Exercise

Exercise: Binomial (Bernoulli) GLM - dolphin behavioural plasticity

1. The data for this exercise were collected by the Cromarty Lighthouse team between 2010 and 2016, using underwater sound recorders (CPOD) to continuously monitor the pattern of presence and foraging behaviour of bottlenose dolphins at key sites in the Moray Firth. Variables:

- `X` index of the observations
- `presence`: 0 for absence, 1 for presence
- `year`
- `julianday`: day of the year
- `tideangle_deg`: tidal state‘
- `mh`: hour of the day (integer)
- `mon`: month (integer)
- `Per2`: Bin year into two periods (May+June vs rest of year)
- `Per4`: 3 periods of 20 days from early May to end of June vs rest of the year
- `Time6`: Bin time of day into 6 4h periods (first centered on midnight)
- `Tide4`: Bin tide angle into 4 quadrants with peaks 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. For example, **sightings at Sutors are thought to be more frequent around May-June, although the odds of detecting dolphins may depend on various factors, including tidal state and time of day**. 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 (particularly May/June vs. rest of the year), with possible interactions between them.

The data have been aggregated as presence/absence at a 1h resolution. You will focus on one of the sites, “Sutors”, a subset which will leave you with just under 50000 presence/absence records to play with. 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.

Background on the data and the study can be found here, 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.

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 full data file 'dolphin.csv' into R (50000 records) by running the following chunk of code (please unfold and copy/paste). The code will take a reproducible random subset of 5000 observations which is well enough to give similar results as the full one, without working your computer too hard (graphs are what it could struggle with mostly).

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

# re-ordering factor levels for convenience:
dat$Per2<- factor(dat$Per2, levels= c("RestOfYear", "MayJun"))
# (making "RestOfYear" the reference level)
dat$Per4<- factor(dat$Per4, levels= c("RestOfYear", "MayJun1", "MayJun2", "MayJun3"))
dat$Time6<- factor(dat$Time6, levels= c("MNight", "AM1", "AM2", "MDay", "PM1", "PM2")) # reordering chr

str(dat)
```

```
## 'data.frame':    47335 obs. of  11 variables:
##  $ X            : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ presence     : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ year         : int  2010 2010 2010 2010 2010 2010 2010 2010 2010 2010 ...
##  $ julianday    : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ tideangle_deg: int  354 23 52 81 109 138 167 196 225 253 ...
##  $ mh           : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ mon          : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Per2         : Factor w/ 2 levels "RestOfYear","MayJun": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Per4         : Factor w/ 4 levels "RestOfYear","MayJun1",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Time6        : Factor w/ 6 levels "MNight","AM1",..: 1 1 2 2 2 2 3 3 3 3 ...
##  $ Tide4        : int  1 1 2 2 2 3 3 3 3 4 ...
```

The code for converting the original publicly available data into the 'dolphin.csv' file is given at the end of the practical. It includes converting the original numeric variables into categories that you can specify (binning). Binning is done easily using the `cut()` function (examples in the code chunk at the end, if you want to create your own categories).

3. Take a look at the structure of this dataframe. Start with an initial data exploration to look at any imbalance between the predictors, and factors affecting presence of dolphins. Which ones are continuous or categorical? Which ones would your intuition guide you to use for data exploration? For modelling? Hints:

- Presence/absence data (Bernoulli) are more difficult to explore.

- One approach 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 proportion of time present, you could calculate mean presence per category `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, `matplot(tapply(dat$presence, list(dat$Group1, dat$Group2), mean), type= "l", ylim= c(0, 1), xlab= "Group1", ylab= "presence", lty= 1)` produces one line per Group2.

4. Let's start with a Binomial (Bernoulli) GLM (using `glm()` and the appropriate `family` argument) with all interactions between numerical time of day, tide angle and day of the year as predictors: `tideangle_deg * mh * julianday`. Which individual interactions are implied in this formula? (Hint: if unsure, the summary of the model at the next question will list them).

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?

6. Are all the terms significant? if not, simplify the model. Remember to choose the correct ANOVA method (sequential or not), and the appropriate test.

7. Let's now validate the model, using deviance residuals. The easiest tool is the `binnedplot()` in the `arm` package, if you can. If not, use the DIY code chunk below.

```
library(car)
vif(PA4)
# High for terms involved in the interaction, as expected. No concern.

par(mfrow= c(2, 2))
plot(PA4, col= dat$presence + 1) # red is presence, black is absence
# Not very useful statistical art. Not worth framing either.
```

```
# plot against predictors:
res4.p<- resid(PA4, type= "pearson")

par(mfrow= c(2, 2))
plot(res4.p ~ dat$tideangle_deg, col= dat$presence + 1)

plot(res4.p ~ dat$mh, col= dat$presence + 1)

plot(res4.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$tideangle_deg, y= res4.p, xlab= "Tide angle", nclass= 100)
binnedplot(x= dat$mh, y= res4.p, xlab= "hour")
binnedplot(x= dat$julianday, y= res4.p, xlab= "Day of the year", nclass= 100)
```

In case you can't use `binnedplot`, here is a home-made version:

```
par(mfrow= c(2, 2))
plot(res4.p ~ dat$tideangle_deg, col= dat$presence + 1)
tide.means<- tapply(res4.p, list(dat$tideangle_deg), mean)
tide.vals<- as.numeric(names(tide.means))
lines(tide.means ~ tide.vals, col= 3)
abline(h= 0, lty= 3, col= grey(0.5))

plot(res4.p ~ dat$mh, col= dat$presence + 1)
hour.means<- tapply(res4.p, list(dat$mh), mean)
lines(hour.means ~ as.numeric(names(hour.means)), col= 3)
abline(h= 0, lty= 3, col= grey(0.5))

plot(res4.p ~ dat$julianday, col= dat$presence + 1)
day.means<- tapply(res4.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.
```

8. 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 a solution, before unfolding 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.

```
# 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.

# Each of the predictors we started with already has one or more
# categorical counterpart in the data set.
# I suggest you try Tide4 * Per2 * Time6
# You can choose something else or cut your own predictors, too.
```

9. Apply the usual model selection approach for the new version of the model. What are the main sources of variation in the data? What is the proportion of deviance explained?

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

11. Assuming that the model is fine as it is, let's plot the predictions with their confidence intervals for the probability of presence in relation to time of day, in both the May/June period and the rest of the year. For tide, assume it is fixed at a level of your choice, e.g. "1". Suggested approach:

- create a `data.frame` called `X` containing the data to predict for. This can be done by hand following previous examples or using the function `expand.grid` for creating all the combinations of the variables of interest: `expand.grid(NameOfVar1 = levels(data$NameOfVar1), NameOfVar2= levels(data$NameOfVar2), NameOfVar3= "1"))`
- 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`.
- plot fitted values, extracted using `Z$fit`, against the appropriate column of X (you can use different symbols or colours for groups).
- in X, add columns for the fitted values and their confidence intervals, on the response scale (to be calculated).
- use the function `segments` or `arrows` to add confidence intervals to the fitted values (see the help page for the respective function).

The code is available below for you to unfold if you don't want to try yourself.

```
PA13.dat4pred<- expand.grid(Time6= levels(dat$Time6),
                            Per2= levels(dat$Per2),
                            fTide4= "1")
```

```
PA13.pred<- predict(PA13, PA13.dat4pred, type= "link", se.fit= T)

PA13.dat4pred$fit.resp<- exp(PA13.pred$fit)/(1+exp(PA13.pred$fit))
# or plogis(PA13.pred$fit)

# lower 95% CI
PA13.dat4pred$LCI<- plogis(PA13.pred$fit - 1.96*PA13.pred$se.fit)
# upper 95% CI
PA13.dat4pred$UCI<- plogis(PA13.pred$fit + 1.96*PA13.pred$se.fit)
```

12. Repeat the model selection using AIC, with `step()`. Do you obtain the same minimal adequate model? Then replace `Per2` by month `mon` (as a factor) for a finer seasonal resolution, and apply a model selection with `step()` again. Is the same model structure preferred? Which of the `Per2` or `mon` models is favoured by AIC? Do the residuals look better?

13. 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? **Optional** 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?

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

Appendix. Code for converting the original publicly available data (10 Mb) [https://datadryad.org/stash/dataset/doi:10.5061/dryad.k378542] into the 'dolphin.csv' file. Includes converting numeric variables into categories that you can define to suit your needs (binning), including making more bins if you wish. Binning is done easily using the `cut()` function. For example, creating 5 regular bins is done using `cut(MyVector, breaks= 5)`. Note here that `cut` is used in a non-standard way to make the beginning and end of a cyclic variable belong to the same bin, which may be more biologically meaningful (you can decide, you are the expert!).

```
fulldat<- read.delim("./data/FineScale_Dataset_GAMM_OFB2019.txt")

str(fulldat)

dat<- fulldat[fulldat$site == "Sutors", c("presence", "year", "julianday", "tideangle_deg", "mh")]

dat$mon<- as.numeric(cut(dat$julianday, seq(1, 370, by= 30.5)))
```

```r
dat$tideangle_deg<- round(dat$tideangle_deg)
# count number of data per year/month combination and represent as mosaicplot
plot(table(dat$year, dat$mon))

# remove 2016
dat<- dat[dat$year != 2016, ]

# Bin year into two periods (May+June vs rest of year)
dat$Per2<- cut(dat$julianday, breaks= c(-1, 120, 180, 400),
            labels= c("RestOfYear", "MayJun", "RestOfYear"))
dat$Per2<- factor(dat$Per2, levels= c("RestOfYear", "MayJun"))
# (making "RestOfYear" the reference level)

# check this is working as intended:
plot(as.numeric(dat$Per2) ~ dat$julianday)

# Bin year into 4 periods:
# 3 periods of 20 days from early May to end of June vs rest of the year
dat$Per4<- cut(dat$julianday, breaks= c(0, 120, 140, 160, 180, 400),
            labels= c("RestOfYear", "MayJun1", "MayJun2", "MayJun3", "RestOfYear"))
dat$Per4<- factor(dat$Per4, levels= c("RestOfYear", "MayJun1", "MayJun2", "MayJun3"))
# (reordering levels)

# check this is working as intended:
plot(as.numeric(dat$Per4) ~ dat$julianday)

# Bin time of day into 6 4h periods (first centered on midnight)
dat$Time6<- cut(dat$mh, breaks= c(-1, seq(2, 22, by= 4), 24),
            labels= c("MNight", "AM1", "AM2", "MDay", "PM1", "PM2", "MNight"))
dat$Time6<- factor(dat$Time6, levels= c("MNight", "AM1", "AM2", "MDay", "PM1", "PM2")) # reordering chr

# check this is working as intended:
table(dat$Time6, dat$mh)

# Bin tide angle into 4 quadrants with peaks in middle of respective bin
dat$Tide4<- cut(dat$tideangle_deg, breaks= c(-1, 45, 135, 225, 315, 360),
            labels= c(1:4, 1))

# check this is working as intended:
plot(as.numeric(dat$Tide4) ~ dat$tideangle_deg)

# unless you desperately want to test the performance of your computer,
# play safe and reduce the size of the data set from 50000 to 5000:
set.seed(74) # makes the random sampling reproducible
# This means you will get the same random sample as the solutions to
# the exercises and the same results.
dat<- dat[sample(1:nrow(dat), size= 5000), ] # random subset or rows

write.csv(dat, "dolphin.csv")
```