CREEM, Univ of St Andrews: Distance sampling online workshop

Analysis in R: Analysis of classic duck nest data

March 2022

The Monte Vista National Wildlife Refuge.

1 Using survey data

We use field data to fit different detection function models and estimate density and abundance. The data were collected during line transect surveys of duck nests in Monte Vista National Wildlife Refuge, Colorado, USA in 1967 and 1968. Twenty transects, each 25.75km in length were walked 5 times over the two years. Total transect length of 128.75km (25.75 $\times$ 5) and a distance out to 2.4m was searched. Consult Anderson and Pospahala (1970) for a description of the survey. Distances of detected nests have been provided in a 'csv' text file in a basic format required by 'Distance'. The columns in the the file are:

- Study.Area - name of the study region (Monte Vista NWR)
- Region.Label - identifier of regions or strata (in this case there is only one region and it is set to 'Default')
- Area - size of the stratum
- Sample.Label - line transect identifier
- Effort - length of each transect
- distance - perpendicular distances (m).

The distances allow different key functions/adjustments to be fitted in the detection function model and, by including the transect lengths and area of the region, density and abundance can be estimated.

2 Objectives of the practical

1. Import a text file
2. Understand the structure of a data frame
3. Fit different key functions/adjustments in the detection function model using `ds`
4. Examine the results of an analysis, i.e. `ddf` and `dht` components of a `dsmodel` object

3 Importing the data

The file containing the duck nest survey data is located on the online workshop website. Either download the file following this link or the content of the data file can be read directly from the web into an R (R Core Team, 2018) data frame named `nests` via the following command

```
nests <- read.csv(file="https://workshops.distancesampling.org/online-course/exercisepdfs/Ch7/datase
                  header=TRUE)
```

The URL of the file location is quite long, we repeat it here so it is more legible.
{https://workshops.distancesampling.org/online-course/exercisepdfs/Ch7/datasets/ducks-area-effort.csv}

This command is made up of several components:

- `read.csv` is a function to read data files of type 'csv' (comma-separated values),
- the function has two arguments specified; `file` specifies the name of the data file and `header=TRUE` specifies that the first row of the data file contains the names of the data columns.
- the `<-` symbol has assigned the data set to an object called `nests`. Note that there is now an object called `nests` listed on the 'Environment' tab.

To check that the data file has been read into `R` correctly, use the `head` and `tail` 'functions' to look at the top and bottom rows of the data, respectively. To look at the first few rows of `nests` type the following command.

```
head(nests, n=2)
```

The `head` function as used above displays the first 6 records of the named object. The argument `n` controls the number of rows to display. To display the *last* 2 records in the data, type the command:

```
tail(nests, n=2)
```

The object `nests` is a dataframe object made up of rows and columns. Use the function `dim` to find out the dimensions of the data set (i.e. the total number of rows and columns):

```
dim(nests)
```

Question: How many rows and columns are there in the dataframe `nests`?

Another way to look at a data frame is to move to the 'Environment tab' in R-Studio and click on the rectangle (with the grid); this opens a new tab showing the data.

4 Summarising the perpendicular distances

To access an individual column within a data frame use the `$` symbol, for example to summarise the distances:

```
summary(nests$distance)
```

Similarly to plot the histogram of distances, the command is:

```
hist(nests$distance, xlab="Distance (m)")
```

5 Fitting different models

To use the `ds` function, load the `Distance` package (Miller, 2017).

The function `ds` requires a data frame to have a column called `distance`, we specify the name of the data frame as follows:

```
library(Distance)
```

```
## Loading required package: mrds

## This is mrds 2.2.6
## Built: R 4.1.3; ; 2022-03-17 18:30:31 UTC; windows

##
## Attaching package: 'Distance'

## The following object is masked from 'package:mrds':
##
##     create.bins
```

```
conversion <- convert_units("meter", "kilometer", "square kilometer")
nest.model1 <- ds(nests, key="hn", adjustment=NULL, convert_units = conversion)
```

```
## Fitting half-normal key function

## Key only model: not constraining for monotonicity.

## AIC= 928.134
```

The `convert_units` argument ensures that the correct units are specified - in this example, distances are in metres, lengths in km and the area in km^2. Think of this argument as a divider used to transform units of transect effort into units of perpendicular distance (e.g., 1km / 0.001 = 1000m). The `convert_units` function performs these calculation for you if you provide the units in which perpendicular distances, transects and study area size are recorded in the data.

This call to `ds` fits a half-normal key function with no adjustment terms. Summarise the fitted model:

```
summary(nest.model1$ddf)
```

Plot the detection function with the histogram having 12 bins:

```
plot(nest.model1, nc=12)
```

To fit different detection functions, change the `key` and `adjustment` arguments. For example to fit a half-normal key function with cosine adjustment terms, use the command:

```
nest.model2 <- ds(nests, key="hn", adjustment="cos", convert_units = conversion)
```

By default, AIC selection will be used to fit adjustment terms of up to order 5.

Question: Have any adjustment terms been selected?

To fit a hazard rate key function with Hermite polynomial adjustment terms, then use the command:

```
nest.model3 <- ds(nests, key="hr", adjustment="herm", convert_units = conversion)
```

```
summary(nest.model3$ddf)
```

Use the `help` command to find out what other key functions and adjustment terms are available.

Question: Is there much difference in the probability of detecting a nest given it is within the maximum detection distance between the three models fitted to the duck nest data?

6 The `ds` object

The objects created with `ds` (e.g. `nest.model1`) are made up of two parts. We can list them using the `names` function as below:

```
names(nest.model1)
```

The detection function information is in the `ddf` part and the density and abundance estimates are stored in the `dht` part. To access each part, then the `$` can be used (as with columns in a data frame). For example to see what information is stored in the `ddf` part, we can use the `names` function again:

```
names(nest.model1$ddf)
```

7 Goodness of fit

Before making inference from the detection function we have fitted, we should evaluate the model. First assessment is goodness of fit, accomplished using the function `gof_ds`:

```
gof_ds(nest.model1)
```

Calling the function `gof_ds` with the default arguments and exact distance data, a *q-q* plot is produced along with the unweighted Cramer-von Mises goodness of fit test.

Question: Interpret the q-q plot and CvM test results for the duck nest data.

8 Estimating density and abundance

So far, we have concentrated on the detection function but, with more information such as transect lengths and the area of the region, we can estimate density and abundance. The second component of a `ds` object, contains this additional information. This information can be viewed with:

```
str(nest.model1$dht$individuals, max=1)
```

This `dht` object contains considerable information. However, focus upon three tables generated by the `summary()` function: summary, abundance and density. Dig more deeply into the content of these tables.

8.1 Summary information

This provides information about the survey:

- size of study area,
- area covered by sampling effort
- length of all transects
- number of detections
- number of transects
- encounter rate (ER) number of detections per unit transect length and its associated variability

```
nest.model1$dht$individuals$summary
```

8.2 Abundance estimates

Estimated density multiplied by the size of the study area.

```
nest.model1$dht$individuals$N
```

8.3 Density estimates

Density estimated using the formula

$$\hat{D} = \frac{n}{a\hat{P}_a}$$

where n (number of nests)=534, a (covered area)=12.36 and $\hat{P}_a$ (probability of detection)=0.8693

```
nest.model1$dht$individuals$D
```

Question: Compute (by hand) the density estimate resulting from the estimated probability of detection arising from the hazard rate detection function: $\hat{P}_a$= 0.8891.

References

Anderson, D. R. and R. S. Pospahala. 1970. Correction of bias in belt transect studies of immotile objects. The Journal of Wildlife Management, **34**:141–146. URL http://www.jstor.org/stable/3799501.

Buckland, S. T., E. A. Rexstad, T. A. Marques, and C. S. Oedekoven. 2015. Distance Sampling: Methods and Applications. Springer. URL https://www.springer.com/gb/book/9783319192185.

Miller, D. L. 2017. Distance: Distance Sampling Detection Function and Abundance Estimation. URL https://CRAN.R-project.org/package=Distance. R package version 0.9.7.

R Core Team. 2018. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.