Exercise

Binomial (Bernoulli) GLM - dolphin behavioural plasticity

1. The data for this exercise were collected by the Cromarty Lighthouse team, using underwater sound recorders (CPOD) to continuously monitor the pattern of presence and foraging behaviour of bottlenose dolphins at Sutors, in the Moray Firth, between 2010 and 2016. Additional background for this study is provided at the end of the exercise, in case of interest.

- Variables:
  - `X` index of the observations
  - `presence`: 0 for absence, 1 for presence in 1h time slot. Note that "absence" refers to the absence of a detection, not to the absence of dolphins. We can ignore this in the analysis, but we should keep it in mind when interpreting the results.
  - `year`
  - `julianday`: day of the year
  - `tideangle_deg`: continuous tidal state, from high to ebb, low and flood
  - `mh`: hour of the day (integer)
  - `mon`: month (integer)
  - `Time6`: Bin time of day into 6 4h periods: MNight (2200-0200); AM1 (0200-0600); AM2 (06:00-10:00); MDay (10:00-14:00); PM1 (14:00-18:00); PM2 (18:00-22:00)
  - `Tide4`: Bin tide angle into 4 quadrants with tide peaks (high, descending, low, rising) in middle of respective bin
- It has been suggested that the patterns of use of coastal foraging sites by this dolphin population is quite variable over time. The goal of this exercise is to describe variation in dolphin probability of presence in relation to factors like tidal state, time of day and season.
- As in previous exercises, either create a new R script (perhaps call it GLM_PresAbs) or continue with your previous R script in your RStudio Project. Again, make sure you include any metadata you feel is appropriate (title, description of task, date of creation etc) and don't forget to comment out your metadata with a `#` at the beginning of the line.

2. Import the data file 'dolphin.csv' into R (a "small" 5000 records-long subset of the original data set) by running the following chunk of code (please unfold and copy/paste - adjust the path as required).

```
dat<- read.csv("./data/dolphin.csv", stringsAsFactors= T)

dat$Time6<- factor(dat$Time6, levels= c("MNight", "AM1", "AM2", "MDay", "PM1", "PM2"))
# reordering chronologically

str(dat)
```

```
## 'data.frame':    5000 obs. of  9 variables:
##  $ X            : int  31458 14027 40551 40456 15894 13109 23797 6053 23445 34584 ...
##  $ presence     : int  0 1 0 0 1 0 0 0 0 0 ...
##  $ year         : int  2014 2011 2015 2015 2011 2011 2013 2010 2012 2014 ...
##  $ julianday    : int  59 226 80 76 312 188 102 256 327 192 ...
##  $ tideangle_deg: int  247 356 176 299 127 75 44 73 180 103 ...
##  $ mh           : int  8 13 7 8 3 7 3 6 14 15 ...
##  $ mon          : int  2 8 3 3 11 7 4 9 11 7 ...
##  $ Time6        : Factor w/ 6 levels "MNight","AM1",..: 3 4 3 3 2 3 2 2 4 5 ...
##  $ Tide4        : int  4 1 3 4 2 2 1 2 3 2 ...
```

3. Take a look at the structure of this dataframe, and do an initial data exploration.

- Some things you could focus on are:
  - look at any correlation/data imbalance (different sample sizes between portions of a predictor variable) for all predictors, or useful combinations of predictors (for example `year` and `month`, `Tide4` and time of day `mh`)
  - look for factors affecting the probability of presence of dolphins (proportion of time present). Which ones are continuous or categorical? Which ones would your intuition guide you to use for modelling?
- Notes:
  - Presence/absence data (Bernoulli) are more difficult than most to explore.
  - One approach for data imbalance is to count observations per categories of interest.
  - `table()` is a useful way to count the number of observations per category or combinations of categories, e.g. `ObsPerMonthYear<- table(dat$year, dat$mon)`
  - `plot(ObsPerMonthYear)` returns a "mosaic plot" where the area of each rectangle is proportional to the count.
  - For factors affecting the proportion of time present, you could calculate mean presence per category, which is the proportion of time present: `bla<- tapply(dat$presence, list(dat$GroupOfInterest), mean)` and plot this using `plot(bla, type= "b", ylim= c(0, 1), xlab= "GroupOfInterest", ylab= "presence")`
  - In more than one dimension, `tmp<- tapply(dat$presence, list(dat$Group1, dat$Group2), mean)` calculates the proportion of time present for each combination of Group1 and Group2, and `matplot(tmp, type= "l", ylim= c(0, 1), xlab= "Group1", ylab= "presence", lty= 1)` plots the proportion of time present against Group1, with a separate line per Group2 categories.

4. We will start with a toy model, to get you started thinking about the problem. You will need to specify a Binomial (Bernoulli) GLM (using `glm()` and the appropriate `family` argument). Let's include the main effects of numerical time of day, tide angle and day of the year as predictors: `tideangle_deg + mh + julianday`.

5. Obtain summaries of the model output using the `summary()` function. Make sure you understand the mathematical and biological interpretation of the model, by writing down the complete model on paper (with distribution and link function). What biological hypothesis does each term imply, qualitatively? Is this model biologically sensible?

6. Let's now validate the model, using deviance residuals. The easiest tool is the `binnedplot()` in the `arm` package, if you can. If you cannot install the arm package and access its binnedplot, use the "DIY" alternative code chunk further down.

```
library(car)
vif(PA1)
# No concern.

par(mfrow= c(2, 2))
plot(PA1, col= dat$presence + 1) # red is presence, black is absence
# Not very useful or pretty statistical art. Not worth framing.

# plot against predictors:
res1.p<- resid(PA1, type= "pearson")

par(mfrow= c(2, 2))
plot(res1.p ~ dat$tideangle_deg, col= dat$presence + 1)

plot(res1.p ~ dat$mh, col= dat$presence + 1)

plot(res1.p ~ dat$julianday, col= dat$presence + 1)

# Can't see anything useful.

# Use arm if you can:
library(arm)
par(mfrow= c(2, 2))
binnedplot(x= dat$tideangle_deg, y= res1.p, xlab= "Tide angle", nclass= 100)
binnedplot(x= dat$mh, y= res1.p, xlab= "hour")
binnedplot(x= dat$julianday, y= res1.p, xlab= "Day of the year", nclass= 100)
```

In case needed, a home-made alternative to the `binnedplot` function:

```
par(mfrow= c(2, 2))
# plot the residuals against tideangle_deg
plot(res1.p ~ dat$tideangle_deg, col= dat$presence + 1)
# get the mean of the residuals for each 1 degree bin of tideangle_deg
```

```
tide.means<- tapply(res1.p, list(dat$tideangle_deg), mean)
# convert ordered bin labels into numbers (1 to 360)
tide.vals<- as.numeric(names(tide.means))
# plot residual means against bin number
lines(tide.means ~ tide.vals, col= 3)
# add horizontal line at y= 0 for reference
abline(h= 0, lty= 3, col= grey(0.5))

# same idea for hour of the day:
plot(res1.p ~ dat$mh, col= dat$presence + 1)
hour.means<- tapply(res1.p, list(dat$mh), mean)
lines(hour.means ~ as.numeric(names(hour.means)), col= 3)
abline(h= 0, lty= 3, col= grey(0.5))

# same for julianday:
plot(res1.p ~ dat$julianday, col= dat$presence + 1)
day.means<- tapply(res1.p, list(dat$julianday), mean)
lines(day.means ~ as.numeric(names(day.means)), col= 3)
abline(h= 0, lty= 3, col= grey(0.5))

# Same story.
```

7. Are you happy with the diagnostic plots? Is there something you could do to improve the model while addressing the initial question(s)? Spend some time looking at the available predictors, and working out what your model should look like, before reading the hints in the code chunk below. If you have relevant biological information, or insight from your data exploration that suggests a better approach than what is indicated below, feel free to try it for comparison.

```
# Please take the time to think before unfolding the next code chunk
```

```
# The issue is that the effects of these predictors are not linear
# on the logit (link) scale.

# There are several ways the non-linearity could be addressed.
# one of the most straightforward with glm() is to discretize
# continuous predictors into bins and to treat them as factors.
# In this way, a mean is estimated per category of the variable,
# and no assumption is made about the shape of the relationship.

# Each of the predictors we started with already has one or more
# categorical counterpart in the data set.
# I suggest you try fTide4 + fMonth + Time6, with fTide4 and
# fMonth being the factor version of Tide4 and mon (both need creating).
```

8. Fit the new version of the model. Are all the terms significant? If not, simplify the model. Remember to choose the correct ANOVA method (sequential or not), and the appropriate test. What is the proportion of deviance explained?

9. Do the model validation for the minimal adequate model. Is everything looking good?

10. Assuming that the model is fine as it is, let's plot the predictions for the probability of presence in relation to time of day `fTime6`. You will need to set the value of other predictors `fTide4`, `fMonth` at a fixed level of your choice, e.g. "1". Optionally, you can add the confidence intervals around the predictions (highly recommended in a report).

- Calculation of confidence intervals:
  - As for the Poisson GLM, you will need to calculate the lower and upper bounds of the 95% CI on the link scale (same method)
  - Only then convert these to the response scale
  - Don't forget that the link function is different though, and so is the function for the back-transformation
  - In R you can do the back-transformation yourself using the equation provided in the lecture, or use the pre-made `plogis` function.
- Suggested approach:
  - create a `data.frame` called `X` containing the data to predict for.
  - use `predict()` with the appropriate options to obtain the fitted values on the link scale and for being able to calculate the confidence intervals later. Store in object `Z`.
  - I suggest you plot the predictions for each level of your categorical predictor as dots using `plot(Z$fit)`. (Some people prefer bars using `barplot(Z$fit)`, but this can make drawing confidence intervals slightly harder)
  - in X, add columns for the fitted values and their confidence intervals, on the response scale (to be calculated).
  - Because in the model the predictions are for categorical predictors, you will need to draw a vertical error bar for each predicted value
  - This can be done using the `arrows` function, with arguments `x0` and `y0`, the X,Y coordinates of the starting point of the arrows, and `x1` and `y1`, the X,Y coordinates of the end point of the arrows. See `?arrows` for further formatting options.
  - Note that for vertical bars like we want, `x0` and `x1` should the same, and `y0` and `y1` are the lower and upper bounds of the confidence intervals that you calculated.

The code is available below for you to unfold, if you don't want to try yourself (you are always welcome to ask demonstrators for help).

```
PA10.dat4pred<- data.frame(Time6= levels(dat$Time6),
                                fMonth= "10", fTide4= "1")

PA10.pred<- predict(PA10, PA10.dat4pred, type= "link", se.fit= T)

# Convert predictions to the response (probability) scale.
# And add them to the prediction data frame (that bit is optional)
PA10.dat4pred$fit.resp<- exp(PA10.pred$fit)/(1+exp(PA10.pred$fit))
# or plogis(PA10.pred$fit)

# lower 95% CI
```

```
PA10.dat4pred$LCI<- plogis(PA10.pred$fit - 1.96*PA10.pred$se.fit)
# upper 95% CI
PA10.dat4pred$UCI<- plogis(PA10.pred$fit + 1.96*PA10.pred$se.fit)
```

```
par(mfrow= c(1, 1))
plot(x= 1:6, y= PA10.dat4pred$fit.resp,
      pch= 16, cex= 1.4, xlab= "Section of day",
          ylab= "Fitted probability", ylim= c(0, 1),
      main= "Predictions for time of day\n(assuming Tide = 1 and Month = 10)")

arrows(x0= 1:6, x1= 1:6,
          y0= PA10.dat4pred$LCI, y1= PA10.dat4pred$UCI,
          length= 0.02, angle= 90, code= 3)
```

11. **Optional**: Repeat question 10 for the predictions according to levels of `fTide4`, and then `fMonth`, each time fixing other variables at a value of your choice.

```
par(mfrow= c(2, 2)) # we will need 3 plots

# repeat plotting of predictions for Time6
PA10.dat4pred<- data.frame(Time6= levels(dat$Time6),
                                fMonth= "10", fTide4= "1")

PA10.pred<- predict(PA10, PA10.dat4pred, type= "link", se.fit= T)

PA10.dat4pred$fit.resp<- plogis(PA10.pred$fit)
# lower 95% CI
PA10.dat4pred$LCI<- plogis(PA10.pred$fit - 1.96*PA10.pred$se.fit)
# upper 95% CI
PA10.dat4pred$UCI<- plogis(PA10.pred$fit + 1.96*PA10.pred$se.fit)

plot(x= 1:6, y= PA10.dat4pred$fit.resp,
      pch= 16, cex= 1.4, xlab= "Section of day",
          ylab= "Fitted probability", ylim= c(0, 1),
      main= "Predictions for time of day\n(assuming Tide = 1 and Month = 10)",
      xaxt= "n") # supress automatic x axis (we will draw our own improved axis)

arrows(x0= 1:6, x1= 1:6,
          y0= PA10.dat4pred$LCI, y1= PA10.dat4pred$UCI,
          length= 0.02, angle= 90, code= 3)

axis(side= 1, at= 1:6, label= levels(dat$Time6))

# plotting of predictions for fMonth
PA10.dat4pred<- data.frame(fMonth= levels(dat$fMonth),
                              Time6 = "PM2", fTide4= "1")

PA10.pred<- predict(PA10, PA10.dat4pred, type= "link", se.fit= T)
```

```
PA10.dat4pred$fit.resp<- plogis(PA10.pred$fit)
# lower 95% CI
PA10.dat4pred$LCI<- plogis(PA10.pred$fit - 1.96*PA10.pred$se.fit)
# upper 95% CI
PA10.dat4pred$UCI<- plogis(PA10.pred$fit + 1.96*PA10.pred$se.fit)

plot(x= 1:12, PA10.dat4pred$fit.resp,
      pch= 16, cex= 1.4, xlab= "Month",
          ylab= "Fitted probability", ylim= c(0, 1),
      main= "Predictions per month\n(assuming Time = PM2 and Tide = 1)")

arrows(x0= 1:12, x1= 1:12,
          y0= PA10.dat4pred$LCI, y1= PA10.dat4pred$UCI,
          length= 0.02, angle= 90, code= 3)

# plotting of predictions for fTide4
PA10.dat4pred<- data.frame(fTide4= levels(dat$fTide4),
                           Time6 = "PM2", fMonth= "10")

PA10.pred<- predict(PA10, PA10.dat4pred, type= "link", se.fit= T)

PA10.dat4pred$fit.resp<- plogis(PA10.pred$fit)
# lower 95% CI
PA10.dat4pred$LCI<- plogis(PA10.pred$fit - 1.96*PA10.pred$se.fit)
# upper 95% CI
PA10.dat4pred$UCI<- plogis(PA10.pred$fit + 1.96*PA10.pred$se.fit)

plot(1:4, PA10.dat4pred$fit.resp,
      pch= 16, cex= 1.4, xlab= "Tidal phase",
          ylab= "Fitted probability", ylim= c(0, 1),
      main= "Predictions for tide\n(assuming Time = PM2 and Month = 10)",
      xaxt= "n") # supress x axis (we will draw our own)

axis(side= 1, at= 1:4, label= levels(dat$fTide4))

arrows(x0= 1:4, x1= 1:4,
          y0= PA10.dat4pred$LCI, y1= PA10.dat4pred$UCI,
          length= 0.02, angle= 90, code= 3)
```

12. How satisfied are you with the model, and with all the assumptions being met? What have you learned from it, with respect to the initial aims of the study? Are there areas of improvement?

13. **If you would like to go further**: Re-fit the model with interactions between all categorical predictors, two by two. What hypotheses do these interactions correspond to?

14. **If you would like to go further**: Perform model selection "by hand", using the AIC from the model summary, or using `AIC(YourModel)`, and construct an AIC table. There are 18 possible nested models in

total, including the full model above. The list of models to evaluate is up to you, from a fully exploratory approach using all 18 models, to a more targeted list of models based on your specific research questions or predictions.

15. **If you would like to go further**: Perform model validation

16. **If you would like to go even further**: Interpret the model biologically.

End of the Binomial (Bernoulli) GLM - dolphin behavioural plasticity exercise

For info, background on the data and the study can be found in this short video, courtesy of Paul Thompson. The exercise can be done entirely without consulting this. I recommend you watch this or any companion material (the referenced paper) outside the synchronous session, to make the most of the time you have with demonstrators to progress on the exercises.

For info, the publication here offers a different approach to analysing these data, using slightly fancier GLMs with smooth terms (called GAMs, for Generalized Additive Models), and a few additional refinements: [https://www.nature.com/articles/s41598-019-38900-4]. What assumptions differ between this and your approach?

For info, the full original data (10 Mb) are publicly available here: [https://datadryad.org/stash/dataset/doi:10.5061/dryad.k378542], in case of interest.