

David Borchers, Eric Rexstad, Ben Stevenson,
Eric Howe and Greg Distiller

Spatial Capture-Recapture by Maximum Likelihood, with R

October 14, 2016

Springer

Contents

1	Spatial encounter rates and detection probabilities	3
1.1	A simple spatial encounter rate model	3
1.1.1	Spatial capture histories and encounter functions	5
1.1.2	The capture history likelihood component: count data	8
1.2	A simple spatial binary detection model	9
1.2.1	The capture history likelihood component: binary data	10
1.3	Should you model $p(d, T)$ or $\lambda(d)$?	10
1.4	Count and binary data example	11
1.4.1	Example fits to count data	12
1.4.2	Example fits to binary data	16
1.5	Occasions: recaptures across space and time	17
1.6	Strata: no recaptures across space and time	18
1.7	Modelling covariate effects	19
1.7.1	Link functions and linear predictors	19
1.8	Detector effects	21
1.9	Occasion effects	21
1.10	Stratum effects	21
1.11	Effort effects	21
1.12	Individual effects (behaviour and heterogeneity)	21
1.13	Habitat effects	21
	References	21
	Index	23

||||| Updated upstream ===== | Stashed changes

Chapter 1

Spatial encounter rates and detection probabilities

Abstract This chapter concerns models for spatial encounter rates and detection probabilities, and how they are related. It includes details of how encounter rates and detection probabilities can be made to depend on survey effort, individual-level variables, detector-specific variables, occasion-specific variables, and habitat or environmental variables. A key concept for all of these is the expected encounter rate.

1.1 A simple spatial encounter rate model

If an individual's activity centre is far from you, you are less likely to encounter it than if it is near you. This simple fact is central to all SCR models and incorporating it into capture-recapture models is what turns them from non-spatial into *spatial* capture-recapture models. In this chapter we explain how this is done.

A key concept here is the expected **encounter rate** – the number of times an individual is expected to encounter a detector per time unit (day/hour/year/whatever), given the distance of its activity centre from the detector.

Of course, it is not only the individual's distance from the detector that determines the expected encounter rate. It also depends on

1. the detector's ability to detect (some detectors will be better than others),
2. how susceptible the individual in question is to detection (some individuals behave more conspicuously than others),
3. how conducive to detection the prevailing conditions are (individuals may be easier to detect in good weather conditions, for example), and
4. how individuals use their habitat (if there is a lot of unsuitable habitat between a detector and an individual's activity centre, it may venture near the detector less often than otherwise, for example).

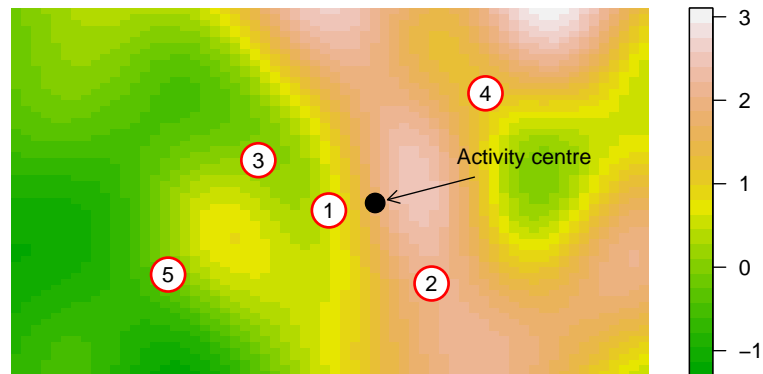
We will deal with each of these sources of difference in expected encounter rate in turn below, but we start by expanding a bit on the idea of a spatial encounter rate.

Firstly, and importantly, by “encounter” we mean an interaction between individual and detector that results in a detection. So if an individual walks in front of a camera trap but is not photographed, no encounter occurred. Or if an individual makes a noise that is not detected by a microphone on an acoustic survey, no encounter occurred.

Need to deal with issue of correlated encounters over short time somewhere (Andy’s book pp249-50).

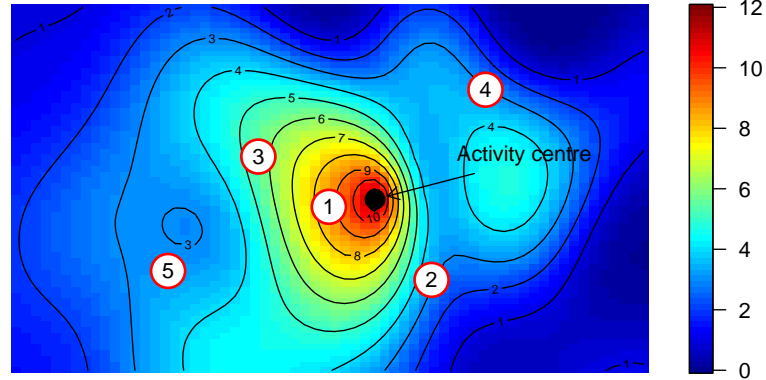
Figures 1.1 and 1.2 illustrate schematically how expected encounter rate varies in space. Figure 1.1 shows the location of five camera traps at various distances from an individual’s activity centre. The shading shows some spatial variable (a habitat variable, for example) that affects encounter rate, with lower values being associated with higher encounter rates. Figure 1.2 is a map of hypothetical encounter rate at all points in space for this individual. This rate is a combination of the individual’s habitat usage (it tends to use habitat with low values of the spatial variable more than those with higher values) and distance from its activity centre (the farther a point is, the less it is used). For simplicity, we assume in this figure that all detectors are equally effective.

Fig. 1.1 Five detectors placed at various distances from an animal’s activity centre. The shading represents habitat type.



We don’t get to see maps like that in Figure 1.2 on a survey, all we see are the detections at each detector, and from these we try to reconstruct something like the map, in addition to estimating animal density. In this chapter we focus on estimation

Fig. 1.2 A plot showing an animal's average space usage over the survey period, expressed in terms of the number of times it is expected to encounter a detector at any point in the region during the survey. The locations of five detectors used on the survey are also shown.



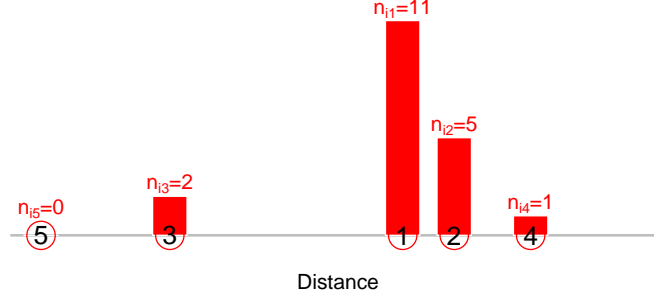
the expected number of encounters as a function of distance from activity centre and leave density estimation aside until Chapter ???. And to keep things simple, we're initially going to (a) ignore habitat and consider expected encounter rate as a function of distance only, and (b) work in one distance dimension rather than two.

1.1.1 Spatial capture histories and encounter functions

In one dimension, detectors are located at points along a line rather than points on a plane. Figure 1.3 shows the five detectors of Figure 1.2 in one dimension, together with the number of times individual i was detected at each detector. These data comprise a spatial capture history for individual i , which we can write as $\omega_i = (n_{i1}, n_{i2}, n_{i3}, n_{i4}, n_{i5}) = (11, 5, 2, 1, 0)$. Spatial capture histories like this (albeit with locations in two dimensions, not one) are all you get from a camera trap survey. Everything we estimate is based on these spatial capture histories.

Although it is nothing like a map of expected encounters, Figure 1.3 actually tells us quite a bit about the expected number of encounters of this individual at any point along the line. Firstly, it tells us something about the location of the individual's activity centre: this is likely nearer detectors 1 and 2 than detector 3, because there were more encounters with 1 and 2 than with 3. So the activity centre is likely

Fig. 1.3 Illustrative example of camera trap data in one dimension. The number of detections of individual i on each of $j = 1, \dots, 5$ cameras is shown, together with the location of each camera.



between detectors 1 and 3, and closer to 1 than to 3. Secondly, it suggests that the individual does not range as far as the distance from detector 1 to detector 5 (because we estimate that it is close to 1 and it had no encounters with detector 5).

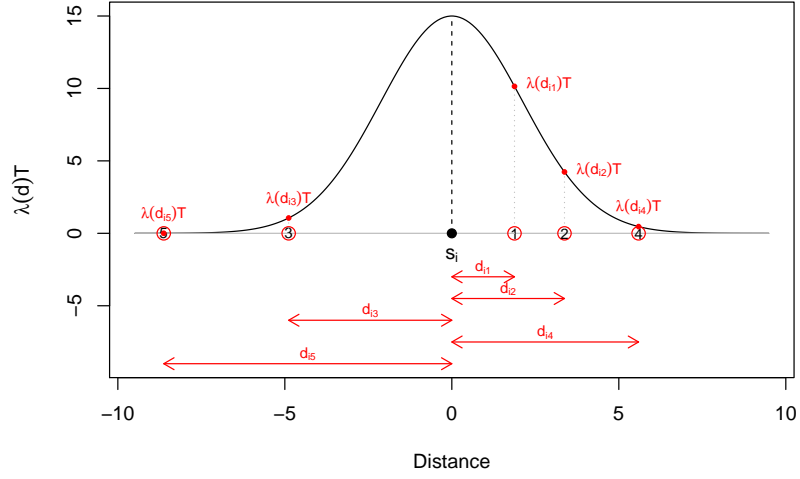
We can formalise these observations by fitting an encounter rate function to the spatial capture history data shown in Figure 1.3. To do this, let's assume that the encounter rate function has the shape of a Gaussian function, centred on an unknown activity centre at \mathbf{s}_i , with an unknown “scale parameter” σ that determines its range, and an unknown intercept λ_0 (the encounter rate with a detector placed at the activity centre). With these assumptions, the expected encounter rate at a point \mathbf{x}_j that is a distance $d_{ij} = \|\mathbf{s}_i - \mathbf{x}_j\|$ from \mathbf{s}_i is

$$\lambda(d_{ij}) = \lambda_0 \exp \left\{ \frac{-d_{ij}^2}{2\sigma^2} \right\}. \quad (1.1)$$

We can estimate λ_0 and σ by fitting this function to the data in Figure 1.3. However, we need to take account of the survey duration to do this since the longer the survey, the higher the number of encounters will be. (Recall that $\lambda(d)$ is an encounter *rate*, i.e. number of encounters *per unit time*.) This is very easily done, since the expected number of encounters is proportional to the time spent surveying: if you survey for twice as long, you expect twice as many encounters. So the expected *number* of encounters on a survey of duration T time units is just $\lambda(d)T$. This is shown in Figure 1.4.

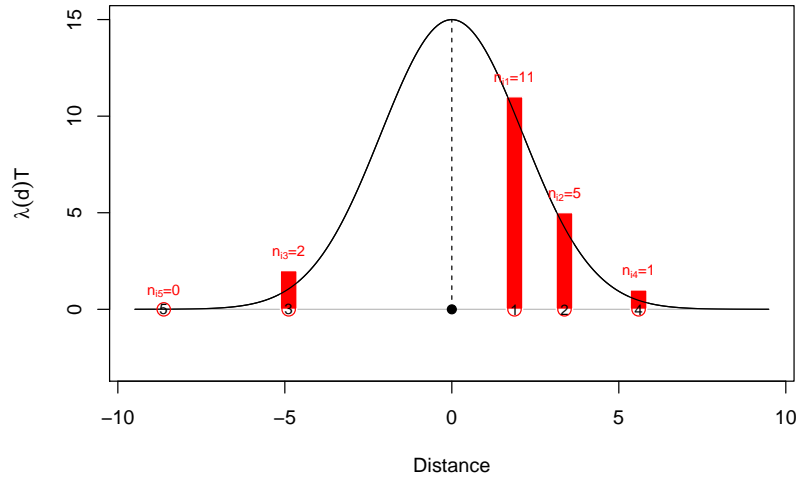
If we fit this function to the observed numbers of encounters shown in Figure 1.3, we get Figure 1.4. In summary, given the locations of detectors and an expected encounter rate model with unknown intercept and range, we can use spatial capture histories to estimate the expected number of encounters of individuals at any distance from their activity centres. This also allows us to estimate the probability of

Fig. 1.4 The encounter rate function $\lambda(d)$ scaled by survey duration T and centred at s_i . Distances from x_i to each camera trap are shown below the curve and the expected number of encounters at each camera trap ($\lambda(d_{ij})T$ to camera trap $j = 1, \dots, 5$) is marked on the curve.



detecting an individual at all, since the probability of detection is just the probability of getting more than zero encounters – see Section ??

Fig. 1.5 The time-scaled encounter rate function $\lambda(d)T$ fitted to the data in Figure 1.3. The black dot is the estimated activity centre location.



We use exactly the same methods to estimate encounter rate functions in two dimensions. In fact, since the distance measure used in the one-dimensional example above was the two-dimensional distance from the activity centre to the detectors, this example was actually two-dimensional - its just that we plotted the encounter function in the distance dimension rather than on the plane.

1.1.2 The capture history likelihood component: count data

The encounter rate model $\lambda(d)$ on its own is not enough to fit it to the observed capture history. An appropriate model also needs to explain why the observed number of encounters differs from the expected number. The model needs a random component.

Now whenever events occur independently at some fixed average rate ($\lambda(d)$ in our case), the number of events (encounters in our case) occurring over a time period T has a Poisson distribution with parameter $\lambda(d)T$. So our model for the observed number of encounters (under the assumption that encounters are independent events) is if individual i 's activity centre is a distance d_{ij} from detector j then n_{ij} , the number of encounters of individual i with detector j in a time period T , is a Poisson random variable with parameter $\lambda(d_{ij})T$. We write this as

$$[n_{ij}|\mathbf{s}_i] = \text{Po}(n_{ij}; \lambda(d_{ij})T). \quad (1.2)$$

(Note that we need the activity centre \mathbf{s}_i to be known in order to know d_{ij} .)

Assuming that detections occur independently at each of J detectors (as might be reasonable for camera trap surveys, or acoustic surveys), the capture history for individual i is distributed as the product of independent Poisson random variables:

$$[\boldsymbol{\omega}_i|\mathbf{s}_i] = \prod_{j=1}^J \text{Po}(n_{ij}; \lambda(d_{ij})T). \quad (1.3)$$

And if encounters by different individuals with detectors are also independent then the set of n observed capture histories $\boldsymbol{\Omega}_n = (\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_n)$, given the activity centres $\mathbf{S}_n = (\mathbf{s}_1, \dots, \mathbf{s}_n)$ of all n detected individuals, is also distributed as the product of independent Poisson random variables, and constitutes

The capture history likelihood component for count data

$$[\boldsymbol{\Omega}_n|\mathbf{S}_n] = \prod_{i=1}^n \prod_{j=1}^J \text{Po}(n_{ij}; \lambda(d_{ij})T) \quad (1.4)$$

This is the probability of observing capture histories $\omega_1, \dots, \omega_n$ for individuals $i = 1, \dots, n$, given that the i th individual's activity centre is at distances d_{i1}, \dots, d_{iJ} from detectors $j = 1, \dots, J$.

1.2 A simple spatial binary detection model

In the above example, which is appropriate for detectors like camera traps or acoustic arrays, the capture history consisted of the number of encounters at each detector. But not all kinds of detectors generate this kind of capture history. Suppose that you did a survey with hair snares rather than camera traps or microphones.

With hair snares you can't tell how often individuals encountered detectors, only whether or not they did encounter them (whether or not there is at least one identifiable hair in the snare). In this case, using 1 to denote encounter and 0 to denote no encounter, the capture history from the example above would be $(1, 1, 1, 1, 0)$ instead of $(11, 5, 2, 1, 0)$. Nevertheless, the Poisson encounter rate model is still applicable, because the probability that individual i encountered detector j is just the probability that n_{ij} is greater than zero, i.e.

$$\mathbb{P}(n_{ij} > 0) = 1 - \mathbb{P}(n_{ij} = 0) = 1 - e^{-\lambda(d_{ij})T}. \quad (1.5)$$

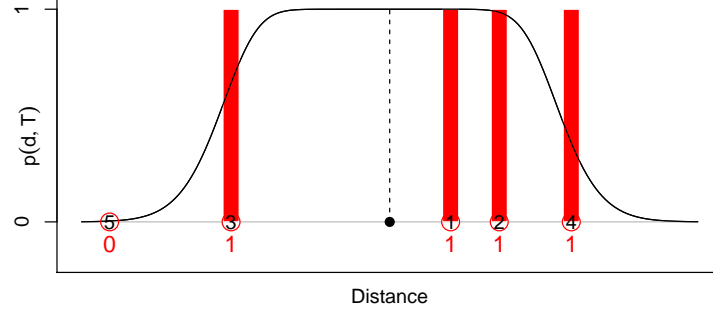
We can write the probability that individual i is detected by (i.e. encounters) detector j as

$$p(d_{ij}, T) = 1 - e^{-\lambda(d_{ij})T}. \quad (1.6)$$

Notice that this probability depends on the survey duration, T . If the same detector is used for different periods of time on two occasions, it will have different detection probabilities on the two occasions, and if identical detectors are used for different periods of time on the same occasion, they will have different detection probabilities. It is important to take account of this when doing SCR inference.

The binary detection function $p(d_{ij}, T)$ for the encounter rate model $\lambda(d_{ij})$ shown in Figure 1.5 is shown in Figure 1.6. Notice that while the encounter rate model was specified to have a Gaussian function shape (see Eqn (1.1) and Figure 1.5), this leads to a detection function $p(d_{ij}, T)$ that does not have a Gaussian function shape (see Eqn (1.6) and Figure 1.6). Conversely, if we specify that the detection function $p(d_{ij}, T)$ has a Gaussian function shape, then the corresponding encounter rate function would not have a Gaussian function shape.

Fig. 1.6 The detection probability function (black curve) corresponding to the encounter function shown in Figure 1.5, with binary capture history given by the bars and the binary data below the x -axis.



1.2.1 The capture history likelihood component: binary data

Using δ_{ij} to indicate whether or not individual i encountered detector j ($\delta_{ij} = 1$ if detected, $\delta_{ij} = 0$ if not), we can write the capture history as $\boldsymbol{\omega}_i = (\delta_{i1}, \dots, \delta_{iJ})$. And assuming that encounters occur independently on each detector, then an individual's capture history is distributed as the product of Bernoulli random variables with "success" probabilities $p(d_{ij}, T)$ ($j = 1, \dots, J$), and we write

$$[\boldsymbol{\omega}_i | \mathbf{s}_i] = \prod_{j=1}^J \text{Bern}(\delta_{ij}; p(d_{ij}, T)). \quad (1.7)$$

And if encounters by different individuals are also independent then the set of n observed capture histories $\boldsymbol{\Omega}_n = (\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_n)$ is distributed as the product of independent Bernoulli random variables:

The capture history likelihood component for independent binary data

$$[\boldsymbol{\Omega}_n | \mathbf{S}_n] = \prod_{i=1}^n \prod_{j=1}^J \text{Bern}(\delta_{ij}; p(d_{ij}, T)) \quad (1.8)$$

1.3 Should you model $p(d, T)$ or $\lambda(d)$?

It doesn't matter. You can model either. Eqn (1.6) gives us a way of converting between encounter rate $\lambda(d)$ and detection probability $p(d, T)$:

$$p(d, T) = 1 - e^{-\lambda(d)T} \quad (1.9)$$

or making $\lambda(d)$ the subject of the equation:

$$\lambda(d) = \frac{-\log \{1 - p(d, T)\}}{T}. \quad (1.10)$$

So if you have hair snare data, and choose to specify a model for $\lambda(d)$ (e.g. specify that it is has a Gaussian function shape), then given the survey duration T , this is easily converted into a model for $p(d, T)$ using Eqn (1.9). Or if you have camera trap data, you can choose to specify a model for $p(d, T)$ for a given T (e.g. specify that it is has a Gaussian function shape for this T), and this is easily converted into an encounter rate model $\lambda(d)$ using Eqn (1.10).

For example if you specified that when $T = 1$, $p(d, T)$ has a Gaussian shape with intercept paramter g_0 (the probability of detection when the activity centre is at the detector) and range parameter σ :

$$p(d, T = 1) = g_0 \exp \left\{ \frac{-d^2}{2\sigma^2} \right\}. \quad (1.11)$$

then the corresponding $\lambda(d)$ is

$$\lambda(d) = \frac{-\log \{1 - p(d, T = 1)\}}{1} = -\log \left\{ 1 - g_0 \exp \left\{ \frac{-d^2}{2\sigma^2} \right\} \right\} \quad (1.12)$$

Note, however, that if $p(d, T)$ has some specific shape (Gaussian, for example) for a survey duration T , it will *not* have this shape for any other T . So if you specify that $p(d, T)$ has a Gaussian shape for your default survey duration and then some detectors fail halfway through the survey, these detectors will not have a Gaussian detection function shape. This is usually not a big deal because (a) unless T changes a lot, the shapes will not be very different, and (b) you never know the true shape of $p(d, T)$ for any T anyway. But it is as well to be aware of that the shape of $p(d, T)$ depends on T .

We usually have very little idea what the detection or encounter rate function shape should be. We have a range of functional forms for each and choosing between them is a model selection question. Whether it is better to specify a form for $p(d, T)$ or for $\lambda(d)$ is also a model selection question.

1.4 Count and binary data example

To illustrate the application of the above ideas, we will use the package `secr` to fit simple models to count data from a camera trap survey of leopards in the Kruger National Park, and to a binary version of these data.

Need to introduce this dataset in the introduction (?), and acknowledge source.

Here we focus on the expected encounter rate function and detection function specification. To fit an SCR model you also need to specify (a) a model for activity centre density and distribution, and (b) because activity centres are unobserved, you need to specify a region within which the activity centres of the detected individuals could be, and outside of which they could not be. For (a) will use the default density model here, which assumes unknown but equal density of activity centres everywhere, and we will use a pre-prepared “mask” (an `secr` term) that specifies the region (b). Density models are dealt with in detail in Chapter ?? and masks are explained further in Section ??.

The necessary data and associated fitted objects are obtained by loading the library `scrmlebook` package and getting the data and other R objects for this chapter, as follows:

```
library(scrmlbook)
data("ERdetfun-data")
data("ERdetfun-fits")
```

The first data command loads the count capture history in R object `kruger.capt`, the corresponding binary capture history in `kruger.capt.bin` and the mask object in `kruger.mask`. The second loads all the models that are fitted in this chapter (the code some of which is given below). These are all objects that were created using the `secr` library. Details of the data and the fits can be viewed in the `scrmlebook` help files as follows

```
help("ERdetfun-data")
help("ERdetfun-fits")
```

1.4.1 Example fits to count data

To specify an encounter rate model we need to specify the functional form to use and to say how its parameters depend on covariates. We postpone consideration of covariates until later and so use the syntax `1`, which indicates no dependence on covariates.

In `secr`-speak a model with Gaussian shape is called a “half-normal” model (a name derived from the distance sampling literature), and encounter rate functions are called “hazard” functions (a name that derives from survival analysis literature). So specifying that the model is “HHN” in `secr`-speak, is specifying that it is an encounter rate model (the first H is for “hazard”) with half-normal shape (“HN”). And as `secr`-speak for λ_0 and σ is “lambda0” and “sigma”, the specification for an encounter rate model with Gaussian shape is as shown below.

We fit this model using the `secr` function `secr.fit`, by passing the function the capture history object `kruger.fit` (which contains in it details of detector locations), the model specification in the form of `er.model` and `detfn` below, and information on the region that could contain the activity centres of detected individuals (`kruger.mask`).

insert citation?


```
detfn="HHN"
er.hn.model=list(lambda0~1,sigma~1)
er.hn.fit=secr.fit(kruger.capt,model=er.model,mask=kruger.mask,
                  detectfn=detfn)
```

We will also fit an encounter rate model with “hazard rate” shape. This is defined as follows

$$\lambda(d) = \lambda_0 \left(1 - \exp \left\{ - \left[\frac{d}{\sigma} \right]^{-b} \right\} \right). \quad (1.13)$$

It has one more parameter (b) than a Gaussian model and is hence more flexible. Its shape can be seen in Figure 1.7.

An encounter rate model with hazard rate shape is called “HHR” in *secr*-speak, and we can fit it as follows:

```
detfn="HHR"
er.hr.model=list(lambda0~1,sigma~1)
er.hr.fit=secr.fit(kruger.capt,model=er.model,mask=kruger.mask,
                  detectfn=detfn)
```

The Gaussian and hazard models can be compared on the basis of AIC_c as follows

```
aics = AIC(er.hn.fit,er.hr.fit)
aics[,c("model", "detectfn", "dAICc", "AICcwt")]

##              model              detectfn dAICc AICcwt
## er.hr.fit D~1 lambda0~1 sigma~1 z~1 hazard hazard rate 0.000 0.6735
## er.hn.fit  D~1 lambda0~1 sigma~1 hazard halfnormal 1.448 0.3265
```

How to put in neat table?

With a ΔAIC_c of only about 1.45, this suggests that the hazard rate model is slightly better than the Gaussian model. Point estimates and upper and lower 95% confidence interval bounds for density, λ_0 , σ and the hazard rate parameter b (called z in *secr*-speak) are shown in Table ??.

```
predict(er.hr.fit)[,c("estimate", "lcl", "ucl")]

##          estimate          lcl          ucl
## D      2.015677e-04 1.303595e-04 3.116729e-04
## lambda0 1.910467e+00 5.875132e-01 6.212428e+00
## sigma   1.262702e+03 5.032888e+02 3.167993e+03
## z        2.112622e+00 1.654037e+00 2.698352e+00
```

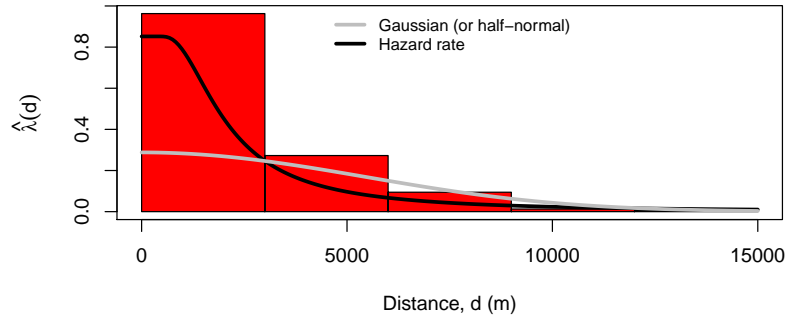
how make this into a table?

A plot similar to Figure 1.5 would show how the capture frequencies have informed the encounter rate model, as explained above. However, because we have 62 detectors here rather than just 5, a plot like Figure 1.5 is too messy to be useful. So instead of using exact distances to detectors, we will use distance intervals and count the average number of detections for all detectors in each distance interval. And because we are operating in two spatial dimensions, negative and positive distances are not well defined, so we will do a plot only on positive distances (as if the

negative distances had been folded over onto the positive ones). Doing this with the `scrmlbook` library function `plot.er.fit`, we get Figure 1.7.

```
plot.er.fit(er.hr.fit,dmax=15000,binwidth=1000,lwd=3,
            xlab="Distance, d (m)",ylab=expression(hat(lambda)(d)))
plot(er.hn.fit,xval=seq(0,15000,length=200),lwd=3,add=TRUE,col="gray")
legend("top",legend=c("Gaussian (or half-normal)", "Hazard rate"),
      col=c("gray", "black"),lwd=3,bty="n",cex=0.8)
```

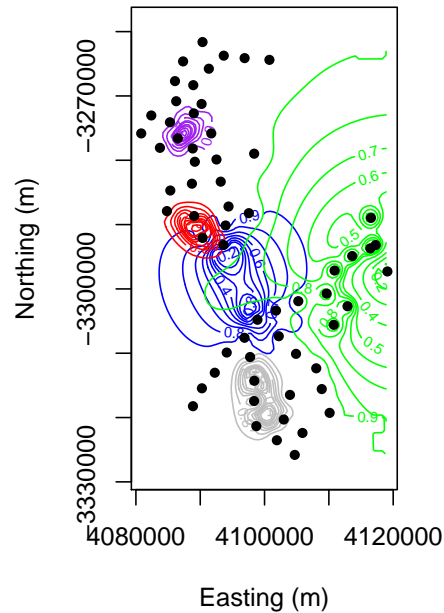
Fig. 1.7 Histogram of average number of detections per detector against distance of detectors from estimated activity centre locations, by 3,000m distance interval. The estimated Gaussian and hazard-rate encounter rate models are plotted over the histogram.



The Gaussian model seems to be a very poor fit to the average encounter data, and in this plot looks a lot worse fit than the hazard rate model. However, we need to be cautious about over-interpreting this figure because it is based on the *estimated* locations of activity centres, and these have substantial uncertainty associated with them, as shown in Figure 1.8. This uncertainty is not incorporated into Figure 1.7. We can't really conclude a lot more from this plot than that the encounter rate does decline with distance, as expected, and that animals were very unlikely to venture more than about 15km from their activity centres over the survey period.

Interestingly, the estimated leopard density from the hazard rate model is only about 5% smaller than that from the Gaussian model, despite their very different shapes (Figure 1.7). This is a fairly typical feature of SCR models: different models can give quite different encounter rate (or detection function) shapes but very similar density estimates. The root cause of this is that we have rather poor information on the distances of detectors from activity centres because we do not observe the activity centres, and as a result there is relatively little information in the data to tell us about the shape of the encounter rate function or detection function. There is usually quite good information on *area under the encounter rate or detection function*. In our case the areas under the Gaussian and hazard rate curves differ by only 6%.

Fig. 1.8 Kruger leopard study camera traps (black dots) and estimated probability contours of the activity centres of 5 detected individuals. (The average distance between the cameras is 3264m.)



A moral of this story is “Be careful not to over-interpret encounter rate or detection functions.”

We could also specify Gaussian or hazard-rate forms for the detection function rather than the encounter rate function. For example, a detection function with hazard rate form is fitted thus:

```
detfn="HR"
df.hr.model=list(lambda0~1,sigma~1)
df.hr.fit=seccr.fit(kruger.capt,model=df.model,mask=kruger.mask,
                    detectfn=detfn)
```

Note that this detection function applies over the duration of the survey. So we are implicitly saying that $T = 1$ for the duration of the survey, i.e., that time units are “this-survey-length”. Specifying the same detection function for a survey of a different length would be inconsistent.

Having fitted models with both Gaussian and hazard-rate forms for the encounter rate function and the detection function, we can compare models using AIC_c again:

```
aics = AIC(er.hn.fit,er.hr.fit,df.hn.fit,df.hr.fit)
aics[,c("model", "detectfn", "dAICc", "AICcwt")]
```

##	model	detectfn	dAICc	AICcwt
----	-------	----------	-------	--------

How to put in neat table?

```
## er.hr.fit D~1 lambda0~1 sigma~1 z~1 hazard hazard rate 0.000 0.3375
## df.hr.fit D~1 g0~1 sigma~1 z~1 hazard rate 0.119 0.3180
## df.hn.fit D~1 g0~1 sigma~1 halfnormal 1.246 0.1810
## er.hn.fit D~1 lambda0~1 sigma~1 hazard halfnormal 1.448 0.1636
```

It is apparent from this that it makes very little difference whether you specify the shape of the encounter rate function or the shape of the detection function. The main difference is that the shape of the encounter rate function is independent of the survey duration, whereas the shape of the detection function changes with survey length. This makes the encounter rate function a “common currency” for all survey durations.

1.4.2 Example fits to binary data

Here we suppose that instead of camera-traps, the detectors on the kruger leopard survey were hair snares (and that identification was by way of DNA). This leads to some loss of information because instead of counts we get only binary detected/undetected data at each detector.

We can fit models to these binary capture history data (in `kruger.capt.bin`) much as we did with the count data, but having specified that detector are what `secr` calls “proximity” detectors when we created the capture history object. You can see what kind of detectors are in a capture history object thus:

```
detector(kruger.cams)
## [1] "count"
detector(kruger.cams.bin)
## [1] "proximity"
```

Here is code to fit an encounter rate function model with hazard rate form to the binary data, for example:

```
detfn="HHN"
er.hn.model=list(g0~1,sigma~1)
er.bin.hn.fit=secr.fit(kruger.capt.bin,model=er.bin.hn.model,
                      mask=kruger.mask,detectfn=detfn)
```

And to compare various models fitted to these binary data:

```
aics = AIC(er.bin.hn.fit,er.bin.hr.fit,df.bin.hn.fit,df.bin.hr.fit)
aics[,c("model","detectfn","dAICc","AICcwt")]

## model detectfn dAICc AICcwt
## df.bin.hn.fit D~1 g0~1 sigma~1 halfnormal 0.000 0.2846
## er.bin.hn.fit D~1 lambda0~1 sigma~1 hazard halfnormal 0.016 0.2823
## df.bin.hr.fit D~1 g0~1 sigma~1 z~1 hazard rate 0.529 0.2184
## er.bin.hr.fit D~1 lambda0~1 sigma~1 z~1 hazard hazard rate 0.563 0.2147
```

1.5 Occasions: recaptures across space and time

A notable feature of the example above was that the word “occasion” was not used. This might be surprising to you if you have a capture-recapture (CR) background since occasions are absolutely central to conventional (non-spatial) CR. In non-spatial CR, recaptures occur on different occasions in time and you can’t get recaptures unless you have occasions. This is not true of SCR, where you can get *recaptures over space* rather than, or in addition to, recaptures over time.

Instead of detections (or not) on each of a number of occasions in *time*, we had detections (or not) at each of a number of detectors in *space* in the example above. This allowed us to estimate the expected number of encounters (and hence detection probability, since detection is just at least one encounter) when we have only a single sampling occasion. But you can’t do this if your detectors are traps that hold animals when they detect them.

We delay dealing with such traps until Chapter ???. But here we extend the above models to include occasions, which we index by $k = 1, \dots, K$. The count and binary likelihood functions of Eqns (1.4) and (1.8) are easily extended to include multiple occasions like this:

The capture history likelihood component for count data (multi-occasion)

$$[\boldsymbol{\Omega}_n | \mathbf{S}_n] = \prod_{i=1}^n \prod_{j=1}^J \prod_{k=1}^K \text{Po}(n_{ijk}; \lambda(d_{ijk})T_k) \quad (1.14)$$

and this:

The capture history likelihood component for independent binary data (multi-occasion)

$$[\boldsymbol{\Omega}_n | \mathbf{S}_n] = \prod_{i=1}^n \prod_{j=1}^J \prod_{k=1}^K \text{Bern}(\delta_{ijk}; p(d_{ijk}, T_k)) \quad (1.15)$$

where

n_{ijk} is the number of detections of individual i at detector j on occasion k ,
 δ_{ijk} is 1 if individual i was detected at detector j on occasion k ,
 d_{ijk} is the distance of the activity centre of individual i from detector j on occasion k . Note that this is a latent variable - we do not observe it.
 T_k is the duration of the k th occasion.

1.6 Strata: no recaptures across space and time

It is sometimes useful to analyse multiple survey units that might share common encounter rate or detection function parameters, but between which recaptures cannot occur. These might be distinct spatial units (regions separated by larger distances than individuals would travel over the survey period, for example), or they might be surveys of the same region at times that are so far apart that no marks from one occasion are identifiable on other occasions. In survey parlance these are commonly called “strata”. In *secur*-speak they are “sessions”. (Royle *et al.* (2014) use both “session” and “occasion” to refer to what we call “occasion” in this book.)

Can't get citations working!

In this case, each stratum is like a separate mini-survey that could be analysed on its own, without the other strata. But if some parameters may be shared across strata (if they might have a common σ , for example), then the analysis must combine the strata. Because no recaptures (as opposed to parameters) are shared across strata, the capture histories are independent between strata and the corresponding multi-stratum likelihood is just the product of the likelihoods for each stratum. Indexing stratum by $l = 1, \dots, L$, the stratified count and binary likelihood functions are:

The capture history likelihood component for count data (multi-occasion, stratified)

$$[\mathbf{\Omega}_n | \mathbf{S}_n] = \prod_{l=1}^L \prod_{i=1}^{n_l} \prod_{k=1}^{K_l} \prod_{j=1}^{J_{kl}} \mathbf{Po}(n_{ijkl}; \lambda(d_{ijkl})T_{kl}) \quad (1.16)$$

and this:

The capture history likelihood component for independent binary data (multi-occasion, stratified)

$$[\mathbf{\Omega}_n | \mathbf{S}_n] = \prod_{l=1}^L \prod_{i=1}^{n_l} \prod_{k=1}^{K_l} \prod_{j=1}^{J_{kl}} \mathbf{Bern}(\delta_{ijkl}; p(d_{ijkl}, T_{kl})) \quad (1.17)$$

where

n_{ijk} is the number of detections of individual i at detector j on occasion k ,
 δ_{ijkl} is 1 if individual i in stratum l was detected at detector j on occasion k in that stratum,
 d_{ijkl} is the distance of the activity centre of individual i in stratum l from detector j on occasion k in that stratum. Note that this is a latent variable - we do not observe it.
 T_{kl} is the duration of the k th occasion in the l th stratum.
 n_l is the number of individuals detected in stratum l .
 J_{kl} is the number of detectors used on occasion k in stratum l .
 K_l is the number of sampling occasions in stratum l .

1.7 Modelling covariate effects

As noted at the beginning of this chapter, encounter rates and detection probabilities often depend on variables associated with the detectors, the environment, the individual animals and other things. We need to be able to model this dependence. While there is no single correct way to do this, there are two ways in which it is commonly done:

- by making the intercept of the encounter rate function (λ_0) or the detection function (g_0) depend on covariates and/or
- by making the scale parameter (σ) of the encounter rate function or the detection function depend on covariates.

1.7.1 Link functions and linear predictors

We make parameters depend on covariates by writing them as functions of the covariates. One might, for example, think of making the probability g_0 depend on say the elevation, x , at which the detector is placed, in this way: $g_0 = \beta_0 + \beta_1 x$, where β_0 and β_1 are unknown parameters. This would not be a good idea because there is nothing in this equation to stop g_0 going out of bounds (above 1 or below 0). Link functions are used to keep parameters within their bounds. As with generalised linear models (GLMs), the link function relates a “linear predictor” (often denoted η , and equal to $\beta_0 + \beta_1 x$ in the example above) to the parameter of interest (g_0 in this example).

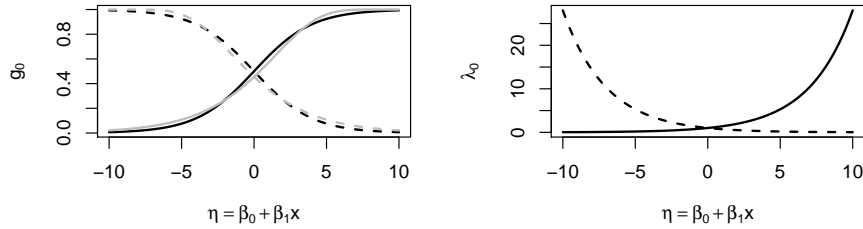
In the case of g_0 we need a link function that keeps it between 0 and 1, while in the case of λ_0 (a rate, and hence bounded below by zero but unbounded above) and σ (also bounded below by zero but unbounded above). Common choices are a logit

link or complimentary log-log link (“cloglog” link for short) for g_0 , and log links for λ_0 and σ .

By convention, link functions specify η as a function of (in our case) g_0 , λ_0 or σ . In our context, it is easier to interpret the inverse link functions, which give g_0 , λ_0 or σ as functions of the linear predictor η . Examples of the inverse link function forms as functions of a linear predictor $\eta = \beta_0 + \beta_1 x$ are shown for positive and negative β_1 in Figure 1.9. The equations for the inverse link functions are as follows:

$$\begin{aligned} g_0 &= \frac{e^\eta}{1+e^\eta} && \text{(inverse logit link)} \\ g_0 &= 1 - \exp\{e^{-\eta}\} && \text{(inverse cloglog link)} \\ \lambda_0 &= e^\eta && \text{(inverse log link)} \\ \sigma &= e^\eta && \text{(inverse log link)} \end{aligned}$$

Fig. 1.9 Inverse logit and cloglog (left) and log (right) link functions of a linear predictor $\eta = \beta_0 + \beta_1 x$. Solid lines are for positive β_1 while dashed lines are for negative β_1 . The left plot shows the inverse logit link in black and the inverse cloglog link in grey.



Dependence on more covariates is modelled by adding terms to the linear predictor: $\eta = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_m x_m$. The `secr` syntax for this follows the syntax for GLMs in R:

```
g0    x1 + x2 + ... + xm
lambda0  x1 + x2 + ... + xm
sigma  x1 + x2 + ... + xm
```

The `secr` package automatically selects the log link for σ and λ_0 , and the logit link for g_0 . Interaction terms and main effects with interactions can be added using “.” and “*”, respectively, as with GLMs in R.

Should we mention regression splines here, and add a plot explaining?

1.8 Detector effects

When there are differences between detectors that may make some more efficient than others ... (Hmmm, it is the data structure that is different between detector effects, occasion effects and stratum effects, not the model specification. Maybe these three sections need to be a running example, with code since there is not really anything methodological to say?)

1.9 Occasion effects

1.10 Stratum effects

1.11 Effort effects

The encounter rate function is the key to adjusting for changes in effort between traps, occasions or sessions. With it we already have a link function between the expected encounter rate $\lambda(d)$ or detection probability $p(d)$ on the one hand, and the time a detector operated (T_{jkl} for detector j on occasion j in stratum l) on the other:

1.12 Individual effects (behaviour and heterogeneity)

1.13 Habitat effects

References

Royle, J. A., R. B. Chandler, R. Sollmann, and B. Gardner (2014). *Spatial Capture-Recapture*. Academic Press.

Index

Animal-level variables, *see also* Heterogeneity
Capture function, *see* Detection function
Capture probability, *see* Detection probability
Detection probability, *see also* Detection function
Encounter rate, 3
Expected encounter rate, *see* encounter rate
Hazard rate, *see also* encounter rate
Survey-level variables, *see also* Heterogeneity