GLM Assignment

You will need to use either data you have collected from your thesis, data from Dryad, or some other biological dataset from the internet. You will put together two worked examples showing the typical use of GLMs in data analysis for data that cannot be analyzed use traditional ANOVA/OLS techniques.

Examples can have categorical predictors, continuous predictors, or both (as relevant). Each should include relevant data visualizations and explanation. You will be responsible for presenting and interpreting your analyses in-class using .html files from a Github repository containing your analyses (one or more .html reports as necessary). Each person should limit their presentation to 5-6 minutes.

Table

Description automatically generated

Additionally, you should organize data, files, visualizations, objects, etc. using the ‘best-practices’ we discussed early in the semester.

1. Condense loose files into sub-folders to avoid cluttering up the root directory containing the .Rproj file. (e.g. data, docs, figures, tables, reference\_info, etc.)
2. Name files intuitively with appropriate structure and organization (e.g. 01\_data\_winter, 02\_data\_summer, etc.)
3. Use intuitive object names (e.g. model\_1\_binomial, figure\_4\_seasonal, table\_3\_cort\_baseline)
4. Use library(janitor) to help clean dirty data and enforce a consistent format.
5. The entire workflow should be repeatable; if we were to fork your repository, we should only have to run the .Rmd file to replicate all steps in the analysis and visualization.

Potential datasets. These are relatively ‘clean’ and do not require much pre-processing

data() #lists datasets from various installed packages; good for practice

data(package = .packages(all.available = TRUE)) #lists the data sets in all \*available\* packages.

install.packages(“datasets”) # package that has tons of practice data

library(datasets)

data(package="datasets")

# I prefer this package because the datasets tend to have more variables

install.packages("dslabs")

library(dslabs)

data(package="dslabs")

Tips for Success:

1. If you are going to use a **binomial** error structure, your response variable should be binary (presence/absence, yes/no, species1/species2, etc.) Ideally, (but not necessarily) there would not be complete overlap in the ranges of the binary groups; some overlap is fine. Test this by making boxplots showing the distribution of the response variable between the two groups.
2. If you are going to use a **Poisson** error structure, your response variable should be counts (i.e. you can put counts into bins of a histogram; whole numbers, not continuous data). Test by making a histogram of the counts.
3. If you have continuous data for both predictor and response variables, use **boxcox** transform to suggest which error structure would be best.