Distance sampling online workshop

Analysis in R: Covariates in the detection function

CREEM, Univ of St Andrews – October 2018

1   Covariates in the detection function

We illustrate fitting multiple covariate distance sampling (MCDS) models
to point transect data using a bird survey in 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, Section 5.3.2) which
duplicates the analysis presented in Marques et al. (2007). This set of data
is included in 'Distance for Windows' as one of the Sample Projects: you can
open this project (entitled `amakihi.zip`) in the 'Sample projects' directory
in the 'My Distance projects' directory residing under 'My Documents'. We
describe the analysis of these data using Distance in R (R Core Team, 2018).

Hawaii amakihi *(Hemignathus virens)*

2   Objectives of this practical

1. Introduce different types of plots to explore covariates
2. Add covariates to the detection function
3. Plot the detection functions.

3   Importing the data

Analysis begins by importing the data from a comma-delimited file. This file
was created by copying the data from the amakihi Distance project. The file
resides on a website associated with Buckland et al. (2015) and it can be
read directly from the website rather than downloaded to your computer.

```
amakihi <- read.csv(file = "https://synergy.st-andrews.ac.uk/ds-manda/files/2016/11/amakihi.csv")
```

Check that it has been imported correctly.

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

These data consist of eight columns:

- 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 they are visited a single time)
- distance - radial distance of detection from observer
- OBS - initials of the observer
- MAS - minutes after sunrise
- HAS - hour after sunrise

Note there is no `Area` field in the dataset. Detection functions can be
fitted to the data, but bird density and abundance cannot be estimated.

The latter three columns are the covariates to be considered for possible inclusion into the detection function.

There a couple of records with missing distances and so can be deleted with the following command:

```
amakihi <- amakihi[!is.na(amakihi$distance), ]
```

In this command,

- records in `amakihi` are selected using the square brackets `[]`
- `amakihi` is a data frame and so selection can be performed on either rows or columns i.e. `[rows, columns]`. In this case, the selection is performed on the rows (because the selection criteria is before the comma) and all columns will be retained
- the rows selected as those where the distances (stored in `amakihi$distance`) are not missing. The function `is.na` selects elements that are missing; the symbol `!` means 'not', and so `!is.na` selects elements that are not missing.

4   Exploratory data analysis

It is important to gain an understanding of the data prior to fitting detection functions. 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 by plotting histograms with different number of bins and different truncation.

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.

This format is probably not as useful as a histogram in a distance sampling context but boxplots can be useful to compare the distances for discrete groups in the data. Here we use boxplots to display the distribution of distances recorded by each observer and for each hour after sunrise. Note how the ~ symbol is used to define the groups.

Boxplots of distances by observer:

Question: Examine the distribution of radial distances of the point transect data of the amakihi.

Basic syntax will be

```
ggplot(amakihi, aes(x=distance))
+ geom_histogram(binwidth=1)
```

Examine the full dataset, then truncate the data to 82.5m.

```
ggplot(droplevels(amakihi), aes(x = OBS, y = distance)) +
    geom_boxplot() + labs(x = "Observer initials",
    y = "Radial distance (m)")
```

Boxplot of distances for each hour after sunrise:

```
ggplot(amakihi, aes(x = factor(HAS), y = distance)) +
    geom_boxplot() + labs(x = "Hours after sunrise",
    y = "Radial distance (m)")
```

For minutes after sunrise (a continuous variable), we create a scatterplot of MAS (on the x-axis) against distances (on the y-axis).

```
ggplot(amakihi, aes(x = MAS, y = distance)) +
    geom_point(alpha = 0.25, size = 1.6) + labs(x = "Hours after sunrise",
    y = "Radial distance (m)")
```

5   Colinearity

Estimating the parameters of a detection function when covariates are involved is complex. You will recall from multiple linear regression that problems in estimation arise when two covariates in the model are highly correlated. In the exploratory data analysis, it is useful to look for colinearity in potential covariates.

To alleviate the potential colinearity difficulty, hours after sunrise could be transformed to a discrete, rather than a continuous variable.

Question: Compute the correlation of minutes after sunrise and hours after sunrise using the `cor()` function.

6   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, indicate it is 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 by `Distance for Windows`. By default R uses the first factor level but by using the `relevel` function, this can be changed:

```
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 entertaining 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.

```
amakihi$MAS <- amakihi$MAS/sd(amakihi$MAS, na.rm = TRUE)
```

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

```
range(amakihi$MAS)
```

7   Candidate models

With three potential covariates, there are 8 possible combinations for including them in the detection function:

- No covariates
- OBS
- HAS
- MAS
- OBS + HAS
- OBS + MAS
- HAS + MAS
- OBS + HAS + MAS

Even without considering covariates there are a number of possible key function/adjustment term combinations and 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 as an illustration of how to incorporate covariates in the detection function.

If it is not already loaded, load the `Distance` package (Miller, 2017).

Fit a hazard rate model with no covariates or adjustment terms. By default, line transects are assumed and because our data are point transects, the argument `transect="point"` is specified:

```
library(Distance)
hr.model0 <- ds(amakihi, transect = "point", key = "hr",
    truncation = 82.5, adjustment = NULL, order = 0)
```

The fitted model can be investigated using the `summary` function. Make a note of the AIC for this model.

```
summary(hr.model0)
```

Question: 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?

```
hr.obs <- ds(amakihi, transect = "point", key = "hr",
    formula = ~OBS, truncation = 82.5)
print(hr.obs$ddf$criterion)
```

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

```
hr.obshas <- ds(amakihi, transect = "point", key = "hr",
    formula = ~OBS + HAS, truncation = 82.5)
print(hr.obshas$ddf$criterion)
```

Question: Fit the other candidate models including covariates shown in the list above and decide which model is best in terms of AIC.

```
hr.has <- ds(amakihi, transect = "point", key = "hr",
    formula = ~HAS, truncation = 82.5)
hr.mas <- ds(amakihi, transect = "point", key = "hr",
    formula = ~MAS, truncation = 82.5)
hr.obsmas <- ds(amakihi, transect = "point", key = "hr",
    formula = ~OBS + MAS, truncation = 82.5)
hr.hasmas <- ds(amakihi, transect = "point", key = "hr",
    formula = ~HAS + MAS, truncation = 82.5)
hr.hasmasobs <- ds(amakihi, transect = "point",
    key = "hr", formula = ~HAS + MAS + OBS, truncation = 82.5)
```

A useful function for summarising a candidate model set is `summarize_ds_models`. The arguments to the function is an enumeration of the candidate model objects.

```
summarize_ds_models(hr.model0, hr.obs, hr.has,
    hr.mas, hr.obshas, hr.obsmas, hr.has.mas,
    hr.hasmasobs)
```

8   Plotting the detection functions

The detection functions can be investigated using the `plot` function as shown below. A few different plotting options are illustrated.

```
# Plot simple model
plot(hr.model0, nc = 20, main = "No covariates",
    pch = 20, pdf = TRUE)
```

```r
# Plot model with OBS
plot(hr.obs, nc = 10, main = "Model with OBS covariate",
    pch = 1, cex = 0.5, pdf = TRUE)
```

What does the detection function look like for your selected model?

```r
# Fit best model
hr.best <- ds(amakihi, transect = "point", key = "hr",
    truncation = 82.5, quiet = TRUE, formula = ~OBS +
        MAS)
# Plot model with OBS and MAS
plot(hr.best, nc = 10, main = "Model with OBS and MAS covariates",
    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.

References

Buckland, S. T., E. A. Rexstad, T. A. Marques, and C. S. Oedekoven. 2015. Distance Sampling: Methods and Applications. Springer. URL https://www.springer.com/gb/book/9783319192185.

Marques, T. A., L. Thomas, S. G. Fancy, and S. T. Buckland. 2007. Improving estimates of bird density using multiple covariate distance sampling. The Auk, **124**:1229–1243. URL https://doi.org/10.1642/0004-8038(2007)124[1229:ieobdu]2.0.co;2.

Miller, D. L. 2017. Distance: Distance Sampling Detection Function and Abundance Estimation. URL https://CRAN.R-project.org/package=Distance. R package version 0.9.7.

R Core Team. 2018. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.