CREEM, Univ of St Andrews: Distance sampling on-line workshop

Analysis in R: Analysis of multi-species surveys

March 2022

1 More complex analyses

(Solutions)

This practical is based on the Montrave songbird case study in Buckland et al. (2015, Section 5.2.2.3), with computer code under Montrave songbird case study. Both point and line transect surveys were conducted and here we use the data from the **line transect** data, although the issues (and solutions) will be similar.

These data are provided in a 'flat file' format (i.e. it contains all the necessary columns to estimate a detection function, density and abundance). While both formats are equally valid, the 'flat file' approach has a particular idiosyncrasy which we exploit here to introduce more functions and data manipulation.

Several species of birds were identified but not all species were detected on all transects. If a simple data selection is performed to select records for a particular species, then not all of the transects will be included in the resulting data (because that species may not have been seen). This doesn't matter if we are only interested in fitting detection functions, but will matter if we wish to estimate density and abundance because the effort will be too low since some of the transects are missing. To correct for this, some data frame manipulation is required. There is generally more than one way to do something in R (Miller et al., 2019) - for an alternative way see the computer code 'Montrave song bird case study' associated with Buckland et al. (2015), as well as Section 9 below.

European robin *(Erithacus rubecula)*; one of the species in the Montrave study (Buckland, 2006).

Aerial view of Montrave study area. White diagonal lines represent transects walked for data analysed here.

2 Objectives of the practical

1. Data frame selection and manipulation
2. Extracting estimates from `dht` object
3. Customising detection function plots
4. Improve re-usability of code with functions

3 Importing the data

The data is in a 'flat file' format and contains the following columns:

- Region.Label - name of study
- Area - size of study region (hectares)
- repeats - number of visits to transect
- Sample.Label - line transect identifier
- Effort - length of transect (km)
- distance - perpendicular distance (m)
- species - species of bird (c=chaffinch, g=great tit, r=robin and w=wren)
- visit - on which visit bird was detected.

Use the following command to import the data from the website associated with Buckland et al. (2015) and then use the `head` command to examine it.

```
birds <- read.csv(file="https://raw.githubusercontent.com/distanceworkshops/online-course/gh-pages/ex
```

{file="https://raw.githubusercontent.com/distanceworkshops/online-course/gh-pages/exercisepdfs/Ch7/datasets/montrave-line.csv"}

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

```
##   Region.Label Area repeats Sample.Label Effort distance species visit
## 1     Montrave 33.2       2            1  0.208       75       c     1
## 2     Montrave 33.2       2            1  0.208       40       c     1
```

Question: Explore the data. How many transects are there?

```
length(unique(birds$Sample.Label))
```

```
## [1] 19
```

For now, save the transect labels to a new object as we will use them later on:

```
tran.lab <- unique(birds$Sample.Label)
```

The `table` command is a quick way to determine how many detections there are of each species:

```
table(birds$species)
```

```
##
##   c   g   r   w
##  73  32  82 156
```

As a hint of things to come, create a two-way table showing the number of detections by transect and by species. If there are zeroes in this table, it will create a challenge.

```
with(birds, table(species, Sample.Label))
```

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
c	4	7	7	5	9	7	5	1	1	3	1	0	2	2	4	3	4	7	1
g	0	2	3	5	5	1	3	2	1	1	0	0	2	0	2	0	3	1	1
r	3	8	11	5	8	7	5	7	3	1	0	2	5	6	4	0	4	3	0
w	10	11	12	11	14	12	13	11	6	3	1	4	9	12	9	2	7	6	3

Each of the line transects was visited twice which is not taken into account at present. However, it is straightforward to do so:

```
birds$Effort <- birds$Effort * birds$repeats
```

4 Getting the effort right for the robin data

For the purposes of this practical, we are interested in estimating the density of robins and so we select only these records:

```
robins <- birds[birds$species=="r", ]
```

Question: On how many transects were robins detected?

```
length(unique(robins$Sample.Label))
```

```
## [1] 16
```

If we were to use the `robins` data as it is at present to estimate density, then density would be **incorrect** because the search effort associated with three transects is missing. Resolution of this problem is to use additional arguments to the `ds()` function. These arguments carry information about the survey design that cannot be deduced from the flatfile `montrave-lines.csv`.

4.1 Additional arguments

Rather than proceeding with the `robins` data set, which will produce erroneous results, let's back up and work with the entire multiple species data set.

We will create two data frames to contain information about Buckland's survey. These data frames will be passed to `ds()` via the arguments `region_table` and `sample_table`. The contents of the data frames are described in the help for the `ds()` function

- region_table data.frame with two columns:
  - Region.Label label for the region
  - Area area of the region
  - region_table has one row for each stratum. If there is no stratification then region_table has one entry with Area corresponding to the total survey area. If Area is omitted density estimates only are produced.
- sample_table data.frame mapping the regions to the samples (i.e. transects). There are three columns:
  - Sample.Label label for the sample
  - Region.Label label for the region that the sample belongs to.
  - Effort the effort expended in that sample (e.g. transect length).

The code in the next chunk performs the magic we need.

```
bird.regiontab <- data.frame(Region.Label=as.factor(c(1,2)), Area=c(33.2,33.2))
bird.sampletab <- data.frame(Region.Label=as.factor(rep(c(1,2), each=19)),
                             Sample.Label=rep(1:19, times=2),
                             Effort=c(0.208, 0.401, 0.401, 0.299, 0.350,
                                      0.401, 0.393, 0.405, 0.385, 0.204,
                                      0.039, 0.047, 0.204, 0.271, 0.236,
                                      0.189, 0.177, 0.200, 0.020))
```

4.2 Dissecting the code

The `region_table` data frame is the most simple. We have indicated there are two strata (the two visits; estimation is not required for each visit, but it does represent how the sampling was carried out). Because the strata simply repeat visits to the same transects, the size of the strata is identical and equal to the size of the study area.

The `sample_table` data frame needs to know the "names" of each stratum (here simply "1" and "2") and the names of the transects (numbers 1-19) and the stratum to which each transect belongs. Finally the length of each transect is required. When completed, the data frame contains 38 rows (the 19 transects duplicated once). The 19 transects belonging to stratum 1 (first visit) and again the same 19 transects belonging to stratum 2 (second visit).

When these data frames are provided to `ds()` the integrity of the survey design will be preserved regardless of the data filtering that might take place.

5 Analysis

Before we fit any models, have a quick look at the histogram of distances:

```
hist(robins$distance, breaks=20)
```

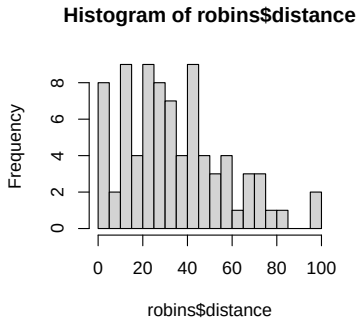


Figure 1: Perpendicular distances of robins in Montrave study.

Consistent with Buckland et al. (2015), three detection functions are fitted using the `ds()` function in the R package `Distance` (Miller et al., 2019) using the truncation distance used by Prof Buckland:

```
library(Distance)
robins$Region.Label <- robins$visit
robin.hn.herm <- ds(robins, truncation=95, transect="line",
                    region_table = bird.regiontab, sample_table = bird.sampletab,
                    key="hn", adjustment="herm", convert_units=0.1)
robin.uni.cos <- ds(robins, truncation=95, transect="line",
                    region_table = bird.regiontab, sample_table = bird.sampletab,
                    key="unif", adjustment="cos", convert_units=0.1)
robin.haz.simp <- ds(robins, truncation=95, transect="line",
                     region_table = bird.regiontab, sample_table = bird.sampletab,
                     key="hr", adjustment="poly", convert_units=0.1)
```

Question: What is the preferred model for the robin data?

```
summarize_ds_models(robin.hn.herm, robin.uni.cos, robin.haz.simp)
```

Table 2: Model selection for robin data from Montrave line transect survey.

Model	C-vM p-value	$\hat{P}_a$	se($\hat{P}_a$)	ΔAIC
`robin.uni.cos`	0.510	0.636	0.103	0.000
`robin.hn.herm`	0.435	0.609	0.070	0.406
`robin.haz.simp`	0.732	0.679	0.053	0.565

Note: All three detection function fit the data (based upon the C-vM test of exact distances). The estimated detection probability is very similar for all models, and the ΔAIC values of all models is < 1. Hence all models will give very similar estimates of density.

6 Examining the `dht` object

The fitted model object (e.g. `robin.uni.cos`) is made up of two parts; the detection function in the `ddf` part and the estimates in the `dht` part. In this section, we look at the `dht` part.

To list the elements that are contained in `dht`, use the `names` function:

```
names(robin.uni.cos$dht)
```

```
## [1] "individuals"
```

Detections were of individual birds and so group size was not included in these data - if it had been included (in a column called `size`), then as well as `individuals` there would have been elements `clusters` and `Expected.S`.

The estimates stored in the `individuals` object can be listed in a similar manner:

```
names(robin.uni.cos$dht$individuals)
```

```
## [1] "bysample"       "summary"        "N"              "D"
## [5] "average.p"      "cormat"         "vc"             "Nhat.by.sample"
```

To collect together the density estimates (and estimates of precision) from all the fitted models, we can use the following command:

```
model.results <- rbind(robin.uni.cos$dht$individuals$D[3,],
                       robin.haz.simp$dht$individuals$D[3,],
                       robin.hn.herm$dht$individuals$D[3,])
```

```
model.results
```

Question: Examine the three sets of density estimates to see if the previous suggestion (that the density estimates are similar) is confirmed.

Table 3: Density estimates for Montrave robins under three fitted detection functions.

Label	Estimate	se	cv	lcl	ucl	df
Total	0.6857	0.1248	0.1820	0.4795	0.9806	108.3286
Total	0.6419	0.0731	0.1138	0.5124	0.8041	93.2373
Total	0.7152	0.1014	0.1418	0.5408	0.9458	113.5515

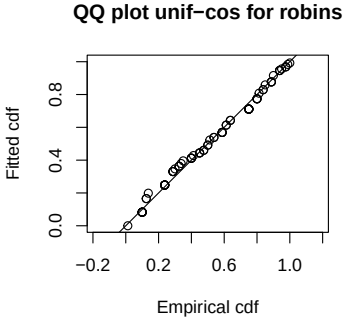
7 Goodness of fit

Here we look at goodness of fit test with unequal bin intervals and just consider one of the fitted models. First we specify the required bin intervals.

```
robin.brks <- c(0, 12.5, 22.5, 32.5, 42.5, 52.5, 62.5, 77.5, 95.0)
```

Perform the tests using both exact distance data for the Cramer-von Mises test and specified breakpoints for χ^2 test for the uniform-cosine model that had the (slightly) smallest AIC score.

```
testout <- gof_ds(robin.uni.cos, breaks=robin.brks,
                  chisq = TRUE, main="QQ plot unif-cos for robins")
```




```
nice <- data.frame(statistic=c(testout$chisquare$chi1$chisq,
                               testout$dsgof$CvM$W),
                   pvalue=c(testout$chisquare$chi1$p,
                            testout$dsgof$CvM$p))
row.names(nice) <- c("Chisquare", "CvM test")
knitr::kable(nice, row.names = TRUE, digits=3,
             caption="Goodness of fit tests for unicos model.")
```

Table 4: Goodness of fit tests for unicos model.

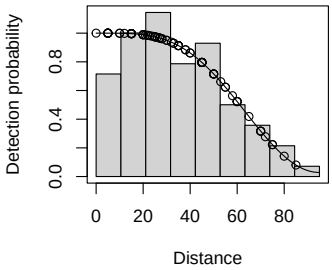
	statistic	pvalue
Chisquare	3.804	0.578
CvM test	0.117	0.510

Note: The detections fall close to the diagonal line of the qq plot, suggesting an adquate fit for the uniform cosine model. The *p-value* of the Cramer-von Mises test (at bottom of printout) confirms this. Similarly the *p-value* for the χ^2 test also suggests an adequate fit.

8 Customising the detection function plot

The `plot` function provides a basic plot of the fitted detection function overlaid onto the scaled distribution of distances:

```
plot(robin.uni.cos)
```



9 Advanced: modularising R code to work with multiple species

When analysing a multi-species survey, it is likely that the invesigator will want to analyse all (or at least many) of the species encountered during the survey. This will necessitate some repetitive calculation fitting multiple detection functions for each species.

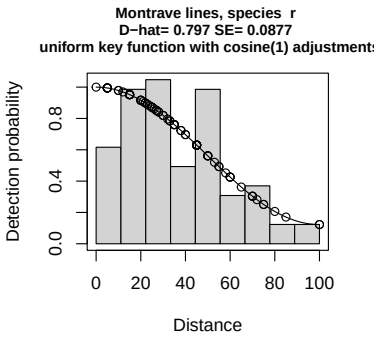
To facilitate the repetitive nature of such analyses, it is useful to take advantage of the programmatic nature of the R language to create *functions* that can be called repeatedly with arguments to accommodate changing circumstances. The code below demonstrates such a modular approach with `fit.hn.uni.haz()` defined to aide in the repeated analyses.

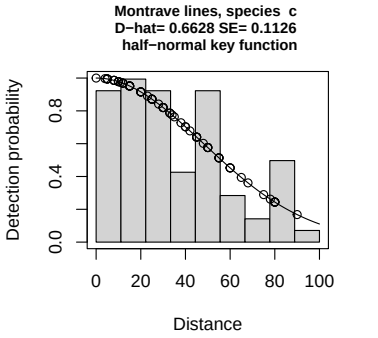
`fit.hn.uni.haz()` fits three candidate models to a dataset provided as the first argument. The second argument is the truncation distance. The final argument determines whether the `summarize_ds_models()` table is printed.

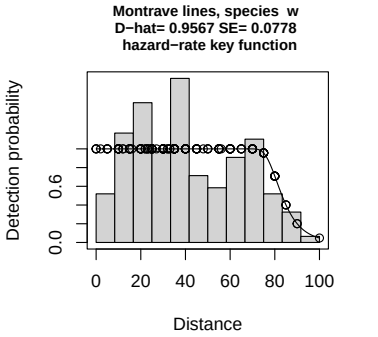
```
fit.hn.uni.haz <- function(data, trunc, print=TRUE) {
  #  Purpose:  fit three key functions to transect data,
  #            perform model selection and
  #            print model selection table
  #  Input: data to analyse, truncation distance, print flag
  #  Output: fitted model object (class `dsmodel`)
  #  Rexstad August 2018
  hn.herm <- ds(data, trun=trunc, key="hn", adj="herm",
                convert_units = .1)
  uni.cos <- ds(data, trun=trunc, key="unif", adj="cos",
                convert_units = .1)
  haz.simp <- ds(data, trun=trunc, key="hr", adj="poly",
                 convert_units =.1)
  mods <- summarize_ds_models(hn.herm, uni.cos, haz.simp, output="plain")
  if(print) print(knitr::kable(mods))
  names(mods) <- c("mod","key","form","fit","pa","sepa","daic")
  if(mods[1,1]=="hn.herm") {
    result <- hn.herm
  } else {
    if(mods[1,1]=="uni.cos") {
    result <- uni.cos
  } else {
    result <- haz.simp
  }
  }
  return(result)
}
```

The function is used in the calling code below. Note the `for` loop that iterates through three of the four species detected in the Montrave survey (great tit not analysed because there were few detections). Note finally, that even though the model with the smallest AIC for the winter wren was the hazard rate model, this was not the model used for inference by Buckland (2006) as the shape (perfect detectability to 70m) was determined to be biologically implausible. This underscores one of the challenges associated with "assembly line" analysis of multiple species; biological insight needs to be maintained in the model selection process.

```
for(thisspecies in c("r", "c", "w")) {
  best.model <- fit.hn.uni.haz(birds[birds$species==thisspecies, ],
                               trunc= 100, print=FALSE)
  plot(best.model,
       main=paste("Montrave lines, species ", thisspecies,
       "\nD-hat=", round(best.model$dht$individuals$D$Estimate,4),
       "SE=", round(best.model$dht$individuals$D$se, 4),
       "\n",best.model$ddf$name.message))
}
```







References

Buckland, S. T. 2006. Point transect surveys for songbirds: robust methodologies. The Auk, **123**:345. URL https://doi.org/10.1642/0004-8038(2006)123[345:psfsrm]2.0.co;2.

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., E. Rexstad, L. Thomas, L. Marshall, and J. L. Laake. 2019. Distance sampling in R. Journal of Statistical Software, **89**. URL https://doi.org/10.18637/jss.v089.i01.