MT5751 Projects

David Borchers, Stephen Buckland, Eric Rexstad

January 19, 2017

Chapter 1

Distance Sampling

1. **Line transect estimation from the a minke whale survey**: The survey from which these data were obtained is described in Branch, T.A. and D.S. Butterworth (2001) Southern Hemisphere minke whales: standardised abundance estimates from the 1978/79 to 1997/98 IDCR-SOWER surveys. Journal of Cetacean Research and Management 3(2): 143-174, which is on MMS. The survey region is shown in Figure 1(e) on page 146 of that paper.

   a) **Getting started**: Open `RStudio` and in it open a new `R script` window (using the button top left). Load the `R` package `Distance` by typing

   ```
   library(Distance)
   ```

   in the `R script` window and then clicking on the Run button with your cursor on the line that you just entered. Now do the same for the command

   ```
   data(minke)
   ```

   which loads a minke whale survey dataset. You can find out a bit about it by typing

   ```
   ?minke
   ```

   which opens the `minke` help pages in `RStudio`'s Help window.

   Have a look at the data by typing `head(minke)` or clicking on `minke` in the top right window (Figure 1.1).

   Plot a histogram of the observed perpendicular distances (`minke$distance`) using the `histogram` command.

   b) **Fitting a model**: Typing

   ```
   ?Distance
   ```

   opens the `Distance` help pages in `RStudio`'s Help window. Click on the "Index" link at the bottom of this page and browse the topics. Look in particular at the `ds` command. (You should be looking at something like Figure 1.1.)

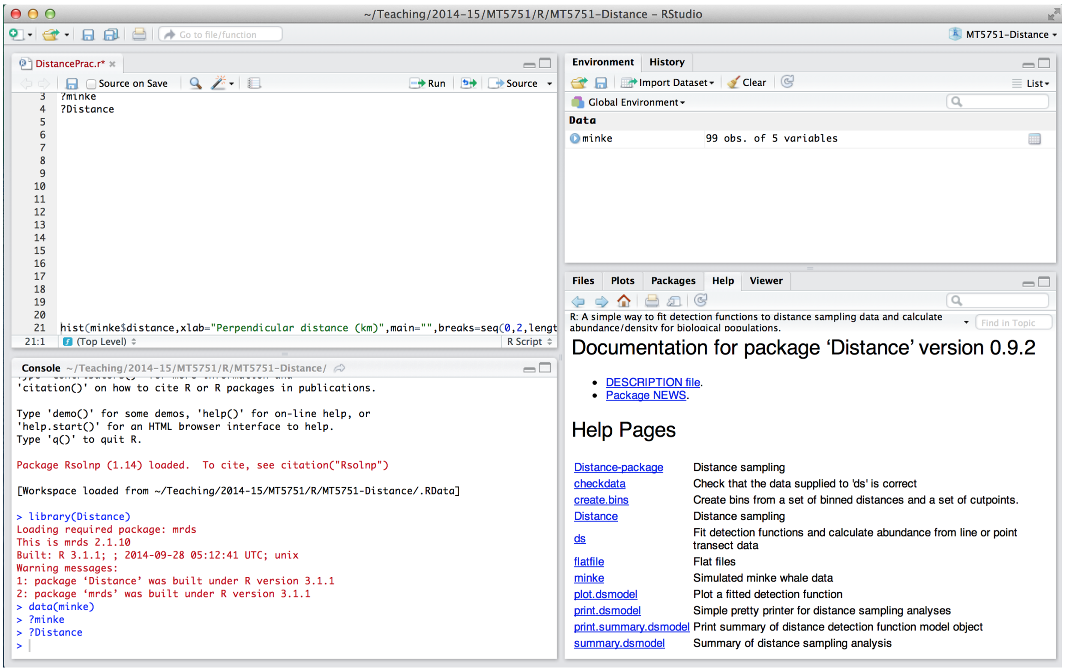
   Use the function `ds` to fit a half-normal detection function model and estimate the number of minke whales in the area covered by this survey.

   Having fitted a model with the command `ds`, you can look at the estimates using the `summary` command. (For example, if your fitted model is in an object called `fit.hn`, type `summary(fit.hn)` to look at the estimates obtained in fitting.)

   Note that in order to keep them positive, detection function scale parameters are parameterised as $\sigma = \exp(\beta)$ in the package `Distance`, and it is the estimate of β, not σ^2 that `summary` reports. Using this fact, verify that $\hat{\sigma}^2$ is about 0.495

   c) **Find the best estimate**: Now try more than one detection function form, select a model using some statistically valid criterion, and check goodness-of-fit of the model, using the function `ddf.gof`. You can get the help page for `ddf.gof` by typing `?ddf.gof`. *Note: You will need to pass the* `$ddf` *component of the object returned by* `ds` *into* `ddf.gof` *– if you pass the whole object it will not work. (This is a bug in that is currently being addressed.)*

Figure 1.1: RStudio window.



```
?minke
?Distance
















hist(minke$distance,xlab="Perpendicular distance (km)",main="",breaks=seq(0,2,lengt
```

```
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

Package Rsolnp (1.14) loaded.  To cite, see citation("Rsolnp")

[Workspace loaded from ~/Teaching/2014-15/MT5751/R/MT5751-Distance/.RData]

> library(Distance)
Loading required package: mrds
This is mrds 2.1.10
Built: R 3.1.1; ; 2014-09-28 05:12:41 UTC; unix
Warning messages:
1: package ‘Distance’ was built under R version 3.1.1
2: package ‘mrds’ was built under R version 3.1.1
> data(minke)
> ?minke
> ?Distance
> 
```

Data

```
minke                 99 obs. of 5 variables
```

R: A simple way to fit detection functions to distance sampling data and calculate abundance/density for biological populations.

Documentation for package ‘Distance’ version 0.9.2

- DESCRIPTION file.
- Package NEWS.

Help Pages

Distance-package	Distance sampling
checkdata	Check that the data supplied to 'ds' is correct
create.bins	Create bins from a set of binned distances and a set of cutpoints.
Distance	Distance sampling
ds	Fit detection functions and calculate abundance from line or point transect data
flatfile	Flat files
minke	Simulated minke whale data
plot.dsmodel	Plot a fitted detection function
print.dsmodel	Simple pretty printer for distance sampling analyses
print.summary.dsmodel	Print summary of distance detection function model object
summary.dsmodel	Summary of distance sampling analysis

Chapter 2

Capture-Recapture

This document is to help you get started with the R package secr. The package has extensive help, which you should use. In particular, you should read the vignettes secr.overview.pdf and secr-datainput.pdf to help you get going. You can get these by typing

```
library(secr)
?secr
```

in your R window and then scrolling down in the help window that this opens, until you find links to these documents.

We will play with the stoatDNA dataset to get used to secr.

```
data(stoatDNA)   # get stoat hair snare dataset
?stoatDNA        # find out something about it
summary(stoatCH) # look at the capture histories
plot(stoatCH)    # plot it
```

What kind of detectors were used (single-catch traps, multi-catch traps, binary detectors or count detectors)?
How many capture occasions were there?
How many stoats were detected over the whole survey?
How many stoats were captures once, twice, three times, ...?
Extract the trap data from stoatCH and plot it:

```
traps=traps(stoatCH)
plot(traps)
```

Now fit a model to the data:

```
stoat.model.HN=secr.fit(stoatCH, buffer = 1000, detectfn = 0)
```

Try that again, with the additional argument print.level=0 (which is an argument of the optimiser nlm that secr.fit uses to find the maximum of the likelihood function):

```
stoat.model.HN <- secr.fit(stoatCH, buffer = 1000, detectfn = 0,trace=0)
```

Do you see what changed while fitting the model with this extra argument?

Now look at the output from the fit:

```
stoat.model.HN       # look at the estimates
```

Verify that you understand what the "Beta parameters" and "Fitted (real) parameters" are by calculating the latter manually from the former.

Plot the detection function:

```
plot(stoat.model.HN,xval=0:1000,sigmatick=TRUE,limits=TRUE,ylim=c(0,0.12))
```

Do you understand what the solid, dashed and vertical lines are? See the help for `plot.secr` if you do not.

Now try fitting models with some other detection functions. (See the `secr` help pages on `detectfn` for the options available to you - don't try them all, there are way too many!)

Try fitting models of type Mt, Mb, Mh for σ and/or g_0. (Table 5 in the vignette `secr.overview.pdf` is useful here - but only consider `t`, `T`, `b`, `B` and `h2`.)

Chapter 3

SECR

Chapter 4

Occupancy