observation
 8053 representing “trap of capture”. The matrix `Ycat` here has the maximal dimension N
 8054 and so, to do an analysis that mimics a real situation, we would have to discard the
 8055 uncaptured individuals. The function `simMnSCR` in the package `scrbook` will also simulate
 8056 data that includes a behavioral response which will be the typical situation in small-
 8057 mammal trapping problems (see ?, for details).

8058 Here we use our function `simMnSCR` to simulate a data set with $K = 7$ occasions. We’ll
 8059 run the model using `JAGS` which we have found is much more effective for this class of
 8060 models. We get the data set-up for analysis by augmenting the size of the data set to
 8061 $M = 200$. In addition we choose starting values for `s` and the data augmentation variables
 8062 `z`. For starting values of `s` we cheat a little bit here and use the true values for the observed
 8063 individuals and then augment the $M \times 2$ matrix `S` with $M - n$ randomly selected activity
 8064 centers. Our function `spiderplot` returns the mean observed location of individuals for
 8065 use as starting values for the `nind` encountered individuals. The parameters input to
 8066 `simMnSCR` are the intercept α_0 , $\sigma = \sqrt{1/(2\alpha_1)}$ for the Gaussian encounter probability
 8067 model, and α_2 is the behavioral response parameter. The data simulation and set-up
 8068 proceeds as follows:

```

8069 > set.seed(2013)
8070 > parms <- list(N=100,alpha0= -.40, sigma=0.5, alpha2= 0)
8071 > data <- simMnSCR(parms, K=7, ssbuff=2)
8072 > nind <- nrow(data$Ycat)
8073
8074 > M <- 200
8075 > Ycat <- rbind(data$Ycat,matrix(nrow(data$X)+1,nrow=(M-nind),ncol=data$K))
8076 > Sst <- rbind(data$S,cbind(runif(M-nind,data$xlim[1],data$xlim[2]),
8077                         runif(M-nind,data$ylim[1],data$ylim[2])))
8078 > zst <- c(rep(1,160),rep(0,40))
```

8079 The model specification is not much more complicated than the binomial or Poisson

models given previously. The main consideration is that we define the cell probabilities for each trap $j = 1, 2, \dots, J$ and then define the last cell probability, $J+1$, for “not captured”, to be the complement of the sum of the others. The code is shown in Panel ???. In the last lines of code here we specify N and density, D , as derived parameters.

To fit the model, we need to package everything up (inits, parameters, data) and send it off to **JAGS** to build an MCMC simulator for us (these commands are executed in the help file for **simMnSCR**). In addition to the usual data objects, we also pass the limits of the assumed rectangular state-space (`ylim`, `xlim`, both 1×2 vectors) and the scale of the standardized units, called `trap.space` here because we typically will define the trap coordinates to be an integer grid. If the trap spacing is 10 m and we want units of density computed in terms of individuals per meter-squared, then we input `trap.space=10`. The analysis is carried out as follows:

```

8092 > inits <- function(){ list (z=zst,sigma=runif(1,.5,1) ,S=Sst) }
8093
8094 # Parameters to monitor
8095 > parameters <- c("psi","alpha0","alpha1","sigma","N","D")
8096
8097 # Bundle the data. Note this reuses "data"
8098 > data <- list (X=data$X,K=data$K, trap.space=1,Ycat=Ycat,M=M,
8099           ntraps=nrow(data$X),ylim=data$ylim,xlim=data$xlim)
8100
8101 > library(R2jags)
8102 > out <- jags (data, inits, parameters, "model.txt", n.thin=1,
8103   n.chains=3, n.burnin=1000, n.iter=2000)
```

The posterior summaries are provided in the following **R** output (recall that $N = 100$, $\alpha_0 = -0.40$, and $\sigma = 0.5$):

```

8106 > out
8107 Inference for Bugs model at "model.txt", fit using jags,
8108 3 chains, each with 2000 iterations (first 1000 discarded)
8109 n.sims = 3000 iterations saved
8110      mu.vect sd.vect    2.5%     25%     50%     75%   97.5% Rhat n.eff
8111 D        1.873  0.189  1.531   1.750   1.859   2.000   2.250 1.006  1300
8112 N       119.867 12.107 98.000 112.000 119.000 128.000 144.000 1.006  1300
8113 alpha0   -0.435  0.151  -0.738  -0.535  -0.439  -0.331  -0.146 1.004   580
8114 alpha1    2.195  0.286   1.658   2.004   2.180   2.372   2.785 1.003  2400
8115 psi      0.599  0.069   0.465   0.552   0.599   0.645   0.739 1.006  1400
8116 sigma    0.480  0.032   0.424   0.459   0.479   0.500   0.549 1.003  2400
8117 deviance 892.164 21.988 850.922 877.417 891.561 906.246 937.728 1.003   950
8118
8119 [... output deleted ...]
```

9.2.3 Multinomial relationship to the Poisson

The multinomial is related to the Poisson encounter rate model by a conditioning argument. Let y_{ij} be the number of encounters for individual i in trap j . If $y_{ij} \sim \text{Poisson}(\lambda_{ij})$,

```

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`.

then, conditional on the *total* number of captures (i.e., across all traps), $y_i = \sum_j y_{ij}$, the trap encounter frequencies are multinomial with probabilities

$$\pi_{ij} = \frac{\lambda_{ij}}{\sum_j \lambda_{ij}}$$

for $j = 1, 2, \dots, J$. Or equivalently the *trap of capture* is categorical with probabilities π_{ij} as given above. Under the Gaussian kernel model, these probabilities are:

$$\pi_{ij} = \frac{\exp(-\alpha_1 d(\mathbf{x}_i, \mathbf{s}_i)^2)}{\sum_j \exp(-\alpha_1 d(\mathbf{x}_i, \mathbf{s}_j)^2)} \quad (9.2.2)$$

where, we note, the intercept α_0 has canceled from both the numerator and denominator. This makes sense because, here, these probabilities describe the trap-specific capture probabilities *conditional on capture*. Therefore, the model is not completely specified, absent a model for the “overall” probability of encounter or the expected frequency of captures, say ϕ_i . Depending on how we specify a model for this quantity ϕ_i , we can reconcile it directly with the Poisson model. Let y_i be the total number of encounters for individual i and suppose y_i has a Poisson distribution with mean ϕ_i . Then, marginalizing Eq. ?? over the Poisson distribution for y_i produces the original set of *iid* Poisson frequencies with probabilities:

$$\lambda_{ij} = \phi_i \pi_{ij}$$

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 the marginal distribution of y_{ij} is Poisson with mean $\exp(\alpha_0 - \alpha_1 d(\mathbf{x}, \mathbf{s})^2)$, equivalent to Eq. ??.

In summary, the Poisson and multinomial models are equivalent in how they model the distribution of captures among traps. It stands to reason that, if the encounter rate of individuals is low, we could use the Poisson and multinomial models interchangeably. In fact, based on our discussion in Sec. ?? 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

We analyze data from a mist-netting study of ovenbirds, conducted at the Patuxent Wildlife Research Center, Laurel MD, by D.K. Dawson and M.G. Efford. The data from this study are available in the `secr` package, and have been analyzed previously by ?, see also ?. Forty-four mist nets spaced 30 m apart on the perimeter of a 600-m x 100-m rectangle were operated on 9 or 10 non-consecutive days in late May and June for 5 years from 2005-2009. The ovenbird data can be loaded as follows:

```
> library(secr)
> data(ovenbird)
```

The data set consists of adult ovenbirds caught during sampling in each of 5 years, 2005-2009. (one ovenbird was killed in 2009, indicated by a negative net number in the encounter data file). As with most mist-netting studies, nets are checked multiple times during a day (e.g., every hour during a morning session). However, for this data set, the within-day recaptures are not included so each bird has at most a single capture per day. Therefore

8158 the multinomial model (detector type ‘multi’ in **secr**) is appropriate. Although several
 8159 individuals were captured in more than one year, this information is not used in the models
 8160 presently offered in **secr**, but we do make use of it in the development of open models in
 8161 Chapt. ??.

8162 **Multiple sample sessions**

8163 Up to this point we have only dealt with a basic closed population sampling situation
 8164 consisting of repeated sample occasions on a single population of individuals using a single
 8165 array of traps. In practice, many studies produce repeated samples over longer periods
 8166 of time over which demographic closure isn’t valid, or at different locations where the
 8167 populations are completely distinct. We adopt the **secr** terminology of *session* for such
 8168 replication by groups of time or space, and the models are *multi-session* models, although
 8169 we think of such models as being relevant to any stratified population (see Chapt. ??).
 8170 We introduced **secr**’s multi-session models in Sec. ???. In the case of the ovenbird data,
 8171 sampling was carried out in multiple years, with a number of sample occasions within
 8172 each year (9 or 10), a type of data structure commonly referred to as “the robust design”
 8173 (?). In this context, it stands to reason that there is recruitment and mortality happening
 8174 across years. In Chapt. ?? we model these processes explicitly but, here, we provide an
 8175 analysis of the data that does not require explicit models for recruitment and survival,
 8176 regarding the yearly populations as independent strata, and fitting a multi-session model.

8177 When the sessions represent explicit time periods, the multi-session model of **secr** can
 8178 be thought of as a type of open population model. In particular, a special case of open
 8179 models arises when we assume N_t (time-specific population sizes) are independent from
 8180 one time period or session to the next – this can be thought of as a “random temporary
 8181 emigration” model of the ? variety, and this is the multi-session model implemented
 8182 in **secr**. In particular, by assuming that N_t is Poisson with mean Λ_t , one can model
 8183 variation in abundance among sessions based on the Poisson-integrated likelihood in which
 8184 parameters of Λ_t appear directly in the likelihood as we noted in Sec. ???. We provide
 8185 an analysis (below) of the ovenbird data here using the multi-session models in **secr**.
 8186 We formalize the multi-session model approach from a Bayesian perspective using data
 8187 augmentation in Chapt. ?? (??).

8188 A 3rd way to develop models for stratified or grouped populations, not based on
 8189 multi-session models, but that is convenient in **BUGS**, is to regard the data from each
 8190 session as an independent data set with its own N_t parameter, and do T distinct data
 8191 augmentations. Because each N_t is regarded as a free parameter, independent of the
 8192 other parameters, we’ll call this the nonparametric multi-session model to distinguish it
 8193 from the multi-session model which assumes the N_t are related to one another by having
 8194 been generated from a common Poisson distribution. We can analyze this model in the
 8195 normal context of data augmentation by augmenting each year separately in the same
 8196 **BUGS** model specification. This approach avoids making explicit model assumptions
 8197 about the N_t parameters. This is distinct from the model implemented in **secr** in that
 8198 **secr** is removing the N_t parameters by integrating the conditional-on- N_t likelihood over
 8199 the Poisson prior for N_t ¹.

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.

8200 We demonstrate these 3 approaches to analyzing grouped/stratified data using the
 8201 ovenbird data: (1) In the following section, we provide the nonparametric multi-session
 8202 model with unconstrained N_t ; (2) we demonstrate the Poisson model-based multi-session
 8203 models from **secr** both here (following section) and in Chapt. ?? from a Bayesian stand-
 8204 point; (3) later, in Chapt. ??, we provide a fully dynamic “spatial Jolly-Seber” model and
 8205 apply it to the ovenbird data.

8206 Analysis in JAGS

8207 The ovenbird data are provided as a multi-session **capthist** object **ovenCH** which, by
 8208 regarding years as independent strata, or sessions, allows for the fitting of the multi-
 8209 session model. For doing a Bayesian analysis in one of the **BUGS** engines (we use **JAGS**
 8210 here) there are a number of ways to structure the data and describe the model. We can
 8211 analyze either a 2-d data set with all years (data augmented) “stacked” into a data set of
 8212 dimension $(5 * M) \times 10$ (5 years, M = size of the augmented data set, K = 10 replicate
 8213 sample occasions). Or, we could produce a 3-d array ($M \times J \times K$). We adopted the
 8214 former approach, analyzing the data as a 2-d array and creating an additional categorical
 8215 variable for “year” to indicate which stratum (year) each record goes with.

8216 Data on individual sex is included with **secr**, but we provide an analysis of a single
 8217 model for all adults, constant σ across years, constant p_0 , and year-specific values of N_t
 8218 (and hence D_t). There is a habitat mask provided with the data but the mask appears
 8219 to just be a modified rectangle around the net locations, clipped to have rounded corners,
 8220 and so we don’t use it here. Instead, we used a rectangular state-space buffer of 200 meters
 8221 for our analysis. There was a single loss-on-capture which we accounted for by fixing $p = 0$
 8222 for all subsequent encounters of that individual (indicated by the binary variable **dead**,
 8223 as shown in Panel ??). We have an **R** script in **scrbook** package called **SCRovenbird**, so
 8224 you can see how to set-up the data and run the model. Executing the script **SCRovenbird**
 8225 produces the posterior summaries given in Table ???. Here, density is in units of birds per
 8226 ha. The posterior mean of σ is about 76 meters, and there is considerable variability in
 8227 density over the 5 year period with density peaking at 1.2 birds/ha in year 3, although
 8228 there is considerable posterior uncertainty. The R-hat’s look a little bit peaked and so we
 8229 might consider running the MCMC analysis longer.

8230 Analysis in secr

8231 Included with the ovenbird data are a number of models fitted as examples. Those include:

```
8232 ovenbird.model.1    fitted secr model -- null
8233 ovenbird.model.1b   fitted secr model -- g0 net shyness
8234 ovenbird.model.1T   fitted secr model -- g0 time trend within years
8235 ovenbird.model.h2   fitted secr model -- g0 finite mixture
8236 ovenbird.model.D   fitted secr model -- trend in density across years
```

8237 The model fit objects provided in **secr** are based on the use of the habitat mask.
 8238 To make the analyses consistent with our previous analysis in **JAGS**, we refit all of the
 8239 models here without the habitat mask. The re-analysis proceeds as follows, changing the
 8240 “trend in density across years” model to allow for year-specific density:

```
8241 ## Fit constant-density model
8242 > ovenbird.model.1 <- secr.fit(ovenCH)
```

```

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

```

8243 ## Fit net avoidance model
8244 > ovenbird.model.1b <- secr.fit(ovenCH, model = list(g0 ~ b))
8245 ## Fit model with time trend in detection
8246 > ovenbird.model.1T <- secr.fit(ovenCH, model = list(g0 ~ T))
8247 ## Fit model with 2-class mixture for g0
8248 > ovenbird.model.h2 <- secr.fit(ovenCH, model = list(g0 ~ h2))
8249 ## Fit a model with session (year)-specific Density
8250 > ovenbird.model.DT <- secr.fit(ovenCH, model = list(D ~ session))

```

8251 All of these can be fitted easily in **JAGS** but the model we fitted previously is roughly
 8252 equivalent to the last model, `ovenvbird.model.DT`, because we allowed for year-specific
 8253 population sizes (and hence density). So, we'll compare our results from **JAGS** to that
 8254 model. The `secr` output is extensive and so we do not reproduce it completely here. By
 8255 default, it summarizes the trap information for each year, encounter information, and then
 8256 output for each year. Here is an abbreviated version for `ovenvbird.model.DT`:

```

8257 > print(ovenvbird.model.DT,digits=2)
8258
8259 secr.fit( capthist = ovenCH, model = list(D ~ session), buffer = 300 )
8260 secr 2.3.1, 14:46:52 23 Jan 2013

```

```

8261
8262 $'2005'
8263 Object class      traps
8264 Detector type    multi
8265 Detector number   44
8266 Average spacing   30.27273 m
8267 x-range           -50 49 m
8268 y-range           -285 285 m
8269
8270 [... deleted ...]
8271
8272          2005 2006 2007 2008 2009
8273 Occasions       9   10   10   10   10
8274 Detections      35   42   52   30   33
8275 Animals         20   22   26   19   16
8276 Detectors        44   44   44   44   44
8277
8278 Model            : D~session g0~1 sigma~1
8279 Fixed (real)     : none
8280 Detection fn     : halfnormal
8281 Distribution      : poisson
8282 N parameters     : 7
8283 Log likelihood   : -1119.845
8284 AIC              : 2253.689
8285 AICc             : 2254.868
8286
8287 [... deleted ...]

```

8288 To do model selection we use the handy helper-function `AIC` as follows (output edited
 8289 to fit on the page):

```

8290 AIC (ovenbird.model.1, ovenbird.model.1b, ovenbird.model.1T,
8291           ovenbird.model.h2, ovenbird.model.DT)
8292
8293          model detectfn npar logLik     AIC     AICc     dAICc
8294 ovenbird.model.1T [edited output] 4 -1111.850 2231.700 2232.109 0.000
8295 ovenbird.model.1b      ....      4 -1117.615 2243.229 2243.637 11.528
8296 ovenbird.model.h2      ....      3 -1121.164 2248.327 2248.570 16.461
8297 ovenbird.model.1      ....      5 -1119.762 2249.524 2250.143 18.034
8298 ovenbird.model.DT      ....      7 -1119.845 2253.689 2254.868 22.759

```

8299 We see that our `DT` model is way down at the bottom of the list. Instead, the model with
 8300 a time-trend (within-season) in detection probability is preferred, followed by a behavioral
 8301 response. We encourage you to adapt the **JAGS** model specification for such models which
 8302 is easily done (see Chapt. ?? for many examples). We provide the summary results for
 8303 the model having `D ~ session` as follows:

```
8304 > print(ovenbird.model.DT,digits=2)
```

```

8305
8306 secr.fit( capthist = ovenCH, model = list(D ~ session), buffer = 300 )
8307 secr 2.3.1, 14:46:52 23 Jan 2013
8308
8309 [...deleted....]
8310
8311 Fitted (real) parameters evaluated at base levels of covariates
8312
8313 session = 2005
8314      link estimate SE.estimate     lcl     ucl
8315 D      log      0.920        0.228  0.571  1.484
8316 g0     logit     0.028        0.004  0.021  0.037
8317 sigma   log     78.566       6.379 67.025 92.095
8318
8319 session = 2006
8320      link estimate SE.estimate     lcl     ucl
8321 D      log      0.963        0.238  0.598  1.553
8322 g0     logit     0.028        0.004  0.021  0.037
8323 sigma   log     78.566       6.379 67.025 92.095
8324
8325 session = 2007
8326      link estimate SE.estimate     lcl     ucl
8327 D      log      1.139        0.282  0.706  1.836
8328 g0     logit     0.028        0.004  0.021  0.037
8329 sigma   log     78.566       6.379 67.025 92.095
8330
8331 session = 2008
8332      link estimate SE.estimate     lcl     ucl
8333 D      log      0.832        0.206  0.516  1.341
8334 g0     logit     0.028        0.004  0.021  0.037
8335 sigma   log     78.566       6.379 67.025 92.095
8336
8337 session = 2009
8338      link estimate SE.estimate     lcl     ucl
8339 D      log      0.701        0.173  0.435  1.130
8340 g0     logit     0.028        0.004  0.021  0.037
8341 sigma   log     78.566       6.379 67.025 92.095

```

8342 The point estimates (MLEs) of density are uniformly lower than the Bayesian estimates
8343 (posterior means) shown in Table ???. We expect some difference in this direction due to
8344 small-sample skew of the posterior. In addition, there may be slight differences due to
8345 the fact that **secr** multi-session model assumes that the N_t have a Poisson prior, but the
8346 implementation in **JAGS** using data augmentation is based on a binomial prior. The
8347 estimated σ is very similar between the **JAGS** analysis and **secr**.

9.3 SINGLE-CATCH TRAPS

8348 The classical animal trapping experiment is based on a physical trap which captures a
 8349 single animal and holds that individual until subsequent molestation by a biologist. This
 8350 type of observation model – the “single-catch” trap – was the original situation considered
 8351 in the context of spatial capture-recapture by ?. Nowadays, capture-recapture data are
 8352 more often obtained by other methods (DNA from hair snares, or scat sampling, camera
 8353 traps etc...) but nevertheless the single-catch traps are still widely used in small mammal
 8354 studies (??) and other situations.

8355 The single-catch model is basically a multinomial model but one in which the number
 8356 of available traps is reduced as each individual is captured. As such, the constraints on the
 8357 joint likelihood for the sample of n encounter histories are very complicated. As a result,
 8358 at the time of this writing, there has not been a formal development of either likelihood or
 8359 Bayesian analysis of this model and applications of SCR models to single-catch systems
 8360 have used the independent multinomial model as an approximation (see below).

8361 Nevertheless, we can make some progress to describing the basic observation model
 8362 formally. In particular, if we imagine that all of the individuals captured queued up at
 8363 the beginning of the capture session to draw a number indicating their order of capture,
 8364 then there is a nice conditional structure resulting from a “removal process” operating on
 8365 the traps. The first individual captured has the multinomial observation model:

$$\mathbf{y}_1 \sim \text{Multinomial}(\boldsymbol{\pi}_1)$$

8366 whereas the 2nd individual captured also has a multinomial encounter probability model
 8367 but with the trap which captured the first individual removed. We might express this as:

$$\mathbf{y}_2 \sim \text{Multinomial}(\boldsymbol{\pi}_2)$$

8368 where

$$\pi_{2j} = \frac{(1 - y_{1j}) * \exp(\alpha_0 - \alpha_1 d_{ij}^2)}{\sum_j (1 - y_{1j}) * \exp(\alpha_0 - \alpha_1 d_{ij}^2)}$$

8369 and so on for $i = 3, 4, \dots, n$. In a certain way, this model is a type of local behavioral
 8370 response model but where the response is to other individuals being captured. Evidently,
 8371 the **order of capture** is relevant to the construction of these multinomial cell probabil-
 8372 ities. More generally, the *time* of capture of an individual in any trapping interval will
 8373 affect the encounter probability of subsequently captured individuals, but we think that
 8374 order of capture might lead to a practical approximation to the single-catch process (this
 8375 is how we simulate the data in our function `simScSCR`). In the simulation of single catch
 8376 data, we randomly ordered the population of individuals for each sample occasion, and
 8377 then cycled through them, turning off each trap if an individual was captured in it.

8378 9.3.1 Inference for single-catch systems

8379 For the single-catch model, we argued that the observations have a multinomial type of
 8380 observation model, but the multinomial observations have a unique conditional dependence
 8381 structure among them owing to the “removal” of traps as they fill-up with individuals.
 8382 Thus, competition for single-catch traps renders the independence assumptions for the
 8383 independent multinomial model invalid. However, as ? noted, we expect “bias to be small

when trap saturation (the proportion of traps occupied) is low. Trap saturation will be higher when population density is high..." relative to trap density, or when net encounter probability is high. ? did a limited simulation study and found essentially no effective bias and concluded that estimators of density from the misspecified independent multinomial model are robust to the mild dependence induced when trap saturation is low. Naturally then, we expect that the Poisson model could also be an effective approximation under the same set of circumstances.

In the **R** package **scrbook** we provide a function for simulating data from a single-catch system (function **simScSCR**) and fitting the misspecified model (**example(simScSCR)**) in **JAGS** so that you can evaluate the effectiveness of this misspecified model for situations that interest you.

9.3.2 Analysis of Efford's possum trapping data

We provide an analysis here of data from a study of brushtail possums in New Zealand. The data are available with the **R** package **secr** (?); see the help file **?possum** after loading the **secr** package. Originally the data were analyzed by ?, and a detailed description of the data set is available in the help file, from which we summarize:

*Brushtail possums (*Trichosurus vulpecula*) are an unwanted invasive species in New Zealand. Although most abundant in forests, where they occasionally exceed densities of 15/ha, possums live wherever there are palatable food plants and shelter.*

To load the possum data, execute the following commands:

```
> library(secr)
> data(possum)
```

The study area encompasses approximately 300 ha, and 180 live traps were organized in 5 distinct grids, shown in Fig. ???. Each square arrangement of traps consisted of 36 traps with a spacing of 20 m. Thus the squares are 180 m on a side. Individuals were captured, tagged, and released over 5 days during April, 2002. A noteworthy aspect of this study is that it involves replicated grids selected in some fashion from within a prescribed region. From an analysis standpoint, we could adopt the use of the multi-session models which we used previously to analyze the ovenbird data. This would be useful if we had covariates at the trapping grid level that we wanted to model. Alternatively, we could pool the data from all of the grids and analyze them jointly as if they were based on a single trapping grid (with 180 traps) which is clearly a reasonable view in this case. In doing this sort of pooling, there is an implicit assumption that N_t (t indexing trapping grid in this case) is Poisson distributed, with constant mean (??) which we also address in Chapt. ??.

The data file **possumCH** contains 112 encounter histories, and we analyze those here although the last 8 of those are recaptures treated as new individuals². The encounter process is not strictly a single-catch multinomial process because, as noted in the **possum** help file "One female possum was twice captured at two sites on one day, having entered a second trap after being released; one record in each pair was selected arbitrarily and discarded." which is a similar situation to what might happen in bird mist net studies, as a bird might fly into a net upon release from another. By discarding the two extra-capture

²M. Efford, personal communication

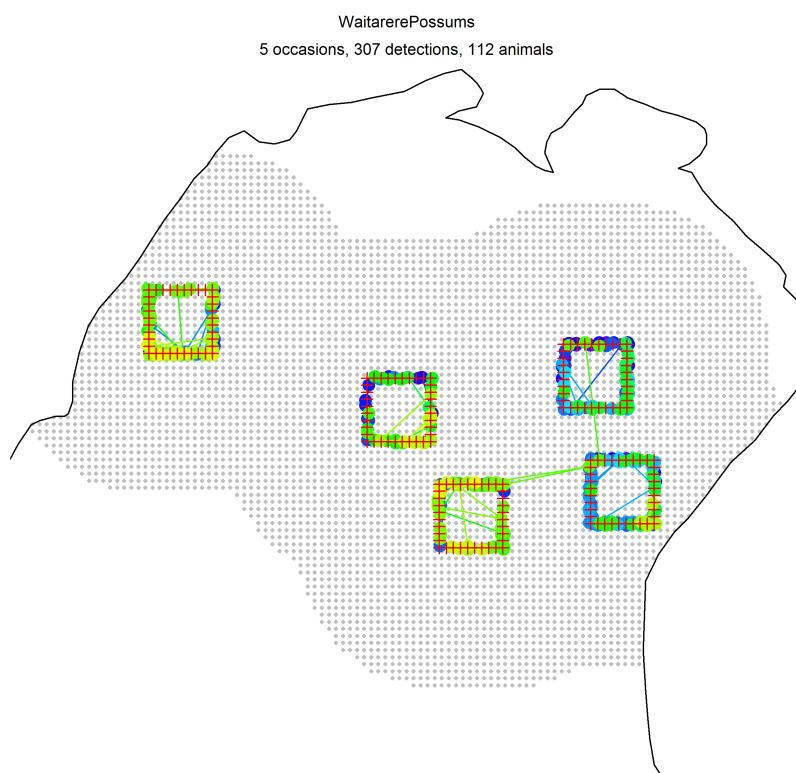


Figure 9.2. Trapping grids used in possum study from ?, data are contained in the R package `secr` (?), refer to the help file `?possum` for additional details of this study.

events, we can satisfactorily view these data as single-catch data, for which **secr** uses the independent multinomial likelihood (M. Efford, pers. comm.). If multiple, same-session captures were common, then it might be worth developing a model for $n_{i,k}$ = the number of captures of individual i during sample occasion k , in order to make use of all captures.

For our Bayesian analysis here, we used a rectangular state-space which doesn't account for any geographic boundaries of the survey region, but we note that a habitat mask is included in **secr** and it could be used in a Bayesian analysis. Whether or not we use the mask is probably immaterial as long as we understand the predictions of N or D over the water don't mean anything biological and we probably wouldn't report such predictions. The **JAGS** model specification is based on that of the ovenbird analysis given previously, and so we don't reproduce the model here. The **R/JAGS** script is called **SCRpossum**, which is in the **scrbook** package. The results are summarized in Table ??.

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
Dha	1.549	0.115	1.343	1.547	1.777	1.009	340
α_0	-0.935	0.167	-1.270	-0.934	-0.605	1.007	870
α_1	0.000	0.000	0.000	0.000	0.000	1.001	2800
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

The estimated density (posterior mean) is about 1.53 possums/ha. To obtain the **secr** results for the equivalent null model, we execute the following command

> **secr.fit(capthist = possumCH, trace = F)**

which produces (edited) summary output:

```
8441 [...] some output deleted ...
8442
8443 Fitted (real) parameters evaluated at base levels of covariates
8444 link estimate SE.estimate lcl ucl
8445 D log 1.6988930 0.17352645 1.3913904 2.0743547
8446 g0 logit 0.1968542 0.02256272 0.1563319 0.2448321
8447 sigma log 51.4689114 2.59981905 46.6204139 56.8216500
8448
8449 [...] some output deleted ...]
```

As we've discussed previously, there are many reasons for why there might be differences between Bayesian and likelihood estimates. But even among likelihood estimates – any time you run a model there is some numerical integration going on which requires some specific choices of how to do the integration (see Chapt. ??). For now we just observe

8454 that the estimated density is certainly in the ballpark (compared to those in Table. ??),
 8455 and so too is the estimated σ .

9.4 ACOUSTIC SAMPLING

8456 The last decade has seen an explosion of technology that benefits the study of animal
 8457 populations. This includes DNA sampling methods that allow for identification from
 8458 hair or scat, camera trapping and identification software that allow efficient sampling of
 8459 many mammals, and the resulting statistical technology that helps us to make sense of
 8460 such data (??????). One other extremely promising technology area is that of acoustic
 8461 sampling using microphones or recording devices. That is, instead of having cameras
 8462 record encounters, or humans pick up scat, we can establish an array of (usually) electronic
 8463 recording devices which, instead of establishing a visual identity of individuals, record a
 8464 vocal expression of each individual. In this context, ? referred to audio recorders as “signal
 8465 strength proximity detectors” to distinguish them from other types of proximity detections,
 8466 including camera traps, which are *visual* proximity detector. Using audio records, the
 8467 spatial pattern of the *signal strength* at the different audio recorders or microphones can be
 8468 used for inference about density (??) in the same way as the spatial pattern of detections is
 8469 used in the types of SCR models we have discussed so far. The basic technical formulation
 8470 of these models comes from ?, and it was applied to field study of birds by ?. In that
 8471 study, recording devices were organized in groups of 4 (in a square pattern), with an array
 8472 of 5×15 such clusters of 4, separated by 100 m (300 total recorder locations). This data
 8473 set, called **signalCH**, is provided with the **secr** package along with some sample analyses
 8474 and help files. See ?, a version of the document **secr-sound.pdf** (that also comes with
 8475 the **secr** package) which you can access directly from the main help file (?**secr**).

8476 Our development here mostly follows ?, but we change some notation to be consistent
 8477 with our previous material. Let $S(\mathbf{x}, \mathbf{u})$ be the strength of a signal emanating from signal
 8478 location \mathbf{u} , as recorded by a device at location \mathbf{x} . Just as ordinary SCR models represent
 8479 a model of *encounter frequency* as a function of distance, in acoustic models, the acoustic
 8480 SCR model is a model of sound attenuation as a function of distance. In particular, the
 8481 acoustic models assumes that S (or a suitable transformation) declines with distance d
 8482 from the origin of the sound, to the recording device. In the context of spatial sampling
 8483 of animals, the origin is the actual location of some individual animal, and the recording
 8484 device is something we nailed to a tree, or mounted on a post. For example, a model of
 8485 sound attenuation used by ? is the following:

$$S(\mathbf{x}, \mathbf{u}) = \alpha_0 + \alpha_1 d(\mathbf{x}, \mathbf{u}) + \epsilon \quad (9.4.1)$$

8486 where $\epsilon \sim \text{Normal}(0, \sigma_s^2)$. In many standard situations, S will be measured in decibels,
 8487 which can be any value on the real line. In the conduct of acoustic sampling and the
 8488 development of custom models for your own situation, it would probably be helpful to know
 8489 something about sound dynamics and signal processing. In this model, the parameters
 8490 α_0 , α_1 and σ_s^2 are to be estimated. We abbreviate the set of parameters by $\boldsymbol{\theta}$ for short.

8491 The basic structure of an acoustic SCR study is not really much different from ordinary
 8492 SCR studies. Just as ordinary SCR models require that individuals be encountered at > 1
 8493 trap, these acoustic models require that individuals be heard at > 1 recorder. Therefore,
 8494 the acoustic signals (calls or vocalizations) must be reconcilable and, in fact, reconciled

8495 successfully by the investigator. In practice, this would require associating signals that
 8496 occur at the same instant with the same individual (or making a decision one way or the
 8497 other). Further, if individuals are actively moving during the sample period (that recorders
 8498 are functioning) then individuals might be double-counted, thereby biasing estimates of
 8499 density. In general, the models produce an estimate of density of *sources*, and how that is
 8500 interpreted depends on whether individuals are stationary or mobile, and other things. In
 8501 particular, if multiple survey occasions are used (e.g., on different days), then modeling
 8502 movement of individuals would be essential in order to interpret estimates of density
 8503 meaningfully. Models that allow some movement should be possible (see Sec. ?? below,
 8504 and Chaps. ?? and ??).

8505 **9.4.1 The signal strength model**

8506 We assert that an individual is detected if S exceeds a threshold, c . The reason for intro-
 8507 ducing this threshold c is that sound recorders will always record some background sound,
 8508 and so effective use of the acoustic SCR models requires specification of the threshold of
 8509 measured signal below which the record is censored (non-detection occurs) because the
 8510 recorded sound is assumed to be background noise. So we assert that an individual is
 8511 detected if $S > c$ which occurs with probability $\Pr(S > c)$, the encounter probability. To
 8512 expand on and formalize this, let S_{ij} be the observed value of S for animal i at detector
 8513 j . The encounter probability is $\Pr(S_{ij} > c)$ which is $\Pr(S_{ij} > c) = 1 - \Pr(S_{ij} < c)$, so
 8514 that, if we standardize the variate we have

$$1 - \Pr\left(\frac{(S_{ij} - \mathbb{E}(S))}{\sigma_s} < \frac{(c - \mathbb{E}(S))}{\sigma_s}\right)$$

8515 This probability calculation requires evaluation of the CDF of a standard normal variate
 8516 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
 8517 the parameters α_0 , α_1 , σ_s^2 and also the individual location \mathbf{u} and trap location \mathbf{x} . We'll
 8518 identify it by $\gamma(\boldsymbol{\theta}, \mathbf{x}, \mathbf{u})$ when we need to be explicit about those things. We can compute
 8519 $\Pr(S_{ij} > c) = 1 - \Pr(\eta < \gamma(\boldsymbol{\theta}, \mathbf{x}, \mathbf{u}))$ easily using any software package including R which
 8520 has a standard function, `pnorm`, for computing the normal cdf. To be more precise, we'll
 8521 use the $\Phi()$ to represent the normal cdf. Therefore, an individual is encountered whenever
 8522 $S_{ij} > c$ which happens with probability $\Pr(S_{ij} > c) = 1 - \Phi(\gamma(\boldsymbol{\theta}, \mathbf{x}, \mathbf{u}))$.

8523 Naturally this quantity should depend on *where* an individual is located at the time
 8524 of recording – what we call its instantaneous location, say \mathbf{u} , to distinguish it from its
 8525 home-range center \mathbf{s} (but we outline a model below that contains both \mathbf{u} and \mathbf{s}), and
 8526 also the trap \mathbf{x} , so we index the quantity γ by those two quantities, in addition to the
 8527 parameters α_0 , α_1 and σ_s . The probability of detection is therefore

$$p_{ij} = p(\alpha_0, \alpha_1, \sigma | \mathbf{x}_j, \mathbf{u}_i) = 1 - \Phi(\gamma(\cdot))$$

8528 where \mathbf{u}_i is the instantaneous location of individual i and \mathbf{x}_j is the location of trap j .
 8529 We'll suppose here that the random variables \mathbf{u}_i have state-space \mathcal{U} ³.

8530 How do we interpret this probability? Well, two things have to happen for an individual
 8531 to be encountered by a trap: (1) it has to vocalize; (2) the microphone has to record a

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

8532 signal $> c$. These two things together are a product of biological and environmental factors
 8533 which could include time of day, wind direction and speed, or maybe rain, humidity and
 8534 other things. The bottom line is a lot of factors are balled up in whether or not the
 8535 microphone records a sound greater than the threshold.

8536 The observations from an acoustic survey are the signal strength measurements, and
 8537 the likelihood of the observed signal strength from individual i at detection device j can
 8538 be specified by noting that the likelihood is the normal pdf for the observed signal if the
 8539 signal strength is $> c$ and, otherwise, the contribution to the likelihood is $\Phi(\gamma(\cdot))$ (see Eq.
 8540 8 of ??):

$$\Pr(S_{ij}|\mathbf{u}_i) = \Phi(\gamma(\cdot))^{1-I(S_{ij}>c)} \text{Normal}(S_{ij}; \alpha_0, \alpha_1, \sigma_s, \mathbf{x}_j, \mathbf{u}_i)^{I(S_{ij}>c)}$$

8541 We can use this as the basis for constructing the binomial-form of the likelihood as
 8542 we did in Chapt. ??, which involves the number of individuals not encountered, n_0 . The
 8543 probability that an individual is *not* captured is equal to the probability that its signal
 8544 strength doesn't exceed c at any microphone. The probability of not being captured at a
 8545 microphone \mathbf{x}_j is:

$$1 - p_{\mathbf{u},j} = \Phi(\gamma(\cdot))$$

8546 and therefore the probability of not being captured at any microphone is:

$$\Pr(\text{all } S_{\mathbf{u},j} < c | \mathbf{u}) = \prod_{j=1}^J (1 - p_{\mathbf{u},j}) = \prod_{j=1}^J \Phi(\gamma(\cdot, \mathbf{x}_j, \mathbf{u}))$$

8547 and therefore the marginal probability of not being captured is

$$\pi_0 = [\text{all } S_{\mathbf{u},j} < c | \boldsymbol{\alpha}] = \int_{\mathcal{U}} \left\{ \prod_{j=1}^J \Phi(\gamma(\boldsymbol{\theta}, \mathbf{x}_j, \mathbf{u})) \right\} d\mathbf{u}$$

8548 which can be used to construct the binomial form of the likelihood as we did in Chapt.
 8549 ?? (see Eq. ??).

8550 9.4.2 Implementation in secr

8551 Fitting acoustic encounter models in **secr** is no more difficult than other SCR models.
 8552 There is a handy manual (**secr-sound.pdf**) with examples (?) which comes with the **secr**
 8553 package. The basic process is that **make.capthist** will make a **capthist** object from a 3-
 8554 dimensional encounter array – which is a binary array indicating whether each individual
 8555 was detected or not at each recorder/microphone. In the case of signal strength data,
 8556 **secr** handles the case where `# occasions = 1`, i.e., the recorders obtained data for a single
 8557 sample occasion, but this is not a general requirement of the model for signal strength
 8558 data (see next section). The “signal” attribute of the **capthist** object contains the signal
 8559 strength in decibels. The best way to include the signal attribute is to use **make.capthist**
 8560 in the usual way, providing it with the encounter data and trap data and, in addition, the
 8561 variable “cutval” (which is c in our notation above) and then provide the signal strength
 8562 data as an extra column of the **capthist** object. See `?make.capthist` for details.

8563 9.4.3 Implementation in BUGS

8564 We don't know of any Bayesian applications of acoustic SCR models, although we imagine
 8565 that implementation of such models in the **BUGS** engines should be achievable. It seems
 8566 easy enough to write down a general hierarchical model that would accommodate sampling
 8567 on repeated occasions. Let \mathbf{s}_i be the home range center, and let \mathbf{u}_{ik} the instantaneous
 8568 location of individual i during sample occasion k (see Chapt. ?? for similar models). The
 8569 model for \mathbf{u}_{ik} can be specified conditional on \mathbf{s}_i . For example, we could assume that \mathbf{u}_{ik}
 8570 are bivariate normal draws with mean \mathbf{s}_i and some variance σ_u^2 . Then, conditional on \mathbf{u}_{ik}
 8571 an individual produces a signal according to the signal attenuation model (Eq. ??), or
 8572 perhaps some other model. Then we generate the binary encounter data by truncating the
 8573 observed signal at c . This general model then is an example of an SCR model in which
 8574 parameters of a movement model are identifiable (see Sec. 2.6) because there is direct
 8575 information about movement outcomes from the sampling method, unlike other types of
 8576 encounter methods (e.g., camera traps) for which animal locations are restricted to a set of
 8577 fixed, pre-determined points where traps are located. Other types of SCR methods allow
 8578 for movement information too, including some of the search-encounter models (Chapt.
 8579 ??).

8580 Instead of developing a Bayesian version of this model here, we leave it to the reader
 8581 to explore simulating data and devising a Bayesian implementation of the acoustic model
 8582 in one of the **BUGS** engines. Note that for a single occasion, you can simulate the data
 8583 using the two stage model (having both \mathbf{s} and \mathbf{u}) or you can simulate \mathbf{u} uniformly without
 8584 dealing with \mathbf{s} in the model. The kernel of the **BUGS** model specification should resemble
 8585 the following snippet:

```
8586 model {
  8587   # Ignoring loops and data augmentation
  8588   u[i,1] ~ dunif(xlim[1], xlim[2])
  8589   u[i,2] ~ dunif(ylim[1], ylim[2])
  8590   mu[i,j] <- alpha0 + alpha1*d[i,j]
  8591   ###
  8592   ### JAGS has this T() truncation feature
  8593   S[i,j] ~ dnorm(mu[i,j], 1/sigma^2)T(c,Inf)
  8594   ###
  8595   gamma[i,j] <- (c - mu[i,j])/sigma
  8596   p[i,j] <- 1 - pnorm(gamma[i,j], 0, 1) # JAGS has pnorm() function
  8597   y[i,j] ~ dbern(p[i,j])
  8598 }
```

8599 9.4.4 Other types of acoustic data

8600 ? noted that various other types of acoustic data might arise for which SCR-like models
 8601 would be useful⁴. For example, we could measure the *time of arrival* of a vocal queue of
 8602 some sort at multiple recorders to estimate the number and origin of N queues. Another
 8603 example is that where we measure *direction* to a queue from multiple devices and do,

⁴Some of the following is also related to material presented by D.L. Borchers at the ISEC 2012 conference in Norway.

effectively, a type of statistical triangulation to the multiple but unknown number of sources. This has direct relevance to types of double or multiple-observer sampling that people do in field studies of birds. Normally 2 observers stand in close proximity and record birds, reconciling their detections after data collection. An SCR-based formulation of the double-observer method has two observers (or more) standing some distance apart, e.g., 50 or 100 meters, and marking individual birds on a map (or at least a direction) and a time of detection. The SCR/double-observer method could be applied to such data.

9.5 SUMMARY AND OUTLOOK

In this chapter we extended SCR models to accommodate alternative models for the observation process, including Poisson and multinomial models. Along with the binomial model described in Chapt. ??, this sequence of models will accommodate a substantial majority of contemporary spatial capture-recapture problems, including the 4 main types of encounter data: binary encounters, multinomial trials from “multi-catch” and “single-catch” (???) trap systems, and Poisson encounter frequency data from devices that can record multiple encounters of the same individual at a device. We summarize the standard observation models and the corresponding `secr` terminology in Table ???. What we refer to as search-encounter (or area-search) models (see Chapt. ??) are distinct from most of the other classes in that the observation location can also be random (in contrast to traps, where the location is fixed by design). This auxiliary data is informative about an intermediate process related to movement (?).

Table 9.4. Different observation models, where we discuss them in this book, and what the corresponding `secr` terminology is

observation model	Where in this book?	<code>secr</code> name
Bernoulli	Chapt. ??	<code>proximity</code>
Poisson	Sec. ??	<code>count</code>
Multinomial (ind)	Sec. ??	<code>multi-catch</code>
Multinomial (dep)	Sec. ??	<code>single-catch</code>
Acoustic	Sec. ??	<code>signal</code>
Search-encounter	Chapt. ??	<code>polygon</code> (in part)

There is a need for other types of encounter models that arise in practice. We identify a few of them here, although we neglect a detailed development of them at the present time or, in some cases, put that off until later chapters: (1) Removal systems – Sometimes traps kill individuals and SCR models can handle that. This can be viewed as a kind of open model, with mortality only, and we handle such models (in part) in Chapt. ??; (2) There are models for which only specific summary statistics are observable (??) which we cover in Chaps. ?? - ??; (3) We can have multiple observation methods working together as in ?.

There remains much research to be done to formalize models for certain observation systems. For example, while we think one will usually be able to analyze single-catch systems using the multi-catch model, or even the Bernoulli model if encounter probability is sufficiently low, a formalization of the single-catch model would be a useful development and, we believe, it should be achievable using one or another of the **BUGS** engines. In

8636 addition, classical “trapping webs” (?????) have been around for quite some time and it
8637 seems like they are amenable to formulation as a type of SCR model although we have
8638 not pursued that development simply because trapping webs are rarely used in practice.

8639

8640

8641

10

SAMPLING DESIGN

10.1 GENERAL CONSIDERATIONS

8642 10.1.1 Model-based not design-based

8643 10.1.2 Sampling space or sampling individuals?

8644 10.1.3 Scope of inference vs. state-space

10.2 STUDY DESIGN FOR (SPATIAL) CAPTURE-RECAPTURE

10.3 TRAP SPACING AND ARRAY SIZE RELATIVE TO ANIMAL MOVEMENT

8645 10.3.1 Example: Black bears from Pictured Rocks National Lakeshore:

8646 **10.3.2 Final musings: SCR models, trap spacing and array size**

10.4 SPACING OF TRAPS WITH TELEMETERED INDIVIDUALS

10.5 SAMPLING OVER LARGE SCALES

10.6 MODEL-BASED SPATIAL DESIGN

8647 **10.6.1 Formalization of the Design Problem for SCR Studies**

8648 **10.6.2 An Optimal Design Criterion for SCR**

(10.6.1)

(10.6.2)

(10.6.3)

8649 **10.6.3 Optimization of the criterion**

8650 **10.6.4 Illustration**

10.7 COVARIATE MODELS

10.8 SUMMARY AND OUTLOOK

8651

Part III

8652

8653

Advanced SCR Models

8654

8655

8656

8657

11

MODELING SPATIAL VARIATION IN DENSITY

11.1 HOMOGENEOUS POINT PROCESS REVISITED

11.2 INHOMOGENEOUS POINT PROCESSES

(11.2.1)

(11.2.2)

(11.2.3)

(11.2.4)

(11.2.5)

(11.2.6)

11.3 OBSERVED POINT PROCESSES**11.4 FITTING INHOMOGENEOUS POINT PROCESS SCR MODELS**

8658 **11.4.1 Continuous space**

8659 **11.4.2 Discrete space**

11.5 ECOLOGICAL DISTANCE AND DENSITY COVARIATES**11.6 THE JAGUAR DATA****11.7 SUMMARY AND OUTLOOK**

8660
8661

12

8662

MODELING LANDSCAPE CONNECTIVITY

12.1 SHORTCOMINGS OF EUCLIDEAN DISTANCE MODELS

(12.2.1)

(12.2.2)

(12.2.3)

8663 12.2.1 Example of Computing Cost-weighted distance

12.3 SIMULATING SCR DATA USING ECOLOGICAL DISTANCE

12.4 LIKELIHOOD ANALYSIS OF ECOLOGICAL DISTANCE MODELS

(12.4.1)

8664 **12.4.1 Example of SCR with Least-Cost Path**

12.5 BAYESIAN ANALYSIS

12.6 SIMULATION EVALUATION OF THE MLE

8665 **12.6.1 Simulation Results**

12.7 DISTANCE IN AN IRREGULAR PATCH

8666 **12.7.1 Basic Geographic Analysis in R**

12.8 SUMMARY AND OUTLOOK

8667
8668

13

8669
8670
8671

INTEGRATING RESOURCE SELECTION WITH SPATIAL CAPTURE-RECAPTURE MODELS

8672
8673
8674

13.1 A SIMPLE MODEL OF SPACE USAGE

13.1.1 Poisson use model

13.1.2 Thinning

13.1.3 Capture-recapture Data

13.2 THE JOINT RSF/SCR LIKELIHOOD

13.3 APPLICATION: NEW YORK BLACK BEAR STUDY

13.4 SIMULATION STUDY

13.5 SUMMARY AND OUTLOOK

8675
8676

14

8677
8678

STRATIFIED POPULATIONS: MULTI-SESSION AND MULTI-SITE DATA

14.1 DATA STRUCTURE

14.2 MULTINOMIAL ABUNDANCE MODELS

(14.2.1)

(14.2.2)

(14.2.3)

(14.2.4)

(14.2.5)

8679
8680
8681
8682

14.2.1 Observation Models

14.2.2 Simulating group structured capture-recapture data

14.2.3 Fitting in BUGS

14.2.4 Approach B modeling ψ

14.3 SPATIAL CAPTURE-RECAPTURE

(14.3.1)

14.4 APPLICATION**8683 14.4.1 Results**

14.5 TOPICS IN MULTI-SESSION MODELS

8684 **14.5.1 Temporal models**

8685 **14.5.2 Dependence – is it a problem?**

14.6 MULTI-SESSION MODELS IN SECR

8686 **14.6.1 Ovenbird data in WinBUGS?**

8687 **14.6.2 Converse data in secr?**

14.7 SUMMARY AND OUTLOOK

8688
8689

15

8690
8691

MODELS FOR SEARCH-ENCOUNTER DATA

15.1 SEARCH-ENCOUNTER SAMPLING DESIGNS

15.2 A MODEL FOR SEARCH-ENCOUNTER DATA

(15.2.1)

(15.2.2)

(15.2.3)

8692 15.2.1 Ecological process model

8693 15.2.2 Other stuff

15.3 EXAMPLES

8694 15.3.1 Hard plot boundaries

8695 15.3.2 Analysis of other protocols

15.4 DESIGN 3: AD HOC IMPLEMENTATION OF DESIGN 1.

15.5 CAPRICAILLIE CRAP

8696 15.5.1 model

15.6 DESIGN 4 – NO LOCATION INFO

15.7 SUMMARY AND OUTLOOK

8697
8698
8699

16

OPEN POPULATION MODELS

16.1 INTRODUCTION

- 8700 16.1.1 Overview of Population Dynamics
- 8701 16.1.2 Animal movement related to population demography
- 8702 16.1.3 Basic assumptions of JS and CJS models

16.2 TRADITIONAL JOLLY-SEBER MODELS

- 8703 16.2.1 Data Augmentation for the Jolly-Seber Model
-
-

- 8704 16.2.2 Mist-netting example
- 8705 16.2.3 Shortcomings of the traditional JS models

16.3 SPATIAL JOLLY-SEBER MODELS

(16.3.1)

8706 **16.3.1 Mist-netting example**

16.4 TRADITIONAL CJS MODELS

8707 **16.4.1 Migratory fish example**

16.5 MULTI-STATE CJS MODELS

8708 **16.5.1 Migratory fish example**

16.6 SPATIAL CJS MODELS

8709 **16.6.1 Migratory fish example**

16.7 MOVING ACTIVITY CENTERS

8710 **16.7.1 Migratory Fish Example Notes**

16.8 SUMMARY AND OUTLOOK



8711

Part IV

8712

8713

Super-Advanced SCR Models

8714
8715

8716
8717

17

DEVELOPING MARKOV CHAIN MONTE CARLO SAMPLERS

8718 **17.0.1 Why build your own MCMC algorithm?**

17.1 MCMC AND POSTERIOR DISTRIBUTIONS

(17.1.1)

(17.1.2)

17.2 TYPES OF MCMC SAMPLING

8719 **17.2.1 Gibbs sampling**

(17.2.1)

8720 **17.2.2 Metropolis-Hastings sampling**

8721 **17.2.3 Metropolis-within-Gibbs**

8722 **17.2.4 Rejection sampling and slice sampling**

17.3 MCMC FOR CLOSED CAPTURE-RECAPTURE MODEL MH

17.4 MCMC ALGORITHM FOR MODEL SCR0

8723 **17.4.1 SCR model with binomial encounter process**

8724 **17.4.2 Looking at model output**

8725 **Markov chain time series plots**

8726 **17.4.3 Posterior density plots**

8727 **17.4.4 Serial autocorrelation and effective sample size**

8728 **17.4.5 Summary results**

8729 **17.4.6 Other useful commands**

17.5 MANIPULATING THE STATE-SPACE

17.6 INCREASING COMPUTATIONAL SPEED

8730 **17.6.1 Parallel computing**

8731 **17.6.2 Using C++**

17.7 SUMMARY AND OUTLOOK

8732
8733

8734
8735

18

SPATIAL CAPTURE-RECAPTURE FOR UNMARKED POPULATIONS

**18.1 EXISTING MODELS FOR INFERENCE ABOUT DENSITY IN
UNMARKED POPULATIONS**

18.2 SPATIAL CORRELATION AS INFORMATION

18.3 DATA

18.4 MODEL

(18.4.1)

(18.4.2)

18.5 NORTHERN PARULA EXAMPLE

18.6 IMPROVING PRECISION WITH PRIOR INFORMATION

18.7 DESIGN ISSUES

8736 **18.7.1 How Much Correlation Is Enough?**

8737 **18.7.2 Linear Designs**

8738 **18.7.3 Quadrat counts**

18.8 ALTERNATIVE OBSERVATION MODELS

8739 **18.8.1 Spatial point process models**

18.9 CONCLUSION

8740
8741

19

8742
8743
8744

SPATIAL MARK-RESIGHT MODELS FOR PARTIALLY IDENTIFIABLE POPULATIONS

8747

19.1 BACKGROUND

- 8745 19.1.1 Types of partial ID data
8746 19.1.2 A short history of mark-resight models

(19.1.1)

8748
8749

19.2 KNOWN NUMBER OF MARKED INDIVIDUALS

- 19.2.1 MCMC for a spatial mark-resight model
19.2.2 Binomial encounter model

19.3 UNKNOWN NUMBER OF MARKED INDIVIDUALS

19.4 IMPERFECT IDENTIFICATION OF MARKED INDIVIDUALS

19.5 HOW MUCH INFORMATION DO MARKED AND UNMARKED INDIVIDUALS CONTRIBUTE?

19.6 INCORPORATING TELEMETRY DATA**19.7 SUMMARY AND OUTLOOK**

8750
8751

20

8752
8753

2012: A SPATIAL CAPTURE-RECAPTURE ODYSSEY

20.1 10 THESIS OR DISSERTATION TOPICS

20.2 THREE DIMESIONAL SPACE

20.3 GREGARIOUS SPECIES

8754

Part V

8755

8756

Appendices

8757 **APPENDIX I - USEFUL SOFTWARE AND**
8758 **R PACKAGES**

8760 Throughout this book we have used a suite of software and R packages, all of which are
8761 freely available online. To make life a little easier for you, here we provide you with a list
8762 of all software and R packages, download links and some (hopefully) helpful tips regarding
8763 their installation.

20.4 WINBUGS

8764 Although **WinBUGS** (?) is becoming increasingly obsolete with the faster and more
8765 flexible **OpenBUGS** and **JAGS**, there are still situations in which the program comes in
8766 handy. The .exe file can be downloaded from <http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/contents.shtml>.
8767 On 32 bit machines you can just go ahead and double-click on the .exe file and follow
8768 the installation instructions on the screen. On 64 bit machines, according to the BUGS
8769 project you should download a zip file (from the same page) and unzip it into a folder
8770 of your choice. There are a couple of additional steps to make BUGS run. First, you
8771 need to obtain a key (which is free and valid for life) here: 'http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/WinBUGS14_immortality_key.txt'. The key comes with in-
8772 structions on how to activate it. Second, you need to update the basic **WinBUGS**
8773 version to the most current one (which is from August 2007) following the instructions
8774 given here: 'http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/WinBUGS14_cumulative_patch_No3_06_08_07_RELEASE.txt'. **WinBUGS** is ready to use after quitting and re-
8775 opening it. Remember that **WinBUGS** only runs on Windows machines. Also, there
8776 appears to be a problem installing the program in Vista, although we have no personal
8777 experience with this.
8778

20.4.1 WinBUGS through R

8780 While you can run **WinBUGS** as a standalone application, we recommend you access it
8781 from within **R** using the package **R2WinBUGS** (?), so you can conveniently process your out-
8782 put, make graphs etc. **R2WinBUGS** also allows you to run models in **OpenBUGS** (see be-
8783 low). You can install the package from within **R** directly from a cran mirror. In addition to
8784 the usual package help document (<http://cran.r-project.org/web/packages/R2WinBUGS/R2WinBUGS.pdf>)
8785 you can also download a short manual with some examples (''http://voteweb.com/bayes_beach/R2WinBUGS.pdf'').
8786

20.5 OPENBUGS

8788 **OpenBUGS** is the up-to-date version of **WinBUGS** and can be downloaded here:
 8789 ''<http://www.openbugs.info/w/Downloads>'' (Windows, Mac and Linux versions are
 8790 available). The name '**OpenBUGS**' refers to the software being open source, so users
 8791 do not need to download a license key, like they have to for **WinBUGS** (although the
 8792 license key for **WinBUGS** is free and valid for life). For Windows, install by double-
 8793 clicking on the .exe file and following the instructions on the installer screen. Compared
 8794 to **WinBUGS**, **OpenBUGS** has more built-in functions. The method of how to deter-
 8795 mine the right updater for each model parameter has changed and the user can manually
 8796 control the MCMC algorithm used to update model parameters. Several other changes
 8797 have been implemented in **OpenBUGS** and a detailed list of differences between the two
 8798 **BUGS** versions, can be found at <http://www.openbugs.info/w/OpenVsWin>. We have
 8799 encountered convergence problems with simple scr models in this program. There is an
 8800 extensive help archive for both **WinBUGS** and **OpenBUGS** and you can subscribe to
 8801 a mailing list, where people pose and answer questions of how to use these programs at
 8802 <http://www.mrc-bsu.cam.ac.uk/bugs/overview/list.shtml>

8803 **20.5.1 OpenBUGS through R**

8804 Like **WinBUGS**, **OpenBUGS** can be used as a standalone application or through **R**.
 8805 There are several packages that allow **R** to interface with **OpenBUGS**, all of which can
 8806 be installed directly from a cran mirror:

8807 **R2WinBUGS**: One of the options in the `bugs()` call is `program`, which lets you specify either
 8808 **WinBUGS** or **OpenBUGS**. This is a convenient option because after having worked
 8809 through some of this book you will likely be familiar with the format of `bugs()` output
 8810 and other functions of the **R2WinBUGS** package.

8811 **R2openBUGS**: **R2openBUGS** (?) is very similar to, and actually based on, **R2WinBUGS** and
 8812 it is unclear to us what can be gained by using the former over the latter. Arguments of
 8813 the `bugs()` call differ slightly between the two packages and given that **R2WinBUGS** allows
 8814 for the use of both **OpenBUGS** and **WinBUGS** it is probably easiest to stick with it.

8815 **BRugs**: **BRugs** (?) can be installed from within **R** directly from a cran mirror. In addi-
 8816 tion to the help document at ''<http://www.biostat.umn.edu/~brad/software/BRugs/>
 8817 [BRugs_9_21_07.pdf](http://www.biostat.umn.edu/~brad/software/BRugs/BRugs_9_21_07.pdf)'' there is a **WinBUGS** style manual you can access at ''<http://www.rni.helsinki.fi/openbugs/OpenBUGS/Docu/BRugs%20Manual.html>'' . **BRugs** has
 8818 the convenient feature that all pieces of a **BUGS** analysis can be run from within **R**,
 8819 including checking the model syntax, something that requires opening the **BUGS** GUI
 8820 with other packages.

20.6 JAGS

8822 **JAGS** (Just Another Gibbs Sampler) (?) runs scr models considerably faster than **Win-**
 8823 **BUGS**, does not have the convergence problem with simple scr models we have encoun-
 8824 tered in **OpenBUGS** but similar to the latter program, is flexible and constantly updated.

8825 Writing a **JAGS** model is virtually identical to writing a **WinBUGS** model. However,
8826 some functions may have slightly different names and you can look up available func-
8827 tions and their use in the **JAGS** manual. One potential downside is that **JAGS** can
8828 be very particular when it comes to initial values. These may have to be set as close to
8829 truth as possible for the model to start. Although **JAGS** lets you run several parallel
8830 Markov chains, this characteristic interferes with the idea of using overdispersed initial
8831 values for the different chains. Also, we have found that when running models, sometimes
8832 **JAGS** crashes for unclear reasons, taking **R** down with it. Oftentimes, in order to make
8833 it run again you'll have to go through downloading and installing it again (remove the
8834 non-functioning version first).

8835 **JAGS** has a variety of functions that are not available in **WinBUGS**. For example,
8836 **JAGS** allows you to supply observed data for some deterministic functions of unobserved
8837 variables. In **BUGS** we cannot supply data to logical nodes. Another useful feature is
8838 that the adaptive phase of the model (the burn-in) is run separately from the sampling
8839 from the stationary Markov chains. This allows you to easily add more iterations to the
8840 adaptive phase if necessary without the need to start from 0. There are other, more
8841 subtle differences and there is an entire manual section on differences between **JAGS** and
8842 **OpenBUGS**.

8843 **JAGS** is available for download at '<http://sourceforge.net/projects/mcmc-jags/files/>', together with the R package **rjags** (?), which allows running **JAGS** through **R**,
8844 user and installation manuals and examples. At this site **JAGS** is available for Windows
8845 and Mac; Linux binaries are distributed separately and you can find links to various sources
8846 here: '<http://mcmc-jags.sourceforge.net/>'. **JAGS** comes with a 32 bit and a 64
8847 bit version and can be installed by double-clicking on the .exe file and following the in-
8848 structions on the installer screen. For questions and problems concerning **JAGS** there is a
8849 forum online at <http://sourceforge.net/projects/mcmc-jags/forums/forum/610037>.

8851 20.6.1 JAGS through R

8852 Unlike the two **BUGS** programs, **JAGS** does not have a GUI interface but a command
8853 line interface that can be used to run the program as a standalone application. **JAGS**
8854 will solely perform the MCMC simulation; analyzing and summarizing the output has to
8855 be done outside of **JAGS**. To run **JAGS** through **R** you have two options.

8856 **rjags**: As mentioned above, **rjags** (?) can be found together with **JAGS** and was
8857 developed/is being maintained by the inventor of **JAGS**, which means it is guaranteed
8858 to stay up to date when/as **JAGS** changes. The package can be installed from a cran
8859 mirror and the help document can be accessed at '<http://cran.r-project.org/web/packages/rjags/rjags.pdf>'

8861 **R2jags**: Alternatively, the package **R2jags** (?) provides a means of accessing **JAGS**
8862 through **R**. We prefer **rjags** for the reason named above, as well as because it stores data
8863 in a more memory-efficient way and has better **plot()** and **summary()** methods.

20.7 R

8864 At the time of the preparation of this list, **R** for Windows is at version 2.15.0, which can be
8865 downloaded at <http://cran.r-project.org/bin/windows/base/> This site also contains help-

8866 ful tips on how to install **R** in Windows Vista, how to update **R** packages etc. Installation
8867 of **R** in Windows is straightforward: download the .exe file, double-click on it and follow
8868 the instructions of the Windows installer. The later versions of **R** come with versions for
8869 both 64 bit and 32 bit machines. The **R** site (''<http://mirrors.softliste.de/cran/>'')
8870 has an extensive FAQ section ?, which includes instructions on how to install R on Unix
8871 and Mac computers.

8872 **20.7.1 R packages**

8873 This section provides an alphabetical list of useful **R** packages. There is a large number
8874 of **R** packages and by no means is this list intended to be complete in terms of what is
8875 useful. Rather, we list packages that we are familiar with and that we employ at one point
8876 or the other in this book. Unless explicitly stated otherwise, all packages can be installed
8877 directly from within **R** through a cran mirror.

8878 **adapt:** `adapt` (?) is a package for multidimensional numerical integration. The package
8879 has been removed from the CRAN repository but can be obtained from ''<http://cran.r-project.org/src/contrib/Archive/adapt/>''.

8881 **coda:** `coda` (?) lets you summarize and perform diagnostics on mcmc output. For a
8882 list and description of functions, see the manual at ''<http://cran.r-project.org/web/packages/coda/coda.pdf>''.

8884 **gdistance:** `gdistance` (?) is a package for calculating distances and routes on geographical
8885 grids and can be used to calculate least cost path surfaces. Manual at ''<http://cran.r-project.org/web/packages/gdistance/gdistance.pdf>''..

8887 **igraph:** `igraph` (?) provides routines for graphs and network analysis. Manual at ''<http://cran.r-project.org/web/packages/igraph/igraph.pdf>''.

8889 **inline:** `inline` (?) allows the user to define R functions with in-lined **C**, **C++** or
8890 **Fortran** code. Manual at <http://cran.r-project.org/web/packages/inline/inline.pdf>.

8892 **maps:** `maptools` (?) is a library of maps. Manual at ''<http://cran.r-project.org/web/packages/maps/index.html>''.

8894 **maptools:** `maptools` (?) provides a set of tools for reading and manipulating spatial data,
8895 especially ESRI shapefiles. Manual at ''<http://cran.r-project.org/web/packages/maptools/maptools.pdf>''.

8897 **R2cuba:** `R2cuba` (?) is another package for multidimensional integration. Manual at
8898 ''<http://cran.r-project.org/web/packages/R2Cuba/R2Cuba.pdf>''.

8899 **raster:** `raster` (?) provides functions for geographic analysis and modeling with raster
8900 data. Manual at ''<http://cran.r-project.org/web/packages/raster/raster.pdf>''.

8901 **Rcpp:** `Rcpp` (?) provides R functions as well as a C++ library which facilitate the
8902 integration of R and C++. Manual at <http://cran.r-project.org/web/packages/Rcpp/Rcpp.pdf>.

8904 **RcppArmadillo:** `RcppArmadillo` (?) is a templated C++ linear algebra library, integrating
8905 the **Armadillo** library and R. Manual at <http://cran.r-project.org/web/packages/RcppArmadillo/RcppArmadillo.pdf>.

8907 **reshape:** `reshape` (?) allows you to easily manipulate, summarize and reshape data.
8908 Manual at '<http://cran.r-project.org/web/packages/reshape/reshape.pdf>'.

8909 **rgeos:** `rgeos` (?) provides many useful functions for spatial operations such as intersecting
8910 or buffering spatial features. Manual at '<http://cran.r-project.org/web/packages/rgeos/rgeos.pdf>'.

8912 **SCRbayes:** (?)

8913 **secr:** `secr` (?)

8914 **shapefiles:** `shapefiles` (?) allows you to read and write ESRI shapefiles (i.e. shapefiles
8915 you would use in ArcGIS). Manual at '<http://cran.r-project.org/web/packages/shapefiles/shapefiles.pdf>'.

8917 **snow, snowfall:** `snow` (?) and `snowfall` (?) provide functionality for parallel computing.
8918 The latter is a more user-friendly wrapper around the former. Manuals at
8919 <http://cran.r-project.org/web/packages/snowfall/snowfall.pdf> and <http://cran.r-project.org/web/packages/snow/snow.pdf>.

8921 **sp:** `sp` (?) is a package for plotting, selecting, subsetting etc. spatial data. `sp` and
8922 `spatstat` (see below) are complementary in many ways and data formats can be easily
8923 converted between the two packages. Manual at '<http://cran.r-project.org/web/packages/sp/sp.pdf>'.

8925 **SPACECAP:** (?)

8926 **spatstat:** `spatstat` (?) is an extensive package for analyzing spatial data. We use it,
8927 for example, to generate random points within a state space that cannot be described
8928 as a rectangle but consists of a (or several) arbitrary polygon(s). Manual at '<http://cran.r-project.org/web/packages/spatstat/spatstat.pdf>'.

8930 **unmarked:**

8931 #####

8932 #####

8933 **References**

8934 #####

BIBLIOGRAPHY

8935

8936

- 8937 Aitkin, M. (1991), "Posterior bayes factors," *Journal of the Royal Statistical Society. Series B (Methodological)*, 53, 111–142.
- 8938 Alho, J. M. (1990), "Logistic regression in capture-recapture models," *Biometrics*, 46, 623–635.
- 8941 Alpízar-Jara, R. and Pollock, K. H. (1996), "A combination line transect and capture-recapture sampling model for multiple observers in aerial surveys," *Environmental and Ecological Statistics*, 3, 311–327.
- 8944 Amstrup, S. C., McDonald, T. L., and Manly, B. F. J. (2005), *Handbook of capture-recapture analysis*, Princeton Univ Pr.
- 8946 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.
- 8949 Arnason, A., Schwarz, C., and Gerrard, J. (1991), "Estimating closed population size and number of marked animals from sighting data," *Journal of Wildlife Management*, 55, 716–730.
- 8952 Baddeley, A. and Turner, R. (2005), "Spatstat: an R package for analyzing spatial point patterns," *Journal of Statistical Software*, 12, 1–42, ISSN 1548-7660.
- 8954 Bales, S., Hellgren, E., Leslie Jr, D., and Hemphill Jr, J. (2005), "Dynamics of a recolonizing population of black bears in the Ouachita Mountains of Oklahoma," *Wildlife Society Bulletin*, 33, 1342–1351.
- 8957 Balme, G. A., Slotow, R., and Hunter, L. T. B. (2010), "Edge effects and the impact of non-protected areas in carnivore conservation: leopards in the Phinda–Mkuze Complex, South Africa," *Animal Conservation*, 13, 315–323.
- 8960 Bartmann, R. M., White, G. C., Carpenter, L. H., and Garrott, R. A. (1987), "Aerial Mark-Recapture Estimates of Confined Mule Deer in Pinyon-Juniper Woodland," *The Journal of Wildlife Management*, 51, 41–46.
- 8963 Berger, J., Liseo, B., and Wolpert, R. (1999), "Integrated likelihood methods for eliminating nuisance parameters," *Statistical Science*, 14, 1–28.
- 8965 Best, N. G., Ickstadt, K., and Wolpert, R. L. (2000), "Spatial Poisson Regression for Health and Exposure Data Measured at Disparate Resolutions," *Journal of the American Statistical Association*, 95, 1076.
- 8968 Bibby, C. J. and Buckland, S. T. (1987), "Bias of bird census results due to detectability varying with habitat," *Acta Ecologica*, 8, 103–112.
- 8970 Bibby, C. J., Burgess, N. D., and Hill, D. A. (1992), "Bird census techniques: Academic Press," London, UK, 257.
- 8972 Bivand, R. and Rundel, C. (2011), *rgeos: Interface to Geometry Engine - Open Source (GEOS)*, r package version 0.1-8.

- 8974 Blair, W. (1940), "A study of prairie deer-mouse populations in southern Michigan,"
8975 *American Midland Naturalist*, 24, 273–305.
- 8976 Bolker, B. (2008), *Ecological models and data in R*, Princeton Univ Pr.
- 8977 Borchers, D. L. (2012), "A non-technical overview of spatially explicit capture–recapture
8978 models," *Journal of Ornithology*, 152, 1–10.
- 8979 Borchers, D. L., Buckland, S. T., and Zucchini, W. (2002), *Estimating animal abundance:*
8980 *closed populations*, vol. 13, Springer Verlag.
- 8981 Borchers, D. L. and Efford, M. G. (2008), "Spatially explicit maximum likelihood methods
8982 for capture–recapture studies," *Biometrics*, 64, 377–385.
- 8983 Borchers, D. L., Zucchini, W., and Fewster, R. M. (1998), "Mark-recapture models for
8984 line transect surveys," *Biometrics*, 54, 1207–1220.
- 8985 Boulanger, J. and McLellan, B. (2001), "Closure violation in DNA-based mark-recapture
8986 estimation of grizzly bear populations," *Canadian Journal of Zoology*, 79, 642–651.
- 8987 Bowden, D. C. (1993), "A simple technique for estimating population size. Technical Re-
8988 port 93/12." Tech. rep., Department of Statistics, Colorado State University, Fort
8989 Collins, Colorado, USA.
- 8990 Brooks, S. P., Catchpole, E. A., and Morgan, B. J. T. (2000), "Bayesian Animal Survival
8991 Estimation," *Statistical Science*, 15, 357–376.
- 8992 Buckland, S., Anderson, D., Burnham, K., Laake, J., Borcher, D., and L, T. (2001), *In-
8993 troduction to distance sampling: estimating abundance of biological populations*, Oxford,
8994 UK: Oxford University Press.
- 8995 Buckland, S. T. (2004), *Advanced distance sampling*, Oxford University Press, USA.
- 8996 Burnham, K., Anderson, D., and Laake, J. (1980), "Estimation of density from line tran-
8997 sect sampling of biological populations," *Wildlife monographs*, 3–202.
- 8998 Burnham, K., Anderson, D., White, G., Brownie, C., and Pollock, K. (1987), "Design and
8999 analysis methods for fish survival experiments based on release-recapture," *American
9000 Fisheries Society Monograph 5. American Fisheries Society, Bethesda Maryland.* 1987.
9001 737.
- 9002 Burnham, K. P. and Anderson, D. R. (2002), *Model selection and multimodel inference:
9003 a practical information-theoretic approach*, Springer Verlag.
- 9004 Burnham, K. P. and Overton, W. S. (1978), "Estimation of the size of a closed population
9005 when capture probabilities vary among animals," *Biometrika*, 65, 625.
- 9006 Burt, W. (1943), "Territoriality and home range concepts as applied to mammals," *Journal
9007 of mammalogy*, 24, 346–352.
- 9008 Casella, G. and Berger, R. L. (2002), *Statistical inference*, Duxbury Press.
- 9009 Chandler, R. B. and Royle, J. A. (In press), "Spatially-explicit models for inference about
9010 density in unmarked or partially marked populations," *Annals of Applied Statistics*.
- 9011 Chandler, R. B., Royle, J. A., and King, D. (2011), "Inference about density and tempo-
9012 rary emigration in unmarked populations," *Ecology*, 92, 1429–1435.
- 9013 Cloeber, J., Danchin, E., Dhondt, A., and Nichols, J. (2001), *Dispersal*, Oxford.
- 9014 Converse, S., White, G., Farris, K., and Zack, S. (2006), "Small mammals and forest fuel
9015 reduction: national-scale responses to fire and fire surrogates," *Ecological Applications*,
9016 16, 1717–1729.
- 9017 Converse, S. J. and Royle, J. A. (2012), "Dealing with incomplete and variable detectabil-
9018 ity in multi-year, multi-site monitoring of ecological populations," in *Design and Anal-
9019 ysis of Long-term Ecological Monitoring Studies*, eds. Gitzen, R. R., Millspaugh, J. J.,

- 9020 Cooper, A. B., and Licht, D. S., Cambridge University Press, pp. 426–442.
- 9021 Cooch, E. and White, G. (2006), “Program MARK: a gentle introduction,” *available online with the MARK programme*, 7.
- 9023 Coull, B. A. and Agresti, A. (1999), “The Use of Mixed Logit Models to Reflect Heterogeneity in Capture-Recapture Studies,” *Biometrics*, 55, 294–301.
- 9025 Cressie, N. (1991), *Statistics for spatial data*, Wiley Series in Probability and Mathematical Statistics.
- 9027 Csardi, G. and Nepusz, T. (2006), “The igraph software package for complex network research,” *InterJournal, Complex Systems*, 1695.
- 9029 Dawson, D. K. and Efford, M. G. (2009), “Bird population density estimated from acoustic signals,” *Journal of Applied Ecology*, 46, 1201–1209.
- 9031 DeGraaf, R. M. and Yamasaki, M. (2001), *New England wildlife: habitat, natural history, and distribution*, University Press of New England.
- 9033 Dice, L. R. (1938), “Some census methods for mammals,” *Journal of Wildlife Management*, 2, 119–130.
- 9035 Dixon, P. (2002), “Bootstrap resampling,” *Encyclopedia of environmetrics*.
- 9036 Dorazio, R. M. (2007), “On the choice of statistical models for estimating occurrence and extinction from animal surveys,” *Ecology*, 88, 2773–2782.
- 9038 Dorazio, R. M. and Royle, J. A. (2003), “Mixture models for estimating the size of a closed population when capture rates vary among individuals,” *Biometrics*, 59, 351–364.
- 9040 Durban, J. and Elston, D. (2005), “Mark-recapture with occasion and individual effects: abundance estimation through Bayesian model selection in a fixed dimensional parameter space,” *Journal of agricultural, biological, and environmental statistics*, 10, 291–305.
- 9043 Eddelbuettel, D. and François, R. (2011), “Rcpp: Seamless R and C++ Integration,” *Journal of Statistical Software*, 40, 1–18.
- 9045 Efford, M. (2011), *secr: Spatially explicit capture-recapture models*, R package version 2.3.1.
- 9047 Efford, M. G. (2004), “Density estimation in live-trapping studies,” *Oikos*, 106, 598–610.
- 9048 Efford, M. G., Borchers, D. L., and Byrom, A. E. (2009a), “Density estimation by spatially explicit capture-recapture: likelihood-based methods,” *Modeling demographic processes in marked populations*, 255–269.
- 9051 Efford, M. G. and Dawson, D. K. (2009), “Effect of distance-related heterogeneity on population size estimates from point counts,” *The Auk*, 126, 100–111.
- 9053 — (2010), “SECR for acoustic data,” .
- 9054 Efford, M. G., Dawson, D. K., and Borchers, D. L. (2009b), “Population density estimated from locations of individuals on a passive detector array,” *Ecology*, 90, 2676–2682.
- 9056 Efford, M. G., Dawson, D. K., and Robbins, C. S. (2004), “DENSITY: software for analysing capture-recapture data from passive detector arrays,” *Animal Biodiversity and Conservation*, 27, 217–228.
- 9059 Efford, M. G. and Fewster, R. M. (2012), “Estimating population size by spatially explicit capture-recapture,” *Oikos*.
- 9061 Efford, M. G., Warburton, B., Coleman, M. C., and Barker, R. J. (2005), “A field test of two methods for density estimation,” *Wildlife Society Bulletin*, 33, 731–738.
- 9063 F Dormann, C., M McPherson, J., B Araújo, M., Bivand, R., Bolliger, J., Carl, G., G Davies, R., Hirzel, A., Jetz, W., Daniel Kissling, W., et al. (2007), “Methods to account for spatial autocorrelation in the analysis of species distributional data: a

- 9066 review," *Ecography*, 30, 609–628.
- 9067 Farnsworth, G. L., Pollock, K. H., Nichols, J. D., Simons, T. R., E., H. J., and R.,
9068 S. J. (2002), "A removal model for estimating detection probabilities from point-count
9069 surveys," *The Auk*, 119, 414–425.
- 9070 Fienberg, S. E., Johnson, M. S., and Junker, B. W. (1999), "Classical multilevel and
9071 Bayesian approaches to population size estimation using multiple lists," *Journal of the
9072 Royal Statistical Society of London A*, 163, 383–405.
- 9073 Fiske, I. J. and Chandler, R. B. (2011), "unmarked: An R package for fitting hierarchical
9074 models of wildlife occurrence and abundance," *Journal Of Statistical Software*, 43, 1–23.
- 9075 Foster, R. J. and Harmsen, B. J. (2012), "A critique of density estimation from camera-
9076 trap data," *The Journal of Wildlife Management*, 76, 224–236.
- 9077 François, R., Eddelbuettel, D., and Bates, D. (2011), *RcppArmadillo: Rcpp integration
9078 for Armadillo templated linear algebra library*, r package version 0.2.25.
- 9079 García-Alaníz, N., Naranjo, E. J., and Mallory, F. F. (2010), "Hair-snares: A non-invasive
9080 method for monitoring felid populations in the Selva Lacandona, Mexico," *Tropical
9081 Conservation Science*, 3, 403–411.
- 9082 Gardner, B., Reppucci, J., Lucherini, M., and Royle, J. (2010a), "Spatially explicit in-
9083 ference for open populations: estimating demographic parameters from camera-trap
9084 studies," *Ecology*, 91, 3376–3383.
- 9085 Gardner, B., Royle, J. A., and Wegan, M. T. (2009), "Hierarchical models for estimating
9086 density from DNA mark-recapture studies," *Ecology*, 90, 1106–1115.
- 9087 Gardner, B., Royle, J. A., Wegan, M. T., Rainbolt, R. E., and Curtis, P. D. (2010b),
9088 "Estimating black bear density using DNA data from hair snares," *The Journal of
9089 Wildlife Management*, 74, 318–325.
- 9090 Garshelis, D. L. and Hristienko, H. (2006), "State and provincial estimates of American
9091 black bear numbers versus assessments of population trend," *Ursus*, 17, 1–7.
- 9092 Gelman, A. (2006), "Prior distributions for variance parameters in hierarchical models,"
9093 *Bayesian analysis*, 1, 515–533.
- 9094 Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004), *Bayesian data analysis,
9095 second edition.*, Bocan Raton, Florida, USA: CRC/Chapman & Hall.
- 9096 Gelman, A., Meng, X. L., and Stern, H. (1996), "Posterior predictive assessment of model
9097 fitness via realized discrepancies," *Statistica Sinica*, 6, 733–759.
- 9098 Gelman, A. and Rubin, D. B. (1992), "Inference from iterative simulation using multiple
9099 sequences," *Statistical Science*, 7, 457–511.
- 9100 Geman, S. and Geman, D. (1984), "Stochastic relaxation, Gibbs distributions, and the
9101 Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine
9102 Intelligence*, PAMI-6, 721–741.
- 9103 Genz, A. S., Meyer, M. R., Lumley, T., and Maechler, M. (2007), "The adapt Package. R
9104 package version 1.0-4," .
- 9105 Gilks, W. R., Thomas, A., and Spiegelhalter, D. J. (1994), "A Language and Program for
9106 Complex Bayesian Modelling," *Journal of the Royal Statistical Society. Series D (The
9107 Statistician)*, 43, 169–177, ArticleType: primary-article / Issue Title: Special Issue:
9108 Conference on Practical Bayesian Statistics, 1992 (3) / Full publication date: 1994 /
9109 Copyright 1994 Royal Statistical Society.
- 9110 Gopalswamy, A. M. (2012), "Capture-recapture models, spatially explicit," in *Encyclo-
9111 pedia of Environmentrics Second Edition*, eds. El-Shaarawi, A. H. and Piegorsch, W.,

- 9112 John Wiley and Sons Ltd, Chichester, UK.
- 9113 Gopalaswamy, A. M., Royle, A. J., Hines, J., Singh, P., Jathanna, D., Kumar, N. S., and
9114 Karanth, K. U. (2012a), "Program SPACECAP: software for estimating animal density
9115 using spatially explicit capture-recapture models," *Methods in Ecology and Evolution*,
9116 online early, r package version 1.0.4.
- 9117 Gopalaswamy, A. M., Royle, J. A., Delampady, M., Nichols, J. D., Karanth, K. U., and
9118 Macdonald, D. W. (2012b), "Density estimation in tiger populations: combining infor-
9119 mation for strong inference," *Ecology*, 93, 1741–1751.
- 9120 Greig-Smith, P. (1964), *Quantitative plant ecology*, Butterworths (Washington).
- 9121 Hahn, T., Bouvier, A., and Kiêu, K. (2010), *R2Cuba: Multidimensional Numerical Inte-
9122 gration*, r package version 1.0-6.
- 9123 Hall, R. J., Henry, P. F. P., and Bunck, C. M. (1999), "Fifty-year trends in a box turtle
9124 population in Maryland," *Biological Conservation*, 88, 165–172.
- 9125 Hanski, I. A. (1999), *Metapopulation Ecology*, Oxford Univiversity Press.
- 9126 Hawkins, C. E. and Racey, P. A. (2005), "Low population density of a tropical forest
9127 carnivore, Cryptoprocta ferox: implications for protected area management," *Oryx*, 39,
9128 35–43.
- 9129 Hayes, R. J. and Buckland, S. T. (1983), "Radial distance models for the line transect
9130 method," *Biometrics*, 39, 29–42.
- 9131 Hayne, D. (1950), "Apparent home range of Microtus in relation to distance between
9132 traps," *Journal of mammalogy*, 31, 26–39.
- 9133 Hayne, D. W. (1949), "An Examination of the Strip Census Method for Estimating Animal
9134 Populations," *The Journal of Wildlife Management*, 13, 145–157.
- 9135 Hedley, S. L., Buckland, S. T., and Borchers, D. L. (1999), "Spatial modelling from line
9136 transect data," *Journal of Cetacean Research and Management*, 1, 255–264.
- 9137 Hestbeck, J. B. and Malecki, R. A. (1989), "Mark-resight estimate of Canada Goose
9138 midwinter numbers," *Journal of Wildlife Management*, 53, 749–752.
- 9139 Higdon, D. (1998), "A process-convolution approach to modelling temperatures in the
9140 North Atlantic Ocean," *Environmental and Ecological Statistics*, 5, 173–190.
- 9141 Hobbs, N. T. (2011), "An Ecological Modeler's Primer on JAGS," .
- 9142 Holdenried, R. (1940), "A population study of the long-eared chipmunk (*Eutamias quadri-
9143 maculatus*) in the central Sierra Nevada," *Journal of Mammalogy*, 21, 405–411.
- 9144 Hornik, K. (2011), "The R FAQ," ISBN 3-900051-08-9.
- 9145 Huggins, R. M. (1989), "On the statistical analysis of capture experiments," *Biometrika*,
9146 76, 133.
- 9147 Hurlbert, S. H. (1984), "Pseudoreplication and the design of ecological field experiments,"
9148 *Ecological monographs*, 54, 187–211.
- 9149 Illian, J., Penttinen, A., Stoyan, H., and Stoyan, D. (2008), *Statistical analysis and mod-
9150 elling of spatial point patterns*, Wiley.
- 9151 Ivan, J. (2012), "Density, demography, and seasonal movements of snowshoe hares in
9152 central Colorado," Ph.D. thesis, Colorado State University.
- 9153 Ivan, J., White, G., and Shenk, T. (2013a), "Using auxiliary telemetry information to
9154 estimate animal density from capture-recapture data," *Ecology*.
- 9155 — (2013b), "Using simulation to compare methods for estimating density from capture-
9156 recapture data," *Ecology*.
- 9157 Jackson, R., Roe, J., Wangchuk, R., and Hunter, D. (2006), "Estimating Snow Leop-

- ard Population Abundance Using Photography and Capture-Recapture Techniques," *Wildlife Society Bulletin*, 34, 772–781.
- Jett, D. A. and Nichols, J. D. (1987), "A Field Comparison of Nested Grid and Trapping Web Density Estimators," *Journal of Mammalogy*, 68, 888–892.
- Johnson (2010), "A Model-Based Approach for Making Ecological Inference from Distance Sampling Data," *Biometrics*, 66, 310318.
- Johnson, D. H. (1999), "The insignificance of statistical significance testing," *The journal of wildlife management*, 63, 763–772.
- Kadane, J. B. and Lazar, N. A. (2004), "Methods and criteria for model selection," *Journal of the American Statistical Association*, 99, 279–290.
- Karanth, K. U. (1995), "Estimating tiger *Panthera tigris* populations from camera-trap data using capture–recapture models," *Biological Conservation*, 71, 333–338.
- Karanth, K. U. and Nichols, J. D. (1998), "Estimation of Tiger Densities in India Using Photographic Captures and Recaptures," *Ecology*, 79, 2852–2862.
- (2000), "Ecological status and conservation of tigers in India," *WCS, US Fish and Wildlife Service. Centre for Wildlife Studies. Bangalore, India*, 123.
- (2002), *Monitoring tigers and their prey: a manual for researchers, managers and conservationists in {T}ropical {A}sia*, Bangalore, India: Centre for Wildlife Studies.
- Kass, R. and Wasserman, L. (1996), "The selection of prior distributions by formal rules," *Journal of the American Statistical Association*, 91, 1343–1370.
- Kays, R. W., Slauson, K. M., Long, R. A., MacKay, P., Zielinski, W. J., and Ray, J. C. (2008), "Remote cameras." *Noninvasive survey methods for carnivores*, 110–140.
- Kelly, M., Noss, A., Di Bitetti, M., Maffei, L., Arispe, R., Paviolo, A., De Angelo, C., and Di Blanco, Y. (2008), "Estimating puma densities from camera trapping across three study sites: Bolivia, Argentina, and Belize," *Journal of Mammalogy*, 89, 408–418.
- Kendall, K. C., Stetz, J. B., Boulanger, J., Macleod, A. C., Paetkau, D., and White, G. C. (2009), "Demography and genetic structure of a recovering grizzly bear population," *The Journal of Wildlife Management*, 73, 3–16.
- Kendall, W. L., Nichols, J. D., and Hines, J. E. (1997), "Estimating Temporary Emigration Using Capture-Recapture Data with Pollock's Robust Design," *Ecology*, 78, 563–578.
- Kéry, M. (2010), *Introduction to WinBUGS for Ecologists: Bayesian Approach to Regression, ANOVA, Mixed Models and Related Analyses*, Academic Press.
- Kéry, M. (2011), "Towards the modelling of true species distributions," *Journal of Biogeography*, 38, 617–618.
- Kéry, M., Gardner, B., and Monnerat, C. (2010), "Predicting species distributions from checklist data using site-occupancy models," *Journal of Biogeography*, 37, 1851–1862.
- Kéry, M., Gardner, B., Stoeckle, T., Weber, D., and Royle, J. A. (2011), "Use of Spatial Capture-Recapture Modeling and DNA Data to Estimate Densities of Elusive Animals," *Conservation Biology*, 25, 356–364.
- Kéry, M., Royle, J. A., and Schmid, H. (2005), "Modeling avian abundance from replicated counts using binomial mixture models," *Ecological Applications*, 15, 1450–1461.
- Kéry, M. and Schaub, M. (2012), *Bayesian Population Analysis Using WinBugs*, Academic Press.
- King, R. and Brooks, S. (2001), "On the Bayesian analysis of population size," *Biometrika*, 88, 317–336.
- King, R., Brooks, S., and Coulson, T. (2008), "Analyzing Complex Capture–Recapture

- 9204 Data in the Presence of Individual and Temporal Covariates and Model Uncertainty,"
9205 *Biometrics*, 64, 1187–1195.
- 9206 Knaus, J. (2010), *snowfall: Easier cluster computing (based on snow)*., r package version
9207 1.84.
- 9208 Kohn, M., York, E., Kamradt, D., Haught, G., Sauvajot, R., and Wayne, R. (1999),
9209 "Estimating population size by genotyping faeces," *Proceedings of the Royal Society of*
9210 *London. Series B: Biological Sciences*, 266, 657–663.
- 9211 Krebs, C. J. (1999), *Ecological methodology*, Menlo Park, CA: Benjamin/Cummings.
- 9212 Kucera and Barrett (2011), "missing," *missing*, missing.
- 9213 Kuo, L. and Mallick, B. (1998), "Variable selection for regression models," *Sankhy: The*
9214 *Indian Journal of Statistics, Series B*, 60, 65–81.
- 9215 Laird, N. M. and Ware, J. H. (1982), "Random-effects models for longitudinal data,"
9216 *Biometrics*, 38, 963–974.
- 9217 Langtimm, C. A., Dorazio, R. M., Stith, B. M., and Doyle, T. J. (2011), "New aerial
9218 survey and hierarchical model to estimate manatee abundance," *The Journal of Wildlife*
9219 *Management*, 75, 399–412.
- 9220 Le Cam, L. (1990), "Maximum likelihood: an introduction," *International Statistical Re-*
9221 *view/Revue Internationale de Statistique*, 58, 153–171.
- 9222 Lebreton, J. D., Burnham, K. P., Clobert, J., and Anderson, D. R. (1992), "Modeling
9223 Survival and Testing Biological Hypotheses Using Marked Animals: A Unified Approach
9224 with Case Studies," *Ecological Monographs*, 62, 67–118.
- 9225 Lele, S. R. and Keim, J. L. (2006), "Weighted distributions and estimation of resource
9226 selection probability functions," *Ecology*, 87, 3021–3028.
- 9227 Lele, S. R., Moreno, M., and Bayne, E. (2012), "Dealing with detection error in site
9228 occupancy surveys: What can we do with a single survey?" *Journal of Plant Ecology*.
- 9229 Lewin-Koh, N. J., Bivand, R., contributions by Edzer J. Pebesma, Archer, E., Baddeley,
9230 A., Bibiko, H.-J., Dray, S., Forrest, D., Friendly, M., Giraudoux, P., Golicher, D., Rubio,
9231 V. G., Hausmann, P., Hufthammer, K. O., Jagger, T., Luque, S. P., MacQueen, D.,
9232 Niccolai, A., Short, T., Stabler, B., and Turner, R. (2011), *maptools: Tools for reading*
9233 *and handling spatial objects*, r package version 0.8-10.
- 9234 Link, W. (2003), "Nonidentifiability of Population Size from Capture-Recapture Data
9235 with Heterogeneous Detection Probabilities," *Biometrics*, 59, 1123–1130.
- 9236 Link, W. A. and Barker, R. J. (1994), "Density Estimation Using the Trapping Web
9237 Design: A Geometric Analysis," *Biometrics*, 50, 733–745.
- 9238 —— (2006), "Model weights and the foundations of multimodel inference," *Ecology*, 87,
9239 2626–2635.
- 9240 —— (2010), *Bayesian Inference: With Ecological Applications*, London, UK: Academic
9241 Press.
- 9242 Link, W. A. and Eaton, M. J. (2011), "On thinning of chains in MCMC," *Methods in*
9243 *Ecology and Evolution*, 3, 112–115.
- 9244 Link, W. A., Yoshizaki, J., Bailey, L. L., and Pollock, K. H. (2010), "Uncovering a latent
9245 multinomial: analysis of markrecapture data with misidentification," *Biometrics*, 66,
9246 178–185.
- 9247 Liu and Wu (1999), "Parameter expansion for data augmentation," *J Am Stat Assoc*, 94,
9248 1264–1274.
- 9249 Lukacs, P. M. and Burnham, K. P. (2005), "Estimating population size from DNA-based

- closed capture-recapture data incorporating genotyping error," *Journal of Wildlife Management*, 69, 396–403.
- Lunn, D., Spiegelhalter, D., Thomas, A., and Best, N. (2009), "The BUGS project: Evolution, critique, and future directions," *Statistics in Medicine*, 28, 3049–3067.
- Mace, R., Minta, S., Manley, T., and Aune, K. (1994), "Estimating grizzly bear population size using camera sightings," *Wildlife Society Bulletin*, 22, 74–83.
- MacEachern, S. N. and Berliner, L. M. (1994), "Subsampling the Gibbs sampler," *American Statistician*, 48, 188–190.
- MacKay, P., Smith, D. A., Long, R. A., Parker, M., Long, R. A., MacKay, P., Zielinski, W. J., and Ray, J. C. (2008), "Scat detection dogs," *Noninvasive survey methods for carnivores*, 183–222.
- MacKenzie, D. I., Nichols, J. D., Lachman, G. B., Droege, S., Royle, J. A., and Langtimm, C. A. (2002), "Estimating site occupancy rates when detection probabilities are less than one," *Ecology*, 83, 2248–2255.
- MacKenzie, D. I., Nichols, J. D., Royle, J. A., Pollock, K. H., Bailey, L. L., and Hines, J. E. (2006), *Occupancy estimation and modeling: inferring patterns and dynamics of species occurrence*, Academic Press.
- 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.
- 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.
- 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. 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.
- Millar, R. B. (2009), "Comparison of hierarchical Bayesian models for overdispersed count

- 9296 data using DIC and Bayes' Factors," *Biometrics*, 65, 962–969.
- 9297 Mills, L. S., Citta, J. J., Lair, K. P., Schwartz, M. K., and Tallmon, D. A. (2000), "Es-
9298 timating animal abundance using noninvasive DNA sampling: promise and pitfalls,"
9299 *Ecological Applications*, 10, 283–294.
- 9300 Minta, S. and Mangel, M. (1989), "A Simple Population Estimate Based on Simulation
9301 for Capture-Recapture and Capture-Resight Data," *Ecology*, 70, 1738–1751.
- 9302 Møller, J. and Waagepetersen, R. P. (2004), *Statistical inference and simulation for spatial
9303 point processes*, Chapman & Hall/CRC.
- 9304 Mohr, C. (1947), "Table of equivalent populations of North American small mammals,"
9305 *American midland naturalist*, 37, 223–249.
- 9306 Moldenhauer, R. R. and Regelski, D. J. (1996), "Northern Parula (*Parula americana*),"
9307 in *The Birds of North America Online*, ed. Poole, A., Ithaca, NY: Cornell Lab of
9308 Ornithology.
- 9309 Molinari-Jobin, A., Kéry, M., Marboutin, E., Marucco, F., Zimmermann, F., Molinari, P.,
9310 Frick, H., Wölfl, S., Bled, F., Breitenmoser-Würsten, C., Fuxjäger, C., Huber, T., I.,
9311 K., Kos, I., Manfred Wölfl, M., and Breitenmoser, U. (2013), "Mapping range dynamics
9312 from opportunistic data: spatio-temporal distribution modeling of lynx *Lynx lynx* L.
9313 in the Alps," *Biological Conservation*, xx, xxxx–xxxx.
- 9314 Neal, A., White, G., Gill, R., Reed, D., and Olterman, J. (1993), "Evaluation of mark-
9315 resight model assumptions for estimating mountain sheep numbers," *Journal of Wildlife
9316 Management*, 57, 436–450.
- 9317 Neal, A. K. (1990), "Evaluation of mark-resight population estimates using simulations
9318 and field data from mountain sheep." M.S. thesis, Colorado State University, Fort
9319 Collins, Colorado, USA.
- 9320 Nelder, J. A. and Wedderburn, R. W. M. (1972), "Generalized linear models," *Journal of
9321 the Royal Statistical Society. Series A (General)*, 135, 370–384.
- 9322 Nichols, J. D., Hines, J. E., Sauer, J. R., Fallon, F. W., Fallon, J. E., and Heglund,
9323 P. J. (2000), "A double-observer approach for estimating detection probability and
9324 abundance from point counts," *The Auk*, 117, 393–408.
- 9325 Nichols, J. D. and Karanth, K. U. (2002), "Statistical concepts: assessing spatial distri-
9326 butions," in *Monitoring tigers and their prey: a manual for researchers, managers and
9327 conservationists in Tropical Asia*, eds. Karanth, K. U. and Nichols, J. D., Bangalore,
9328 India: Centre for Wildlife Studies, pp. 29–38.
- 9329 Niemi, A. and Fernández, C. (2010), "Bayesian Spatial Point Process Modeling of Line
9330 Transect Data," *Journal of Agricultural, Biological, and Environmental Statistics*, 15,
9331 327–345.
- 9332 Norris, J. L. and Pollock, K. H. (1996), "Nonparametric MLE under two closed capture-
9333 recapture models with heterogeneity," *Biometrics*, 52, 639–649.
- 9334 O'Brien, T. (2011), "Abundance, density and relative abundance: A conceptual frame-
9335 work," in *Camera traps in animal ecology: methods and analyses*, eds. O'Connell, A.
9336 F. J., Nichols, J. D., and Karanth, U., Tokyo, Japan: Springer Verlag, pp. 71–96.
- 9337 O'Connell, A. F., Nichols, J. D., and Karanth, U. K. (2010), *Camera traps in animal
9338 ecology: Methods and analyses*, Springer.
- 9339 O'Hara, R. and Sillampää, M. (2009), "A review of Bayesian variable selection methods:
9340 what, how and which," *Bayesian Analysis*, 4, 85–118.
- 9341 Otis, D. L., Burnham, K. P., White, G. C., and Anderson, D. R. (1978), "Statistical

- 9342 inference from capture data on closed animal populations," *Wildlife monographs*, 3–
9343 135.
- 9344 Ovaskainen, O. (2004), "Habitat-specific movement parameters estimated using mark–
9345 recapture data and a diffusion model," *Ecology*, 85, 242–257.
- 9346 Ovaskainen, O., Rekola, H., Meyke, E., and Arjas, E. (2008), "Bayesian methods for
9347 analyzing movements in heterogeneous landscapes from mark-recapture data," *Ecology*,
9348 89, 542–554.
- 9349 Parmenter, R. R. and MacMahon, J. A. (1989), "Animal Density Estimation Using a
9350 Trapping Web Design: Field Validation Experiments," *Ecology*, 70, 169–179.
- 9351 Parmenter, R. R., Yates, T. L., Anderson, D. R., Burnham, K. P., Dunnum, J. L., Franklin,
9352 A. B., Friggens, M. T., Lubow, B. C., Miller, M., Olson, G. S., and Others (2003),
9353 "Small-mammal density estimation: A field comparison of grid-based vs. web-based
9354 density estimators," *Ecological Monographs*, 73, 1–26.
- 9355 Pebesma, E. and Bivand, R. (2011), *Package 'sp'*, r package version 0.9-91.
- 9356 Pledger, S. (2004), "Unified maximum likelihood estimates for closed capture–recapture
9357 models using mixtures," *Biometrics*, 56, 434–442.
- 9358 Plummer, M. (2003), "JAGS: A program for analysis of Bayesian graphical models us-
9359 ing Gibbs sampling," in *Proceedings of the 3rd International Workshop on Distributed
9360 Statistical Computing (DSC 2003)*. March, pp. 20–22.
- 9361 — (2009), "rjags: Bayesian graphical models using mcmc. R package version 1.0 3-12," .
- 9362 — (2011), "rjags: Bayesian graphical models using mcmc. R package version 3-5," .
- 9363 Plummer, M., Best, N., Cowles, K., and Vines, K. (2006), "CODA: Convergence Diagnosis
9364 and Output Analysis for MCMC," *R News*, 6, 7–11.
- 9365 Pollock, K. H. (1982), "A Capture-Recapture Design Robust to Unequal Probability of
9366 Capture." *Journal of Wildlife Management*, 46, 752–757.
- 9367 Pollock, K. H., Nichols, J. D., Brownie, C., and Hines, J. E. (1990), "Statistical inference
9368 for capture-recapture experiments," *Wildlife monographs*, 3–97.
- 9369 Reich, B. J., Gardner, B., and Wilting, A. (2012), "A spatial capture-recapture model for
9370 territorial species," *Biometrics in review*, xx, xx–xxx.
- 9371 Robert, C. P. and Casella, G. (2010), *Introducing Monte Carlo Methods with R*, New York,
9372 USA: Springer.
- 9373 Royle, J., Kéry, M., and Guélat, J. (2011a), "Spatial capture-recapture models for search-
9374 encounter data," *Methods in Ecology and Evolution*, 2, 602–611.
- 9375 Royle, J. A. (2004a), "Generalized estimators of avian abundance from count survey data,"
9376 *Animal Biodiversity and Conservation*, 27, 375–386.
- 9377 — (2004b), "N-Mixture Models for Estimating Population Size from Spatially Replicated
9378 Counts," *Biometrics*, 60, 108–115.
- 9379 — (2006), "Site occupancy models with heterogeneous detection probabilities," *Biomet-
9380 rics*, 62, 97–102.
- 9381 — (2008), "Modeling individual effects in the Cormack–Jolly–Seber model: a state–space
9382 formulation," *Biometrics*, 64, 364–370.
- 9383 — (2009), "Analysis of capture-recapture models with individual covariates using data
9384 augmentation," *Biometrics*, 65, 267–274.
- 9385 Royle, J. A. and Chandler, R. B. (2012), "Integrating Resource Selection Information with
9386 Spatial Capture-Recapture," *arXiv preprint arXiv:1207.3288*.
- 9387 Royle, J. A., Chandler, R. B., Gazenski, K. D., and Graves, T. A. (2012a), "Spatial

- 9388 capture-recapture for jointly estimating population density and landscape connectivity,"
9389 *Ecology*, to appear.
- 9390 Royle, J. A., Chandler, R. B., Sun, C., and Fuller, A. (2012b), "Integrating Resource
9391 Selection Information with Spatial Capture-Recapture," *needs updated MEE*.
- 9392 Royle, J. A. and Converse, S. J. (in review), "Hierarchical spatial capture-recapture mod-
9393 els: Modeling population density from replicated capture-recapture experiments," *Ecol-*
9394 *ogy*.
- 9395 Royle, J. A., Converse, S. J., and Link, W. A. (2012c), "Data Augmentation for Hierar-
9396 chical Capture-recapture Models," *arXiv preprint arXiv:1211.5706*.
- 9397 Royle, J. A., Dawson, D. K., and Bates, S. (2004), "Modeling abundance effects in distance
9398 sampling," *Ecology*, 85, 1591–1597.
- 9399 Royle, J. A. and Dorazio, R. M. (2006), "Hierarchical models of animal abundance and
9400 occurrence," *Journal of Agricultural, Biological, and Environmental Statistics*, 11, 249–
9401 263.
- 9402 — (2008), *Hierarchical modeling and inference in ecology: the analysis of data from pop-*
9403 *ulations, metapopulations and communities*, Academic Press.
- 9404 — (2012), "Parameter-expanded data augmentation for Bayesian analysis of capture-
9405 recapture models," *Journal of Ornithology*, 152, S521–S537.
- 9406 Royle, J. A., Dorazio, R. M., and Link, W. A. (2007), "Analysis of multinomial models with
9407 unknown index using data augmentation," *Journal of Computational and Graphical
9408 Statistics*, 16, 67–85.
- 9409 Royle, J. A. and Dubovský, J. A. (2001), "Modeling spatial variation in waterfowl band-
9410 recovery data," *The Journal of wildlife management*, 65, 726–737.
- 9411 Royle, J. A. and Gardner, B. (2011), "Hierarchical models for estimating density from trap-
9412 ping arrays," in *Camera traps in animal ecology: methods and analyses*, eds. O'Connel,
9413 A. F. J., Nichols, J. D., and Karanth, U., Tokyo, Japan: Springer Verlag, pp. 163–190.
- 9414 Royle, J. A., Karanth, K. U., Gopalaswamy, A. M., and Kumar, N. S. (2009a), "Bayesian
9415 inference in camera trapping studies for a class of spatial capture-recapture models,"
9416 *Ecology*, 90, 3233–3244.
- 9417 Royle, J. A. and Kéry, M. (2007), "A Bayesian state-space formulation of dynamic occu-
9418 pancy models," *Ecology*, 88, 1813–1823.
- 9419 Royle, J. A. and Link, W. A. (2006), "Generalized site occupancy models allowing for
9420 false positive and false negative errors," *Ecology*, 87, 835–841.
- 9421 Royle, J. A., Magoun, A. J., Gardner, B., Valkenburg, P., and Lowell, R. E. (2011b),
9422 "Density estimation in a wolverine population using spatial capture–recapture models,"
9423 *The Journal of Wildlife Management*, 75, 604–611.
- 9424 Royle, J. A. and Nichols, J. D. (2003), "Estimating abundance from repeated presence-
9425 absence data or point counts," *Ecology*, 84, 777–790.
- 9426 Royle, J. A., Nichols, J. D., Karanth, K. U., and Gopalaswamy, A. M. (2009b), "A
9427 hierarchical model for estimating density in camera-trap studies," *Journal of Applied
9428 Ecology*, 46, 118–127.
- 9429 Royle, J. A. and Young, K. V. (2008), "A Hierarchical Model For Spatial Capture-
9430 Recapture Data," *Ecology*, 89, 2281–2289.
- 9431 Russell, R. E., Royle, J. A., Desimone, R., Schwartz, M. K., Edwards, V. L., Pilgrim,
9432 K. P., and McKelvey, K. S. (2012), "Estimating abundance of mountain lions from
9433 unstructured spatial samples," *Journal of Wildlife Management*.

- 9434 Rutledge, M. (2012), "XXX to be determined XXX," Ph.D. thesis, North Carolina State
9435 University.
- 9436 Salom-Pérez, R., Carrillo, E., Sáenz, J., and Mora, J. (2007), "Critical condition of the
9437 jaguar *Panthera onca* population in Corcovado National Park, Costa Rica," *Oryx*, 41,
9438 51.
- 9439 Sanathanan, L. (1972), "Estimating the size of a multinomial population," *The Annals of
9440 Mathematical Statistics*, 43, 142–152.
- 9441 Schofield, M. and Barker, R. (2008), "A unified capture-recapture framework," *Journal of
9442 agricultural, biological, and environmental statistics*, 13, 458–477.
- 9443 Schwartz, M. K. and Monfort, S. L. (2008), *Genetic and endocrine tools for carnivore
9444 surveys*, Island Press Washington, DC, USA.
- 9445 Seber, G. A. F. (1982), *The estimation of animal abundance and related parameters*,
9446 Macmillan Publishing Co.
- 9447 Sepúlveda, M. A., Bartheld, J. L., Monsalve, R., Gómez, V., and Medina-Vogel, G. (2007),
9448 "Habitat use and spatial behaviour of the endangered Southern river otter (*Lontra
9449 provocax*) in riparian habitats of Chile: conservation implications," *Biological Conserva-
9450 tion*, 140, 329–338.
- 9451 Sillett, S., Chandler, R. B., Royle, J. A., Kéry, M., and Morrison, S. (2012), "Hierarchical
9452 distance sampling models to estimate population size and habitat-specific abundance
9453 of an island endemic," *Ecological Applications*.
- 9454 Simons, T. R., Pollock, K. H., Wetrotth, J. M., Alldredge, M. W., Pacifici, K., and
9455 Brewster, J. (2009), "Sources of Measurement Error, Misclassification Error, and Bias
9456 in Auditory Avian Point Count Data," in *Modeling Demographic Processes In Marked
9457 Populations*, eds. Thomson, D. L., Cooch, E. G., and Conroy, M. J., Boston, MA:
9458 Springer US, vol. 3, pp. 237–254.
- 9459 Skalski, J. R., Millspaugh, J. J., and Spencer, R. D. (2005), "Population estimation and
9460 biases in paintball, mark-resight surveys of elk," *Journal of Wildlife Management*, 69,
9461 1043–1052.
- 9462 Sklyar, O., Murdoch, D., Smith, M., Eddelbuettel, D., and François, R. (2010), *inline:
9463 Inline C, C++, Fortran function calls from R*, r package version 0.3.8.
- 9464 Smith, M. H., Blessing, R., Chelton, J. G., Gentry, J. B., Golley, F. B., and McGinnis,
9465 J. T. (1971), "Determining density for small mammal populations using a grid and
9466 assessment lines," *Acta Theriologica*, 16, 105–125.
- 9467 Soisalo, M. K. and Cavalcanti, S. M. C. (2006), "Close-up Space in "Radio-telemetry","
9468 *Biological Conservation*, 129, 487–496.
- 9469 Sollmann, R., Furtado, M. M., Gardner, B., Hofer, H., Jacomo, A. T. A., Trres, N. M., and
9470 Silveira, L. (2011), "Improving density estimates for elusive carnivores: Accounting for
9471 sex-specific detection and movements using spatial capture-recapture models for jaguars
9472 in central Brazil," *Biological Conservation*, 144, 1017–1024.
- 9473 Sollmann, R., Gardner, B., Chandler, R. B., Shindle, D., Onorato, D. P., Royle, J. A., and
9474 O'Connell, A. F. (in revision), "Using multiple data sources provides density estimates
9475 for endangered Florida panther," *Journal of Applied Ecology*.
- 9476 Sollmann, R., Gardner, B., Parsons, A., Stocking, J., McClintock, B., Simons, T., Pollock,
9477 K., and O'Connell, A. (2012), "A spatial mark-resight model augmented with telemetry
9478 data," *Ecology*.
- 9479 Sólymos, P., Lele, S., and Bayne, E. (2012), "Conditional likelihood approach for analyzing

- single visit abundance survey data in the presence of zero inflation and detection error," *Environmetrics*, 23, 197–205.
- Spiegelhalter, D., Best, N., Carlin, B., and Van Der Linde, A. (2002), "Bayesian measures of model complexity and fit," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64, 583–639.
- Stabler, B. (2006), *shapefiles: Read and Write ESRI Shapefiles*, r package version 0.6.
- Stanley, T. and Burnham, K. (1999), "A closure test for time-specific capture-recapture data," *Environmental and Ecological Statistics*, 6, 197–209.
- Stoyan, D. and Penttinen, A. (2000), "Recent Applications of Point Process Methods in Forestry Statistics," *Statistical Science*, 15, 61–78.
- Sturtz, S., Ligges, U., and Gelman, A. (2005), "R2WinBUGS: A Package for Running WinBUGS from R," *Journal of Statistical Software*, 12, 1–16.
- Su, Y.-S. and Yajima, M. (2011), *R2jags: A Package for Running jags from R*, r package version 0.02-17.
- Taberlet, P. and Bouvet, J. (1992), "Bear conservation genetics," *Nature*, 358, 197–197.
- Tanner, M. A. and Wong, W. H. (1987), "The calculation of posterior distributions by data augmentation," *J Am Stat Assoc*, 82, 528–540.
- Thomas, A., O'Hara, B., Ligges, U., and Sturtz, S. (2006), "Making BUGS Open," *R News*, 6, 12–17.
- Thomas, M. (1949), "A generalization of Poisson's binomial limit for use in ecology," *Biometrika*, 36, 18–25.
- Thompson, C., Royle, J. A., and Garner, J. (2012), "A framework for inference about carnivore density from unstructured spatial sampling of scat using detector dogs," *The Journal of Wildlife Management*, 76, 863–871.
- Tierney, L., Rossini, A. J., Li, N., and Sevcikova, H. (2011), *snow: Simple Network of Workstations*, r package version 0.3-7.
- Tilman, D. and Kareiva, P. (1997), *Spatial ecology: the role of space in population dynamics and interspecific interactions*, vol. 30, Princeton University Press.
- Tobler, M. W., Hibert, F., Debeir, L., and Hansen, C. (2012), "Density and sustainable harvest estimates for the lowland tapir in the Amazon of French Guiana using a spatial capture-recapture model," *none yet*, none, none.
- Trolle, M. and Kéry, M. (2003), "Estimation of ocelot density in the Pantanal using capture-recapture analysis of camera-trapping data," *Journal of Mammalogy*, 84, 607–614.
- (2005), "Camera-trap study of ocelot and other secretive mammals in the northern Pantanal," *Mammalia*, 69, 409–416.
- Tyre, A. J., Tenhumberg, B., Field, S. A., Niejalke, D., Parris, K., and Possingham, H. P. (2003), "Improving precision and reducing bias in biological surveys: estimating false-negative error rates," *Ecological Applications*, 13, 1790–1801.
- Valiere, N. and Taberlet, P. (2000), "Urine collected in the field as a source of DNA for species and individual identification," *Molecular Ecology*, 9, 2150–2152.
- van Etten, J. (2011), *Package gdistance*, r package version 1.1-2.
- van Etten, R. J. H. . J. (2012), *raster: Geographic analysis and modeling with raster data*, r package version 1.9-67.
- Venables, W. and Ripley, B. (2002), *Modern applied statistics with S*, Springer verlag.
- Venables, W., Smith, D., and Team, R. D. C. (2012), "An introduction to R," .

- 9526 Ver Hoef, J. and Boveng, P. (2007), "Quasi-poisson vs. negative binomial regression: How
9527 should we model overdispersed count data?" *Ecology*, 88, 2766–2772.
- 9528 Ver Hoef, J. M. (2012), "Who Invented the Delta Method?" *The American Statistician*,
9529 66, 124–127.
- 9530 Wallace, R. B., Gomez, H., Ayala, G., and Espinoza, F. (2003), "Camera trapping for
9531 jaguar (*Panthera onca*) in the Tuichi Valley, Bolivia," *Journal of Neotropical Mammal-*
9532 *ogy*, 10, 133–139.
- 9533 Wegan, M., Curtis, P., Rainbolt, R., and Gardner, B. (2012), "Temporal sampling frame
9534 selection in DNA-based capture mark-recapture investigations." *Ursus*, 23, 42 – 51.
- 9535 Wegan, M. T. (2008), "Aversive conditioning, population estimation, and habitat prefer-
9536 ence of black bears (*Ursus americanus*) on Fort Drum Military Installation in northern
9537 New York," Ph.D. thesis, Cornell University, Jan.
- 9538 White, G. (1996), "NOREMARK: population estimation from mark-resighting surveys,"
9539 *Wildlife Society Bulletin*, 24, 50–52.
- 9540 White, G. and Bennetts, R. (1996), "Analysis of frequency count data using the negative
9541 binomial distribution," *Ecology*, 77, 2549–2557.
- 9542 White, G. and Shenk, M. (2001), "Poplation estimation with radio-marked inividualls." in
9543 *Radio tracking adn animal populations*, eds. Millspaugh, J. and Marzluff, J., San Diego,
9544 USA: Academic Press, pp. 329–350.
- 9545 White, G. C., Anderson, D. R., Burnham, K. P., and Otis, D. (1982), *Capture-recapture*
9546 *and removal methods for sampling closed populations*, Los Alamos: Los Alamos National
9547 Laboratory.
- 9548 White, G. C. and Garrot, R. (1990), *Analysis of wildlife radiolocation data*, New York,
9549 USA: Academic Press.
- 9550 White, G. C. and Shenk, T. M. (2000), *Population estimation with radio-marked animals*,
9551 San Diego, California: Academic Press.
- 9552 Whitman, J., Ballard, W., and Gardner, C. (1986), "Home range and habitat use by
9553 wolverines in southcentral Alaska," *The Journal of wildlife management*, 460–463.
- 9554 Wickham and Hadley (2007), "Reshaping data with the reshape package," *Journal of*
9555 *Statistical Software*, 21.
- 9556 Williams, B. K., Nichols, J. D., and Conroy, M. J. (2002), *Analysis and management of*
9557 *animal populations: modeling, estimation, and decision making*, Academic Pr.
- 9558 Wilson, K. R. and Anderson, D. R. (1985a), "Evaluation of a Density Estimator Based
9559 on a Trapping Web and Distance Sampling Theory," *Ecology*, 66, 1185–1194.
- 9560 — (1985b), "Evaluation of Two Density Estimators of Small Mammal Population Size,"
9561 *Journal of Mammalogy*, 66, 13–21.
- 9562 Wolpert, R. L. and Ickstadt, K. (1998), "Poisson/gamma random field models for spatial
9563 statistics," *Biometrika*, 85, 251 –267.
- 9564 Woods, J. G., Paetkau, D., Lewis, D., McLellan, B. N., Proctor, M., and Strobeck, C.
9565 (1999), "Genetic tagging of free-ranging black and brown bears," *Wildlife Society Bul-*
9566 *letin*, 27, 616–627.
- 9567 Wright, J., Barker, R., Schofield, M., Frantz, A., Byrom, A., and Gleeson, D. (2009), "In-
9568 corporating Genotype Uncertainty into Mark–Recapture-Type Models For Estimating
9569 Abundance Using DNA Samples," *Biometrics*, 65, 833–840.
- 9570 Yang, H. C. and Chao, A. (2005), "Modeling Animals' Behavioral Response by Markov
9571 Chain Models for Capture–Recapture Experiments," *Biometrics*, 61, 1010–1017.

- 9572 Yoshizaki, J., Pollock, K. H., Brownie, C., and Webster, R. A. (2009), "Modeling misiden-
9573 tification errors in capture-recapture studies using photographic identification of evolv-
9574 ing marks," *Ecology*, 90, 3–9.
- 9575 Zuur, A. F., Ieno, E. N., Walker, N. J., Saveliev, A. A., and Smith, G. M. (2009), *Mixed*
9576 *effects models and extensions in ecology with R*, Springer Verlag.
- 9577 Zylstra, E., Steidl, R., and Swann, D. (2010), "Evaluating survey methods for monitoring
9578 a rare vertebrate, the Sonoran desert tortoise," *The Journal of Wildlife Management*,
9579 74, 1311–1318.