Introduction to distance sampling

Workshop, 21-23 August 2019

Centre for Research into Ecological and Environmental Modelling

Exercise 9. Analysis with multipliers

We consider indirect methods to estimate abundance and hence include multipliers in the abundance calculations. The first problem uses data from a dung survey of deer and there are two levels of multipliers that need to be accounted for (dung production rate and dung decay rate). Problems 2 and 3 deal with instantaneous cues and so only cue rate needs to be taken into account.

1 Objectives

The objectives of this exercise are to

1. Fit detection functions to cues
2. Obtain relevant multipliers
3. Use the multipliers in the `dht2` function to obtain animal abundances.

2 Dung survey of deer

The question is how to estimate of the density of sika deer in a number of woodlands in the Scottish Borders. These animals are shy and will be aware of the presence of an observer before the observer detects them, making surveys of this species challenging. As a consequence, indirect estimation methods have been applied to this problem. In this manner, an estimate of density is produced for some sign generated by deer (in this case, faecal or dung pellets) and this estimate is transformed to density of deer (D_{deer} by

$$\hat{D}_{\text{deer}} = \frac{\text{dung deposited daily}}{\text{dung production rate (per animal)}}$$

where the dung deposited daily is given by

$$\text{dung deposited daily} = \frac{\hat{D}_{\text{pellet groups}}}{\text{mean time to decay}}$$

Hence, we use distance sampling to produce a pellet group density estimate, then adjust it accordingly to account for the production and decay processes operating during the time the data were being acquired. We will also take uncertainty in the dung production and decay rates into account in our final estimate of deer density.

Data from 9 woodlands (labelled A-H and J) were collected according to the survey design (Figure 1) but note that data from block D were not included in this exercise.

In addition to these data, we also require estimates of the production rate. From a literature search, we learn that sika deer produce 25 pellet groups daily but this source did

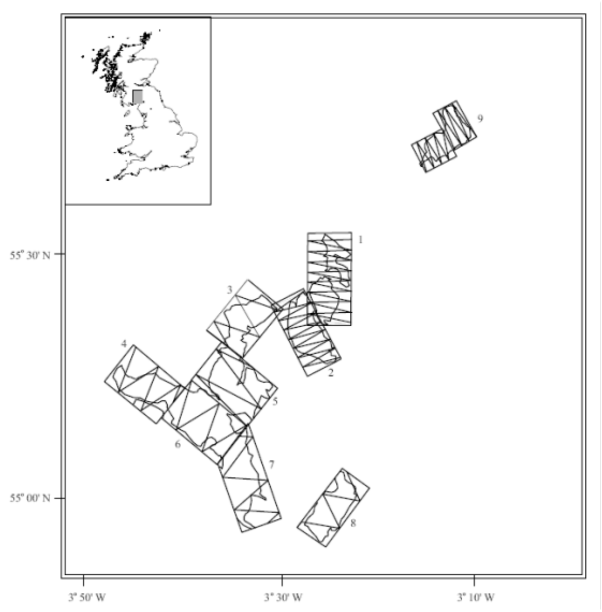


Figure 1: Location of sika deer survey in southern Scotland and the survey design (from Marques et al. (2001). Note the differing amounts of effort in different woodlands based on information derived from pilot surveys.

not provide a measure of variability of this estimate. During the course of our surveys we also followed the fate of some marked pellet groups to estimate the decay (disappearance) rates of a pellet group. A thorough discussion of methods useful for estimating decay rates and associated measures of precision can be found in Laing et al. (2003).

There are many factors that might influence both production and decay rates, and for purposes of this exercise we will make the simplifying assumption that decay rate is homogeneous across these woodlands; with their mean time to decay of 163 days and a standard error of 13 days. (If you were to conduct a survey such as this, you would want to investigate this assumption more thoroughly.)

2.1 Getting started

These data (called `sikadeer`) are available in the `dsdata` package. As in previous exercises the conversion units are calculated. What are the measurement units for these data?

```
library(Distance)
library(dsdata)
# Select data
data(sikadeer)
# Work out conversion units
conversion.factor <- convert_units("centimeter", "kilometer", "square kilometer")
```

2.2 Fit detection function to dung pellets

Fit the usual series of models (i.e. half normal, hazard rate, uniform) models to the distances to pellet groups and decide on a detection function (don't spend too long on this). Call your model `deer.df`. This detection function will be used to obtain $\hat{D}_{\text{pellet groups}}$.

Have a look at the `Summary statistics` for this model - what do you notice about the allocation of search effort in each woodland?

2.3 Multipliers

The next step is to create an object which contains the multipliers we wish to use. We already have estimates of dung production rates but need similar information on dung decay (or persistence) rate.

Data to calculate this has been collected in the file `IntroDS_9.1.csv` in your `data` directory.

```
MIKE.persistence <- function(DATA) {

#  Purpose: calculate mean persistence time (mean time to decay) for dung/nest data
#  Input: data frame with at least two columns:
#         DAYS - calendar day on which dung status was observed
#         STATE - dung status: 1-intact, 0-decayed
#  Output: point estimate, standard error and CV of mean persistence time
#
#  Attribution: code from Mike Meredith website:
#      http://www.mikemeredith.net/blog/2017/Sign_persistence.htm
#   Citing: CITES elephant protocol
#      https://cites.org/sites/default/files/common/prog/mike/survey/dung_standards

  ##   Fit logistic regression model to STATE on DAYS, extract coefficients
  dung.glm <- glm(STATE ~ DAYS, data=DATA, family=binomial(link = "logit"))
  betas <- coefficients(dung.glm)
  ##   Calculate mean persistence time
  mean.decay <- -(1+exp(-betas[1])) * log(1+exp(betas[1])) / betas[2]
  ## Calculate the variance of the estimate
  vcovar <- vcov(dung.glm)
  var0 <- vcovar[1,1]  # variance of beta0
  var1 <- vcovar[2,2]  # variance of beta1
  covar <- vcovar[2,1] # covariance
  deriv0 <- -(1-exp(-betas[1]) * log(1+exp(betas[1])))/betas[2]
  deriv1 <- -mean.decay/betas[2]
  var.mean <- var0*deriv0^2 + 2*covar*deriv0*deriv1 + var1*deriv1^2
  ## Calculate the SE and CV and return
  se.mean <- sqrt(var.mean)
  cv.mean <- se.mean/mean.decay
  out <- c(mean.decay, se.mean, 100*cv.mean)
  names(out) <- c("Mean persistence time", "SE", "%CV")
  plot(decay$DAYS, jitter(decay$STATE, amount=0.10), xlab="Days since initiation",
       ylab="Dung persists (yes=1)",
       main="Eight dung piles revisited over time")
  curve(predict(dung.glm, data.frame(DAYS=x), type="resp"), add=TRUE)
  abline(v=mean.decay, lwd=2, lty=3)
  return(out)
}
dungdecayfile <- system.file("extdata", "IntroDS_9.1.csv", package = "dsdata")
decay <- read.csv(dungdecayfile)
persistence.time <- MIKE.persistence(decay)
print(persistence.time)
```

Running the above command should have produced a plot of dung persistence versus days since produced and fitted a logistic regression (this is like a simple linear regression but restricts the response to taking values between 0 and 1). Note the points can in reality only take values between 0 and 1 but for the purposes of plotting have been 'jittered' to avoid over-plotting.

An estimate of mean persistence time and measure of variability are also provided - make a note of these as they will be required below.

As stated above, we want an object which contains information on the dung production rate (and standard error) and dung decay rate (and standard error). The following command creates a list containing two data frames:

- `creation` contains estimates of the dung production rate and associated standard error
- `decay` contains the dung decay rate and associated standard error where `XX` and `YY` are the estimates you obtained from the dung decay rate analysis.

```
# Create list of multipliers
mult <- list(creation = data.frame(rate=25, SE=0),
#              decay    = data.frame(rate=XX, SE=YY))
mult
```

The final step is to use these multipliers to convert $\hat{D}_{\text{pellet groups}}$ to $\hat{D}_{\text{deer}}$ (as in the equations above) - for this we need to employ the `dht2` function. In the command below the `multipliers=` argument allows us to specify the rates and standard errors. There are a couple of other function arguments that need some explanation:

- `strat_formula=~Region.Label` is specified to take into account the design (i.e. different woodlands or blocks).
- `stratification="effort_sum"` is specified because we want to produce an overall estimate density that is the mean of the woodland specific densities weighted by effort allocated within each block.
- `deer.df` is the detection function you have fitted.

```
# Weight by effort because we have repeats
deer.ests <- dht2(deer.df, flatfile=sikadeer, strat_formula=~Region.Label,
                  convert_units=conversion.factor, multipliers=mult,
                  stratification="effort_sum", total_area=13.9)
deer.ests
```

The function `dht2` also provides information on the components of variance. Make a note of the these (contribution of detection function, encounter rate, decay rate and what happened to production rate component?) in each strata.

3 Cue counting survey of whales

This exercise involves analysing an aerial cue counting survey of whales in the Atlantic and the species in this exercise tend to occur singly. An estimate of mean cue rate and its coefficient of variation have been obtained from tagging studies on a number of whales in the region.

The sample size is relatively small for a cue counting survey (which require larger sample sizes for reliable estimation of the detection function than line transect surveys), but this was the sample that was generated by the (expensive) survey, so you just need to do the best you can with it.

The data are stored in the `dsdata` package under `CueCountingExample`. In the command below, the data object is renamed to a shorter name. In these data, there is a column called search time - this information is copied to a column called `Effort` - this is needed by `ds`.

```
library(Distance)
library(dsdata)
data(CueCountingExample)
# Rename data
cuecountex <- CueCountingExample
head(cuecountex, n=3)
# Create effort column
cuecountex$Effort <- cuecountex$Search.time
```

You can see that these data contain columns containing cue rate information - what is the cue rate and standard error? This information needs to be in a list format for the `dht2` function: this object can be created in a similar way to the `mult` object in problem 1 but here, we create this object from the survey data. Note, there is only one multiplier (for cue rate) required in this problem.

```
# Obtain the cue rates from the survey data
# Select relevant columns
cuerates <- cuecountex[ ,c("Cue.rate", "Cue.rate.SE", "Cue.rate.df")]
# Only save unique values
cuerates <- unique(cuerates)
# Rename columns
names(cuerates) <- c("rate", "SE", "df")

# Create multiplier object
mult <- list(creation=cuerates)
mult
```

Decide on a truncation distance and fit a suitable detection function to these data: call your selected model `whale.df`. Remember that these data are treated as coming from a point transect.

Use the `dht2` function to estimate whale abundance in the survey region, together with a 95% confidence interval. As well as specifying the multipliers, the sampling fraction argument (`sample_fraction=`) also needs to be specified - in this case, half the circle was searched and so what do you think the sampling fraction should be? In the command below, density estimates are obtained unstratified (i.e. ignoring the survey regions).

```
# Estimate density - what is the sampling fraction?
whale.est <- dht2(whale.df, flatfile=cuecountex, strat_formula=~1,
                  multipliers=mult, sample_fraction=?)
whale.est
```

4 Cue counting of songbirds (optional)

Remember the wren data that was introduced in the point transect exercises (Exercise 5): another data collection method that was used was cue counting. In this case, the cue was a song burst. These data are stored in `wren3` in the `dsdata` package.

Following a similar approach to that of the whale data, estimate a detection function of songs, create a multipliers object and include this in the `dht2` function to estimate wren density. Call this `w3.est`). There are a few things that will be useful to know:

- search effort (measured in time) was 2 visits each lasting 5 minutes,
- effort will need to be properly specified before fitting the detection function with `ds`
- multiplier in this case is cue rate and its measure of precision,
- the multiplier needs to be created before making the call to `dht2`

What do you think the sampling fraction will be for these point transects?

```
conversion.factor <- convert_units("meter", NULL, "hectare")
w3.est <- dht2(w3.hr, flatfile=wren3, strat_formula=~1,
               multipliers=mult, convert_units=conversion.factor)
```

To obtain density (rather than abundance) use the following command:

```
print(w3.est, report="density")
```

5 References

Buckland, ST (2006) Point-transect surveys for songbirds: robust methodologies. The Auk 123: 345–357. https://doi.org/10.1642/0004-8038(2006)123%5B345:PSFSRM%5D2.0.CO;2.

Laing SE, ST Buckland, RW Burn, D Lambie and A Amphlett. (2003) Dung and nest surveys: estimating decay rate. Journal of Applied Ecology 40:1102-1111

Marques, FFC, ST Buckland, D Goffin, CE Dixon, DL Borchers, BA Mayle and AJ Peace. (2001) Estimating deer abundance from line transect surveys of dung: sika deer in southern Scotland. Journal of Applied Ecology 38: 349–363. https://doi.org/10.1046/j.1365-2664.2001.00584.x.

Meredith, M. (2017) How long do animal signs remain visible? http://www.mikemeredith.net/blog/2017/Sign_persistence.htm for a thorough description of decay rate estimation.

Solution 9. Analysis with multipliers

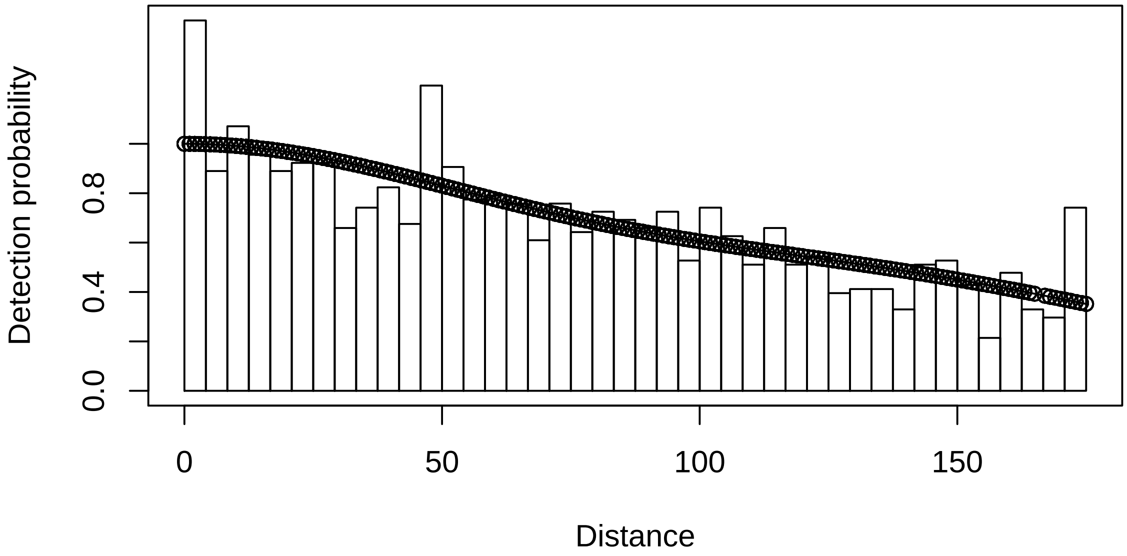
6 Dung survey of deer

The following code loads the relevant packages and data. The perpendicular distances are measured in centimetres, effort along the transects measured in kilometres and areas in square kilometres.

```
library(Distance)
library(dsdata)
# Select data
data(sikadeer)
# Work out conversion units
conversion.factor <- convert_units("centimeter", "kilometer", "square kilometer")
```

Here we did not perform a comprehensive examination of fitting a detection function to the detected pellet groups, however, as a general guideline, we truncated the longest 10% perpendicular distances.

```
# Fit detection function
deer.df <- ds(sikadeer, key="hn", truncation="10%")
# Examine detection function
plot(deer.df)
```



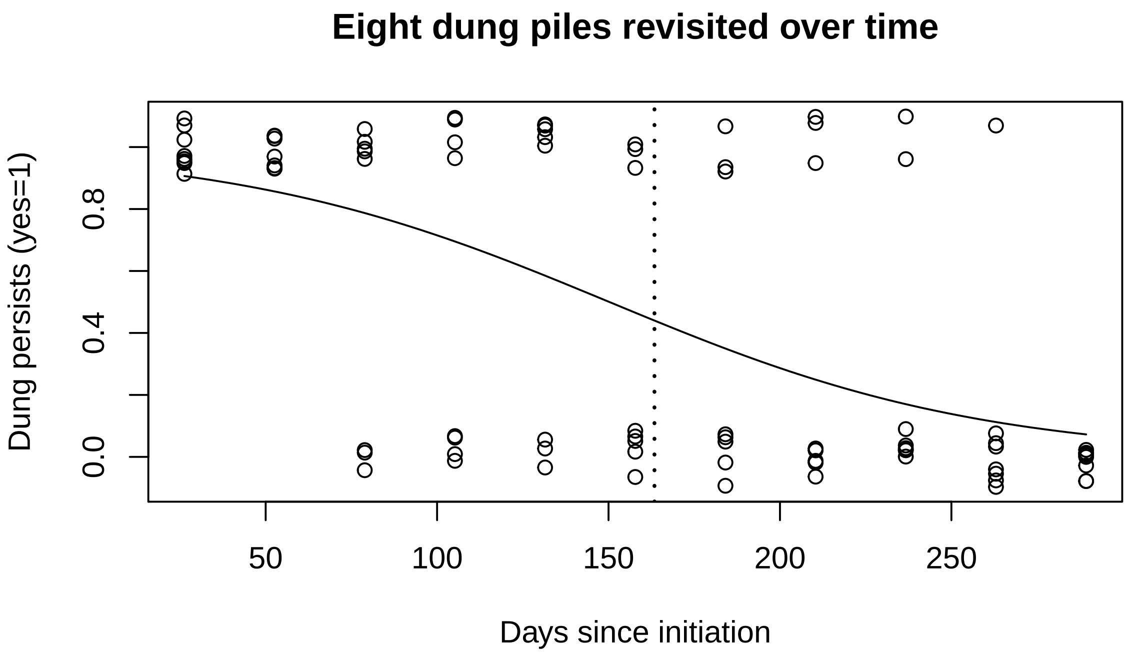
```
# Look at only the summary and encounter rate data
deer.df$dht$individuals$summary
```

```
##   Region Area CoveredArea Effort    n  k        ER      se.ER     cv.ER
## 1      A 13.9       595.0   1.70 1217 13 715.88234 119.918872 0.1675120
## 2      B 10.3       385.0   1.10  396 10 359.99999  86.859289 0.2412758
## 3      C  8.6       157.5   0.45   17  3  37.77778   8.521202 0.2255612
## 4      E  8.0       297.5   0.85   30  5  35.29412  16.568939 0.4694533
## 5      F 14.0        70.0   0.20   29  1 145.00000   0.000000 0.0000000
## 6      G 15.2       140.0   0.40   32  3  80.00000  39.686269 0.4960784
## 7      H 11.3        70.0   0.20    3  1  15.00000   0.000000 0.0000000
## 8      J  9.6        35.0   0.10    7  1  70.00000   0.000000 0.0000000
## 9  Total 90.9      1750.0   5.00 1731 37 346.19999  68.158305 0.1968755
```

The summary above shows that in blocks F, H and J there was only one transect and, as a consequence, it is not possible to calculate a variance empirically for the encounter rate in those blocks.

7 Estimating decay rate from data

A paper by Laing et al. (2003) describes field protocol for collecting data to estimate the mean persistence time of dung or nests to be used as multipliers. The code segment shown earlier analyses a file of such data via logistic regression to produce an estimate of mean persistence time and its associated uncertainty.



```
## Mean persistence time                    SE                   %CV
##            163.396748             14.226998              8.707026
```

Using these results, the multipliers can be specified:

```
# Create list of multipliers
mult <- list(creation = data.frame(rate=25, SE=0),
             decay    = data.frame(rate=163, SE=14))
mult
```

```
## $creation
##   rate SE
## 1   25  0
##
## $decay
##   rate SE
## 1  163 14
```

```
# Obtain animal estimates - overall estimate, weight by effort
deer_ests <- dht2(deer.df, flatfile=sikadeer, strat_formula=~Region.Label,
                  convert_units=conversion.factor, multipliers=mult,
                  stratification="effort_sum",total_area=13.9)
```

```
## Warning in dht2(deer.df, flatfile = sikadeer, strat_formula =
## ~Region.Label, : One or more strata have only one transect, cannot
## calculate empirical encounter rate variance
```

```
deer_ests
```

```
## Summary statistics:
##  Region.Label Area CoveredArea Effort    n  k      ER   se.ER cv.ER
##             A 13.9    0.005950   1.70 1217 13 715.882 119.919 0.168
##             B 10.3    0.003850   1.10  396 10 360.000  86.859 0.241
##             C  8.6    0.001575   0.45   17  3  37.778   8.521 0.226
##             E  8.0    0.002975   0.85   30  5  35.294  16.569 0.469
##             F 14.0    0.000700   0.20   29  1 145.000   0.000 0.000
##             G 15.2    0.001400   0.40   32  3  80.000  39.686 0.496
##             H 11.3    0.000700   0.20    3  1  15.000   0.000 0.000
##             J  9.6    0.000350   0.10    7  1  70.000   0.000 0.000
##         Total 13.9    0.017500   5.00 1731 37 346.200  45.234 0.131
##
## Abundance estimates:
##  Region.Label Estimate      se    cv LCI  UCI        df
##             A     1027 197.474 0.192 691 1527    20.797
##             B      383  99.171 0.259 220  667    11.955
##             C       34   8.200 0.244  15   75     2.759
##             E       29  13.959 0.479   9   99     4.329
##             F      210  19.752 0.094 174  252 60314.198
##             G      126  63.399 0.505  18  858     2.147
##             H       18   1.649 0.094  15   21 60314.217
##             J       69   6.539 0.094  58   83 60314.199
##         Total      497  80.044 0.161 356  694    20.215
##
## Component percentages of variance:
##  Region.Label Detection    ER Multipliers
##             A      4.07 75.96       19.97
##             B      2.24 86.76       10.99
##              C      2.52 85.14       12.34
##              E      0.66 96.13        3.22
##              F     16.93  0.00       83.07
##              G      0.59 96.52        2.89
##              H     16.93  0.00       83.07
##              J     16.93  0.00       83.07
##          Total      5.79 65.78       28.43
```

There are a few things to notice:

- overall estimate of density
  - most effort took place in woodland A where deer density was high. Therefore, the overall estimate is between the estimated density in woodland A and the lower densities in the other woodlands.
- components of variance
  - we now have uncertainty associated with the encounter rate, detection function and decay rate (note there was no uncertainty associated with the production rate) and so the components of variation for all three components are provided.

In woodland A, there were 13 transects on which over 1,200 pellet groups were detected: uncertainty in the estimated density was 19% and the variance components were apportioned as detection probability 4%, encounter rate 76% and multipliers 20%.

In woodland E, there were 5 transects and 30 pellet groups resulting in a coefficient of variation (CV) of 48%: the variance components were apportioned as detection probability 0.7%, encounter rate 96% and multipliers 3%.

In woodland F only a single transect was placed and the CV of density of 9% was apportioned as detection probability 17% and multipliers 83%. Do you trust this assessment of uncertainty in the density of deer in this woodland? We are missing a component of variation because we were negligent in placing only a single transect in this woodland and so are left to 'assume' there is no variability in encounter rate in this woodland.

By the same token, we are left to assume there is no variability in production rates between deer because we have not included a measure of uncertainty in this facet of our analysis.

8 Cue counting survey of whales

```
library(Distance)
library(dsdata)
data(CueCountingExample)
# Rename data
cuecountex <- CueCountingExample
head(cuecountex, n=3)
```

```
##   Region.Label  Area Sample.Label Cue.rate Cue.rate.SE Cue.rate.df object
## 1            B 85000         2.18       25           5           1     NA
## 2            B 85000         2.19       25           5           1     NA
## 3            B 85000         2.20       25           5           1     NA
##   distance Sample.Fraction Sample.Fraction.SE Search.time bss   sp size
## 1       NA             0.5                  0   0.1333333  NA <NA>   NA
## 2       NA             0.5                  0   0.1000000  NA <NA>   NA
## 3       NA             0.5                  0   0.4333333  NA <NA>   NA
##   Study.Area
## 1     whales
## 2     whales
## 3     whales
```

```
# Sort out effort
cuecountex$Effort <- cuecountex$Search.time
# Obtain the cue rates from the survey data
cuerates <- cuecountex[ ,c("Cue.rate", "Cue.rate.SE", "Cue.rate.df")]
cuerates <- unique(cuerates)
names(cuerates) <- c("rate", "SE", "df")
# Create multiplier object
mult <- list(creation=cuerates)
mult
```

```
## $creation
##   rate SE df
## 1   25  5  1
```

The estimated cue rate, $\hat{\nu}$, is 25 cues per unit time (per hour in this case). Its standard error is 5, therefore the CV of cue rate is $5/25 = 0.2$ (20%).

```
# Tidy up data by getting rid of those columns - we don't need them any more
cuecountex[ ,c("Cue.rate", "Cue.rate.SE", "Cue.rate.df", "Sample.Fraction",
                       "Sample.Fraction.SE")] <- list(NULL)
cuecountex$Label <- NULL
# Set truncation distance
trunc <- 1.2
# Half normal detection function
whale.df.hn <- ds(cuecountex, key="hn", transect="point", adjustment=NULL,
                  truncation=trunc)
# Hazard rate detection function
whale.df.hr <- ds(cuecountex, key="hr", transect="point", adjustment=NULL,
                  truncation=trunc)
# Compare models
summarize_ds_models(whale.df.hn, whale.df.hr, output = "plain")
```

```
##         Model Key function Formula C-vM $p$-value Average detectability
## 1 whale.df.hn  Half-normal      ~1      0.8104493             0.2386477
## 2 whale.df.hr  Hazard-rate      ~1      0.7471913             0.2811519
##   se(Average detectability) Delta AIC
## 1                0.04285283  0.000000
## 2                0.06504900  3.069857
```

Half the circle (point transect) was searched and so the sampling fraction $\phi/2\pi = 0.5$. Therefore, $\phi = \pi$ (ϕ must be in radians).

The following commands obtain density estimates assuming no stratification (strat_formula=~1).

```
# Unstratified estimates
whale.est.hn <- dht2(whale.df.hn, flatfile=cuecountex, strat_formula=~1,
                     multipliers=mult, sample_fraction=0.5)
print(whale.est.hn, results="density")
```

```
## Summary statistics:
##  .Label   Area CoveredArea Effort  n  k    ER se.ER cv.ER
##   Total 168000     82.4932  36.47 40 92 1.097 0.303 0.276
##
## Abundance estimates:
##  .Label Estimate       se    cv  LCI   UCI     df
##   Total    13654 5263.071 0.385 6112 30500 13.058
##
## Component percentages of variance:
##  .Label Detection    ER Multipliers
##   Total      21.7 51.38       26.92
```

```
whale.est.hr <- dht2(whale.df.hr, flatfile=cuecountex, strat_formula=~1,
                     multipliers=mult, sample_fraction=0.5)
print(whale.est.hr, results="density")
```

```
## Summary statistics:
##  .Label   Area CoveredArea Effort  n  k    ER se.ER cv.ER
##   Total 168000     82.4932  36.47 40 92 1.097 0.303 0.276
##
## Abundance estimates:
##  .Label Estimate       se    cv  LCI   UCI     df
##   Total    11590 4776.706 0.412 5017 26773 16.589
##
## Component percentages of variance:
##  .Label Detection    ER Multipliers
##   Total     31.51 44.94       23.55
```

A half normal detection function was chosen and whale abundance was estimated to be 13,654 whales with a 95% confidence interval (6,112: 30,500).

Note the large difference between the half normal estimate and the estimate from the hazard rate model, which is 11,590 whales, with 95% confidence interval (5,017; 26773). Remember that the key parameter in a cue counting analysis is $h(0)$, the slope of the fitted pdf to the observed data at distance zero. The difference between the estimates for the different key function is the difference between these slopes for the two models (Fig. 2):

```
# Plot pds
par(mfrow=c(1,2))
# PDFs
plot(whale.df.hn, pdf=TRUE, main="Half normal")
plot(whale.df.hr, pdf=TRUE, main="Hazard rate")
```

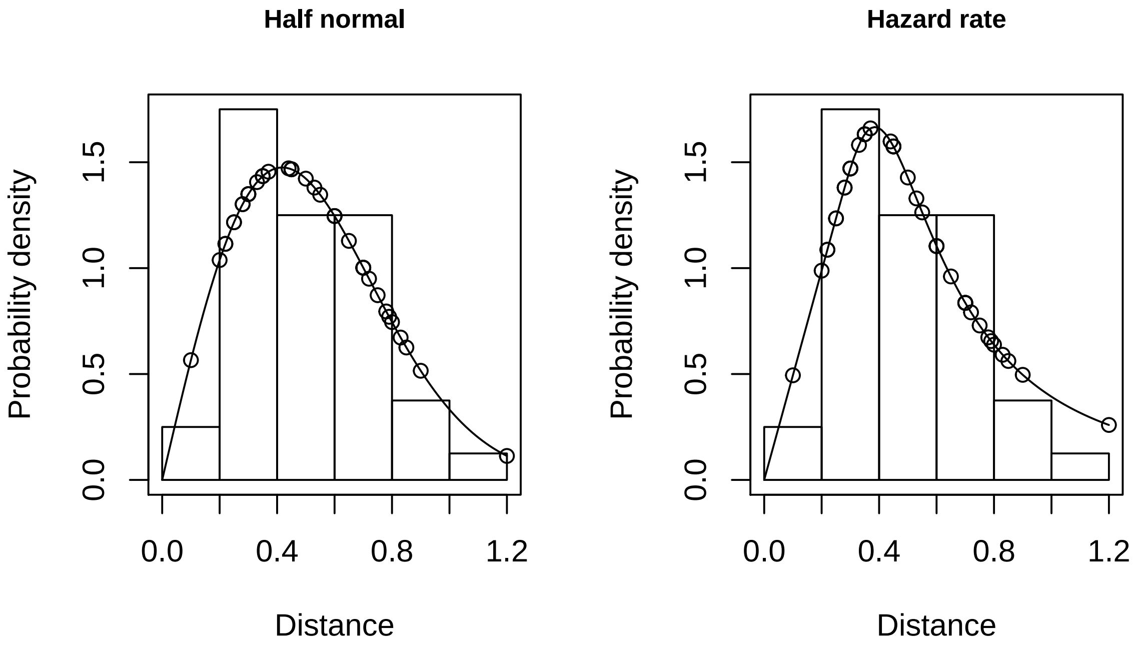


Figure 2: Probability density functions for the two fitted models.

Cue counting estimates of detection probability are more volatile than those from line transect surveys, because on a cue counting survey you have few data where you need it most to estimate $h(0)$ - namely at distances close to zero. As a consequence, cue-counting surveys require higher cue sample size for reliable estimation than samples of animals for line transect surveys.

Don't worry too much about the apparent lack of fit in the first interval, or two, in Figure 2 - remember the sample size is very small in these intervals. Use the plot above and the goodness-of-fit statistics to guide you about the fit of your model.

9 Cue counting survey of songbirds (optional)

```
## Wren data - cue count method
data(wren3)
# Extract the cue rate information
cuerate <- unique(wren3[ , c("Cue.rate","Cue.rate.SE")])
names(cuerate) <- c("rate", "SE")
# Create multipliers list
mult <- list(creation=cuerate)
# Check multipliers
mult
```

```
## $creation
##      rate     SE
## 38 1.4558 0.2428
```

```
# Search time is the effort - this is 2 * 5min visits
wren3$Effort <- wren3$Search.time
# Fit hazard rate detection function model
w3.hr <- ds(wren3, transect="point", key="hr", adjustment=NULL, truncation=92.5)
```

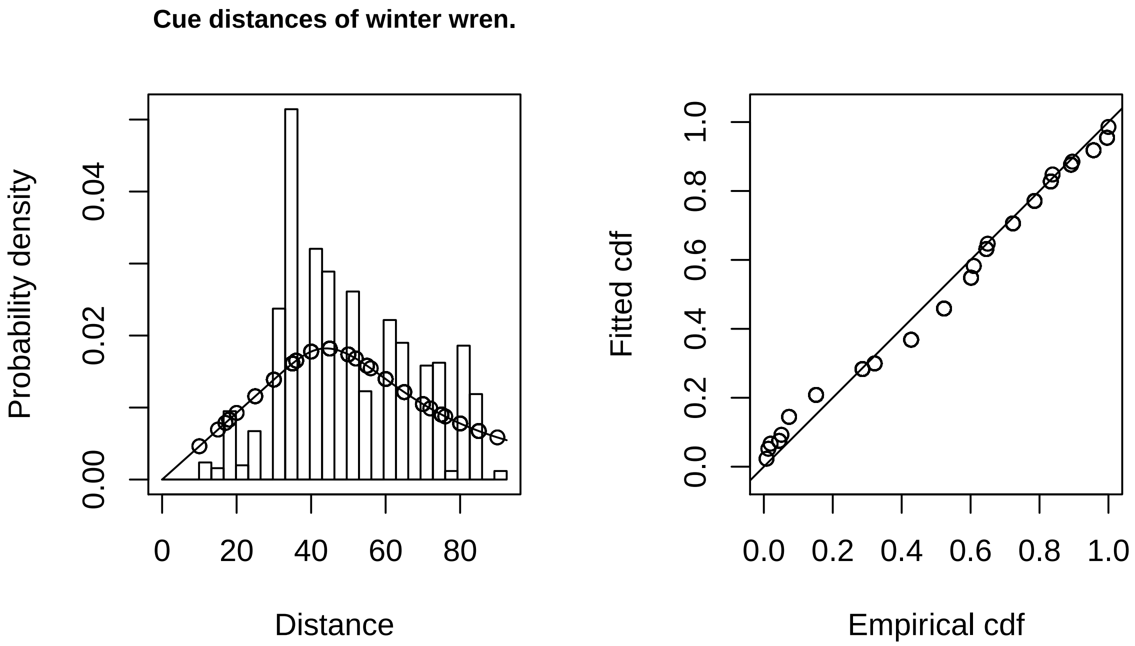
The sampling fraction for these data will be 1 because the whole of the circle was searched.

```
conversion.factor <- convert_units("meter", NULL, "hectare")
w3.est <- dht2(w3.hr, flatfile=wren3, strat_formula=~1,
               multipliers=mult, convert_units=conversion.factor)
# NB "Effort" here is sum(Search.time) in minutes
# NB "CoveredArea" here is pi * w^2 * sum(Search.time)
# Obtain density
print(w3.est, report="density")
```

```
## Summary statistics:
##  .Label Area CoveredArea Effort   n  k    ER se.ER cv.ER
##   Total 33.2    860.1681    320 765 32 2.391 0.236 0.099
##
## Density estimates:
##  .Label Estimate    se  cv    LCI    UCI      df
##   Total   1.2092 0.242 0.2 0.8195 1.7843 522.514
##
## Component percentages of variance:
##  .Label Detection    ER Multipliers
##   Total      6.14 24.33       69.54
```

Note the large proportion of the uncertainty in winter wren density stems from variability in cue (song) rate. Analyses of the cue count data are necessarily rather subjective as the data show substantial over-dispersion (a single bird may give many song bursts all from the same location during a five minute count). In this circumstance, goodness-of-fit tests are very misleading and care must be taken not to over-fit the data (i.e. fit a complicated detection function).

```
par(mfrow=c(1,2))
plot(w3.hr, pdf=TRUE, main="Cue distances of winter wren.")
gof_ds(w3.hr,)
```



```
##
## Goodness of fit results for ddf object
##
## Distance sampling Cramer-von Mises test (unweighted)
## Test statistic = 1.70062 p-value = 6.04794e-05
```