Distance sampling online workshop

Analysis in R: Lure point transects

CREEM, Univ of St Andrews – October 2018

A call station to lure Scottish crossbills *(Loxia scotica)*.

1 Conducting dual surveys: fitting GLMs and bootstrapping

This practical is based on the lure point transect case study in Buckland et al. (2015), Section 10.2.1, which is a simplified version of the analysis in Summers and Buckland (2010). Generalised linear models (GLMs) are used to model the response of Scottish crossbills to a lure in order to estimate their probability of response and hence estimate the density and abundance. To provide a measure of precision for the abundance estimate, 95% confidence intervals are obtained by bootstrapping.

The Scottish crossbill *(Loxia scotica)* is Britain's only endemic bird species. A point transect study was conducted to obtain the number of birds within each point after responding to an audible lure. The probability of responding to the lure was estimated by recording the response of previously detected birds to the lure at different distances Summers and Buckland (2010).

1.1 Objectives of the practical

1. Fit a GLM
2. Obtain predicted values from GLM
3. Calculate abundance
4. Using `for` loop to bootstrap abundance.

1.2 The lure trials

The data provided in the response trials are:

- No. - trial number
- day - days from 1st January
- time - hour of the day
- habitat - habitat type (1=plantation, 2= native pinewood)
- dist - distance of the bird when the lure was played (m)
- behavcode - behaviour code (1=perching and feeding, 2= giving excitement calls, 3=singing)
- numbirds - flock size
- response - response of bird to lure (0=no response, 1=response).

The trials data are in file `lure-trials.csv`. Import the data and check that it has been read correctly:

```
xbill <- read.csv(file = "datasets/lure-trials.csv",
    header = TRUE)
```

```
head(xbill, n = 2)
```

```
##   No. day time habitat dist behavcode
## 1   1  47    9       1  150         1
## 2   2  47   11       1  150         1
##   numbirds response
## 1        1        0
## 2        2        0
```

1.3 Summarising the data

Examine how many birds did, or did not, respond to the lure:

```
table(xbill$response)
```

```
##
##   0   1
##  62 113
```

There are six potential covariates that might affect the probability that a bird responds to the lure (`day`, `time`, `dist`, `numbirds`, `habitat`, and `behavcode`): the latter two are factor type variables and so we need to treat them as factors in models:

```
xbill$habitat <- factor(xbill$habitat)
xbill$behavcode <- factor(xbill$behavcode)
```

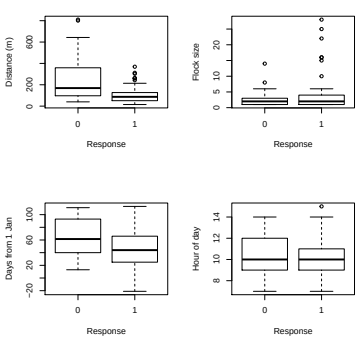
To look at the response in each factor level create a two-way table, for example:

```
addmargins(table(xbill$response, xbill$habitat))
```

```
##
##         1   2 Sum
##   0     6  56  62
##   1    25  88 113
##   Sum  31 144 175
```

Question: Create a set of boxplots to look at the distribution of distances for each response level.

```
# Divide plot window into 4
par(mfrow = c(2, 2))
boxplot(xbill$dist ~ xbill$response, xlab = "Response",
    ylab = "Distance (m)")
boxplot(xbill$numbirds ~ xbill$response, xlab = "Response",
    ylab = "Flock size")
boxplot(xbill$day ~ xbill$response, xlab = "Response",
    ylab = "Days from 1 Jan")
boxplot(xbill$time ~ xbill$response, xlab = "Response",
    ylab = "Hour of day")
```



```
par(mfrow = c(1, 1))
```

Answer: Qualitatively, these boxplots suggest little difference in crossbill behaviour in response to the lure attributable to flock size or time of day. There may be more influence of distance and date upon response to lure. Models will be fitted to make stronger inference.

1.4 Fitting a GLM

Building a model to explain the probability of response in terms of the potential covariates. The dependent variable, `response`, can only take two values (0 and 1) and so rather than fit a linear regression model to these data we fit a GLM. The `glm` function allows us to specify a distribution for the dependent variable in the model with the `family` argument. In effect, this performs a logistic regression, with *success* (animal responding) modelled as a function of explanatory covariates.

We can include all the covariates in a model as follows.

```
model1 <- glm(response ~ dist + numbirds + day +
    time + habitat + behavcode, family = binomial,
    data = xbill)
```

As usual, the `summary` function can be used to display details of the model object.

```
summary(model1)
```

```
##
## Call:
## glm(formula = response ~ dist + numbirds + day + time + habitat +
##     behavcode, family = binomial, data = xbill)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9526  -0.8708   0.5563   0.8168   2.0009
##
## Coefficients:
##              Estimate Std. Error z value
## (Intercept)  2.641601   1.301148   2.030
## dist        -0.010086   0.002323  -4.342
## numbirds     0.076882   0.081172   0.947
## day         -0.008597   0.007405  -1.161
## time        -0.020251   0.113258  -0.179
## habitat2    -0.263713   0.574532  -0.459
## behavcode2   0.070903   0.495256   0.143
## behavcode3   0.962426   0.614541   1.566
##             Pr(>|z|)
## (Intercept)   0.0423 *
## dist        1.41e-05 ***
## numbirds      0.3436
## day           0.2456
## time          0.8581
## habitat2      0.6462
## behavcode2    0.8862
```

```
## behavcode3    0.1173
## ---
## Signif. codes:
##   0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1  ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 219.23  on 166  degrees of freedom
## Residual deviance: 169.10  on 159  degrees of freedom
##   (8 observations deleted due to missingness)
## AIC: 185.1
##
## Number of Fisher Scoring iterations: 5
```

We see that only `dist` has a coefficient that is significantly different from zero. Experiment with dropping non-significant terms. Using a backwards stepping procedure, consistent with the visual inspection of the boxplots above, other covariates remain non-significant, resulting in a simple model:

```
model2 <- glm(response ~ dist, family = binomial,
    data = xbill)
```

```
summary(model2)
```

```
##
## Call:
## glm(formula = response ~ dist, family = binomial, data = xbill)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9521  -0.8066   0.5957   0.7618   1.9355
##
## Coefficients:
##             Estimate Std. Error z value
## (Intercept)  2.16370    0.33944   6.374
## dist        -0.01049    0.00210  -4.993
##             Pr(>|z|)
## (Intercept) 1.84e-10 ***
## dist        5.94e-07 ***
## ---
## Signif. codes:
##   0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1  ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 227.52  on 174  degrees of freedom
## Residual deviance: 179.44  on 173  degrees of freedom
## AIC: 183.44
##
## Number of Fisher Scoring iterations: 5
```

(Degrees of freedom change a little between models because some covariates in the first model have missing values and so these observations are excluded.)

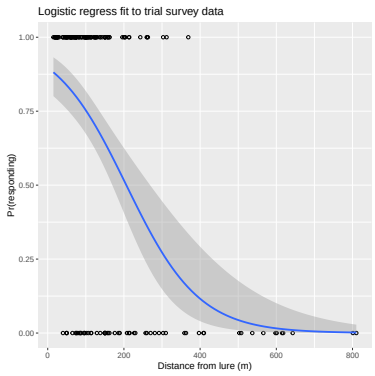
1.5 Prediction

Having fitted a model, we now want to see how the predicted probability of response changes with distance (similar to a detection function model). We assume that the maximum distance that a crossbill will respond to a lure is 850m. Here we create a 'prediction' data frame that has one column called `dist` and this ranges from 0 to 850 (in unit intervals). (Prediction data needs to contain objects with the same names as the explanatory variables in the fitted model.)

```
w <- 850
preddata <- data.frame(dist = 0:w)
phat <- predict.glm(model2, newdata = preddata,
    type = "response")
```

Now we have the estimated probabilities, we can overlay this onto a plot of the observed responses (black circles).

```
ggplot(data = xbill, aes(x = dist, y = response)) +
    geom_point(shape = 1) + geom_smooth(method = "glm",
    method.args = list(family = "binomial")) +
    labs(title = "Logistic regress fit to trial survey data") +
    labs(x = "Distance from lure (m)", y = "Pr(responding)")
```



1.6 Estimating abundance

This section is technical and require understanding of abundance estimation with point transects.

Abundance is obtained from

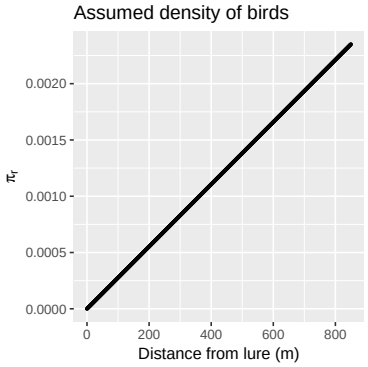
$$\hat{N} = \frac{n \cdot A}{P_a \cdot a}$$

where

- n is the number of detections
- A is the area of the study region (i.e. 3505.8 km^2)
- P_a is the probability of response (or detection) in the covered area, and
- a is the area of the covered region (i.e. $a = k\pi w^2$ where k is the number of points)

First we calculate P_a. To do this we need to specify the function $\pi(r), r \leq w$, which represents the probability density function (pdf) of distances of animals from the point. Assuming crossbills are equally likely at all distances from the point, the pdf is triangular:

```
pi.r <- preddata$dist/sum(preddata$dist)
preddata$pi <- pi.r
ggplot(data = preddata, aes(x = dist, y = pi.r)) +
    geom_point(size = 0.6) + labs(title = "Assumed density of bi
    labs(x = "Distance from lure (m)", y = expression(pi[r]))
```



Then we multiply $\pi(r)$ with the probability of response and (numerically) integrate from 0 to w.

```
Pa <- sum(phat * pi.r)
print(Pa)
```

```
## [1] 0.09709381
```

Assuming that the probability of response is a function of distance only (as in `model2`), then P_a is just less than 10% (i.e. <10% of birds within 850m of a point are detected).

Analysis of main survey data

Read in data from the point transect survey. These data consist of:

- point - point transect identifier
- nscottish - the number of Scottish crossbills detected at the point

Note that detection distances are unknown in the main survey: instead, we have used the trials data to estimate the detection function, and hence the proportion of birds within 850m that are detected.

```
detections <- read.csv("datasets/mainsurveydetections.csv",
    header = TRUE)
n <- sum(detections$nscottish)
```

We now calculate the number of points (k) in the main survey, and hence, the total covered area within 850m of a point, converting from m^2 to km^2. Note that `pi` is a reserved word to represent π (i.e. 3.141593).

```
k <- length(detections$point)
# Covered area (km2)
a <- k * pi * (w/1000)^2
# Size of the study region (km2)
A <- 3505.8
```

We can now estimate the size of the population as:

```
Nscot <- (n * A)/(Pa * a)
```

Question: What is your estimate of the population of Scottish crossbills?

```
## [1] 10007.67
```

1.7 Measure of precision in abundance estimate

We can calculate $1 - \alpha$ confidence interval for true abundance by bootstrapping both trials and points. The steps involved are

1. randomly generate (with replacement) a new set of response data,
2. estimate P_a for the new data,
3. generate (with replacement) a new set of point transect data,
4. estimate abundance,
5. repeat steps 1-4 many times to build a distribution of abundances and
6. use the $\alpha/2$ and $1-\alpha/2$ percentiles of the distribution as confidence interval bounds.

The following code does this:

```
# Initialise parameters Number of bootstraps
nboot <- 999
# Number of trials
m <- length(xbill$dist)
# Create empty vectors to store new sample
bdistances <- vector(length = m)
bresponse <- vector(length = m)
# Create empty vector to store bootstrap
# abundances
bNscot <- vector(length = nboot)
# Create prediction data (w is truncation
# distance defined earlier)
pred <- data.frame(bdistances = 0:w)
# A loop for the bootstraps
for (i in 1:nboot) {
    # Bootstrap trials Generate index of sample
    btindex <- sample(1:m, size = m, replace = TRUE)
    for (j in 1:m) {
        bdistances[j] <- xbill$dist[btindex[j]]
        bresponse[j] <- xbill$response[btindex[j]]
    }
    # Fit GLM
    bmodel <- glm(bresponse ~ bdistances, family = binomial)
    # Predict probability of response
    bphat <- predict.glm(bmodel, newdata = pred,
        type = "response")
    # Calculate Pa
    bPa <- sum(bphat * pi.r)
    # Bootstrap points
    rindex <- sample(1:k, k, replace = TRUE)
    n <- sum(detections$nscottish[rindex])
    # Calculate abundance
    bNscot[i] <- (n * A)/(bPa * a)
}  # End of bootstrap loop
```

Having obtained a distribution of abundances, the $\alpha/2$ and $1-\alpha/2$ percentiles can be obtained:

```
alpha <- 0.05
bounds <- c(alpha/2, 1 - (alpha/2))
plot.this <- as.data.frame(bNscot)
ggplot(data = plot.this, aes(bNscot)) + geom_histogram(fill = "white",
    colour = "black") + labs(title = "Distribution of replicate
    labs(x = expression(hat(N)), y = "Count") +
    geom_vline(xintercept = quantile(bNscot, probs = bounds),
        size = 1.5, linetype = "dotted")
```

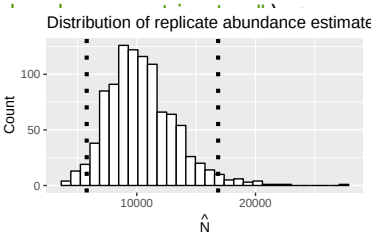


Figure 1: Distribution of abundance estimates from bootstrap.

Note: The distribution of estimates is skewed-right; long tail of the distribution is to the right. This shape is customary in many abundance estimation problems.

```
quantile(bNscot, probs = bounds)
```

```
##      2.5%     97.5%
##  5778.673 16851.491
```

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.

Buckland, S. T., R. W. Summers, D. L. Borchers, and L. Thomas. 2006. Point transect sampling with traps or lures. Journal of Applied Ecology, **43**:377–384. URL https://doi.org/10.1111/j.1365-2664.2006.01135.x.

Summers, R. W. and S. T. Buckland. 2010. A first survey of the global population size and distribution of the scottish crossbill *Loxia scotica*. Bird Conservation International, **21**:186–198. URL https://doi.org/10.1017/S0959270909990323.