

Introduction to distance sampling

Workshop, 21-23 August 2019

Centre for Research into Ecological and Environmental Modelling

Exercise 8. Covariates in the detection function

This exercise consists of four datasets of increasing difficulty. The first problem will show you the rudiments of conducting a multiple covariate distance sampling (MCDS) analysis for a line transect survey. The second problem, MCDS with point transects, is more complicated and (using the functionality available in R) also includes some basic exploratory analysis of the covariates. Problems 3 and 4 are optional but will take you deeper into the heart of understanding multiple covariates.

1 Objectives

The objectives of this exercise are to

1. practice including covariates in the detection function for line and point transects
2. select between candidate models
3. undertake an exploratory analysis of covariates
4. critically appraise the fitted model
5. investigate conversion units.

2 Covariates in line transect detection functions: a whale of a dataset

Rather than relaxing here in the serenity and tranquility of the Scottish coast, imagine instead that you are a research biologist collecting line transect distance sampling data during December on gray whales as they migrated through the Aleutian chain near Unimak Pass en route to their wintering grounds off Baja California (some luckier, more senior researcher, got the job of data collection on their wintering grounds). These data will now be the focus of your attention for this exercise examining the potential utility of covariates in explaining variation in animal detectability.

Detections were of individuals (not groups), and you chose to record not only distance, but also time of observation (at this latitude and at this time of year, the crew was restricted to making observations between 1000 and 1500). However, because of the low sun angles during much of this time, there was some reason to believe that time of day might play a role in whale detectability. Under extreme weather conditions, observer motion sickness can influence the performance of the observers. An additional covariate, ‘motion sickness tablet effective dosage at time of observation (MSTDO)’ was recorded each time a whale was detected.

The data are available for your inspection in the `dsdata` package as follows:

```
library(Distance)
library(dsdata)
# Select data
data(unimak)
# Check data
head(unimak, n=3)
```

Notice the extreme precision with which the perpendicular distances were measured (how do you suppose this could happen on a rolling ship in the Bering Sea?). We can check what units have been used by printing the units associated with these data: a ‘units’ object is available for each dataset - just use `xxx_units` where `xxx` is the name of the dataset as shown below.

```
# What are the units
unimak_units
```

Previously, we have converted metres to kilometres which is a simple conversion. Here we have a mixture of kilometres and miles, but no problem for `convert_units()`.

```
conversion.factor <- convert_units("kilometer", "mile", "square mile")
```

A simple half-normal detection function - this command should be familiar by now.

```
df.hn <- ds(unimak, key="hn", convert.units=conversion.factor)
```

To add a covariate in the detection function, we need to specify the covariate in the `formula` argument. In the command below, `MSTDO` is included - note the ‘~’ in the formula.

```
# Include MSTDO in detection function
df.hn.mstdo <- ds(unimak, key="hn", formula=~MSTDO,
                  convert.units=conversion.factor)
```

The AIC values for these models can be compared using the `AIC` command:

```
AIC(df.hn, df.hn.mstdo)
```

The other possible combinations of covariates are

- Hour
- MSTDO + Hour.

Try a few different models, not forgetting about truncation and decide on a final model.

Don’t forget that you can look at the detection function using the `plot` command:

```
plot(df.hn.mstdo)
```

Why don’t the points lie on the solid line?

If you have been successful in performing the analysis of this dataset (which can now be revealed to have been simulated), you can continue to sharpen your skills in using covariates in your analysis of distance sampling data by exploring the following problems.

3 Covariates in point transect detection functions: Amakihi

In this problem, we illustrate fitting multiple covariate distance sampling (MCDS) models to point transect data using a bird survey from Hawaii: data on an abundant species, the Hawaii amakihi (*Hemignathus virens*) is used. This practical is based on the case study in Buckland et al. (2015) which duplicates the analysis in Marques et al. (2007). This set of data is included in `Distance` as one of the Sample Projects and so can be accessed easily:

```
data(amakihi)
# See what columns it contains
head(amakihi, n=3)
```

These data include:

- `Study.Area` - name of the study area
- `Region.Label` - survey dates which are used as ‘strata’
- `Sample.Label` - point transect identifier
- `Effort` - survey effort (1 for all points because each point was visited once)
- `distance` - radial distance of detection from observer (meters)
- `OBs` - initials of the observer
- `MAS` - minutes after sunrise
- `HAS` - hour after sunrise

Note that the `Area` column is always zero, hence, detection functions can be fitted to the data, but bird abundance cannot be estimated. The covariates to be considered for possible inclusion into the detection function are `OBs`, `MAS` and `HAS`.

3.0.1 Exploratory data analysis

It is important to gain an understanding of the data prior to fitting detection functions (Buckland *et al.* 2015). With this in mind, preliminary analysis of distance sampling data involves:

- assessing the shape of the collected data,
- considering the level of truncation of distances, and
- exploring patterns in potential covariates.

We begin by assessing the distribution of distances to decide on a truncation distance.

```
hist(amakihi$distance)
```

To see if there are differences in the distribution of distances recorded by the different observers and in each hour after sunrise, boxplots can be used. Note how the `~` symbol is used to define the discrete groupings (i.e. observer and hour).

```
# Boxplots by obs
boxplot(amakihi$distance~amakihi$OBs, xlab="Observer", ylab="Distance (m)")
# Boxplots by hour after sunrise
boxplot(amakihi$distance~amakihi$HAS, xlab="Hour", ylab="Distance (m)")
```

The components of the boxplot are:

- the thick black line indicates the median
- the lower limit of the box is the first quartile (25th percentile) and the upper limit is the third quartile (75th percentile)
- the height of the box is the interquartile range (75th - 25th quartiles)
- the whiskers extend to the most extreme points which are no more than 1.5 times the interquartile range.
- dots indicate 'outliers' if there are any, i.e. points beyond the range of the whiskers.

For minutes after sunrise (a continuous variable), we create a scatterplot of MAS (on the x -axis) against distances (on the y -axis). The plotting symbol (or character) is selected with the argument `pch`:

```
# Plot of MAS vs distance (using dots)
plot(x=amakihi$MAS, y=amakihi$distance, xlab="Minutes after sunrise",
      ylab="Distance (m)", pch=20)
```

You may also want to think about potential collinearity (linear relationship) between the covariates - if collinear variables are included in the detection function, they will be explaining some of the same variation in the distances and this will reduce their importance as a potential covariate. How might you investigate the relationship between HAS and MAS?

From these plots can you tell if any of the covariates will be useful in explaining the distribution of distances?

3.1 Adjusting the raw covariates

We would like to treat OBS and HAS as factor variables as in the original analysis; OBS is, by default, treated as a factor variable because it consists of characters rather than numbers. HAS, on the other hand, consists of numbers and so by default would be treated as a continuous variable (i.e. non-factor). That is fine if we want the effect of HAS to be monotonic (i.e. detectability either increases or decreases as a function of HAS). If we want HAS to have a non-linear effect on detectability, then we need to indicate to R to treat it as a factor as shown below.

```
# Convert HAS to a factor
amakihi$HAS <- factor(amakihi$HAS)
```

The next adjustment is to change the *reference* level of the *observer* and *hour* factor covariates - the only reason to do this is to get the estimated parameters in the detection function to match the parameters estimated in Marques et al (2007). By default R uses the first factor level but by using the `relevel` function, this can be changed:

```
# Set the reference level
amakihi$OBS <- relevel(amakihi$OBS, ref="TKP")
amakihi$HAS <- relevel(amakihi$HAS, ref="5")
```

One final adjustment, and more subtle, is a transformation of the continuous covariate MAS. We are considering three possible covariates in our detection function: OBS, HAS and MAS. The first two variables, OBS and HAS, are both factor variables, and so, essentially, we can think of them as taking on values between 1 and 3 in the case of OBS, and 1 to 6 in the case of HAS. However, MAS can take on values from -18 (detections before sunrise) to >300

and the disparity in scales of measure between MAS and the other candidate covariates can lead to difficulties in the performance of the optimizer fitting the detection functions in R. The solution to the difficulty is to scale MAS such that it is on a scale (approx. 1 to 5) comparable with the other covariates.

Dividing all the MAS measurements by the standard deviation (function `sd`) of those measurements accomplishes the desired compaction in the range of the MAS covariate without changing the shape of the distribution of MAS values. The `na.rm=TRUE` argument ensures that any missing values are ignored.

```
# Rescale MAS by dividing by standard deviation
amakihi$MAS <- amakihi$MAS/sd(amakihi$MAS, na.rm=TRUE)
```

Check what this command has done by looking at a summary of the adjusted MAS:

```
summary(amakihi$MAS)
```

3.1.1 Candidate models

With three potential covariates, there are 8 possible models for the detection function:

- No covariates
- OBs
- HAS
- MAS
- OBs + HAS
- OBs + MAS
- HAS + MAS
- OBs + HAS + MAS

Even without considering covariates there are also several possible key function/adjustment term combinations available: if all key function/covariate combinations are considered the number of potential models is large. Note that covariates are not allowed if a uniform key function is chosen and if covariate terms are included, adjustment terms are not allowed. Even with these restrictions, it is not best practice to take a scatter gun approach to detection function model fitting. Buckland et al. (2015) considered 13 combinations of key function/covariates. Here, we look at a subset of these.

Fit a hazard rate model with no covariates or adjustment terms and make a note of the AIC. Note, that 10% of the largest distances are truncated - you may have decided on a different truncation distance.

```
conversion.factor <- convert_units("meter", NULL, "hectare")
amak.hr <- ds(amakihi, transect="point", key="hr", truncation="10%",
             adjustment=NULL, convert.units = conversion.factor)
```

Make a note of the AIC for this model.

Now fit a hazard rate model with OBs as a covariate in the detection function and make a note of the AIC. Has the AIC reduced by including a covariate?

```
conversion.factor <- convert_units("meter", NULL, "hectare")
amak.hr.obs <- ds(amakihi, transect="point", key="hr", formula=~OBs,
```

```
truncation="10%", convert.units = conversion.factor)
```

Fit a hazard rate model with OBs and HAS in the detection function:

```
amak.hr.obs.has <- ds(amakihi, transect="point", key="hr", formula=~OBs+HAS,  
  truncation="10%", convert.units = conversion.factor)
```

Try fitting other possible formula and decide which model is best in terms of AIC. To quickly compare AIC values from different models, use the AIC command as follows (note only models with the same truncation distance can be compared):

```
# AIC values
```

```
AIC(amak.hr, amak.hr.obs, amak.hr.obs.has)
```

Another useful function is `summarize_ds_models` - this has the advantage of ordering the models by AIC (smallest to largest).

```
# Compare models
```

```
summarize_ds_models(amak.hr, amak.hr.obs, amak.hr.obs.has)
```

Once you have decided on a model, plot your selected detection function.

4 More MCDS with line transects: ETP dolphins (optional)

In this problem we have a sample of Eastern Tropical Pacific (ETP) spotted dolphin sightings data, collected by observers placed on board tuna vessels (the data were kindly made available to us by the Inter-American Tropical Tuna Commission - IATTC). In the ETP, schools of yellow fin tuna commonly occur with schools (or groups) of dolphins, and so vessels fishing for tuna often search for dolphins in the hopes of also locating tuna. For each dolphin school detected by the tuna vessels, the observer recorded the species, sighting angle and distance (later converted to perpendicular distance and truncated at 5 nautical miles), school size and a number of covariates associated with each detected school. Many of these covariates potentially affect the detection function, as they reflect how the search was being carried out.

A variety of search methods were used to find the dolphins, and for these data were (the numbers in brackets are the codes used to record the data):

- 20x binoculars from the crow's nest (0)
- 20x binoculars from another location on the vessel (2),
- a helicopter, (3)
- 'bird radar', high power radars which are able to detect seabirds flying above the dolphin schools (5).

Some of these methods may have a wider range of search than the others, and so it is possible that the effective strip width varies according to the method being used.

For each detection the initial cue type was recorded. This included:

- birds flying above the school (1),
- splashes on the water (2),

- floating objects such as logs (4),
- some other unspecified cue (3).

Another covariate that potentially affected the detection function was sea state, as measured by Beaufort. In rougher conditions (i.e. higher Beaufort levels), visibility and/or detectability may be reduced. For this example, Beaufort levels were grouped into two categories, the first including Beaufort values ranging from 0 to 2 (coded as 1) and the second containing values from 3 to 5 (coded as 2).

The sample data encompasses sightings made over a three month period: June, July and August (months 6, 7 and 8, respectively).

4.1 Analysis

The data are available in the `dsdata` package:

```
# Load data
data(ETP_Dolphin)
# Check data
head(ETP_Dolphin, n=3)
```

Start by running a set of conventional distance analyses. Are there any problems in the data and if so how might you mitigate them? (Hint - try dividing the histogram of distances into a large number of intervals.)

As there are a number of potential covariates to be used in this example (i.e. search method, cue, Beaufort class and month), try fitting models with different covariates and combinations of the covariates. All of the covariates in this example are factor covariates except group size and because they have numeric codes, use the `factor` function to let R know to treat them as factors.

Note that both distances and transect lengths were recorded in nautical miles and area in nautical miles squared and so the argument `convert_units` does not need to be specified.

Keep in mind that this is a large dataset (> 1000 observations), and hence estimation may take a while. You will likely end up with quite a few models as there are several potential covariates and no ‘right’ answers. Discuss your choice of final model (or models) with your neighbours - did you make the same choices?

5 More MCDS with point transects: Savannah sparrow (optional)

Point transect surveys were conducted in Arapaho National Wildlife Refuge, Colorado, USA, in 1980 and 1981 and this exercise concerns the data collected on Savannah sparrows. The study area was divided into smaller regions (called ‘pastures’). The data are available in the `dsdata` package as follows:

```
library(dsdata)
data(Savannah_sparrow_1980)
data(Savannah_sparrow_1981)
```

Distances were recorded in metres and area in hectares.

```
conversion.factor <- convert_units("meter", NULL, "hectare")
```

Given that this data object has a long name, these objects can be renamed if you wish: for example,

```
# Rename data  
sav1980 <- Savannah_sparrow_1980
```

For each data set,

1. consider an appropriate truncation distance,
2. fit a detection function with out any covariates
3. include 'pasture' (**Region.Label**) as a covariate in the detection function
4. use AIC to select a model, and
5. estimate density (in birds per hectare) for your selected model.

What would be an alternative to including pasture as a covariate in the detection function to analyse these data?

6 References

Buckland ST, Rexstad ER, Marques TA and Oedekoven CS (2015) Distance sampling: methods and applications. Springer

Knopf FL, Sedgwick JA, and Cannon RW. (1988) Guild structure of a riparian avifauna relative to seasonal cattle grazing. *Journal of Wildlife Management* 52:280–290.

Marques TA, Thomas L, Fancy SG and Buckland ST (2007) Improving estimates of bird density using multiple covariate distance sampling. *The Auk* 124:1229-1243

Solution 8. Covariates in the detection function

7 Covariates in line transect detection functions: a whale of a dataset

```
library(Distance)
library(dsdata)
# Select data
data(unimak)
conversion.factor <- convert_units("kilometer", "mile", "square mile")
```

An example of the sort of analysis you might have performed is given below. I first tried simple half-normal and hazard rate models without covariates, and found that the half-normal model had a lower AIC. I then tried the MSTDO covariate and hour covariates separately (as non-factor covariates). The analysis with MSTDO had a much lower AIC, but the analysis with hour actually had a higher AIC than the analysis without covariates. I tried an analysis with both MSTDO and hour, but this had a higher AIC than MSTDO alone. I concluded that the MSTDO covariate was important and the hour covariate was not.

```
# Fit basic key functions only
df.hn <- ds(unimak, key="hn", convert.units=conversion.factor)
df.hr <- ds(unimak, key="hr", convert.units=conversion.factor)

## Fit covariates in detection functions
# MSTDO
df.hn.mstdo <- ds(unimak, formula=~MSTDO, convert.units=conversion.factor)
# Hour
df.hn.hour <- ds(unimak, formula=~Hour, convert.units=conversion.factor)
# MSTDO + hour
df.hn.mstdo.hour <- ds(unimak, formula=~MSTDO+Hour,
                      convert.units=conversion.factor)
# Compare model fits by AIC
AIC(df.hn, df.hr, df.hn.mstdo, df.hn.hour, df.hn.mstdo.hour)
```

```
##           df      AIC
## df.hn      1 123.2821
## df.hr      2 125.3153
## df.hn.mstdo 2 111.2341
## df.hn.hour  2 154.8669
## df.hn.mstdo.hour 3 156.8669
```

Although these data did not appear to need any truncation, I briefly confirmed that the same results were obtained with 10% truncation.

```

# Try truncating 10% of longest perp. distances
df.hn.trunc <- ds(unimak, key="hn", convert.units=conversion.factor,
                  truncation="10%")
df.hr.trunc <- ds(unimak, key="hr", convert.units=conversion.factor,
                  truncation="10%")

df.hn.mstdo.trunc <- ds(unimak, key="hn", formula=~MSTD0,
                       convert.units=conversion.factor, truncation="10%")
df.hn.hour.trunc <- ds(unimak, key="hn", formula=~Hour,
                       convert.units=conversion.factor, truncation="10%")
df.hn.mstdo.hour.trunc <- ds(unimak, key="hn", formula=~MSTD0+Hour,
                             convert.units=conversion.factor, truncation="10%")

AIC(df.hn.trunc, df.hr.trunc, df.hn.mstdo.trunc, df.hn.hour.trunc,
    df.hn.mstdo.hour.trunc)

```

```

##              df      AIC
## df.hn.trunc      1 80.99018
## df.hr.trunc       2 82.73318
## df.hn.mstdo.trunc  2 76.27399
## df.hn.hour.trunc   2 91.36957
## df.hn.mstdo.hour.trunc 3 93.36957

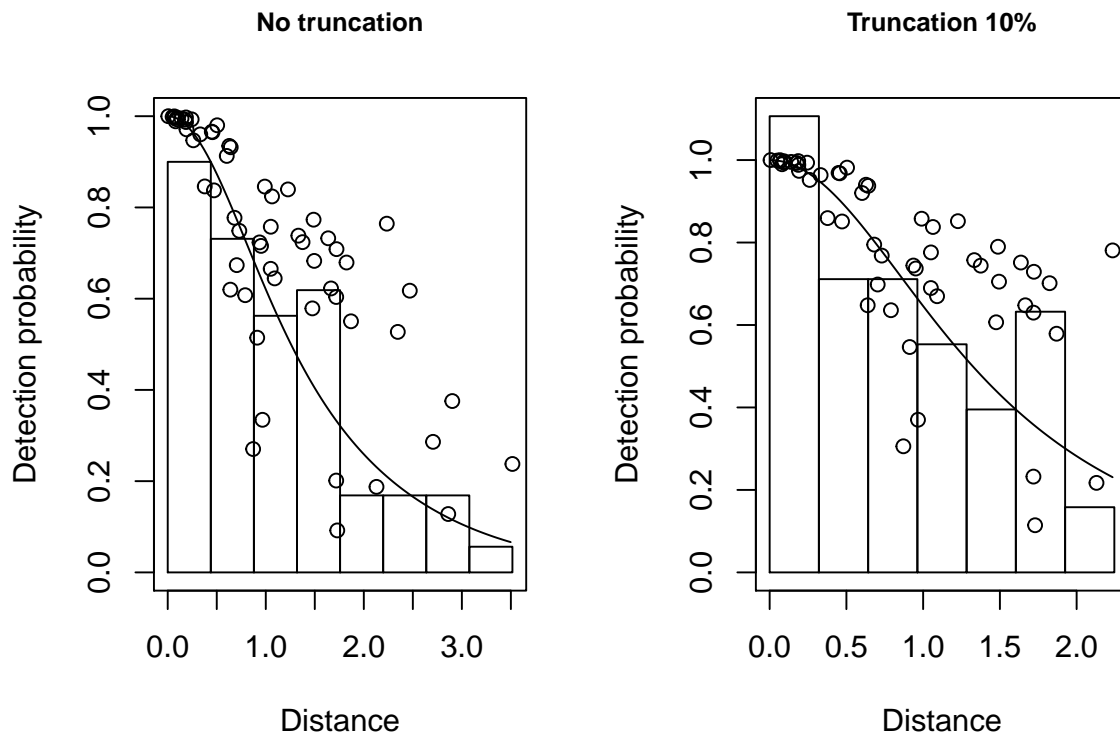
```

Finally, I looked at the fitted detection function plots from the selected models.

```

# Plot detection functions
par(mfrow=c(1,2))
plot(df.hn.mstdo, main="No truncation")
plot(df.hn.mstdo.trunc, main="Truncation 10%")

```



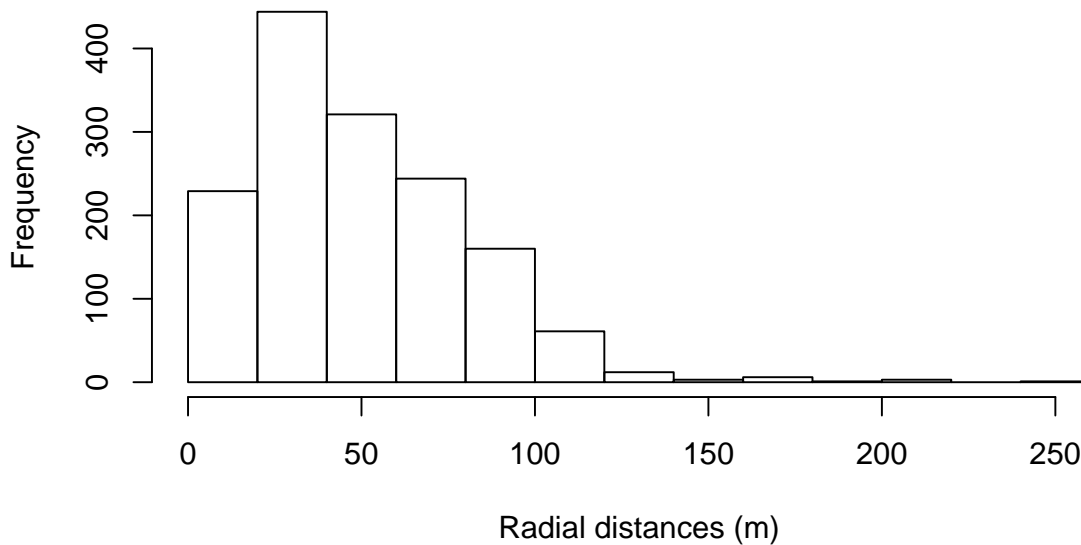
The points indicate the detection probability for individual detections at the recorded perpendicular distance and covariate value (MSTDO in this case).

8 Covariates in point transect detection functions: Amakihi

```
# Load data
data(amakihi)

hist(amakihi$distance, xlab="Radial distances (m)",
     main="Amakihi point transect data.")
```

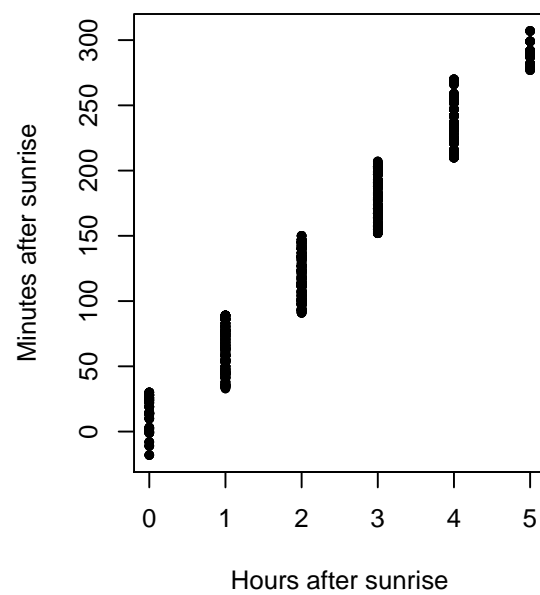
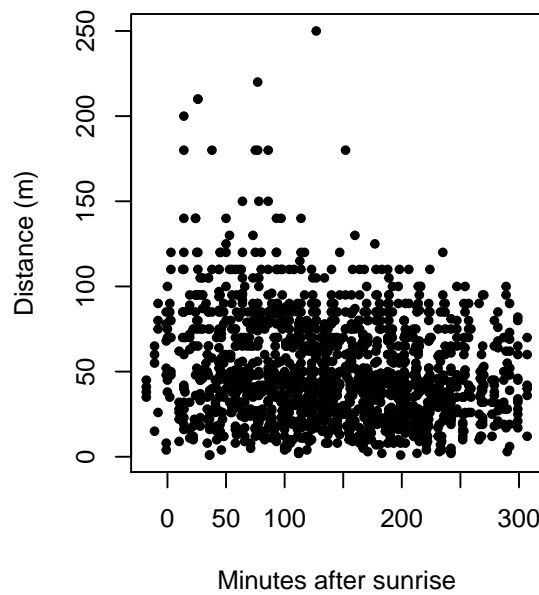
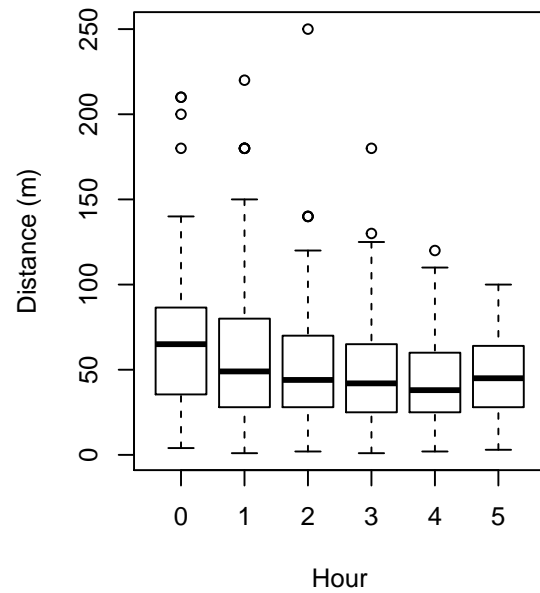
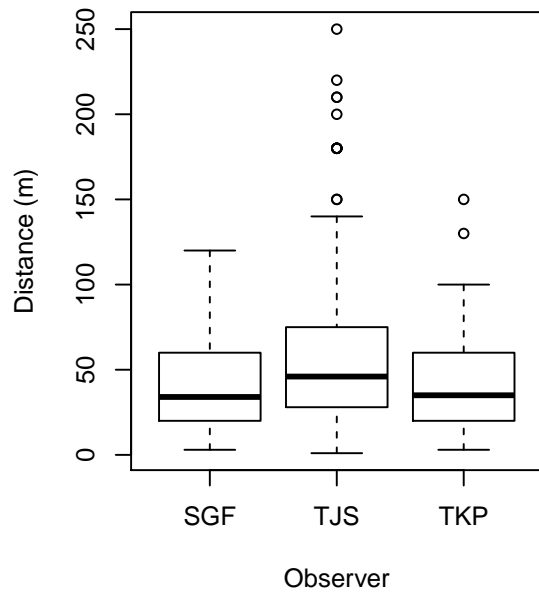
Amakihi point transect data.



A truncation distance of 82.5 m was chosen by Marques et al. (2007).

Plots of the covariates were generated. Not surprisingly, MAS and HAS are correlated and so we need to be cautious of including them in the same model.

```
# Plots of covariates
par(mfrow=c(2,2))
# Boxplots by obs
boxplot(amakihi$distance~amakihi$OBs, xlab="Observer", ylab="Distance (m)")
# Boxplots by hour after sunrise
boxplot(amakihi$distance~amakihi$HAS, xlab="Hour", ylab="Distance (m)")
# Plot of MAS vs distance (using dots)
plot(x=amakihi$MAS, y=amakihi$distance, xlab="Minutes after sunrise",
     ylab="Distance (m)", pch=20)
# Plot of HAS vs MAS (using dots)
plot(x=amakihi$HAS, y=amakihi$MAS, xlab="Hours after sunrise",
     ylab="Minutes after sunrise", pch=20)
```



```
# Adjusting the raw data
# Convert HAS to a factor
amakihi$HAS <- factor(amakihi$HAS)
# Set the reference level
amakihi$OBs <- relevel(amakihi$OBs, ref="TKP")
amakihi$HAS <- relevel(amakihi$HAS, ref="5")
```

The re-scaling of MAS has converted it from a variable with values between -18 and 307 to values between -0.23 and 4.04.

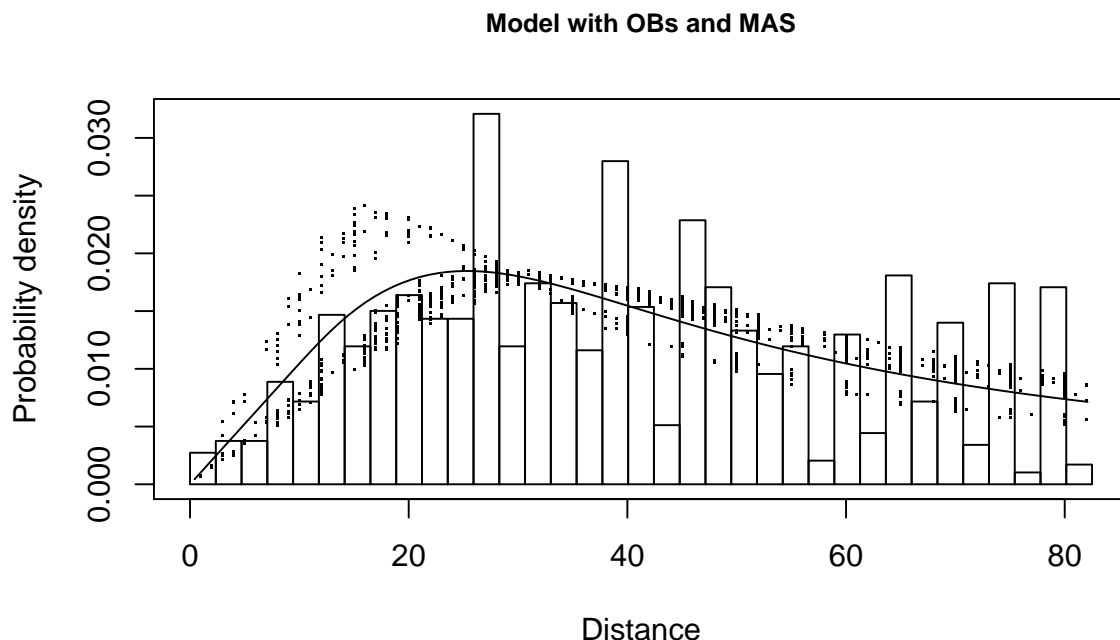
```
# Rescale MAS by dividing by standard deviation
amakihi$MAS <- amakihi$MAS/sd(amakihi$MAS, na.rm=TRUE)
summary(amakihi$MAS)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.     NA's
## -0.2365  1.0248  1.7999  1.8508  2.6145  4.0335         2
```

The model selected by Marques et al. (2007) used a hazard rate key function and included observer and minutes after sunrise - this model is fitted below. The PDF plot is shown.

```
# Fit model selected by Marques et al (2007)
amak.hr.obs.mas <- ds(amakihi, transect="point", key="hr", formula=~OBs+MAS,
                      truncation=82.5)

# Plot selected model
plot(amak.hr.obs.mas, main="Model with OBs and MAS", pch=".", pdf=TRUE)
```



To see more sophisticated examples of plotting the detection function for the selected model, see the code accompanying Buckland et al. (2015) [Hawaiian Amakihi case study](#).

9 More MCDS with line transects: ETP dolphins (optional)

We have not provided a comprehensive analysis of these data but have highlighted a few general feature of these data.

```
# Read in the data
data(ETP_Dolphin)
# Look at data
```

```
head(ETP_Dolphin, n=3)
```

```
##   Region.Label Area Sample.Label Effort object distance LnCluster Month
## 1   Default    0         1      1      1      2.25  6.792344      6
## 2   Default    0         1      1      2      3.15  7.080868      6
## 3   Default    0         1      1      3      3.00  5.181784      6
##   Beauf.class Cue.type Search.method size Study.Area
## 1           2       3             0  891   Dolphin
## 2           2       1             2 1189   Dolphin
## 3           2       1             2  178   Dolphin
```

```
# Check conversion units
```

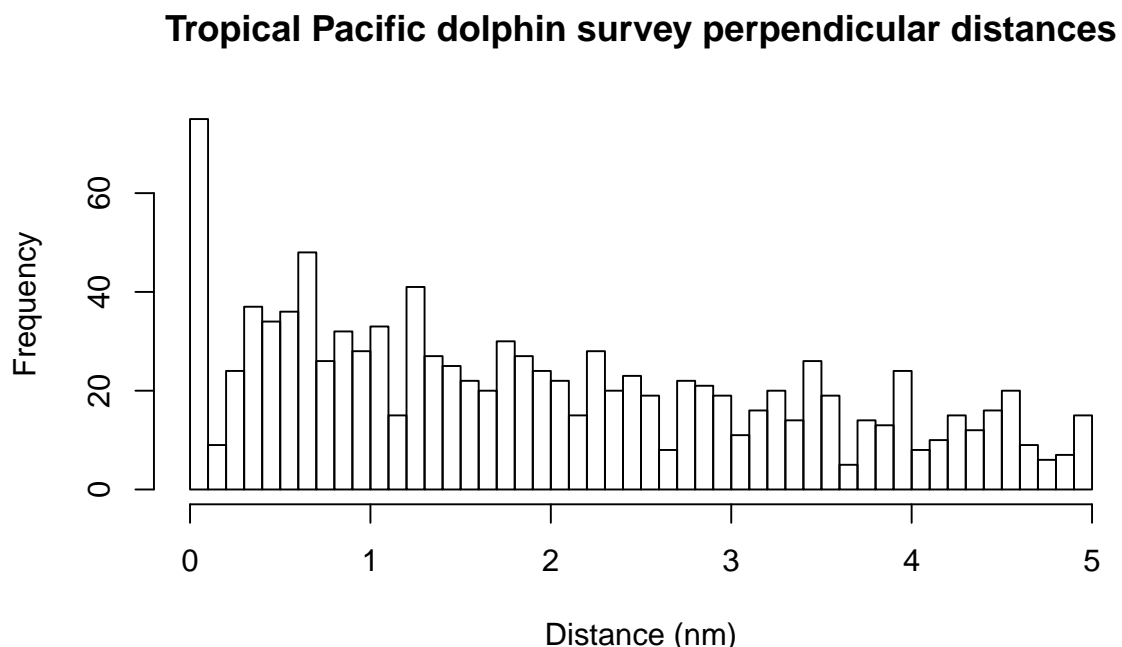
```
ETP_Dolphin_units
```

```
##   Variable          Units Conversion
## 1   Effort      nautical mile      1852
## 2 distance      nautical mile      1852
## 3    Area square nautical mile  3429904
```

To obtain an overall impression of the data, it is useful to fit a histogram with many intervals.

```
# Histogram of distances with lots of intervals
```

```
hist(ETP_Dolphin$distance, nclass=50, xlab="Distance (nm)",
     main="Tropical Pacific dolphin survey perpendicular distances")
```

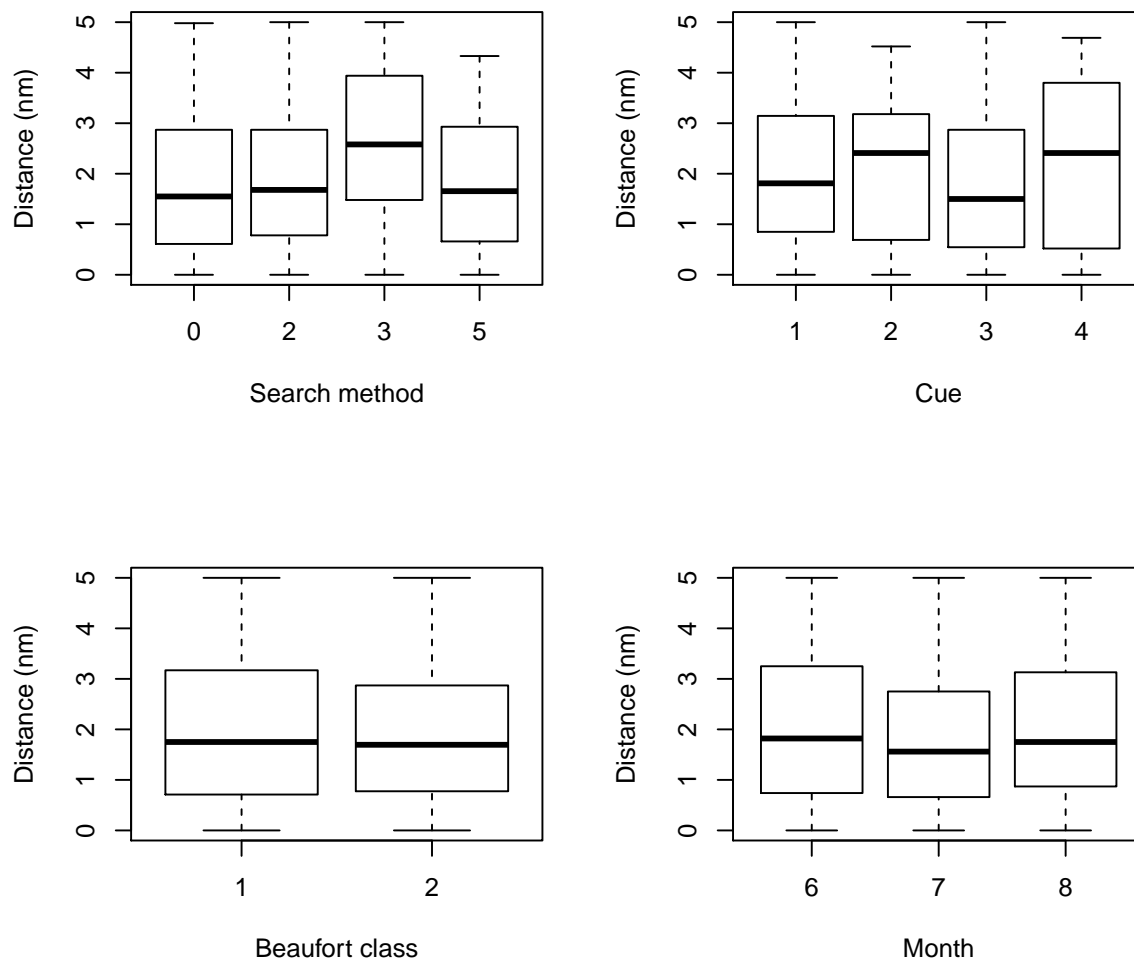


The spikes in the histogram suggest that distances have been rounded to zero and possibly other values. To mitigate these problems, the distances could be binned although we do not do so in the analysis below. The distances have already been truncated at 5 nm and so we will not truncate distances further.

```

# Boxplots of distances against factor covariates
par(mfrow=c(2,2))
# Search method
boxplot(ETP_Dolphin$distance~ETP_Dolphin$Search.method, xlab="Search method",
        ylab="Distance (nm)")
# Cue
boxplot(ETP_Dolphin$distance~ETP_Dolphin$Cue.type, xlab="Cue", ylab="Distance (nm)")
# Beaufort
boxplot(ETP_Dolphin$distance~ETP_Dolphin$Beauf.class, xlab="Beaufort class",
        ylab="Distance (nm)")
# Month
boxplot(ETP_Dolphin$distance~ETP_Dolphin$Month, xlab="Month", ylab="Distance (nm)")

```



To decide whether to fit a half normal or a hazard rate key function, each of these is tried in turn.

```

# Fit basic detection functions
# Half normal
etp.hn <- ds(ETP_Dolphin, key="hn", adjustment=NULL)

```



```
# Hazard rate
etp.hr <- ds(ETP_Dolphin, key="hr", adjustment=NULL)
# Compare these fits
AIC(etp.hn, etp.hr)
```

```
##          df          AIC
## etp.hn    1 3377.489
## etp.hr    2 3365.915
```

The AIC values suggest that hazard rate key function is preferable to the half normal and so this will be used as the key function in the MCDS models. Each covariate is added in turn.

```
# Specify factors as below or include in formula
ETP_Dolphin$Search.method <- factor(ETP_Dolphin$Search.method)
# Add covariates to hazard rate detection function
# Search method (factor)
etp.hr.search <- ds(ETP_Dolphin, key="hr", formula=~Search.method)
# Cue type (factor)
etp.hr.cue <- ds(ETP_Dolphin, key="hr", formula=~factor(Cue.type))
# Beaufort class (factor)
etp.hr.bf <- ds(ETP_Dolphin, key="hr", formula=~factor(Beauf.class))
# Month (factor)
etp.hr.month <- ds(ETP_Dolphin, key="hr", formula=~factor(Month))

# Compare models (using pretty printing)
pander::pander(summarize_ds_models(etp.hr, etp.hr.search, etp.hr.cue, etp.hr.bf,
                                   etp.hr.month, output = "plain"),
               caption="ETP dolphin model selection.")
```

Table 1: ETP dolphin model selection. (continued below)

	Model	Key function	Formula	C-vM p -value
2	etp.hr.search	Hazard-rate	~Search.method	0.3881
3	etp.hr.cue	Hazard-rate	~factor(Cue.type)	0.4494
5	etp.hr.month	Hazard-rate	~factor(Month)	0.4942
1	etp.hr	Hazard-rate	~1	0.4608
4	etp.hr.bf	Hazard-rate	~factor(Beauf.class)	0.4736

	Average detectability	se(Average detectability)	Delta AIC
2	0.5838	0.03602	0
3	0.5875	0.03913	23.31
5	0.5933	0.03893	24.26
1	0.6006	0.03837	26.14
4	0.6015	0.0385	27.99

Based on the AIC, it seems as though the model including search method was preferable and we could continue the model selection process by looking at models with two covariates. However, before going on it is worth looking at the search method model in more detail. If we look at the detection function parameters for this model:

```
# Look at detection function part of the model object
```

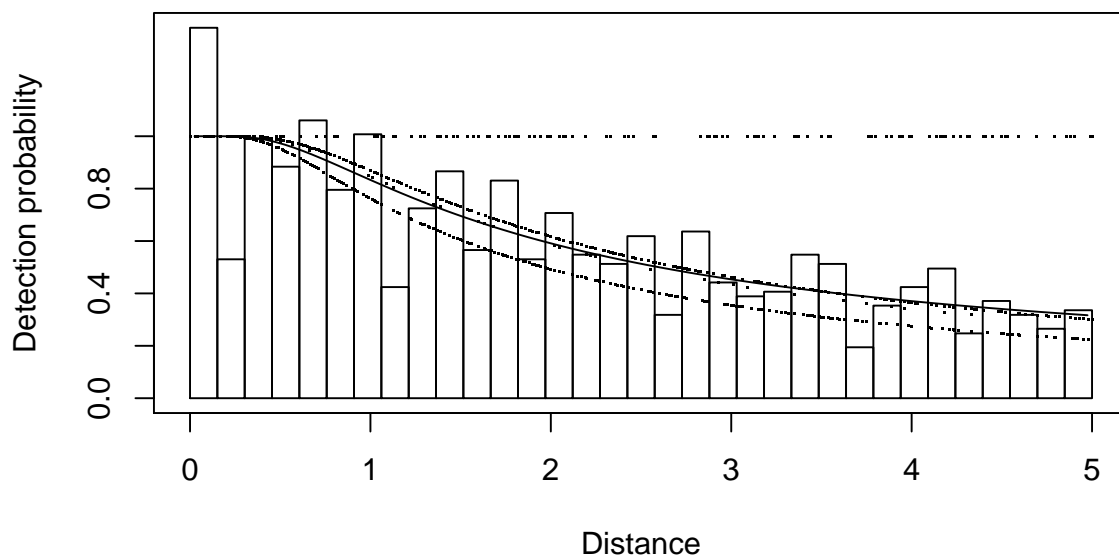
```
etp.hr.search$ddf
```

```
##
## Distance sampling analysis object
##
## Summary for ds object
## Number of observations : 1090
## Distance range       : 0 - 5
## AIC                  : 3339.775
##
## Detection function:
## Hazard-rate key function
##
## Detection function parameters
## Scale coefficient(s):
##           estimate      se
## (Intercept) 0.3355287 0.2072042
## Search.method2 0.3208253 0.2112955
## Search.method3 3.1275278 78.6848803
## Search.method5 0.2430549 0.5378446
##
## Shape coefficient(s):
##           estimate      se
## (Intercept) 0.07828303 0.1387478
##
##           Estimate      SE      CV
## Average p      0.5837971 0.03601616 0.06169295
## N in covered region 1867.0871295 121.16047848 0.06489278
```

we see that the estimated scale coefficient for search method 3 is substantially larger than the estimated scale coefficients for other methods. The effect this has on the detection function is clearly seen in the detection function plot.

```
# Plot search method detection function
```

```
plot(etp.hr.search, pch=".")
```



Search method 3 indicated that the detection was from a helicopter and this detection function suggests that all dolphin schools out to 5 nm were being detected and so detection does not decrease as distance increases. One assumption of MCDS is that the perpendicular distance distributions of the covariate factor levels have the same shape and so it may be worth refitting the models but excluding the observations made by the helicopter.

10 More MCDS with point transects: Savannah sparrow (optional)

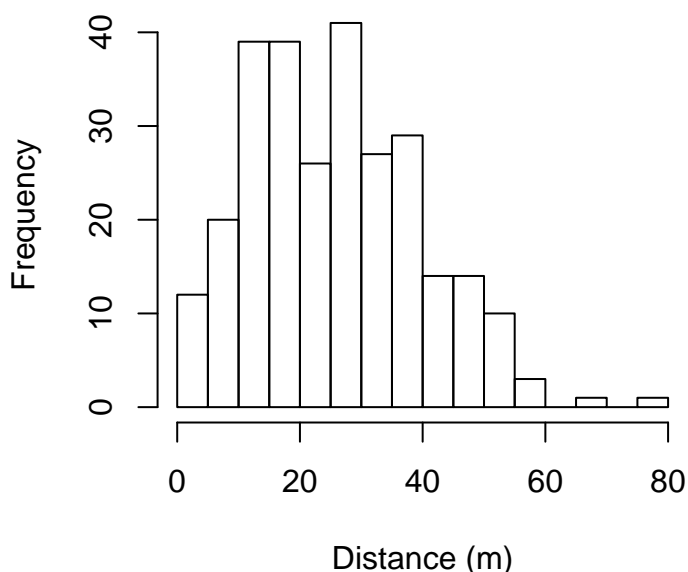
The question here was whether to include pasture as a covariate in the detection function.

```
# Load libraries
library(dsdata)
# Savannah sparrow 1980
data(Savannah_sparrow_1980)
sav1980 <- Savannah_sparrow_1980
# Check data
head(sav1980, n=3)
```

```
##   Region.Label Area Sample.Label Effort object distance Study.Area
## 1   PASTURE 1     1     POINT    1      1    NA      NA   SASP 1980
## 2   PASTURE 1     1     POINT    2      1    NA      NA   SASP 1980
## 3   PASTURE 1     1     POINT    3      1    NA      NA   SASP 1980
```

```
# Histogram of distances with lots of bins
hist(sav1980$distance, nclass=20, xlab="Distance (m)",
     main="Savannah sparrow radial distances 1980")
```

Savannah sparrow radial distances 1980



```
conversion.factor <- convert_units("meter", NULL, "hectare")
```

A truncation distance of 55m was chosen. The half normal and hazard rate functions were tried in turn, allowing AIC selection of cosine adjustment terms, then pasture was included as a covariate in the detection function.

```
# Fit different detection functions, truncation at 55m
# Half-normal
sav1980.hn <- ds(data=sav1980, key="hn", adjustment="cos", truncation=55,
                 transect="point", convert.units=conversion.factor)

# Hazard
sav1980.hr <- ds(data=sav1980, key="hr", adjustment="cos", truncation=55,
                 transect="point", convert.units=conversion.factor)

# Half-normal with pasture covariate
sav1980.hn.region <- ds(data=sav1980, key="hn", truncation=55,
                       transect="point", convert.units=conversion.factor,
                       formula=~Region.Label)

# Hazard with pasture covariate
sav1980.hr.region <- ds(data=sav1980, key="hr", truncation=55,
                       transect="point", convert.units=conversion.factor,
                       formula=~Region.Label)

# Select between these models
AIC(sav1980.hn, sav1980.hr, sav1980.hn.region, sav1980.hr.region)
```

```
##           df      AIC
## sav1980.hn      2 2126.228
## sav1980.hr      3 2127.883
```

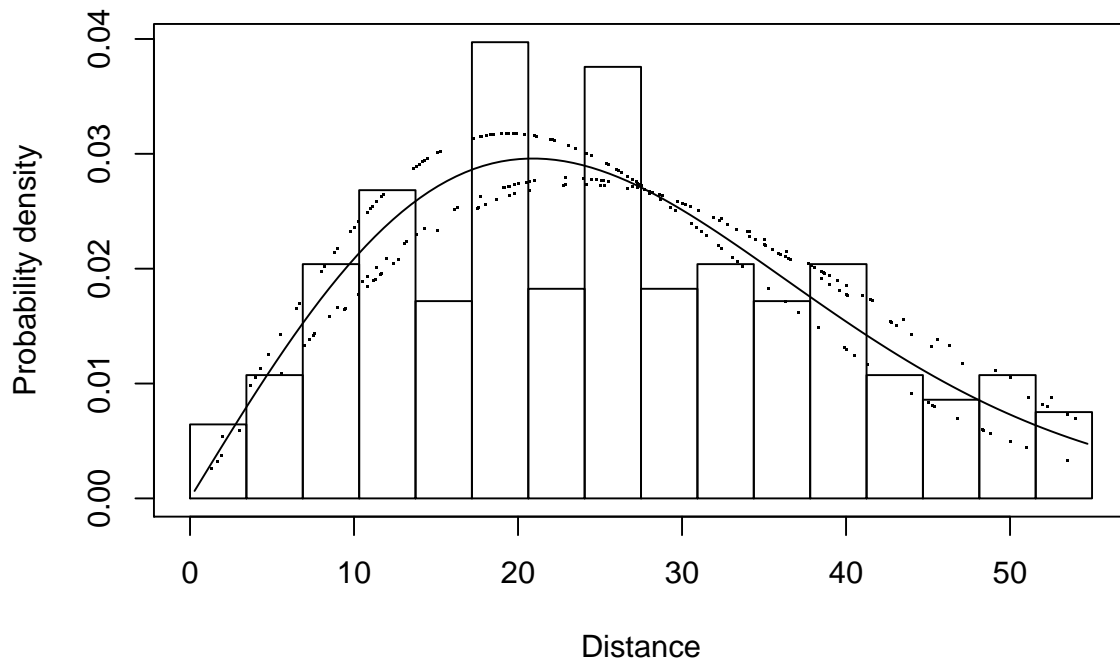


Figure 1: Note different PDF shapes caused by the pasture covariate.

```
## sav1980.hn.region 3 2125.033
## sav1980.hr.region 4 2130.447
```

The half normal model with pasture as a covariate had a marginally smaller AIC than the half normal model without pasture. The plots and estimates are shown below.

```
# Plot results of selected model
plot(sav1980.hn.region, pch=".", pdf=TRUE)
```

```
# Summarise results for selected model
summary(sav1980.hn.region)
```

```
##
## Summary for distance analysis
## Number of observations : 271
## Distance range       : 0 - 55
##
## Model : Half-normal key function
## AIC   : 2125.033
##
## Detection function parameters
## Scale coefficient(s):
##               estimate      se
## (Intercept)    3.17237910 0.09850509
## Region.LabelPASTURE 2 -0.20459998 0.11106008
```

```
## Region.LabelPASTURE 3 -0.03287325 0.12340874
##
##              Estimate          SE          CV
## Average p          0.2896556  0.01973458 0.06813119
## N in covered region 935.5938568 79.88888250 0.08538842
##
## Summary statistics:
##      Region Area CoveredArea Effort   n   k      ER      se.ER
## 1 PASTURE 1      1    117.8411    124  59 124 0.4758065 0.07009015
## 2 PASTURE 2      1    119.7418    126 121 126 0.9603175 0.09521957
## 3 PASTURE 3      1    116.8908    123  91 123 0.7398374 0.07467322
## 4      Total      3    354.4738    373 271 373 0.7265416 0.04772999
##      cv.ER
## 1 0.14730811
## 2 0.09915427
## 3 0.10093193
## 4 0.06569478
##
## Abundance:
##      Label Estimate          se          cv      lcl      ucl      df
## 1 PASTURE 1 1.430165 0.3084015 0.21564058 0.9403512 2.175114 353.1254
## 2 PASTURE 2 4.116174 0.5643816 0.13711317 3.1470035 5.383816 329.2717
## 3 PASTURE 3 2.345638 0.3719019 0.15855041 1.7207006 3.197546 375.0930
## 4      Total 7.891977 0.7427181 0.09411054 6.5625849 9.490665 537.6087
##
## Density:
##      Label Estimate          se          cv      lcl      ucl      df
## 1 PASTURE 1 1.430165 0.3084015 0.21564058 0.9403512 2.175114 353.1254
## 2 PASTURE 2 4.116174 0.5643816 0.13711317 3.1470035 5.383816 329.2717
## 3 PASTURE 3 2.345638 0.3719019 0.15855041 1.7207006 3.197546 375.0930
## 4      Total 2.630659 0.2475727 0.09411054 2.1875283 3.163555 537.6087
```

A similar process was conducted for the 1981 data: a truncation distance of 55m was again used.

```
# Savannah sparrow 1981
data(Savannah_sparrow_1981)
# Rename
sav1981 <- Savannah_sparrow_1981
conversion.factor <- convert_units("meter", NULL, "hectare")

# Fit alternative models
# Half-normal detection function, truncation 55m
sav1981.hn <- ds(data=sav1981, key="hn", adjustment="cos", truncation=55,
                 transect="point", convert.units=conversion.factor)
# Hazard rate
sav1981.hr <- ds(data=sav1981, key="hr", adjustment="cos", truncation=55,
                 transect="point", convert.units=conversion.factor)
# Half normal with pasture
```

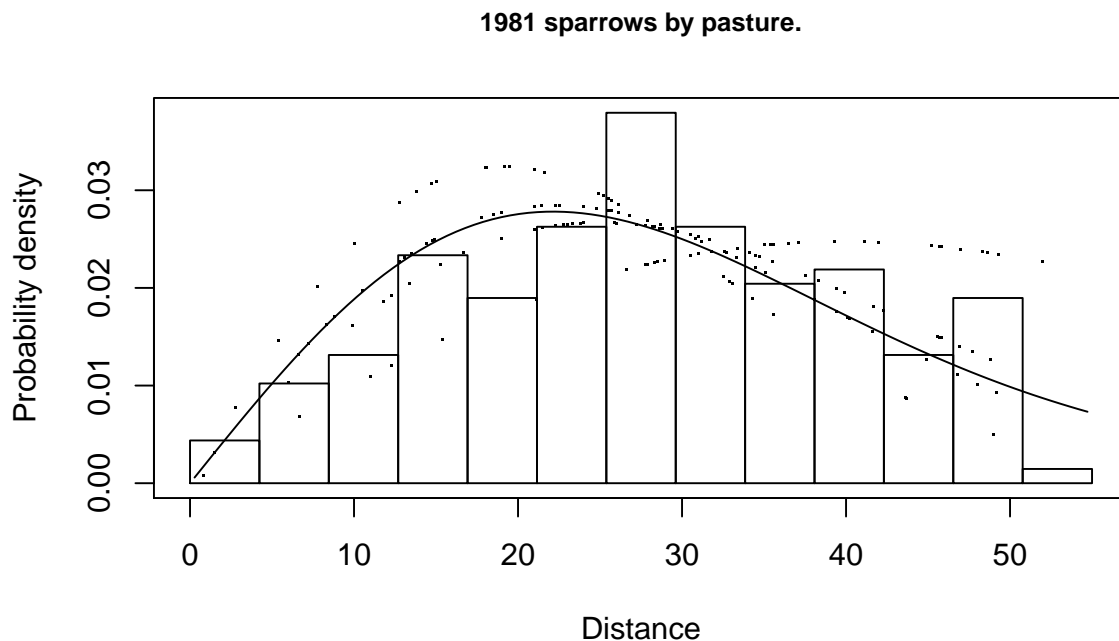


Figure 2: Stronger influence of pasture covariate seen here.

```
sav1981.hn.region <- ds(data=sav1981, key="hn", truncation=55,
                        transect="point", convert.units=conversion.factor,
                        formula=~Region.Label)
# Hazard rate with pasture
sav1981.hr.region <- ds(data=sav1981, key="hr", truncation=55,
                        transect="point", convert.units=conversion.factor,
                        formula=~Region.Label)
# Compare models
AIC(sav1981.hn, sav1981.hr, sav1981.hn.region, sav1981.hr.region)

##           df      AIC
## sav1981.hn      1 1266.358
## sav1981.hr      2 1267.335
## sav1981.hn.region 4 1261.684
## sav1981.hr.region 5 1260.638
```

For 1981, there was a clear preference for including pasture as a covariate in the detection function but little to choose from between the half normal and hazard rate key function. For comparability with 1980, the plots and results below are for the half normal model although AIC showed a slight preference for the hazard rate model. The differences in detection between pastures can easily be seen and this is reflected in the estimated densities (birds per hectare).

```
# Plot results of selected model
plot(sav1981.hn.region, pch=".", pdf=TRUE, main="1981 sparrows by pasture.")
```

```
# Summary of results for selected model
```

```
summary(sav1981.hn.region)
```

```
##
## Summary for distance analysis
## Number of observations : 162
## Distance range       : 0 - 55
##
## Model : Half-normal key function
## AIC   : 1261.684
##
## Detection function parameters
## Scale coefficient(s):
##              estimate      se
## (Intercept)    2.9440439 0.1110264
## Region.LabelPASTURE 1 0.7362729 0.3728560
## Region.LabelPASTURE 2 0.1660991 0.1525319
## Region.LabelPASTURE 3 0.2703068 0.1791860
##
##              Estimate      SE      CV
## Average p      0.3435483 0.03716055 0.1081669
## N in covered region 471.5494106 59.62867919 0.1264527
##
## Summary statistics:
##      Region Area CoveredArea Effort  n  k  ER      se.ER      cv.ER
## 1 PASTURE 0      1    95.03318    100  31 100 0.310 0.05448566 0.17576019
## 2 PASTURE 1      1    95.03318    100  32 100 0.320 0.06175874 0.19299605
## 3 PASTURE 2      1    95.03318    100  51 100 0.510 0.08225975 0.16129363
## 4 PASTURE 3      1    95.03318    100  48 100 0.480 0.07174590 0.14947063
## 5      Total      4   380.13271    400 162 400 0.405 0.03435678 0.08483156
##
## Abundance:
##      Label Estimate      se      cv      lcl      ucl      df
## 1 PASTURE 0 1.3887551 0.3779452 0.2721468 0.8203856 2.350895 255.9018
## 2 PASTURE 1 0.5241861 0.1818325 0.3468855 0.2698808 1.018120 250.9332
## 3 PASTURE 2 1.6980682 0.4058401 0.2390011 1.0674967 2.701119 251.7568
## 4 PASTURE 3 1.3509357 0.3545938 0.2624801 0.8125802 2.245966 253.0328
## 5      Total 4.9619451 0.6827512 0.1375975 3.7903643 6.495655 356.9874
##
## Density:
##      Label Estimate      se      cv      lcl      ucl      df
## 1 PASTURE 0 1.3887551 0.3779452 0.2721468 0.8203856 2.350895 255.9018
## 2 PASTURE 1 0.5241861 0.1818325 0.3468855 0.2698808 1.018120 250.9332
## 3 PASTURE 2 1.6980682 0.4058401 0.2390011 1.0674967 2.701119 251.7568
## 4 PASTURE 3 1.3509357 0.3545938 0.2624801 0.8125802 2.245966 253.0328
## 5      Total 1.2404863 0.1706878 0.1375975 0.9475911 1.623914 356.9874
```

In these models, the detection functions have been fitted to all the detections within the

study region (for each year). An alternative would be to fit separate detection functions within each pasture (specified in `Region.Label`), provided there are enough detections. This would allow different shape detection functions to be fitted in each pasture (providing this is a reasonable thing to do).