Advanced topics in distance sampling

Workshop, 26-30 August 2019

Centre for Research into Ecological and Environmental Modelling

Solutions 8. Mark-Recapture Distance Sampling

In this exercise we look at mark-recapture distance sampling (MRDS) models. The first part of this exercise 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 the R function `mrds` (Laake *et al.* 2019). 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).

To help understand the terminology using in MRDS and the output produced by `mrds`, there is a guide available on the workshop website called 'Interpreting MRDS output: making sense of all the numbers'.

Aims

The aims of this practical are to learn how to model

1. trial and independent-observer configuration
2. full and point independence assumptions,
3. include covariates in the detection function(s) and
4. select between competing models.

1. Golf tee survey

Golf tee data

These data come from a survey of golf tees which 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.

Open R and load the `mrds` package and golf tee dataset (called `book.tee.data`). The elements required for an MRDS analysis are contained within the object dataset. These data are in a hierarchical structure (rather than in a 'flat file' format) so that there are separate elements for observations, samples and regions. In the code below, each of these tables is extracted to avoid typing long names.

```
# 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$book.tee.dataframe # detection information
region <- book.tee.data$book.tee.region # region info
samples <- book.tee.data$book.tee.samples # transect info
obs <- book.tee.data$book.tee.obs # links detections to transects and regions
```

Let's have a close 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
```

The structure of the detection is as follows:

- 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,
- 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 (the same default names as for the `ds` function).

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 will 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 begin by assuming full independence (i.e. detections between observers are independent at all distances): this requires only a mark-recapture (MR) model and, to start with, perpendicular distance will be included as the only covariate.

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 trial configuration with full independence model
fi.mr.dist <- ddf(method='trial.fi', mrmodel=~glm(link='logit',formula=~distance),
                  data=detections, meta.data=list(width=4))
```

Examining `mrds` output

Having fitted the model, we can create tables summarizing the detection data. In the commands below, the tables are created using the `det.tables` function and saved to `detection.tables`.

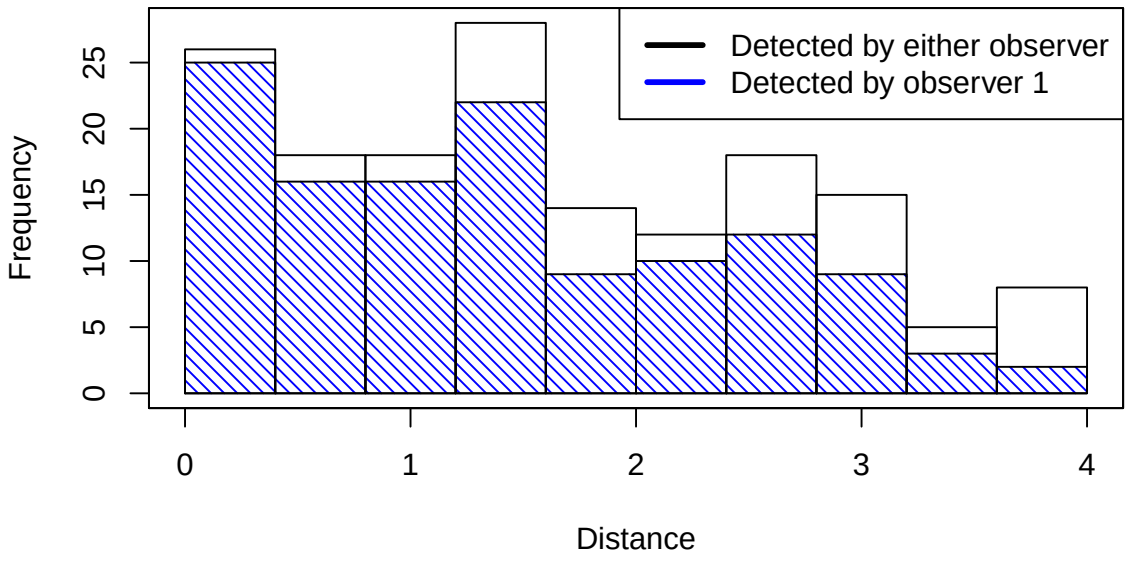
```
# Create a set of tables summarizing the double observer data
detection.tables <- det.tables(fi.mr.dist)
# Print these detection tables
detection.tables
```

```
##
## 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 summary tables can 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.

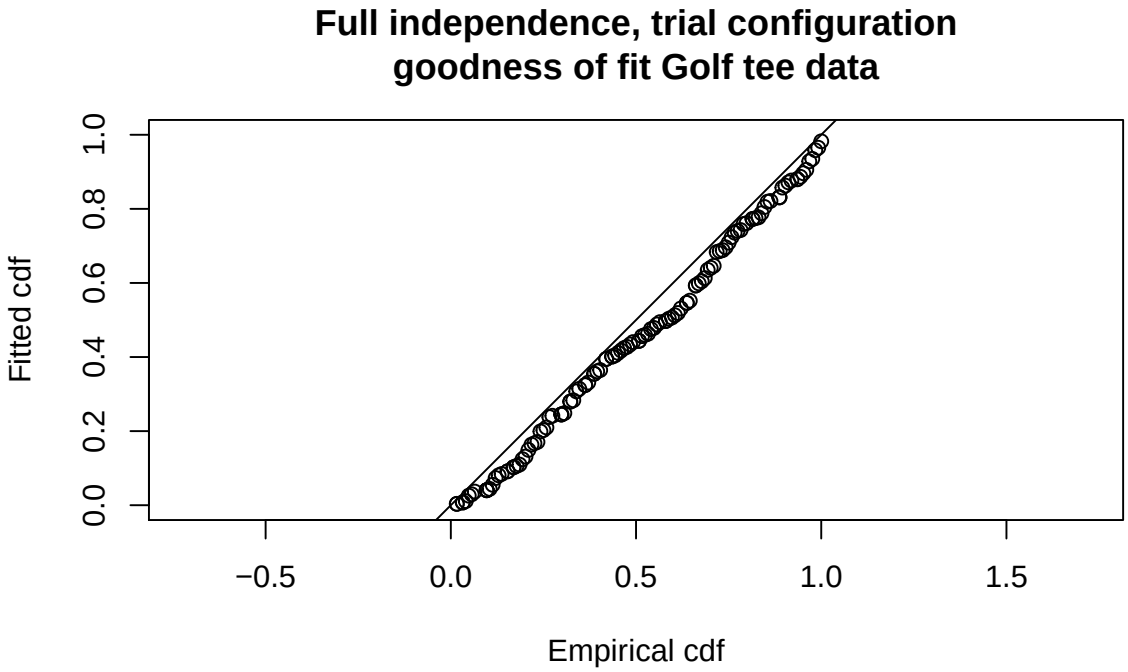
Note that if an independent observer configuration had been chosen, all plots would be available.

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 conventional, 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 configuration\ngoodness of fit G
```

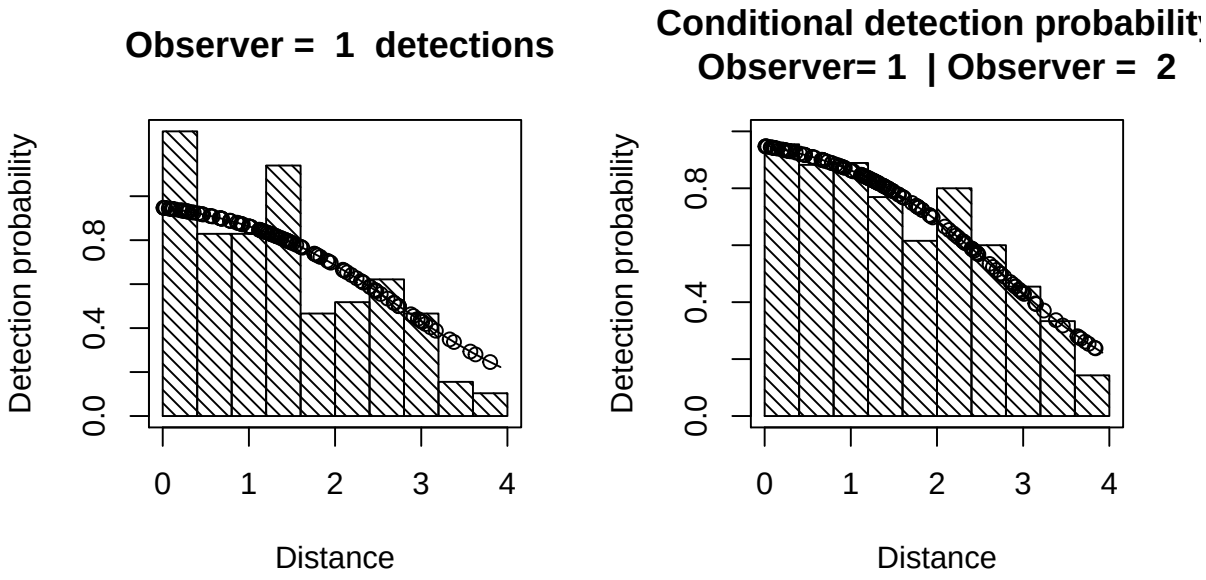



```
# Extract chi-square statistics for reporting
chi.distance <- gof.result$chisquare$chi1$chisq
chi.markrecap <- gof.result$chisquare$chi2$chisq
chi.total <- gof.result$chisquare$pooled.chi
```

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 into 2 columns
par(mfrow=c(1,2))
# 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.
- "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.

There is some evidence of unmodelled heterogeneity in that the fitted line in the left-hand plot declines more slowly than the histogram as the distance increases.

Estimating abundance

Abundance is estimated using the `dht` function. In this function, we need to supply information about the transects and survey regions.

```
# 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
pander::pander(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	1.8	0.12	0.066	3.2	0.21

Region	Area	CoveredArea	Effort	n	ER	se.ER	cv.ER	mean.size	se.mean
2	640	640	80	152	1.9	0.33	0.18	2.9	0.23
Total	1680	1680	210	381	1.8	0.14	0.077	3.1	0.15

```
pander::pander(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	357	32	0.091	295	432	17
2	237	44	0.19	147	380	5.1
Total	593	60	0.1	478	736	16

The estimated abundance is 593 (recall that the true abundance is 760) and so this is somewhat negatively biased. The 95% confidence interval does not include the true value.

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. In the command below, `distance` and `sex` are included in the detection function - remember `sex` was defined as a factor earlier on.

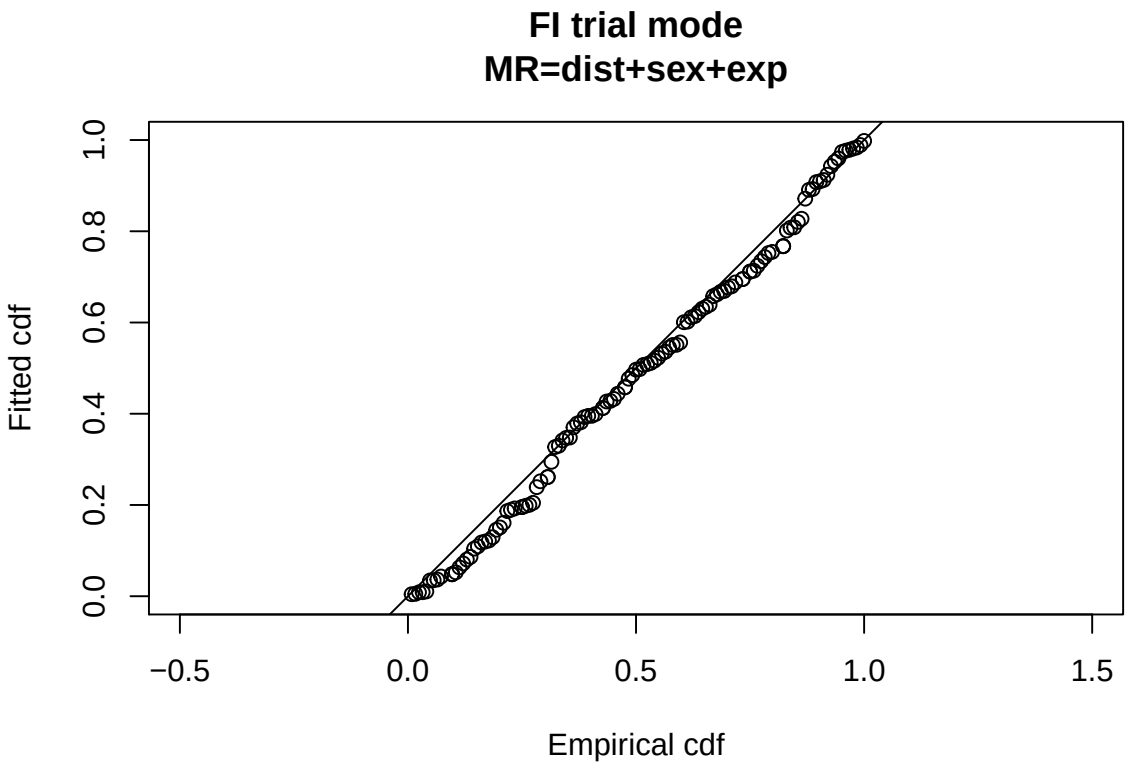
In the code below, all possible models (excluding interaction terms) are fitted.

```
# Full independence model
# Set up list with possible models
mr.formula <- c("~distance","~distance+size","~distance+sex","~distance+exposure","~d
num.mr.models <- length(mr.formula)
# Create dataframe to store results
fi.results <- data.frame(MRmodel=mr.formula, AIC=rep(NA,num.mr.models))
# Loop through all MR models
for (i in 1:num.mr.models) {
  fi.model  <- ddf(method='trial.fi',
                   mrmodel=~glm(link='logit',formula=as.formula(mr.formula[i])),
                  data=detections, meta.data=list(width=4))
  fi.results$AIC[i] <- summary(fi.model)$aic
}
# Calculate delta AIC
fi.results$deltaAIC <- fi.results$AIC - min(fi.results$AIC)
# Order by delta AIC
fi.results <- fi.results[order(fi.results$deltaAIC), ]
# Print results in pretty way
pander::pander(fi.results)
```

	MRmodel	AIC	deltaAIC
7	~distance+sex+exposure	405.7	0
8	~distance+size+sex+exposure	407.4	1.721
4	~distance+exposure	433.7	28.04
3	~distance+sex	434.4	28.74
6	~distance+size+exposure	435.3	29.65
5	~distance+size+sex	436	30.34
1	~distance	452.8	47.13
2	~distance+size	454.6	48.91

We see that the preferred model contains `distance + sex + exposure` and so let's check the goodness-of-fit statistics and detection function plots.

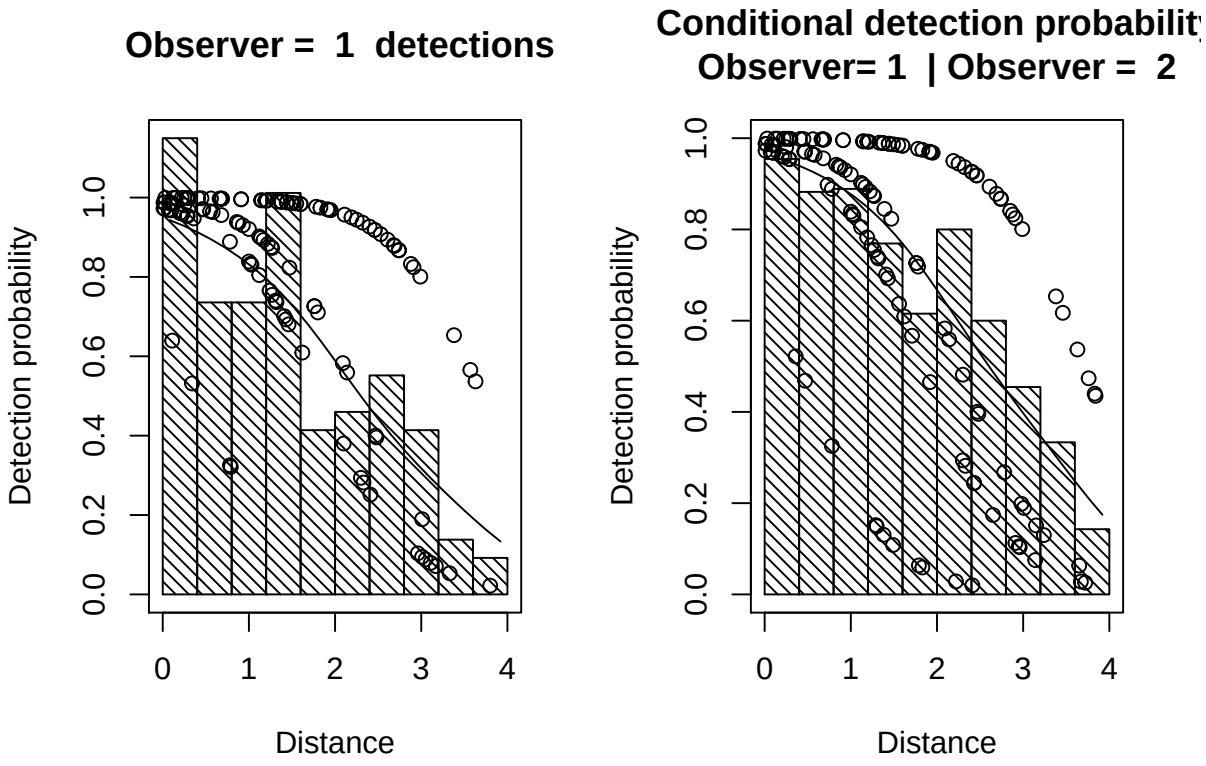
```
# Fit chosen model
fi.mr.dist.sex.exp  <- ddf(method='trial.fi', mrmodel=~glm(link='logit',formula=~dist
                           data=detections, meta.data=list(width=4))

# Check goodness-of-fit
ddf.gof(fi.mr.dist.sex.exp, main="FI trial mode\nMR=dist+sex+exp")
```




```
##
## Goodness of fit results for ddf object
##
## Chi-square tests
##
```

```
## Distance sampling component:
##             [0,0.4]  (0.4,0.8]  (0.8,1.2] (1.2,1.6]   (1.6,2]    (2,2.4]
## Observed  25.000000 16.0000000 16.0000000 22.000000  9.000000 10.0000000
## Expected  20.276214 19.3414709 18.0737423 16.344635 14.082691 11.5105010
## Chisquare  1.100509  0.5772792  0.2379367  1.956798  1.834432  0.1982202
##            (2.4,2.8] (2.8,3.2] (3.2,3.6]  (3.6,4]      Total
## Observed  12.0000000 9.0000000 3.0000000 2.000000 124.000000
## Expected   9.0462873 6.9147393 5.0438158 3.365904 124.000000
## Chisquare  0.9644198 0.6288469 0.8281791 0.554292   8.880913
##
## No degrees of freedom for test
##
## Mark-recapture component:
## Capture History 01
##              [0,0.4]  (0.4,0.8] (0.8,1.2]  (1.2,1.6]   (1.6,2]  (2,2.4]
## Observed  1.00000000 2.00000000 2.0000000 6.00000000 5.0000000 2.000000
## Expected  0.85161169 1.61345653 1.5784634 6.25617270 4.2054953 4.014580
## Chisquare 0.02585579 0.09260606 0.1125735 0.01048955 0.1500983 1.010948
##            (2.4,2.8]  (2.8,3.2] (3.2,3.6]   (3.6,4]     Total
## Observed  6.00000000 6.00000000  2.000000 6.0000000 38.000000
## Expected  6.11790214 6.76552359  1.599467 4.9973276 38.000000
## Chisquare 0.00227217 0.08661952  0.100300 0.2011779  1.792941
## Capture History 11
##                [0,0.4]    (0.4,0.8]   (0.8,1.2]    (1.2,1.6]    (1.6,2]
## Observed  21.000000000 15.000000000 16.00000000 20.000000000 8.00000000
## Expected  21.148388313 15.386543475 16.42153664 19.743827301 8.79450467
## Chisquare  0.001041171  0.009710814  0.01082074  0.003323796 0.07177638
##              (2,2.4]   (2.4,2.8] (2.8,3.2] (3.2,3.6]   (3.6,4]      Total
## Observed  8.0000000 9.000000000 5.0000000 1.0000000 1.0000000 104.000000
## Expected  5.9854201 8.882097863 4.2344764 1.4005328 2.0026724 104.000000
## Chisquare 0.6780697 0.001565048 0.1383941 0.1145468 0.5020052   1.531254
##
##
## Total chi-square = 12.205  P = 0.66344 with 15 degrees of freedom
##
## Distance sampling Cramer-von Mises test (unweighted)
## Test statistic = 0.0976947 p-value = 0.596294
```

```
# Plot detection functions
par(mfrow=c(1,2))
plot(fi.mr.dist.sex.exp)
```




```
# Get abundance estimates
tee.abund.fi <- dht(model=fi.mr.dist.sex.exp, region.table=region, sample.table=sampl

# Print results
tee.abund.fi
```

```
##
## Summary for clusters
##
## Summary statistics:
##   Region Area CoveredArea Effort   n  k        ER      se.ER      cv.ER
## 1      1 1040        1040    130  72  6 0.5538462 0.02926903 0.05284685
## 2      2  640         640     80  52  5 0.6500000 0.08292740 0.12758061
## 3  Total 1680        1680    210 124 11 0.5904762 0.03884115 0.06577936
##
## Abundance:
##   Label  Estimate       se        cv       lcl      ucl        df
## 1     1 119.28976 14.18665 0.1189260  91.64686 155.2704 10.124933
## 2     2  98.17731 18.59356 0.1893876  63.58200 151.5961  7.838438
## 3 Total 217.46707 26.05226 0.1197986 169.90392 278.3451 23.213663
##
## Density:
##   Label  Estimate         se        cv        lcl       ucl        df
## 1     1 0.1147017 0.01364101 0.1189260 0.08812198 0.1492985 10.124933
## 2     2 0.1534020 0.02905244 0.1893876 0.09934687 0.2368689  7.838438
## 3 Total 0.1294447 0.01550730 0.1197986 0.10113328 0.1656816 23.213663
```

```
##
## Summary for individuals
##
## Summary statistics:
##   Region Area CoveredArea Effort   n       ER     se.ER      cv.ER
## 1      1 1040        1040    130 229 1.761538 0.1165805 0.06618107
## 2      2  640         640     80 152 1.900000 0.3342319 0.17591151
## 3  Total 1680        1680    210 381 1.814286 0.1391400 0.07669132
##   mean.size   se.mean
## 1  3.180556 0.2086982
## 2  2.923077 0.2261991
## 3  3.072581 0.1537082
##
## Abundance:
##   Label Estimate       se        cv      lcl      ucl        df
## 1     1 371.0397 37.86856 0.1020607 297.1733 463.2666 11.904078
## 2     2 279.7141 67.25221 0.2404320 154.4960 506.4208  5.482653
## 3 Total 650.7538 82.72648 0.1271241 493.7469 857.6875 11.907386
##
## Density:
##   Label  Estimate         se        cv       lcl       ucl        df
## 1     1 0.3567690 0.03641207 0.1020607 0.2857436 0.4454487 11.904078
## 2     2 0.4370533 0.10508158 0.2404320 0.2414000 0.7912825  5.482653
## 3 Total 0.3873535 0.04924195 0.1271241 0.2938970 0.5105283 11.907386
##
## Expected cluster size
##   Region Expected.S se.Expected.S cv.Expected.S
## 1      1   3.110407     0.2740170    0.08809682
## 2      2   2.849071     0.2211204    0.07761141
## 3  Total   2.992425     0.1758058    0.05875027
```

This model incorporates the effect of more variables causing the heterogeneity. The estimated abundance is 651 which is less biased than the previous estimate and the 95% confidence intervals (494, 858) contain the true value.

The model is a reasonable fit to the data (i.e. non-significant χ^2 and Cramer von Mises tests). This model has a lower AIC (405.7) than the model with only distance (452) and so is to be preferred.

Point independence

A less restrictive assumption than full independence is point independence, which assumes that detections are only independent on the transect centre line i.e. at perpendicular distance zero.

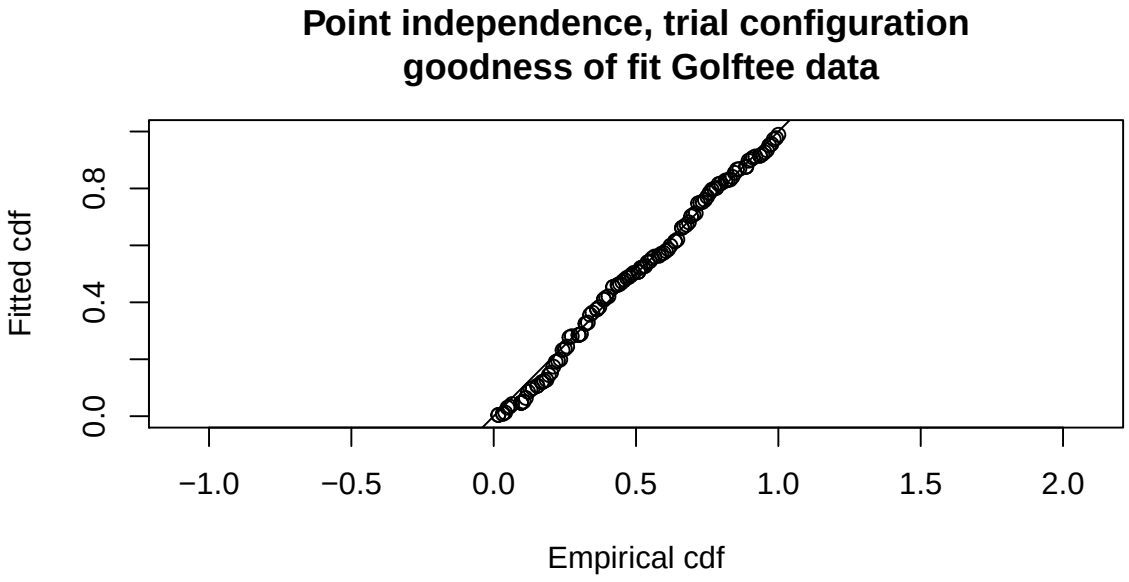
Let's start by seeing if a simple point independence model is better than a simple full independence one. This requires that a distance sampling (DS) model is specified as well a MR model. Here we try a half-normal key function for the DS model.

```r
# Fit trial configuration with point independence 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 configuration\n go
```



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

The AIC for this point independence model is 452.02 which is marginally smaller than the first full independence model that was fitted and hence is to be preferred.

```
# Get abundance estimates
tee.abund.pi <- dht(model=pi.mr.dist, region.table=region, sample.table=samples, obs.

# Print results
tee.abund.pi
```

```
##
## Summary for clusters
##
## Summary statistics:
##   Region Area CoveredArea Effort   n  k        ER      se.ER      cv.ER
## 1      1 1040        1040    130  72  6 0.5538462 0.02926903 0.05284685
## 2      2  640         640     80  52  5 0.6500000 0.08292740 0.12758061
## 3  Total 1680        1680    210 124 11 0.5904762 0.03884115 0.06577936
##
## Abundance:
##   Label  Estimate       se         cv       lcl      ucl        df
## 1     1 130.00869 12.83042 0.09868896 106.66570 158.4601 48.427796
## 2     2  93.89516 14.30894 0.15239269  66.25307 133.0701  8.094139
## 3 Total 223.90385 23.21563 0.10368569 181.78332 275.7840 44.038283
```

```
##
## Density:
##   Label  Estimate         se         cv       lcl       ucl        df
## 1     1 0.1250084 0.01233694 0.09868896 0.1025632 0.1523655 48.427796
## 2     2 0.1467112 0.02235771 0.15239269 0.1035204 0.2079220  8.094139
## 3 Total 0.1332761 0.01381882 0.10368569 0.1082044 0.1641572 44.038283
##
## Summary for individuals
##
## Summary statistics:
##   Region Area CoveredArea Effort   n       ER     se.ER      cv.ER
## 1      1 1040        1040    130 229 1.761538 0.1165805 0.06618107
## 2      2  640         640     80 152 1.900000 0.3342319 0.17591151
## 3  Total 1680        1680    210 381 1.814286 0.1391400 0.07669132
##   mean.size   se.mean
## 1  3.180556 0.2086982
## 2  2.923077 0.2261991
## 3  3.072581 0.1537082
##
## Abundance:
##   Label Estimate       se        cv      lcl      ucl       df
## 1     1 413.4999 44.00745 0.1064268 332.9536 513.5314 30.28937
## 2     2 274.4628 53.42627 0.1946576 171.1754 440.0740  5.98750
## 3 Total 687.9626 79.79845 0.1159924 542.4532 872.5040 25.99319
##
## Density:
##   Label  Estimate         se        cv       lcl       ucl       df
## 1     1 0.3975960 0.04231485 0.1064268 0.3201477 0.4937801 30.28937
## 2     2 0.4288481 0.08347854 0.1946576 0.2674615 0.6876156  5.98750
## 3 Total 0.4095016 0.04749908 0.1159924 0.3228888 0.5193476 25.99319
##
## Expected cluster size
##   Region Expected.S se.Expected.S cv.Expected.S
## 1      1   3.180556     0.2114629    0.06648615
## 2      2   2.923077     0.1750319    0.05987935
## 3  Total   3.072581     0.1391365    0.04528327
```

This results in an estimated abundance of 687. Can we do better if more covariates are included in the DS model?

Covariates in the DS model

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.dist.ds.sex <- 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?

```
# Point independence model
# Include covariates in DS model
# Use selected MR model
# DS models
ds.formula <- c("~size","~sex","~exposure","~size+sex","~size+exposure","~sex+exposur
                "~size+sex+exposure")
num.ds.models <- length(ds.formula)
# Create dataframe to store results
pi.results <- data.frame(DSmodel=ds.formula, AIC=rep(NA,num.ds.models))
# Loop through ds models - use selected MR model from earlier
for (i in 1:num.ds.models) {
  pi.model <- ddf(method='trial', mrmodel=~glm(link='logit',formula=~distance+sex+exp
                  dsmodel=~mcds(key='hn',formula=as.formula(ds.formula[i])),
                  data=detections, meta.data=list(width=4))
  pi.results$AIC[i] <- summary(pi.model)$AIC
}
# Calculate delta AIC
pi.results$deltaAIC <- pi.results$AIC - min(pi.results$AIC)
# Order by delta AIC
pi.results <- pi.results[order(pi.results$deltaAIC), ]
# Print results in pretty way
pander::pander(pi.results)
```

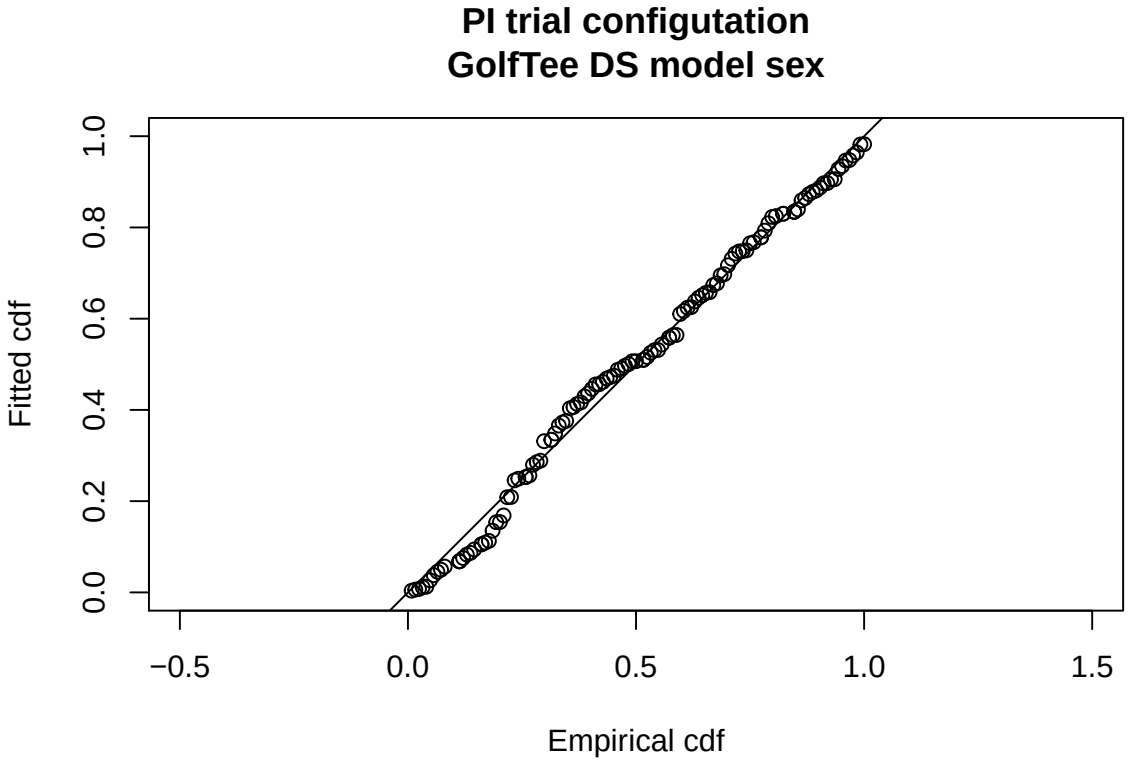
	DSmodel	AIC	deltaAIC
2	~sex	399.3	0
6	~sex+exposure	400.3	1.021
4	~size+sex	401.1	1.802
7	~size+sex+exposure	401.9	2.686
1	~size	407.9	8.657
3	~exposure	408	8.715
5	~size+exposure	409.9	10.63

This indicates that sex should be included in the DS model. We do this and check the goodness of fit and obtain abundance.

```
# Fit chosen model
pi.ds.sex <- ddf(method='trial', mrmodel=~glm(link='logit',formula=~distance+sex+expo
                 dsmodel=~mcds(key='hn',formula=~sex), data=detections,
                 meta.data=list(width=4))
summary(pi.ds.sex)
```

```
##
## 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                                   :  94.89911
##
##
## Conditional detection function parameters:
##                estimate        se
## (Intercept)  0.7870962 0.6774633
## distance    -1.9435496 0.3706866
## sex1         2.8059863 0.6828331
## exposure1    3.6094527 0.7332797
##
##                        Estimate         SE         CV
## Average primary p(0) 0.9697357 0.02018876 0.02081883
##
##
##
## Summary for ds object
## Number of observations :  124
## Distance range         :  0  -  4
## AIC                    :  304.3594
##
## Detection function:
##  Half-normal key function
##
## Detection function parameters
## Scale coefficient(s):
##               estimate        se
## (Intercept) 0.2525377 0.1327279
## sex1        0.5832341 0.2041197
##
##            Estimate         SE         CV
## Average p 0.5605421 0.04616396 0.08235592
##
##
## Summary for trial object
##
## Total AIC value =  399.2585
##                          Estimate          SE         CV
## Average p              0.5435777  0.04643944 0.08543294
## N in covered region 228.1182639 24.21314095 0.10614293
```

```
# Check goodness-of-fit
ddf.gof(pi.ds.sex, main="PI trial configutation\nGolfTee DS model sex")
```




```
##
## Goodness of fit results for ddf object
##
## Chi-square tests
##
## Distance sampling component:
##              [0,0.4] (0.4,0.8]  (0.8,1.2] (1.2,1.6]   (1.6,2]     (2,2.4]
## Observed  25.0000000 16.000000 16.0000000 22.000000  9.000000 10.00000000
## Expected  21.9165416 20.740242 18.6299046 15.975863 13.181194 10.55317365
## Chisquare  0.4338146  1.083396  0.3712525  2.271566  1.326313  0.02899612
##           (2.4,2.8] (2.8,3.2] (3.2,3.6]  (3.6,4]      Total
## Observed  12.000000  9.000000  3.000000 2.000000 124.000000
## Expected   8.260839  6.353756  4.809648 3.578838 124.000000
## Chisquare  1.692483  1.102121  0.680887 0.696519   9.687346
##
## P = 0.20699 with 7 degrees of freedom
##
## Mark-recapture component:
## Capture History 01
##              [0,0.4]  (0.4,0.8] (0.8,1.2]  (1.2,1.6]   (1.6,2]  (2,2.4]
## Observed  1.00000000 2.00000000 2.0000000 6.00000000 5.0000000 2.000000
## Expected  0.85161169 1.61345653 1.5784634 6.25617270 4.2054953 4.014580
## Chisquare 0.02585579 0.09260606 0.1125735 0.01048955 0.1500983 1.010948
##            (2.4,2.8]  (2.8,3.2] (3.2,3.6]   (3.6,4]     Total
## Observed  6.00000000 6.00000000  2.000000 6.0000000 38.000000
## Expected  6.11790214 6.76552359  1.599467 4.9973276 38.000000
```

```
## Chisquare 0.00227217 0.08661952  0.100300 0.2011779  1.792941
## Capture History 11
##                [0,0.4]    (0.4,0.8]   (0.8,1.2]    (1.2,1.6]    (1.6,2]
## Observed  21.000000000 15.000000000 16.00000000 20.000000000 8.00000000
## Expected  21.148388313 15.386543475 16.42153664 19.743827301 8.79450467
## Chisquare  0.001041171  0.009710814  0.01082074  0.003323796 0.07177638
##              (2,2.4]   (2.4,2.8] (2.8,3.2] (3.2,3.6]   (3.6,4]      Total
## Observed  8.0000000 9.000000000 5.0000000 1.0000000 1.0000000 104.000000
## Expected  5.9854201 8.882097863 4.2344764 1.4005328 2.0026724 104.000000
## Chisquare 0.6780697 0.001565048 0.1383941 0.1145468 0.5020052   1.531254
##
## MR total chi-square = 3.3242  P = 0.76719 with 6 degrees of freedom
##
##
## Total chi-square = 13.012  P = 0.44692 with 13 degrees of freedom
##
## Distance sampling Cramer-von Mises test (unweighted)
## Test statistic = 0.081285 p-value = 0.684457
```

```
# Get abundance estimates
tee.abund.pi <- dht(model=pi.ds.sex, region.table=region, sample.table=samples, obs.t
tee.abund.pi
```

```
##
## Summary for clusters
##
## Summary statistics:
##   Region Area CoveredArea Effort   n  k        ER      se.ER      cv.ER
## 1      1 1040        1040    130  72  6 0.5538462 0.02926903 0.05284685
## 2      2  640         640     80  52  5 0.6500000 0.08292740 0.12758061
## 3  Total 1680        1680    210 124 11 0.5904762 0.03884115 0.06577936
##
## Abundance:
##   Label Estimate       se         cv       lcl      ucl       df
## 1     1 125.7678 12.50318 0.09941474 102.97943 153.5991 43.66336
## 2     2 102.3504 17.53163 0.17129022  68.75817 152.3544  7.39421
## 3 Total 228.1183 25.15323 0.11026399 182.12573 285.7254 28.04581
##
## Density:
##   Label  Estimate         se         cv        lcl       ucl       df
## 1     1 0.1209306 0.01202229 0.09941474 0.09901868 0.1476914 43.66336
## 2     2 0.1599226 0.02739317 0.17129022 0.10743464 0.2380538  7.39421
## 3 Total 0.1357847 0.01497216 0.11026399 0.10840817 0.1700746 28.04581
##
## Summary for individuals
##
## Summary statistics:
##   Region Area CoveredArea Effort   n       ER     se.ER      cv.ER
## 1      1 1040        1040    130 229 1.761538 0.1165805 0.06618107
```

```
## 2      2  640         640     80 152 1.900000 0.3342319 0.17591151
## 3  Total 1680        1680    210 381 1.814286 0.1391400 0.07669132
##   mean.size   se.mean
## 1  3.180556 0.2086982
## 2  2.923077 0.2261991
## 3  3.072581 0.1537082
##
## Abundance:
##   Label Estimate       se        cv      lcl      ucl        df
## 1     1 395.0545 36.33952 0.0919861 329.0883 474.2437 79.295718
## 2     2 299.7763 65.43242 0.2182709 175.5600 511.8809  5.685148
## 3 Total 694.8307 84.25554 0.1212605 537.2145 898.6908 15.167370
##
## Density:
##   Label  Estimate         se        cv       lcl       ucl        df
## 1     1 0.3798601 0.03494185 0.0919861 0.3164310 0.4560035 79.295718
## 2     2 0.4684004 0.10223816 0.2182709 0.2743125 0.7998140  5.685148
## 3 Total 0.4135897 0.05015211 0.1212605 0.3197705 0.5349350 15.167370
##
## Expected cluster size
##   Region Expected.S se.Expected.S cv.Expected.S
## 1      1   3.141141     0.2081675    0.06627130
## 2      2   2.928920     0.1866200    0.06371633
## 3  Total   3.045923     0.1371508    0.04502767
```

This model estimated an abundance of 695, which is closest to the true value of all the models - it is still less than the true value indicating, perhaps, some unmodelled heterogeneity on the trackline (or perhaps just bad luck - remember this was only one survey).

Was this complex modelling worthwhile? In this case, the estimated $p(0)$ for the best model was 0.97 (which is very close to 1). If we ran a conventional distance sampling analysis, pooling the data from the two observers, we should get a very robust estimate of true abundance.

2. 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`.

```
crabseal <- read.csv("crabbieMRDS.csv")
```

Crabeater seal analyses

The observer teams acted independently and so an 'independent observer' (IO) configuration can be specified. The code below fits simple models (i.e. distance only) with the full independence assumption and the point independence assumption. For each model, make a note of the estimated values for $p(0)$ for each observer and the observers combined. Check goodness-of-fit and plot the detection function.

Full independence

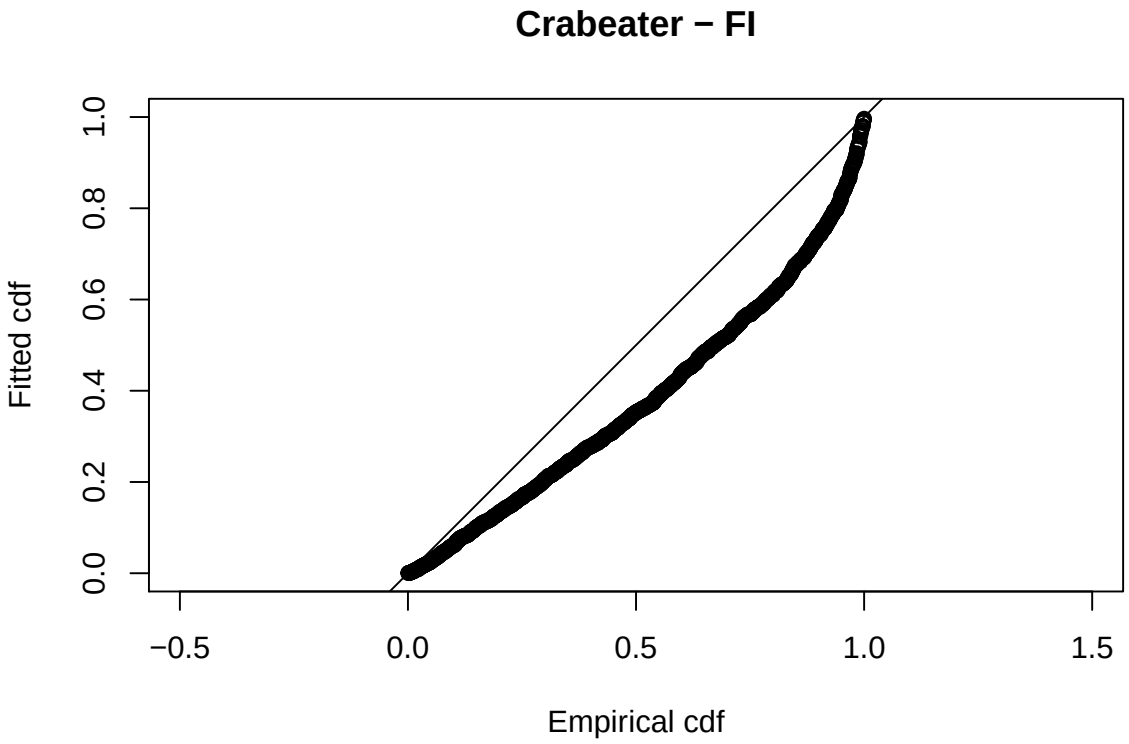
Fit an IO configuration assuming full independence:

```
# IO configuration - full independence
# MR model - distance only
# Truncation 700m
fi.mr.dist <- ddf(method="io.fi", mrmodel=~glm(link="logit", formula=~distance),
                  data=crabseal, meta.data=list(width=700))
summary(fi.mr.dist)
```

```
##
## 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                      :  25681.52
##
```

```
##
## Conditional detection function parameters:
##                  estimate           se
## (Intercept)  2.107762345 0.0994391200
## distance    -0.003087713 0.0003159216
##
##                              Estimate           SE          CV
## Average p                   0.9071952  0.009684936 0.010675691
## Average primary p(0)        0.8916554  0.007472293 0.008380248
## Average secondary p(0)      0.8916554  0.007472293 0.008380248
## Average combined p(0)       0.9882614  0.004402403 0.004454695
## N in covered region    1917.9995344 24.808748709 0.012934700
```

```
# Goodness of fit
ddf.gof(fi.mr.dist, main="Crabeater - FI")
```



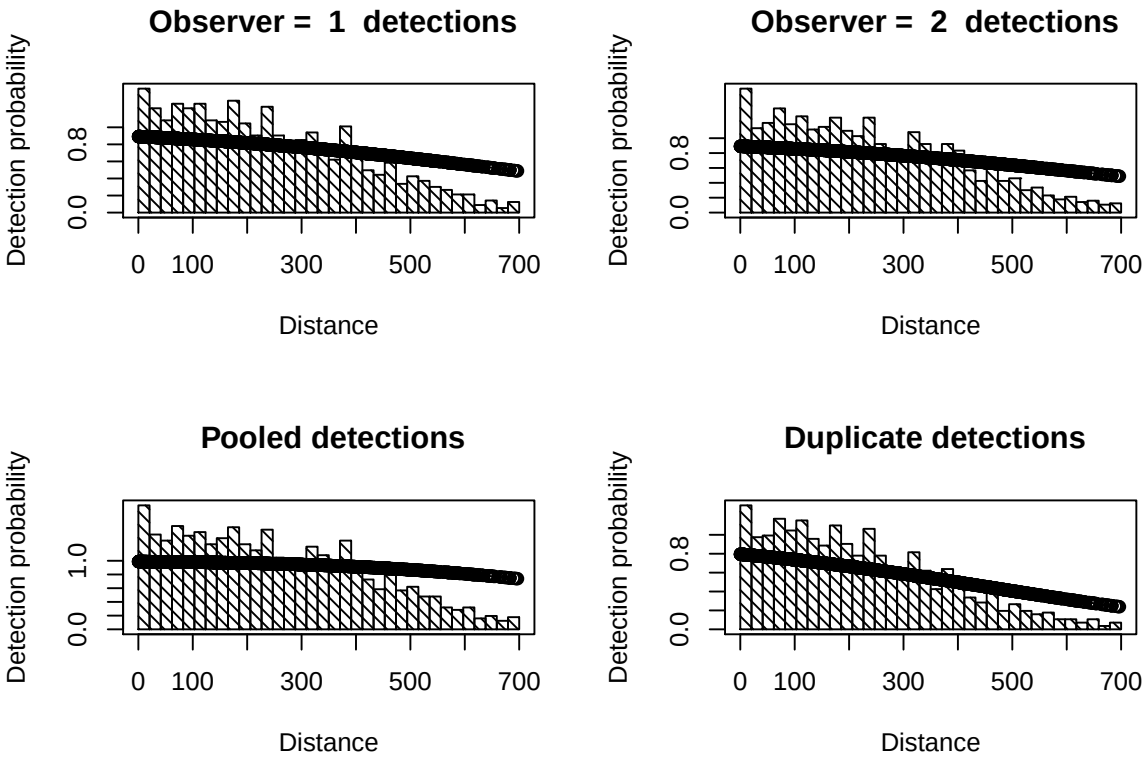

```
##
## Goodness of fit results for ddf object
##
## Chi-square tests
##
## Distance sampling component:
##             [0,20.6] (20.6,41.2] (41.2,61.8] (61.8,82.4] (82.4,103]
## Observed  102.00000   78.000000   73.000000    85.00000  77.000000
## Expected   55.71066   55.627135   55.534399    55.43157  55.317698
## Chisquare  38.46128    8.998218    5.492941    15.77246   8.498586
##           (103,124] (124,144] (144,165] (165,185] (185,206] (206,226]
```

```
## Observed   80.00000 70.000000 75.000000  84.00000 70.000000 65.000000
## Expected   55.19178 55.052752 54.899483  54.73079 54.545434 54.342131
## Chisquare  11.15107  4.058293  7.359464  15.65274  4.378801  2.090278
##           (226,247]  (247,268]  (268,288]  (288,309] (309,329] (329,350]
## Observed   82.00000 59.0000000 51.0000000 56.0000000 68.000000  61.00000
## Expected   54.11955 53.8763369 53.6110994 53.3224417 53.008966  52.66929
## Chisquare  14.36300  0.4872626  0.1271722  0.1344522  4.239492   1.31767
##           (350,371] (371,391]  (391,412] (412,432] (432,453]  (453,474]
## Observed  48.0000000 73.000000 54.0000000 41.000000 33.000000 45.0000000
## Expected  52.3020631 51.905984 51.4798187 51.022424 50.532764 50.0099331
## Chisquare  0.3538627  8.572374  0.1233748  1.968722  6.083138  0.5018889
##           (474,494] (494,515] (515,535] (535,556] (556,576] (576,597]
## Observed  32.000000 35.000000 27.000000 27.000000  18.00000  16.00000
## Expected  49.453179 48.861918 48.235760 47.574521  46.87824  46.14720
## Chisquare  6.159633  3.932567  9.349028  8.897849  17.78976  19.69467
##           (597,618] (618,638] (638,659] (659,679] (679,700]     Total
## Observed   18.00000   9.00000  11.00000   7.00000  10.00000 1740.0000
## Expected   45.38193  44.58319  43.75203  42.88972  41.99781 1740.0000
## Chisquare  16.52133  28.40002  24.51761  30.03219  24.37889  349.8601
##
## No degrees of freedom for test
##
## Mark-recapture component:
## Capture History 10
##             [0,20.6] (20.6,41.2] (41.2,61.8] (61.8,82.4]  (82.4,103]
## Observed   8.0000000   14.000000    5.000000    6.000000 10.00000000
## Expected  10.2048866    8.231391    8.113609    9.910676  9.41082786
## Chisquare  0.4763919    4.042676    1.194852    1.543122  0.03688558
##           (103,124] (124,144] (144,165]   (165,185]  (185,206] (206,226]
## Observed    7.00000 7.0000000 10.000000 12.00000000  8.0000000  7.000000
## Expected   10.25180 9.4180301 10.554005 12.40538306 10.7785958 10.439919
## Chisquare   1.03145 0.6208166  0.029081  0.01324711  0.7162895  1.133442
##           (226,247] (247,268] (268,288] (288,309] (309,329]  (329,350]
## Observed  10.000000  7.000000  4.000000 12.000000  7.000000  9.0000000
## Expected  13.726834 10.333065  9.281080 10.583647 13.417015 12.4813426
## Chisquare  1.011835  1.075124  3.005018  0.189543  3.069094  0.9710291
##             (350,371] (371,391] (391,412]  (412,432] (432,453]  (453,474]
## Observed  11.00000000 21.000000  7.000000 9.00000000 9.0000000 10.0000000
## Expected  10.18502017 16.059675 12.335963 9.67302588 8.0502260 11.3161495
## Chisquare  0.06521265  1.519758  2.308089 0.04682752 0.1120553  0.1530776
##            (474,494]  (494,515]  (515,535]  (535,556]  (556,576] (576,597]
## Observed  8.00000000 9.00000000 10.0000000 8.00000000 5.00000000 6.0000000
## Expected  8.31955348 9.36144247  7.4364943 7.63826938 5.25594828 4.7749107
## Chisquare 0.01227403 0.01395518  0.8836908 0.01713072 0.01246388 0.3143187
##            (597,618] (618,638] (638,659] (659,679]  (679,700]     Total
## Observed  6.00000000  1.000000 2.0000000 1.0000000 3.00000000 269.00000
## Expected  5.53392046  2.823986 3.5330208 2.3030262 3.35058915 307.49333
## Chisquare 0.03925429  1.178095 0.6651964 0.7372375 0.03668392  28.27522
```

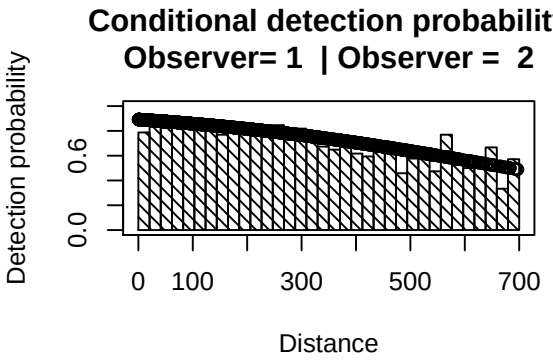
```
## Capture History 01
##            [0,20.6] (20.6,41.2] (41.2,61.8] (61.8,82.4] (82.4,103]
## Observed  20.000000  9.00000000   12.000000  13.0000000  8.0000000
## Expected  10.204887  8.23139113    8.113609   9.9106757  9.4108279
## Chisquare  9.401795  0.07176911    1.861568   0.9629943  0.2115048
##            (103,124]  (124,144] (144,165]  (165,185]    (185,206]
## Observed   8.0000000 9.00000000 15.000000 10.0000000 11.000000000
## Expected  10.2518027 9.41803006 10.554005 12.4053831 10.778595817
## Chisquare  0.4946072 0.01855474  1.872927  0.4663998  0.004547885
##           (206,226]  (226,247]  (247,268]   (268,288]  (288,309] (309,329]
## Observed  14.000000 12.0000000  8.0000000 10.00000000  8.0000000 15.000000
## Expected  10.439919 13.7268340 10.3330651  9.28108040 10.5836469 13.417015
## Chisquare  1.214011  0.2172355  0.5267743  0.05568806  0.6307118  0.186766
##           (329,350]  (350,371]    (371,391] (391,412] (412,432]
## Observed  17.000000 13.0000000 1.600000e+01 18.000000 13.000000
## Expected  12.481343 10.1850202 1.605967e+01 12.335963  9.673026
## Chisquare  1.635903  0.7780163 2.217414e-04  2.600633  1.144291
##             (432,453]   (453,474] (474,494] (494,515] (515,535]  (535,556]
## Observed  8.000000000 12.00000000 13.000000 11.000000  6.000000 10.0000000
## Expected  8.050226002 11.31614950  8.319553  9.361442  7.436494  7.6382694
## Chisquare 0.000313364  0.04132603  2.633144  0.286801  0.277485  0.7302402
##           (556,576] (576,597]  (597,618] (618,638]  (638,659] (659,679]
## Observed  3.0000000 4.0000000 6.00000000 4.0000000 3.00000000  4.000000
## Expected  5.2559483 4.7749107 5.53392046 2.8239856 3.53302079  2.303026
## Chisquare 0.9682939 0.1257587 0.03925429 0.4897368 0.08041593  1.250407
##            (679,700]     Total
## Observed  3.00000000 346.00000
## Expected  3.35058915 307.49333
## Chisquare 0.03668392  31.31678
## Capture History 11
##             [0,20.6] (20.6,41.2] (41.2,61.8] (61.8,82.4]  (82.4,103]
## Observed  74.0000000  55.0000000 56.00000000 66.00000000 59.00000000
## Expected  81.5902269  61.5372177 56.77278148 65.17864860 58.17834429
## Chisquare  0.7061084   0.6944613  0.01051897  0.01035029  0.01160429
##           (103,124]  (124,144]  (144,165]  (165,185]  (185,206]
## Observed  65.000000 54.0000000 50.0000000 62.0000000 51.0000000
## Expected  59.496395 51.1639399 53.8919909 59.1892339 48.4428084
## Chisquare  0.509101  0.1572052  0.2810732  0.1334771  0.1349886
##              (206,226]  (226,247]  (247,268]  (268,288]   (288,309]
## Observed  4.400000e+01 60.0000000 44.0000000 37.0000000 36.00000000
## Expected  4.412016e+01 54.5463321 38.3338699 32.4378392 34.83270625
## Chisquare 3.272588e-04  0.5452703  0.8375108  0.6416368  0.03911768
##            (309,329]   (329,350]  (350,371] (371,391]    (391,412]
## Observed  46.0000000 35.00000000 24.0000000  36.00000 29.000000000
## Expected  41.1659701 36.03731473 27.6299597  40.88065 29.328074375
## Chisquare  0.5676496  0.02985855  0.4768956   0.58269  0.003669958
##            (412,432]   (432,453]   (453,474] (474,494]  (494,515]
## Observed  19.0000000 16.00000000 23.00000000 11.000000 15.0000000
```

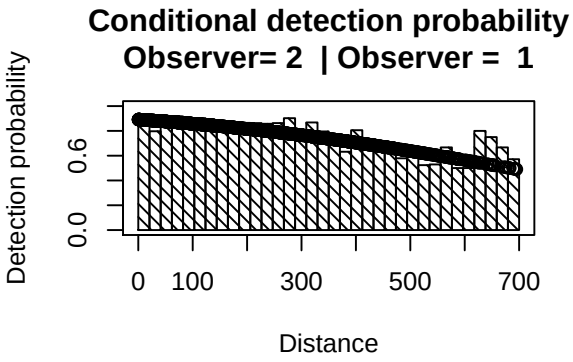
```
## Expected  21.6539482 16.89954800 22.36770099 15.360893 16.2771151
## Chisquare  0.3252728  0.04788214  0.01787408  1.238039  0.1002034
##            (515,535]  (535,556]  (556,576]  (576,597] (597,618] (618,638]
## Observed  11.0000000  9.0000000 10.0000000 6.00000000 6.0000000 4.0000000
## Expected  12.1270114 11.7234612  7.4881034 6.45017864 6.9321591 3.3520287
## Chisquare  0.1047377  0.6326836  0.8426198 0.03141941 0.1253463 0.1252575
##           (638,659]  (659,679] (679,700]      Total
## Observed   6.000000 2.00000000 4.0000000 1125.00000
## Expected   3.933958 2.39394760 3.2988217 1125.01334
## Chisquare  1.085047 0.06482795 0.1490384   11.26376
##
##
## Total chi-square = 420.72  P = 0 with 99 degrees of freedom
##
## Distance sampling Cramer-von Mises test (unweighted)
## Test statistic = 30.1576 p-value = 0.0494425
```

```
# Detection functions
par(mfrow=c(2,2))
plot(fi.mr.dist, which=c(1,2,3,4))
```




```
plot(fi.mr.dist, which=c(5,6))
```





Point independence

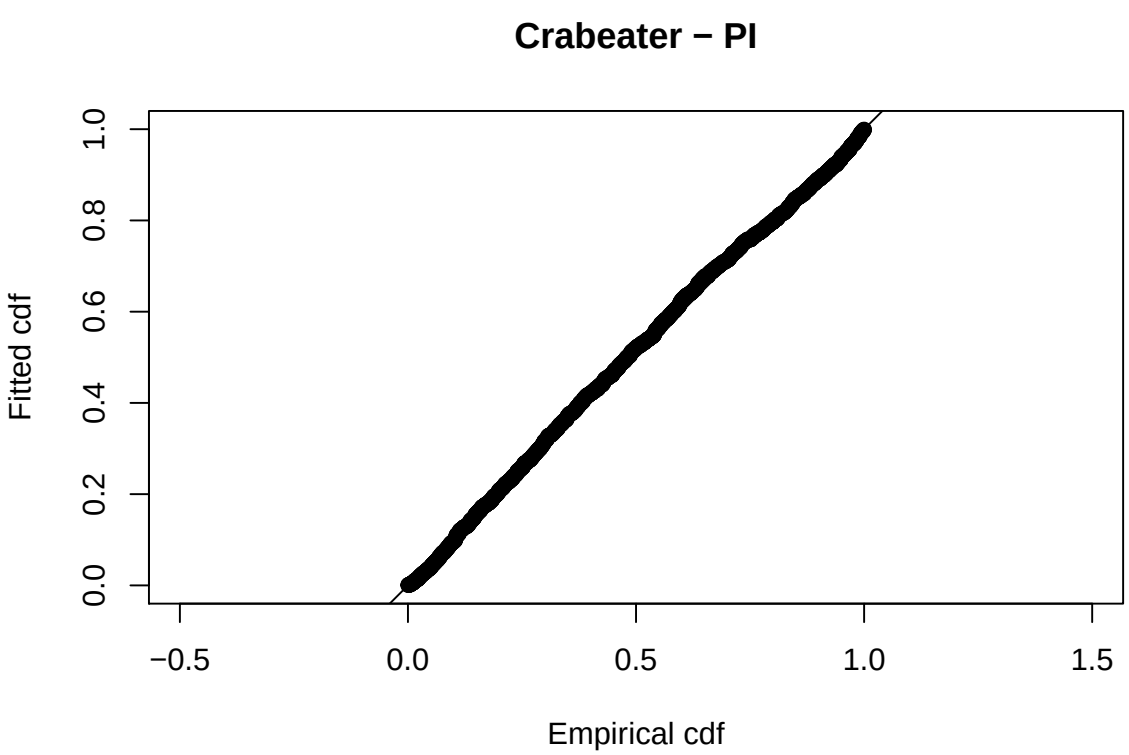
Next fit IO configuration assuming full independence: specify a half-normal key function for the DS model: again only include perpendicular distance in the MR model.

```
# IO configuration - point independence
# MR model - distance only
# DS model - half normal detection function, no additional covars
# Truncation at 700m
pi.mr.dist.ds.hn <- ddf(method="io", dsmodel=~cds(key="hn"),
                  mrmodel=~glm(link="logit", formula=~distance),
                  data=crabseal, meta.data=list(width=700))
summary(pi.mr.dist.ds.hn)
```

```
##
## 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
```

```r
# Goodness of fit
ddf.gof(pi.mr.dist.ds.hn, main="Crabeater - PI")
```



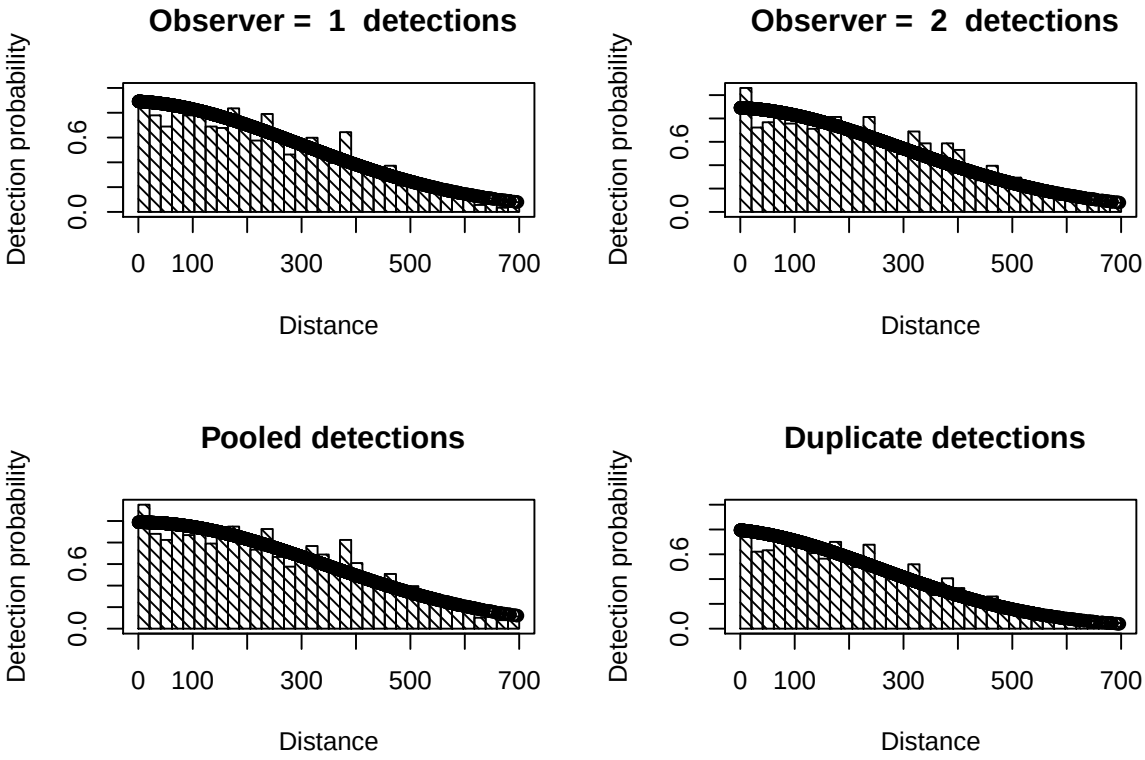

```
##
## Goodness of fit results for ddf object
##
## Chi-square tests
##
## Distance sampling component:
##             [0,20.6] (20.6,41.2] (41.2,61.8]  (61.8,82.4] (82.4,103]
## Observed  102.000000  78.0000000   73.000000 85.000000000 77.0000000
## Expected   87.489432  87.1691605   86.532131 85.585303840 84.3389643
## Chisquare   2.406652   0.9644868    2.116191  0.004002797  0.6386182
##             (103,124] (124,144] (144,165]  (165,185]  (185,206]  (206,226]
## Observed  80.00000000 70.000000 75.000000 84.0000000 70.0000000 65.0000000
## Expected  82.80653211 81.004323 78.951260 76.6685408 74.1792773 71.5081042
## Chisquare  0.09512079  1.494922  0.197748  0.7010737  0.2354614  0.5923164
##           (226,247] (247,268] (268,288]  (288,309] (309,329] (329,350]
## Observed  82.000000 59.000000 51.000000 56.0000000  68.00000 61.000000
## Expected  68.680777 65.723760 62.663820 59.5276308  56.34140 53.130495
## Chisquare  2.582989  0.687863  2.171025  0.2090488   2.41249  1.165604
##             (350,371] (371,391] (391,412]    (412,432]  (432,453]
## Observed  48.00000000  73.00000 54.000000 41.000000000 33.0000000
## Expected  49.91917489  46.73026 43.584921 40.502478263 37.5002530
## Chisquare  0.07378392  14.76772  2.488793  0.006111426  0.5400571
##           (453,474]    (474,494] (494,515]    (515,535]  (535,556]
## Observed  45.000000 32.000000000 35.000000 27.000000000 27.0000000
## Expected  34.593465 31.795173261 29.116260 26.565455124 24.1493915
## Chisquare  3.130533  0.001319508  1.188971  0.007108075  0.3364875
```

```
##           (556,576] (576,597]    (597,618] (618,638]  (638,659] (659,679]
## Observed  18.000000 16.000000 18.000000000  9.000000 11.0000000  7.000000
## Expected  21.872700 19.738123 17.746658325 15.897711 14.1892634 12.618054
## Chisquare  0.685686  0.707948  0.003616568  2.992784  0.7168378  2.501379
##           (679,700]      Total
## Observed  10.0000000 1740.00000
## Expected  11.1797522 1740.00000
## Chisquare  0.1244943   48.94924
##
## P = 0.028069 with 32 degrees of freedom
##
## Mark-recapture component:
## Capture History 10
##             [0,20.6] (20.6,41.2] (41.2,61.8] (61.8,82.4]  (82.4,103]
## Observed   8.0000000   14.000000    5.000000    6.000000 10.00000000
## Expected  10.2048866    8.231391    8.113609    9.910676  9.41082786
## Chisquare  0.4763919    4.042676    1.194852    1.543122  0.03688558
##           (103,124] (124,144] (144,165]   (165,185]  (185,206] (206,226]
## Observed    7.00000 7.0000000 10.000000 12.00000000  8.0000000  7.000000
## Expected   10.25180 9.4180301 10.554005 12.40538306 10.7785958 10.439919
## Chisquare   1.03145 0.6208166  0.029081  0.01324711  0.7162895  1.133442
##           (226,247] (247,268] (268,288] (288,309] (309,329]  (329,350]
## Observed  10.000000  7.000000  4.000000 12.000000  7.000000  9.0000000
## Expected  13.726834 10.333065  9.281080 10.583647 13.417015 12.4813426
## Chisquare  1.011835  1.075124  3.005018  0.189543  3.069094  0.9710291
##             (350,371] (371,391] (391,412]  (412,432] (432,453]  (453,474]
## Observed  11.00000000 21.000000  7.000000 9.00000000 9.0000000 10.0000000
## Expected  10.18502017 16.059675 12.335963 9.67302588 8.0502260 11.3161495
## Chisquare  0.06521265  1.519758  2.308089 0.04682752 0.1120553  0.1530776
##            (474,494]  (494,515]  (515,535]  (535,556]  (556,576] (576,597]
## Observed  8.00000000 9.00000000 10.0000000 8.00000000 5.00000000 6.0000000
## Expected  8.31955348 9.36144247  7.4364943 7.63826938 5.25594828 4.7749107
## Chisquare 0.01227403 0.01395518  0.8836908 0.01713072 0.01246388 0.3143187
##            (597,618] (618,638] (638,659] (659,679]  (679,700]     Total
## Observed  6.00000000  1.000000 2.0000000 1.0000000 3.00000000 269.00000
## Expected  5.53392046  2.823986 3.5330208 2.3030262 3.35058915 307.49333
## Chisquare 0.03925429  1.178095 0.6651964 0.7372375 0.03668392  28.27522
## Capture History 01
##            [0,20.6] (20.6,41.2] (41.2,61.8] (61.8,82.4] (82.4,103]
## Observed  20.000000  9.00000000   12.000000  13.0000000  8.0000000
## Expected  10.204887  8.23139113    8.113609   9.9106757  9.4108279
## Chisquare  9.401795  0.07176911    1.861568   0.9629943  0.2115048
##            (103,124]  (124,144] (144,165]  (165,185]    (185,206]
## Observed   8.0000000 9.00000000 15.000000 10.0000000 11.000000000
## Expected  10.2518027 9.41803006 10.554005 12.4053831 10.778595817
## Chisquare  0.4946072 0.01855474  1.872927  0.4663998  0.004547885
##           (206,226]  (226,247]  (247,268]   (268,288]  (288,309] (309,329]
## Observed  14.000000 12.0000000  8.0000000 10.00000000  8.0000000 15.000000
```

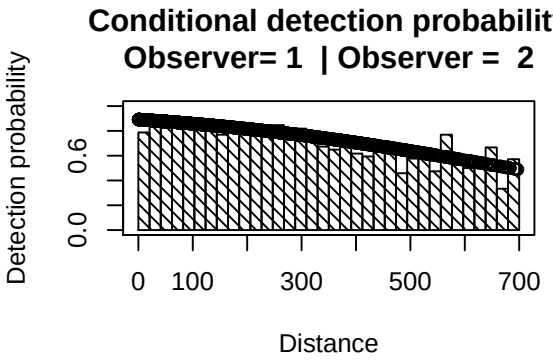
```
## Expected  10.439919 13.7268340 10.3330651  9.28108040 10.5836469 13.417015
## Chisquare  1.214011  0.2172355  0.5267743  0.05568806  0.6307118  0.186766
##           (329,350]  (350,371]    (371,391] (391,412] (412,432]
## Observed  17.000000 13.0000000 1.600000e+01 18.000000 13.000000
## Expected  12.481343 10.1850202 1.605967e+01 12.335963  9.673026
## Chisquare  1.635903  0.7780163 2.217414e-04  2.600633  1.144291
##             (432,453]   (453,474] (474,494] (494,515] (515,535]  (535,556]
## Observed  8.000000000 12.00000000 13.000000 11.000000  6.000000 10.0000000
## Expected  8.050226002 11.31614950  8.319553  9.361442  7.436494  7.6382694
## Chisquare 0.000313364  0.04132603  2.633144  0.286801  0.277485  0.7302402
##           (556,576] (576,597]  (597,618] (618,638]  (638,659] (659,679]
## Observed  3.0000000 4.0000000 6.00000000 4.0000000 3.00000000  4.000000
## Expected  5.2559483 4.7749107 5.53392046 2.8239856 3.53302079  2.303026
## Chisquare 0.9682939 0.1257587 0.03925429 0.4897368 0.08041593  1.250407
##            (679,700]     Total
## Observed  3.00000000 346.00000
## Expected  3.35058915 307.49333
## Chisquare 0.03668392  31.31678
## Capture History 11
##             [0,20.6] (20.6,41.2] (41.2,61.8] (61.8,82.4]  (82.4,103]
## Observed  74.0000000  55.0000000 56.00000000 66.00000000 59.00000000
## Expected  81.5902269  61.5372177 56.77278148 65.17864860 58.17834429
## Chisquare  0.7061084   0.6944613  0.01051897  0.01035029  0.01160429
##           (103,124]  (124,144]  (144,165]  (165,185]  (185,206]
## Observed  65.000000 54.0000000 50.0000000 62.0000000 51.0000000
## Expected  59.496395 51.1639399 53.8919909 59.1892339 48.4428084
## Chisquare  0.509101  0.1572052  0.2810732  0.1334771  0.1349886
##              (206,226]  (226,247]  (247,268]  (268,288]   (288,309]
## Observed  4.400000e+01 60.0000000 44.0000000 37.0000000 36.00000000
## Expected  4.412016e+01 54.5463321 38.3338699 32.4378392 34.83270625
## Chisquare 3.272588e-04  0.5452703  0.8375108  0.6416368  0.03911768
##            (309,329]   (329,350]  (350,371] (371,391]    (391,412]
## Observed  46.0000000 35.00000000 24.0000000  36.00000 29.000000000
## Expected  41.1659701 36.03731473 27.6299597  40.88065 29.328074375
## Chisquare  0.5676496  0.02985855  0.4768956   0.58269  0.003669958
##            (412,432]   (432,453]   (453,474] (474,494]  (494,515]
## Observed  19.0000000 16.00000000 23.00000000 11.000000 15.0000000
## Expected  21.6539482 16.89954800 22.36770099 15.360893 16.2771151
## Chisquare  0.3252728  0.04788214  0.01787408  1.238039  0.1002034
##            (515,535]  (535,556]  (556,576]  (576,597] (597,618] (618,638]
## Observed  11.0000000  9.0000000 10.0000000 6.00000000 6.0000000 4.0000000
## Expected  12.1270114 11.7234612  7.4881034 6.45017864 6.9321591 3.3520287
## Chisquare  0.1047377  0.6326836  0.8426198 0.03141941 0.1253463 0.1252575
##           (638,659]  (659,679] (679,700]      Total
## Observed   6.000000 2.00000000 4.0000000 1125.00000
## Expected   3.933958 2.39394760 3.2988217 1125.01334
## Chisquare  1.085047 0.06482795 0.1490384   11.26376
##
```

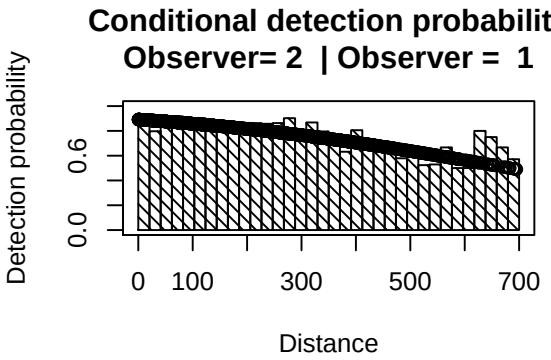
```
## MR total chi-square = 70.856  P = 0.31903 with 66 degrees of freedom
##
##
## Total chi-square = 119.8  P = 0.066573 with 98 degrees of freedom
##
## Distance sampling Cramer-von Mises test (unweighted)
## Test statistic = 0.337147 p-value = 0.106594
```

```
# Detection functions
par(mfrow=c(2,2))
plot(pi.mr.dist.ds.hn, which=c(1,2,3,4))
```




```
plot(pi.mr.dist.ds.hn, which=c(5,6))
```





Which of these models do you prefer?

The FI model had an AIC of 25681 but is not adequate - clearly seen in the goodness-of-fit plot and statistics and the top two detection function plots. Nevertheless, the conditional detection function plots are pretty good (bottom plots) - the model seems adequate for modelling the conditional probability of one observer detecting a group, given the other detected it, but is not adequate for the unconditional probability of detecting a group (which is what we want to estimate). This implies there is something about detected groups that tends to make them more detectable than other groups (i.e. some unmodelled heterogeneity). Treating the conditional detection functions as if they applied to undetected groups will lead to positive bias in estimating detection probability and negative bias in estimating abundance.

The PI model had an AIC of 25325, hence the PI model is preferred on that account and also the goodness-of-fit statistics do not give cause for alarm.

If you have time, try a few models with covariates - warning, this is a large dataset and so fitting models and obtaining model summaries can take a long time with complicated models.

The models given below are those that were used in Borchers *et al.* (2006) and Southwell *et al.* (2007). We do not run them here because they take a long time to fit but are provided for comparison with your models.

```
# Full independence like papers
#mr.model <- "~distance + observer + gscat + fatigue + obsname + exp + ssmi + side +
#fi.paper <- ddf(method="io.fi", mrmodel=~glm(link="logit", formula=as.formula(mr.mo
#                data=crabseal, meta.data=list(width=700))
```

```
# Point independence like papers
#pi.paper <- ddf(method="io.fi", dsmodel=~mcds(key="hn", formula=~side+vis),
#                mrmodel=~glm(link="logit", formula=as.formula(mr.model)),
#                data=crabseal, meta.data=list(width=700))
```

Estimating abundance

Following model criticism and selection, the abundance can be estimated - below we use the simple point independence model. 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. Nevertheless, we illustrate the method to estimate abundance. We require tables of the region, transects and detections - these can easily be created from the data using the `checkdata` function in the `Distance` package (the `:::` is shorthand for accessing a function without loading a package). Using these tables, Horvitz-Thompson-like estimators can be applied to produce estimates of $\hat{N}$. The use of `convert.units=0.001` adjusts the units of perpendicular distance measurement (m) to units of transect effort (km).

```
# 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
pi.abund <- dht(model=pi.mr.dist.ds.hn,
                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
pander::pander(pi.abund$individuals$summary, digits=3,
      caption="Summary information from crabeater seal aerial survey.")
```

Table 5: Summary information from crabeater seal aerial survey. (continued below)

Region	Area	CoveredArea	Effort	n	ER	se.ER	cv.ER
1	1e+06	8594	6139	2053	0.334	0.0325	0.0973

mean.size	se.mean
1.18	0.0127

```
# Pretty tables of estimates of individual abundance
pander::pander(pi.abund$individual$N, digits=3,
      caption="Crabeater seal abundance estimates for study area of arbitrary size.")
```

Table 7: Crabeater seal abundance estimates for study area of arbitrary size.

Label	Estimate	se	cv	lcl	ucl	df
Total	413493	41201	0.0996	339671	503360	129

Crabeater seals with MCDS (optional)

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). These data are available in `crabbieMCDS.csv`.

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.covar <- read.csv("crabbieMCDS.csv")
# Check data imported OK
head(crab.covar, 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.covar$side <- as.factor(crab.covar$side)
crab.covar$vis <- as.factor(crab.covar$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.covar, key="hn", formula=~side, truncation=700)
```

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

Fitting half-normal key function
```

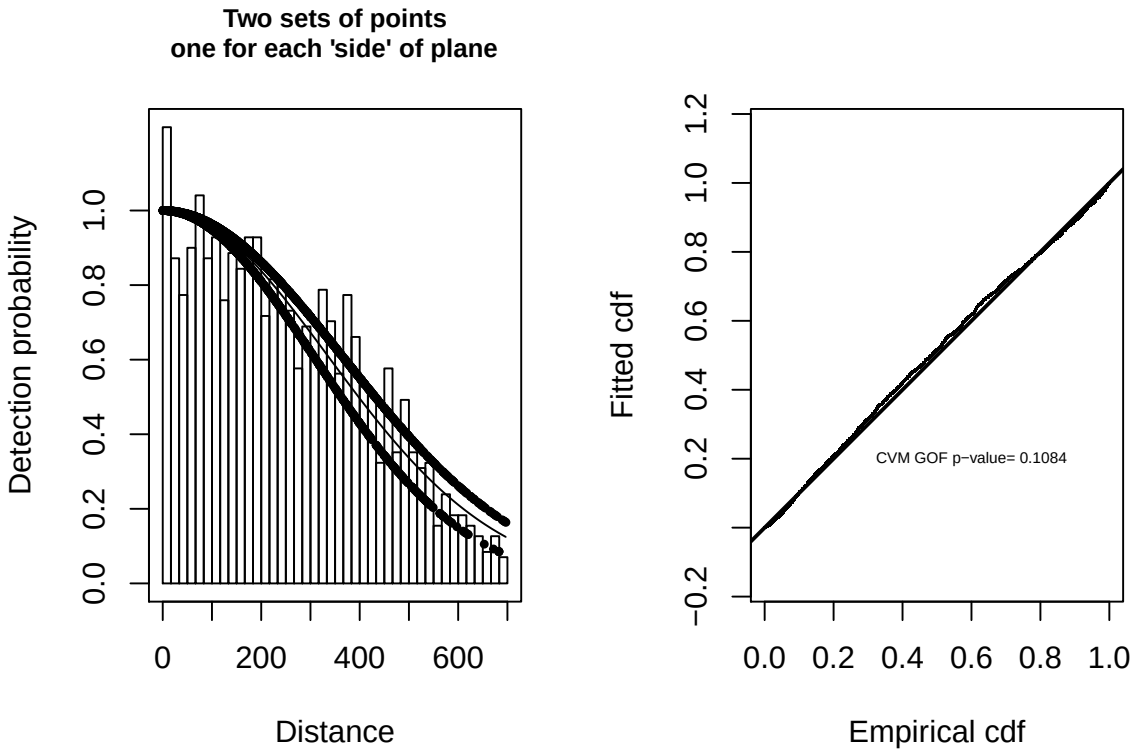


Figure 1: Histogram and fitted half-normal detection function on left. Q-Q plot of detection function and data on right.

```
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 Q-Q 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 and no adjustments
ds.nocov <- ds(crab.covar, key="hn", adjustment=NULL, truncation=700)
```

```
Fitting half-normal key function

Key only model: not constraining for monotonicity.

AIC= 22314.398
```

AIC score for model without covariates is 22314 and AIC score for model with *side* as a covariate is 22305 so the model without *side* as a covariate is very slightly preferred.

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.

In the code below, we fit all possible models and go on examine the selected model, based on AIC.

```
# Create list of formula
mcds.formula <- c("~1", "~side","~vis","~side+vis")
num.mcds.models <- length(mcds.formula)
# Set up table to save results
crab.results <- data.frame(MCDSmodel=mcds.formula, AIC=rep(NA,num.mcds.models))
# Loop through models
for (i in 1:num.mcds.models) {
  ds.model <- ds(crab.covar, key="hn", formula=as.formula(mcds.formula[i]), adjustmen
  truncation=700)
  crab.results$AIC[i] <- summary(ds.model$ddf)$aic
}
```

```
## Fitting half-normal key function

## Key only model: not constraining for monotonicity.

## AIC= 22314.398

## Fitting half-normal key function

## AIC= 22304.742

## Fitting half-normal key function

## AIC= 22311.336

## Fitting half-normal key function

## AIC= 22300.133
```

```
# Calculate delta AIC
crab.results$deltaAIC <- crab.results$AIC - min(crab.results$AIC)
```

```r
# Order by delta AIC
crab.results <- crab.results[order(crab.results$deltaAIC), ]
# Print results in pretty way
pander::pander(crab.results)
```

	MCDSmodel	AIC	deltaAIC
4	~side+vis	22300	0
2	~side	22305	4.61
3	~vis	22311	11.2
1	~1	22314	14.27

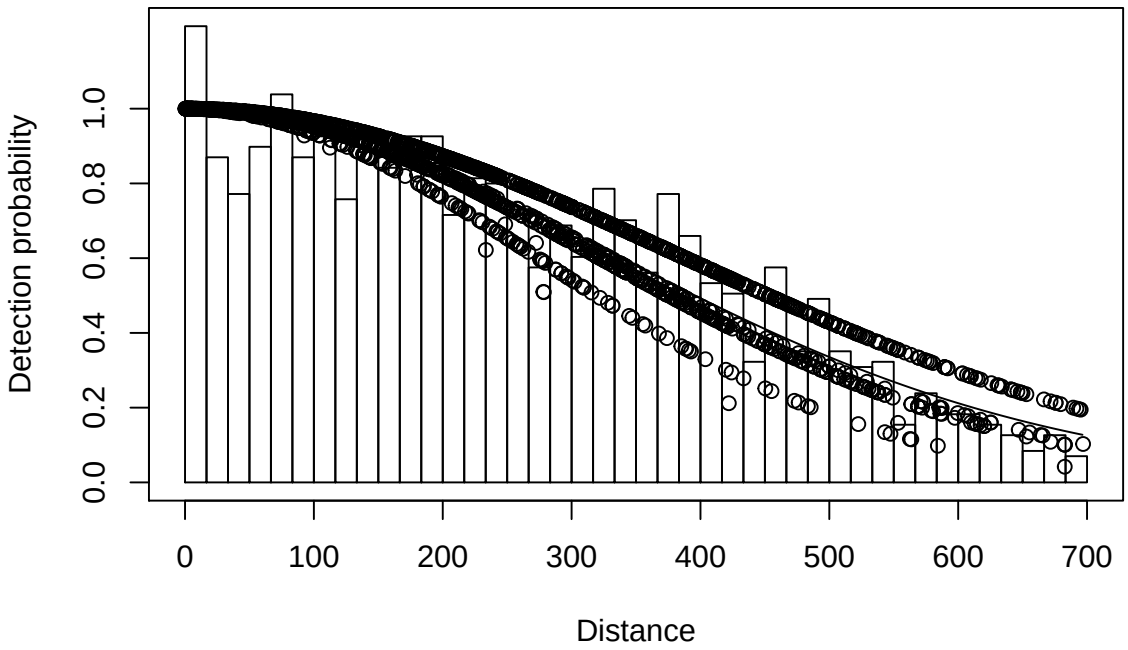
```r
# Fit selected model
crab.mcds <- ds(crab.covar, key="hn", formula=~side+vis, adjustment=NULL,
                truncation=700)
```

```
## Fitting half-normal key function
## AIC= 22300.133
```

```r
summary(crab.mcds$ddf)
```

```
##
## Summary for ds object
## Number of observations :  1740
## Distance range         :  0  -  700
## AIC                    :  22300.13
##
## Detection function:
##  Half-normal key function
##
## Detection function parameters
## Scale coefficient(s):
##               estimate         se
## (Intercept)  5.7889912 0.05438451
## sideR       -0.1866913 0.05394085
## visG         0.1621810 0.05828536
## visP        -0.1235803 0.24749319
##
##                          Estimate          SE         CV
## Average p               0.5812661  0.01243387 0.02139101
## N in covered region 2993.4653237 79.27363082 0.02648223
```

```r
plot(crab.mcds)
```



Rather than fitting an MRDS model, as above, would an MCDS analyses have been adequate?

We can see that, from fitting the MRDS models earlier, the estimate of $p(0)$ for both platforms combined is 0.988 hence, the conventional distance sampling assumption that $p(0) = 1$ is very nearly satisfied. Therefore, in this example the MCDS model seems 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 (2019) mrds: Mark-Recapture Distance Sampling. R package version 2.2.1.

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