Mark-Recapture Distance Sampling using R

Practical 8, Intermediate Distance Sampling workshop, CREEM, 2018

This version of the practical is for those who would like to conduct the analysis using the `mrds` package (Laake *et al.* 2018) in R. There is a separate version describing how to conduct the analysis in Distance (Thomas *et al.* 2010).

The first part of the practical involves analysis of a survey of a known number of golf tees. This is intended mainly to familiarise you with the double-platform data structure and analysis features in R. The second part of the practical involves analysis of the pack-ice seal survey data of Borchers *et al.* (2006) and Southwell *et al.* (2007).

I. Golf tee survey

Golf tee data

These data come from a survey of clusters of golf tees in grass, conducted by statistics students at the University of St Andrews. The data were collected along transect lines, 210 metres in total. A distance of 4 metres out from the centre line was searched and, for the purposes of this exercise, we assume that this comprised the total study area, which was divided into two strata. There were 250 clusters of tees in total and 760 individual tees in total.

The population was independently surveyed by two observer teams. The following data were recorded for each detected group: perpendicular distance, cluster size, observer (team 1 or 2), 'sex' (males are yellow and females are green and golf tees occur in single-sex clusters) and 'exposure'. Exposure was a subjective judgment of whether the cluster was substantially obscured by grass (exposure=0) or not (exposure=1). The lengths of grass varied along the transect line and the grass was slightly more yellow along one part of the line compared to the rest.

The golf tee dataset is provided as part of the `mrds` package (as well as in the Distance for Windows project called 'GolfteesExercise').

Open R and load the `mrds` package and golf tee dataset (called `book.tee.data`). The elements required for an MRDS analysis (i.e. observations, samples, region information) are contained within the object dataset.

```r
# Load libraries
library(knitr)
library(mrds)

# Access the golf tee data
data(book.tee.data)

# Investigate the structure of the dataset
str(book.tee.data)
```

```
List of 4
 $ book.tee.dataframe:'data.frame': 324 obs. of  7 variables:
  ..$ object  : num [1:324] 1 1 2 2 3 3 4 4 5 5 ...
  ..$ observer: Factor w/ 2 levels "1","2": 1 2 1 2 1 2 1 2 1 2 ...
  ..$ detected: num [1:324] 1 0 1 0 1 0 1 0 1 0 ...
  ..$ distance: num [1:324] 2.68 2.68 3.33 3.33 0.34 0.34 2.53 2.53 1.46 1.46 ...
  ..$ size    : num [1:324] 2 2 2 2 1 1 2 2 2 2 ...
  ..$ sex     : num [1:324] 1 1 1 1 0 0 1 1 1 1 ...
  ..$ exposure: num [1:324] 1 1 0 0 0 0 1 1 0 0 ...
 $ book.tee.region   :'data.frame': 2 obs. of  2 variables:
  ..$ Region.Label: Factor w/ 2 levels "1","2": 1 2
  ..$ Area        : num [1:2] 1040 640
 $ book.tee.samples  :'data.frame': 11 obs. of  3 variables:
  ..$ Sample.Label: num [1:11] 1 2 3 4 5 6 7 8 9 10 ...
  ..$ Region.Label: Factor w/ 2 levels "1","2": 1 1 1 1 1 1 2 2 2 2 ...
  ..$ Effort      : num [1:11] 10 30 30 27 21 12 23 23 15 12 ...
 $ book.tee.obs      :'data.frame': 162 obs. of  3 variables:
  ..$ object      : int [1:162] 1 2 3 21 22 23 24 59 60 61 ...
  ..$ Region.Label: int [1:162] 1 1 1 1 1 1 1 1 1 1 ...
  ..$ Sample.Label: int [1:162] 1 1 1 1 1 1 1 1 1 1 ...
```

```
# Extract the list elements from the dataset into easy-to-access objects
detections <- book.tee.data$book.tee.dataframe
region <- book.tee.data$book.tee.region
samples <- book.tee.data$book.tee.samples
obs <- book.tee.data$book.tee.obs
```

Let's have a look at the columns in the detections data because it has a particular structure.

```
# Check detections
head(detections)
```

```
##    object observer detected distance size sex exposure
## 1       1        1        1     2.68    2   1        1
## 21      1        2        0     2.68    2   1        1
## 2       2        1        1     3.33    2   1        0
## 22      2        2        0     3.33    2   1        0
## 3       3        1        1     0.34    1   0        0
## 23      3        2        0     0.34    1   0        0
```

Each detected object (in this case the object was a group or cluster of golf tees) is given a unique number in the `object` column. Notice that each `object` occurs twice - once for observer 1 and once for observer 2. The `detected` column indicates whether the object was seen (`detected=1`) or not seen (`detected=0`) by the observer. Perpendicular distance is in the `distance` column and cluster size is in the `size` column.

To ensure that the variables `sex` and `exposure` are treated correctly, define them as factor variables.

```
# Define sex and exposure as factor variables
detections$sex <- as.factor(detections$sex)
detections$exposure <- as.factor(detections$exposure)
```

Golf tee survey analyses

Estimation of $p(0)$: distance only

We'll start by analysing these data assuming that Observer 2 was generating trials for Observer 1 but not vice versa, i.e. trial configuration where Observer 1 is the primary and Observer 2 is the tracker. (The data could also be analysed in independent observer configuration - you are welcome to try this for yourself). We also assume full independence (i.e. detections between observers are independent at all distances): this requires only a MR model and to start with only perpendicular distance will be included as a covariate. (This is the "FI - MR dist" model in Distance for Windows project. Indeed, if you did fit that model in Distance, you can look in the Log tab at the R code Distance generated and compare it with the code we use here.)

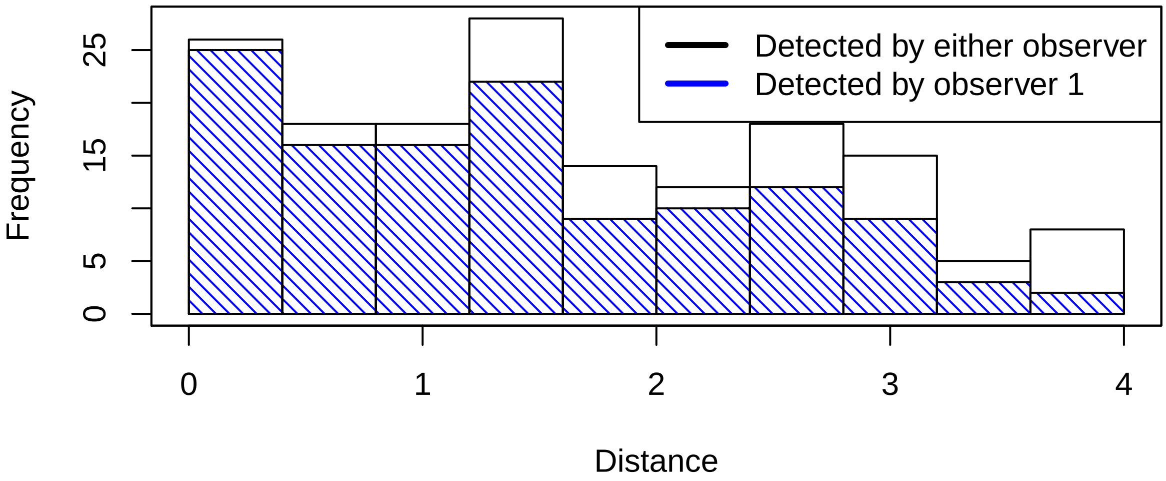
Remember that `?` or `help` can be used to find out more about any of the functions used – e.g., `?ddf` will tell you more about the ddf function.

```
# Fit the FI-trial model
fi.mr.dist <- ddf(method='trial.fi', mrmodel=~glm(link='logit',formula=~distance),
                  data=detections, meta.data=list(width=4))
# Create a set of tables summarizing the double observer data
detection.tables <- det.tables(fi.mr.dist)
# Print these detection tables
detection.tables
```

```
Observer 1 detections
           Detected
            Missed Detected
  [0,0.4]        1       25
  (0.4,0.8]      2       16
  (0.8,1.2]      2       16
  (1.2,1.6]      6       22
  (1.6,2]        5        9
  (2,2.4]        2       10
  (2.4,2.8]      6       12
  (2.8,3.2]      6        9
  (3.2,3.6]      2        3
  (3.6,4]        6        2

Observer 2 detections
           Detected
            Missed Detected
  [0,0.4]        4       22
  (0.4,0.8]      1       17
  (0.8,1.2]      0       18
  (1.2,1.6]      2       26
  (1.6,2]        1       13
  (2,2.4]        2       10
  (2.4,2.8]      3       15
  (2.8,3.2]      4       11
  (3.2,3.6]      2        3
  (3.6,4]        1        7

Duplicate detections

  [0,0.4] (0.4,0.8] (0.8,1.2] (1.2,1.6]   (1.6,2]   (2,2.4] (2.4,2.8]
       21        15        16        20         8         8         9
(2.8,3.2] (3.2,3.6]   (3.6,4]
        5         1         1

Observer 1 detections of those seen by Observer 2
          Missed Detected Prop. detected
[0,0.4]        1       21      0.9545455
(0.4,0.8]      2       15      0.8823529
(0.8,1.2]      2       16      0.8888889
(1.2,1.6]      6       20      0.7692308
(1.6,2]        5        8      0.6153846
(2,2.4]        2        8      0.8000000
(2.4,2.8]      6        9      0.6000000
(2.8,3.2]      6        5      0.4545455
(3.2,3.6]       2        1      0.3333333
(3.6,4]         6        1      0.1428571
```

The information in detection tables could also be plotted, but, in the interest of space, only one (out of six possible plots) is shown below.

```
# Plot detection information, change number to see other plots
plot(detection.tables, which=1)
```



The plot numbers are:

1. Histograms of distances for detections by either, or both, observers. The shaded regions show the number for observer 1.
2. Histograms of distances for detections by either, or both, observers. The shaded regions show the number for observer 2.
3. Histograms of distances for duplicates (detected by both observers).
4. Histogram of distances for detections by either, or both, observers. Not shown for trial configuration.
5. Histograms of distances for observer 2. The shaded regions indicate the number of duplicates - for example, the shaded region is the number of clusters in each distance bin that were detected by Observer 1 given that they were also detected by Observer 2 (the "|" symbol in the plot legend means "given that").
6. Histograms of distances for observer 1. The shaded regions indicate the number of duplicates as for plot 5. Not shown for trial configuration.

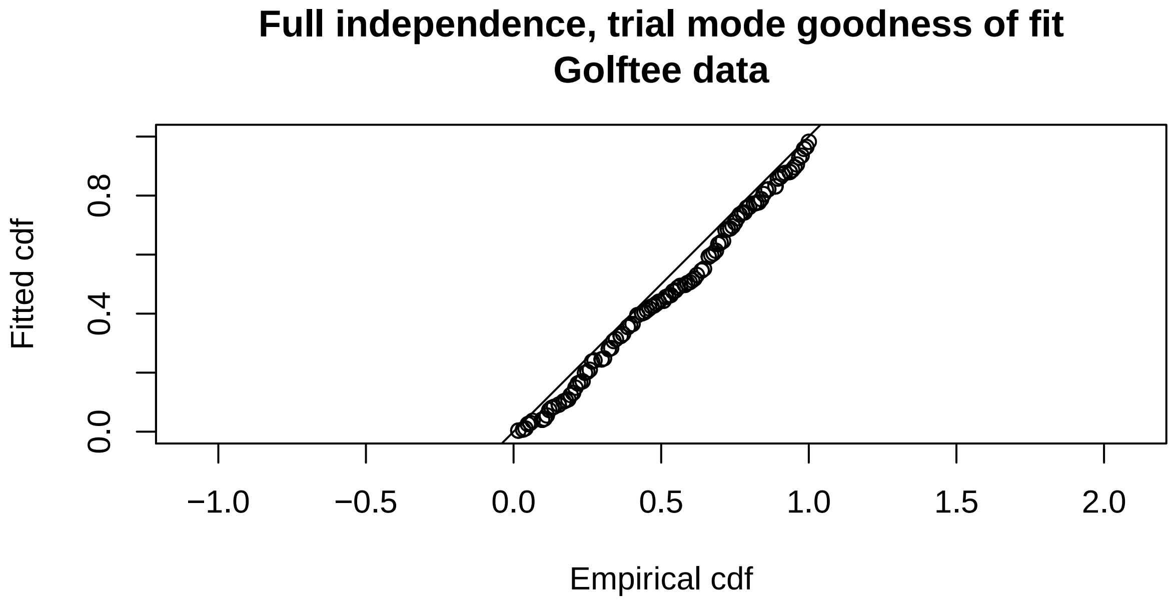
Note that if an independent observer configuration had been chosen, all plots would be available.

A summary of the detection function model is available using the `summary` function. The Q-Q plot has the same interpretation as a Q-Q plot in a single platform analysis.

```
# Produce a summary of the fitted detection function object
summary(fi.mr.dist)
```

```
##
## Summary for trial.fi object
## Number of observations               :  162
## Number seen by primary               :  124
## Number seen by secondary (trials)    :  142
## Number seen by both (detected trials):  104
## AIC                                  :  452.8094
##
##
## Conditional detection function parameters:
##              estimate        se
## (Intercept)  2.900233 0.4876238
## distance    -1.058677 0.2235722
##
##                           Estimate          SE         CV
## Average p                0.6423252  0.04069409 0.06335434
## Average primary p(0)     0.9478579  0.06109655 0.06445750
## N in covered region    193.0486185 15.84826458 0.08209468
```

```
# Produce goodness of fit statistics and a qq plot
gof.result <- ddf.gof(fi.mr.dist,
                      main="Full independence, trial mode goodness of fit\nGolftee data")
```



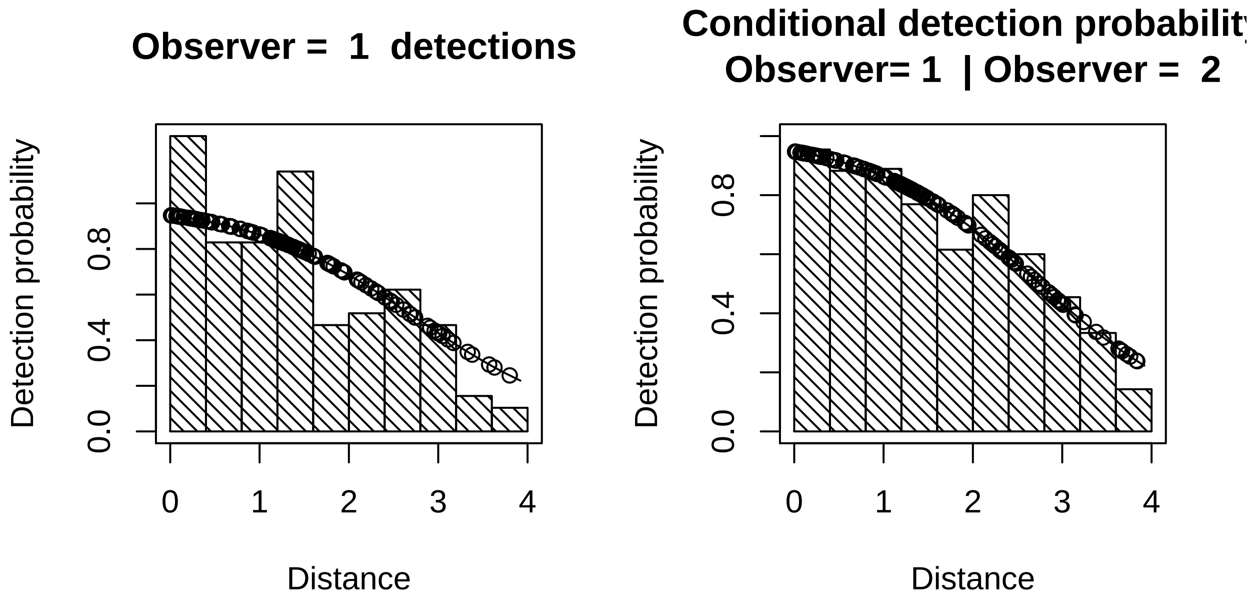
```
# Extract chi-square statistics
chi.distance <- gof.result$chisquare$chi1$chisq
chi.markrecap <- gof.result$chisquare$chi2$chisq
chi.total <- gof.result$chisquare$pooled.chi
```

Abbreviated χ^2 goodness of fit assessment shows the χ^2 contribution from the distance sampling model to be 11.5 and the χ^2 contribution from the mark-recapture model to be 3.4. The combination of these elements produces a total χ^2 of 14.9 with 17 degrees of freedom, resulting in a P-value of 0.604

The (two) detection functions can be plotted:

```
# Divide the plot region
par(mfrow=c(1,2))
```

```r
# Plot detection functions
plot(fi.mr.dist)
```



The plot headed "Observer=1 detections" shows a histogram of Observer 1 detections with the estimated Observer 1 detection function overlaid on it and adjusted for $p(0)$. The dots show the estimated detection probability for all Observer 1 detections.

The plot headed "Conditional detection probability" shows the proportion of Obs 2's detections that were detected by Obs 1 (also see the detection tables). The fitted line is the estimated detection probability function for Obs 1 (given detection by Obs 2) - this is the MR model. Dots are estimated detection probabilities for each Obs 1 detection.

Abundance is estimated using the dht function.

```r
# Calculate density estimates using the dht function
tee.abund <- dht(model=fi.mr.dist, region.table=region, sample.table=samples, obs.table=obs)

# Print out results in a nice format
kable(tee.abund$individuals$summary, digits=2,
      caption="Survey summary statistics for golftees")
```

Table 1: Survey summary statistics for golftees

Region	Area	CoveredArea	Effort	n	ER	se.ER	cv.ER	mean.size	se.mean
1	1040	1040	130	229	1.76	0.12	0.07	3.18	0.21
2	640	640	80	152	1.90	0.33	0.18	2.92	0.23
Total	1680	1680	210	381	1.81	0.14	0.08	3.07	0.15

```r
kable(tee.abund$individuals$N, digits=2,
      caption="Abundance estimates for golftee population with two strata")
```

Table 2: Abundance estimates for golftee population with two strata

Label	Estimate	se	cv	lcl	ucl	df
1	356.52	32.35	0.09	294.54	431.53	17.13
2	236.64	44.14	0.19	147.33	380.09	5.06
Total	593.16	60.38	0.10	478.32	735.57	16.06

Estimation of $p(0)$: distance and other explanatory variables

How about including the other covariates, size, sex and exposure, in the MR model? Which MR model would you use? Don't spend too long on this - just try a couple of models.

Point independence

A less restrictive assumption is point independence, that the detections are only independent on the line i.e. at perpendicular distance zero.

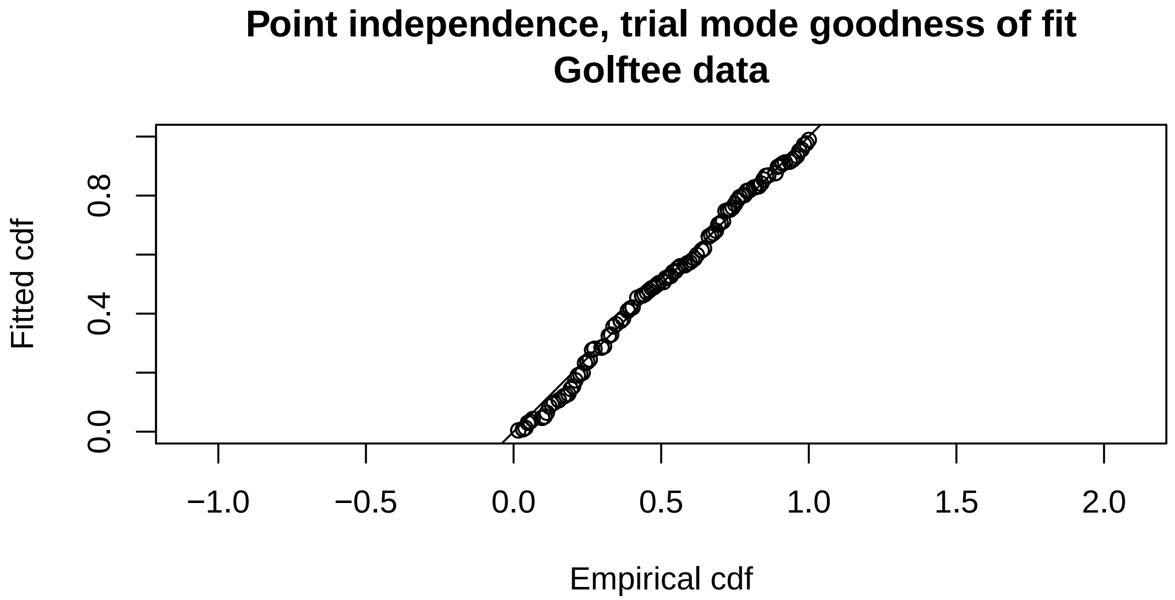
Let's start by seeing if a simple point independence model is better than a simple full independence one. This requires that a distance sampling (DS) model is specified as well a MR model. Here we try a half-normal key function for the DS model.

```r
# Fit the PI-trial model
pi.mr.dist <- ddf(method='trial', mrmodel=~glm(link='logit',formula=~distance),
                  dsmodel=~cds(key='hn'), data=detections,meta.data=list(width=4))

# Summary pf the model
summary(pi.mr.dist)
```

```
##
## Summary for trial.fi object
## Number of observations               :  162
## Number seen by primary               :  124
## Number seen by secondary (trials)    :  142
## Number seen by both (detected trials):  104
## AIC                                  :  140.8887
##
##
## Conditional detection function parameters:
##              estimate        se
## (Intercept)  2.900233 0.4876238
## distance    -1.058677 0.2235722
##
##                         Estimate         SE         CV
## Average primary p(0) 0.9478579 0.02409999 0.02542574
##
##
##
## Summary for ds object
## Number of observations :  124
## Distance range         :  0  -  4
## AIC                    :  311.1385
##
## Detection function:
```

```
##  Half-normal key function
##
## Detection function parameters
## Scale coefficient(s):
##              estimate         se
## (Intercept) 0.6632435 0.09981249
##
##            Estimate         SE         CV
## Average p 0.5842744 0.04637627 0.07937413
##
##
## Summary for trial object
##
## Total AIC value =  452.0272
##                          Estimate          SE         CV
## Average p              0.5538091  0.04615833 0.08334699
## N in covered region 223.9038534 22.99246702 0.10268902
```

```
# Produce goodness of fit statistics and a qq plot
gof.results <- ddf.gof(pi.mr.dist, main="Point independence, trial mode goodness of fit\nGolftee data")
```



Compare the results with the corresponding full independence model. Which has the lower AIC? Which has an estimate closer to known true abundance.

To include covariates in the DS detection function, we need to specify an MCDS model as follows:

```
# Fit the PI-trial model - DS sex and MR distance
pi.mr.dist2 <- ddf(method='trial', mrmodel=~glm(link='logit',formula=~distance),
                   dsmodel=~mcds(key='hn',formula=~sex), data=detections,
                   meta.data=list(width=4))
```

Use the `summary` function to check the AIC and decide if you are going to include any additional covariates in the detection function.

Now try a point independence model that has the preferred MR model from your full independence analyses.

Which has the lower AIC and bias?

II. Crabeater seal survey

Crabeater seal data

This analysis is described in Borchers *et al.* (2006) and Southwell *et al.* (2007). These data come from a helicopter survey of crabeater seals conducted by the Australian Antarctic Division within the pack-ice seals programme. The helicopter could only operate within a relatively short distance from the ice-breaker ship which acted as its base. The ice-breaker could only go where the pack ice was thin enough and so the aerial transects could not be located at random. This means that design-based estimation was not a valid option and so, in the published analysis, abundance was estimated using density surface modelling. For the purposes of this exercise, we concentrate on detection function estimation and create an artificial region as a device to produce abundance estimates.

There were four independent observers in the helicopter, two on each side (front and back). The front observers were considered to be one 'team' and the back observers were considered to be the other 'team'. Various environmental factors were recorded. In addition to perpendicular distance and cluster size, the following explanatory variables are available:

- side - the side of the helicopter from which seal were seen (L and R)
- exp - the experience (in survey hours) of the observer
- fatigue - the number of minutes the observer had been on duty on the current flight
- gscat - group size category (1, 2 and greater than or equal to 3)
- vis - visibility category (Poor, Good and Excellent)
- glare - whether there was glare (Yes or No)
- ssmi - a measure of ice cover
- altitude - the height of the aircraft in metres
- obsname - unique identifier of observer

The data from the survey has been saved in a `.csv` file. This file is read into R using `read.csv`.

```r
library(Distance)
crabseal <- read.csv("crabbieMRDS.csv")
```

Crabeater seal analyses

The observer teams acted independently and so an 'independent observer' configuration can be specified. To start with, we assume point independence and specify a half-normal key function for the DS model and include only perpendicular distance in the MR model.

```r
#  Half normal detection function, 700m truncation distance,
#      logit function for mark-recapture component
crab.ddf.io <- ddf(method="io", dsmodel=~cds(key="hn"),
                 mrmodel=~glm(link="logit", formula=~distance),
                 data=crabseal, meta.data=list(width=700))
summary(crab.ddf.io)
```

```
##
## Summary for io.fi object
## Number of observations   :  1740
## Number seen by primary   :  1394
## Number seen by secondary :  1471
## Number seen by both      :  1125
## AIC                      :  3011.463
##
##
## Conditional detection function parameters:
##                  estimate           se
## (Intercept)  2.107762345 0.0994391200
## distance    -0.003087713 0.0003159216
##
##                         Estimate          SE          CV
## Average primary p(0)   0.8916554 0.009606428 0.010773701
## Average secondary p(0) 0.8916554 0.009606428 0.010773701
## Average combined p(0)  0.9882614 0.002081614 0.002106339
##
##
## Summary for ds object
## Number of observations :  1740
## Distance range         :  0  -  700
## AIC                    :  22314.4
##
## Detection function:
##  Half-normal key function
##
## Detection function parameters
## Scale coefficient(s):
##             estimate        se
## (Intercept) 5.828703 0.0268578
##
##            Estimate         SE         CV
## Average p 0.5845871 0.01247837 0.02134562
##
##
## Summary for io object
## Total AIC value :  25325.86
##
##                         Estimate          SE         CV
## Average p              0.5777249  0.01239179 0.02144929
## N in covered region 3011.8139211 79.84197966 0.02650960
```

Make a note of the estimated values for $p(0)$ for each observer and the observers combined.

Goodness of fit could be examined in the same manner as the golf tees by the use of `ddf.gof(crab.ddf.io)` but I have not shown this step.

Following model criticism and selection, estimation of abundance ensues. The estimates of abundance for the study area are arbitrary because inference of the study was restricted to the covered region. Hence the estimates of abundance here are artificial. For illustration, the `checkdata()` function produces the region, sample, and observation tables. From these tables, Horvitz-Thompson like estimators can be applied to produce estimates of $\hat{N}$. The use of `convert.units` adjusts the units of perpendicular distance measurement (m) to units of transect effort (km). Be sure to perform the conversion correctly or your abundance estimates will be off by orders of magnitude.

```
# Create tables for estimating abundance
# Selecting observer==1 ensures that observations in the obs.table are unique
tables <- Distance:::checkdata(crabseal[crabseal$observer==1,])

# Estimate abundance in covered region
```

```
crab.ddf.io.abund <- dht(model=crab.ddf.io,
                         region=tables$region.table,
                         sample=tables$sample.table, obs=tables$obs.table,
                         se=TRUE, options=list(convert.units=0.001))

# Pretty tables of data summary
kable(crab.ddf.io.abund$individuals$summary, digits=3,
      caption="Summary information from crabeater seal aerial survey.")
```

Table 3: Summary information from crabeater seal aerial survey.

Region	Area	CoveredArea	Effort	n	ER	se.ER	cv.ER	mean.size	se.mean
1	1e+06	8594.082	6138.63	2053	0.334	0.033	0.097	1.18	0.013

```
# Pretty tables of estimates of individual abundance
kable(crab.ddf.io.abund$individual$N, digits=3,
      caption="Crabeater seal abundance estimates for study area of arbitrary size.")
```

Table 4: Crabeater seal abundance estimates for study area of arbitrary size.

Label	Estimate	se	cv	lcl	ucl	df
Total	413493.2	41201.49	0.09964248	339670.9	503359.6	128.6257

Crabeater seals with MCDS

We can also analyse the crabeater seals data as if it were single platform data (i.e. ignoring that $p(0)$ is less than 1).

Data identical to that available in the Distance project `CrabbieMCDSExercise.zip` has been ported to `crabbieMCDS.csv`, as if you had entered these data yourself into a spreadsheet.

This short exercise guides you through the import of these data into R and fits a simple half-normal detection function examining the possible improvement of the model by incorporating *side of plane* and *visibility* covariates.

```
# Load Distance for MCDS
library(Distance)
# Read in data
crab.covariate <- read.csv("crabbieMCDS.csv")
# Check data imported OK
head(crab.covariate, n=3)
```

```
    Study.area Region.Label    Area Sample.Label Effort distance size side
1 Nominal_area            1 1000000        99A21  59.72   144.49    1    R
2 Nominal_area            1 1000000        99A21  59.72   125.16    1    L
3 Nominal_area            1 1000000        99A21  59.72   421.40    1    L
    exp fatigue gscat vis glare ssmi altitude obsname
1   0.0   61.90     1   G     N   79 43.05763      YH
2 211.7   62.61     1   G     N   79 43.05763      MF
3   0.0   62.86     1   G     N   79 43.05763      MH
```

After checking that the data have been read into R appropriately, we are ready to fit a detection function.

As before, *side of plane* and *visibility* are assigned characters and so we need to tell R to treat them as factors.

```
# Define factor variables
crab.covariate$side <- as.factor(crab.covariate$side)
crab.covariate$vis <- as.factor(crab.covariate$vis)
```

With two potential explanatory variables, there are a number of possible models. We start by fitting a detection function with *side of plane* as a covariate using a half-normal key function.

```
# Fit HN key function with side of plane
ds.side <- ds(crab.covariate, key="hn", formula=~side, truncation=700)
```

```
Model contains covariate term(s): no adjustment terms will be included.

Fitting half-normal key function

AIC= 22304.742
```

We would now like to assess the fit of this function to our data. Two visual assessments are provided by the panels below: histogram and fitted function on the left and QQ plot on the right.

```
# Divide plot region
par(mfrow = c(1, 2))
# Create a title for the plot
plot.title <- "Two sets of points\none for each 'side' of plane"
# Plot model
plot(ds.side, pch=19, cex=0.5, main=plot.title)
# Plot qq plot
gof.result <- ds.gof(ds.side, lwd = 2, lty = 1, pch = ".", cex = 0.5)
# Extract gof statistics
message <- paste("CVM GOF p-value=", round(gof.result$dsgof$CvM$p, 4))
# Add gof stats to plot
text(0.6, 0.2, message, cex=0.5)
```

The code below fits the model without any covariates.

```
# Fit HN key function with no covars
ds.nocov <- ds(crab.covariate, key="hn", formula=~1, truncation=700)
```

```
Starting AIC adjustment term selection.

Fitting half-normal key function

Key only model: not constraining for monotonicity.

AIC= 22314.398

Fitting half-normal key function with cosine(2) adjustments

AIC= 22308.645

Fitting half-normal key function with cosine(2,3) adjustments

AIC= 22304.015

Fitting half-normal key function with cosine(2,3,4) adjustments

AIC= 22305.943


Half-normal key function with cosine(2,3) adjustments selected.
```

AIC score for model without covariates is 22304 and AIC score for model with *side* as a covariate is 22305 so the model without *side* as a covariate is preferred.

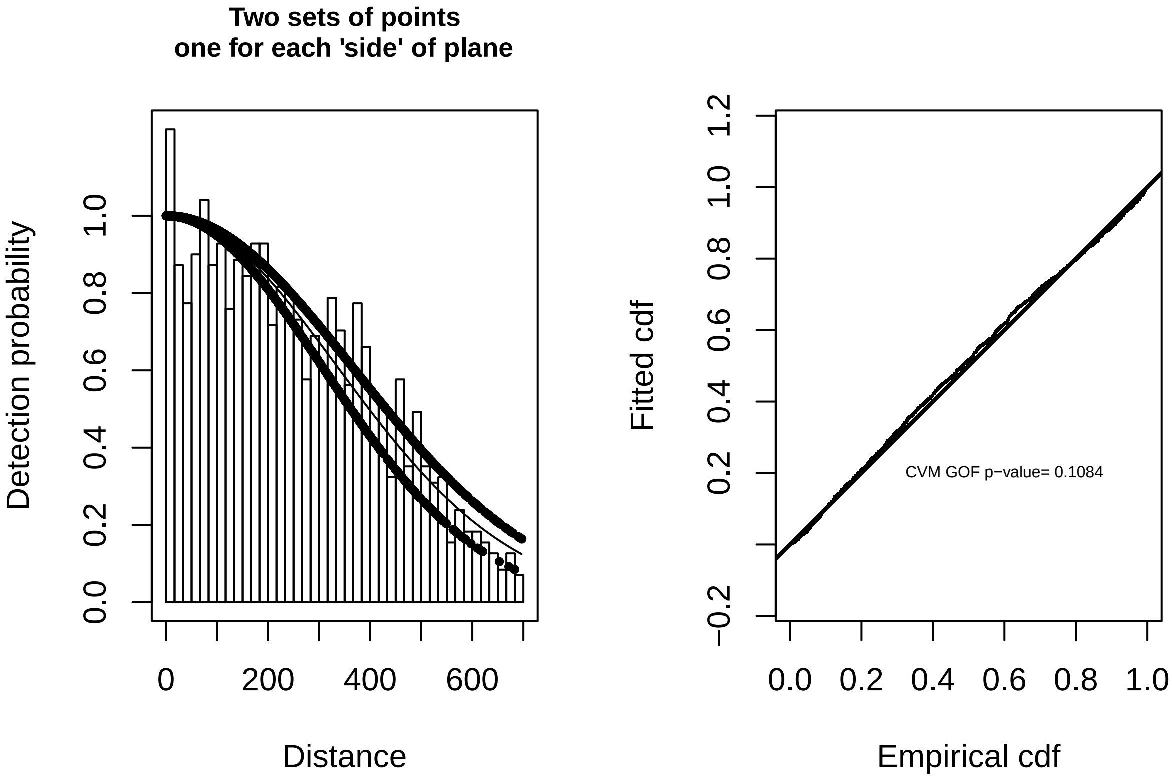


Figure 1: Histogram and fitted half-normal detection function on left. Q-Q plot of detection function and data on right.

We could also fit further detection functions and contrast the resulting models:

- with *visibility* only
- with *side of plane* and *visibility* (excluding an interaction).

Out of the four possible models which is to be preferred?

We could go on to produce abundance estimates from our preferred model using the `dht` function if we had provided information about the size of the crabeater seal study area.

For these data would an MCDS analyses have been adequate?

References

Borchers DL, Laake JL, Southwell C and Paxton CGM (2006) Accommodating unmodeled heterogeneity in double-observer distance sampling surveys. Biometrics 62: 371-378

Laake JL, Borchers DL, Thomas L, Miller DL and Bishop JRB (2018) mrds: Mark-Recapture Distance Sampling. R package version 2.2.0. https://CRAN.R-project.org/package=mrds

Southwell C, Borchers DL, Paxton CGM, Burt ML and de la Mare W (2007) Estimation of detection probability in aerial surveys of Antarctic pack-ice seals. Journal of Agricultural, Biological and Environmental Statistics 12:41-54

Thomas L, Buckland ST, Rexstad EA, Laake JL, Strindberg S, Hedley SL, Bishop JRB, Marques TA, and Burnham KP (2010) Distance software: design and analysis of distance sampling surveys for estimating population size. Journal of Applied Ecology 47: 5-14. DOI: 10.1111/j.1365-2664.2009.01737.x