Coursework Assignment

2023-10-09

QUESTION 01: Data Visualisation for Science Communication

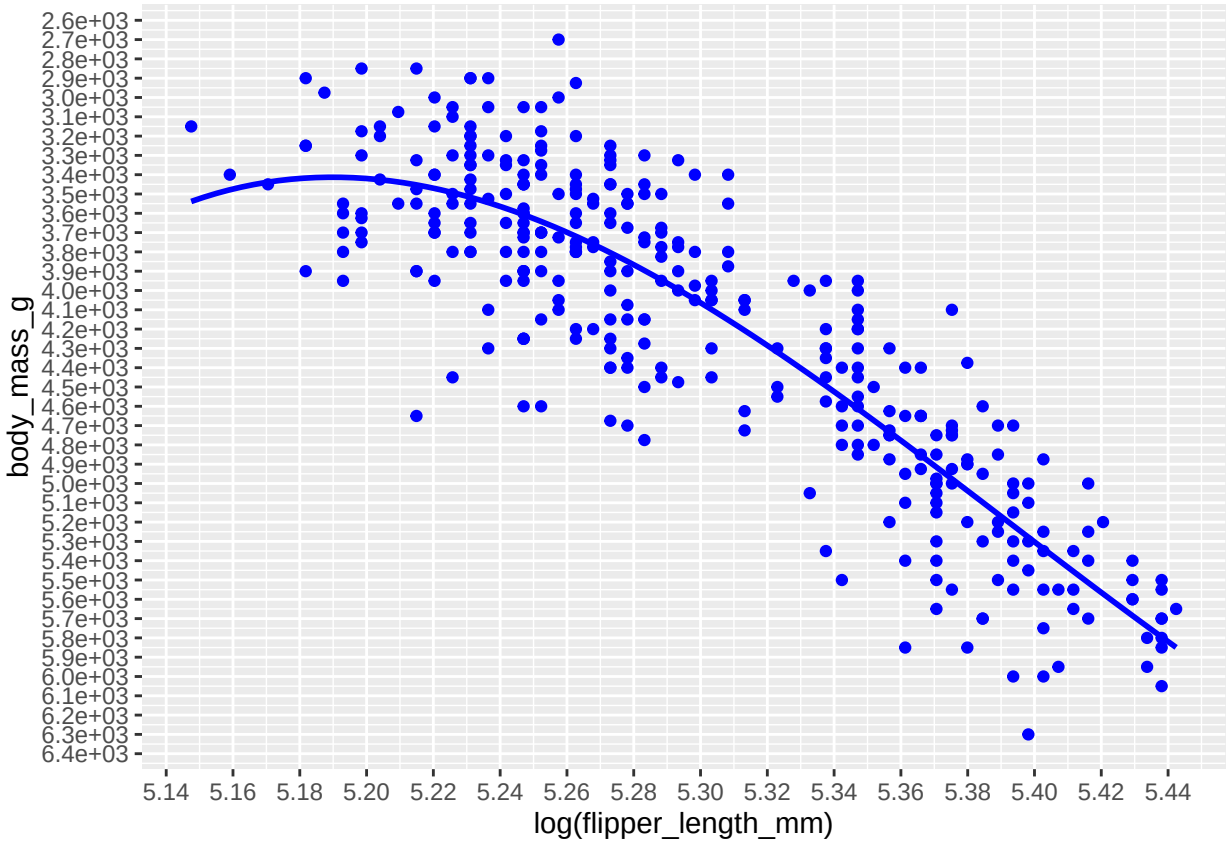
Create a figure using the Palmer Penguin dataset that is correct but badly communicates the data. ***Do not make a boxplot.***

Use the following references to guide you:

- *https://www.nature.com/articles/533452a*
- *https://elifesciences.org/articles/16800*

Note: Focus on visual elements rather than writing misleading text on it.

a) Provide your figure here:



b) Write about how your design choices mislead the reader about the underlying data (200-300 words).

The design of this graph is misleading for a variety of reasons, firstly the trend line is the same colour as the data points obscuring some of them. This makes it harder to read the actual points on the graph. The

trend line is also over fitted to a polynomial formula implying a greater significance to the relationship than is honestly shown in the data. Moreover, the x axis is needlessly placed on a log scale compressing the data points and making the real world values harder to see. The x axis also has too many ticks and guidelines on it, many of which do not align with the data, this makes the graph much busier and seeing the values more difficult. Similarly, the y axis also contains too many ticks, and all the values are needlessly written using scientific notation reducing the ability of the graph to be understood intuitively. This can also have the effect of making the data seem more spaced out than it is. The final confusing factor is that the y axis is plotted upside down with the smallest values at the top of the graph. This has the effect of making it intuitively seem as if penguins with longer flippers weight less when in reality the reverse is true.

Tips on How to Display Data Badly

Exploring Different Smooths

Scales for Continuous Data

QUESTION 2: Data Pipeline

Write a data analysis pipeline in your .rmd RMarkdown file. You should be aiming to write a clear explanation of the steps, the figures visible, as well as clear code.

Your code should include the steps practiced in the lab session:

- *Load the data*
- *Appropriately clean the data*
- *Create an Exploratory Figure (**not a boxplot**)*
- *Save the figure*
- ***New**: Run a statistical test*
- ***New**: Create a Results Figure*
- *Save the figure*

An exploratory figure shows raw data, such as the distribution of the data. A results figure demonstrates the stats method chosen, and includes the results of the stats test.

Between your code, communicate clearly what you are doing and why.

Your text should include:

- *Introduction*
- *Hypothesis*
- *Stats Method*
- *Results*
- *Discussion*
- *Conclusion*

You will be marked on the following:

a) Your code for readability and functionality

b) Your figures for communication

c) Your text communication of your analysis

Below is a template you can use.

Introduction

To begin the analysis the common functions for cleaning and plotting must be sourced so they are accessible in future code blocks.

```
source("functions/Cleaning.r")
source("functions/Plotting.r")
```

Next a copy of the original raw data is saved so it can be accessed "as is" if it is needed for future studies or to validate this analysis.

```
write_csv(penguins_raw, "data/penguins_raw.csv")
```

Next the raw penguin data is cleaned in a variety of ways to make it more intuitive and human readable

- Edits the column names to make them lower snake_case
- Shorten the species names so they can be access with one word, in a consistent format, without knowing the full genus and species
- Removes columns and rows which lack data
- Removes the comment column as it is not useful for analysis and therefore simplifies the dataset

Finally this block saves the cleaned data so that is can be accessed and used in other analysis

```
penguins_clean <- penguins_raw %>%
  clean_column_names() %>%
  shorten_species() %>%
  remove_empty_columns_rows() %>%
  remove_comments_row()

write_csv(penguins_clean, "data/penguins_clean.csv")
```

Hypothesis

My hypothesis that penguin body mass can be used to predict penguin culmen depth. My alternative hypothesis is that there is no correlation between body mass and culmen depth

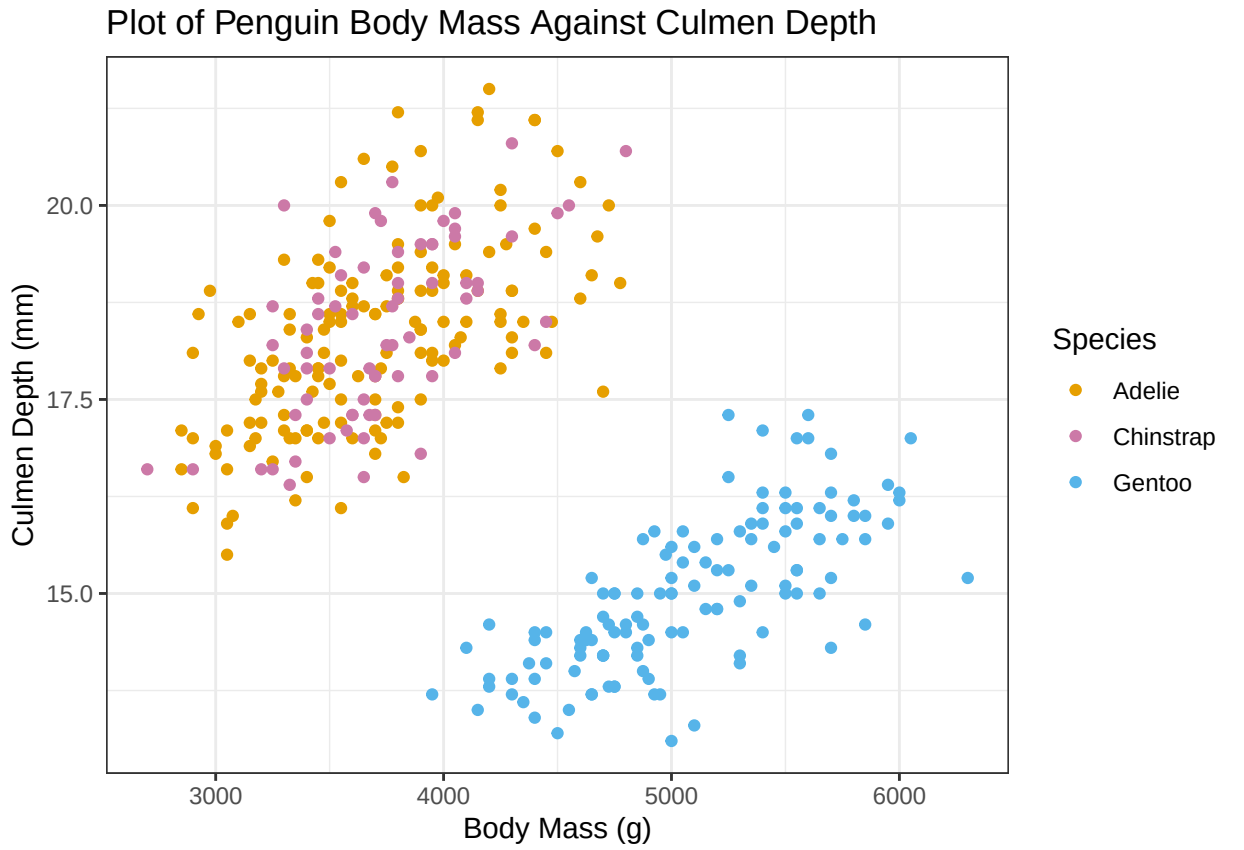
Statistical Methods

To test this hypothesis, it is important to separate out the section of clean data which is of interest to the study. Any cells with NA values are no longer required and so are removed from this hypothesis testing dataset

```
penguins_hypothesis <- penguins_clean %>%
  subset_columns(c("species", "culmen_depth_mm", "body_mass_g")) %>%
  remove_NA() %>%
  group_by_species()
```

To visually see if their is any interaction between culmen depth and body mass it is important to firstly make an exploratory plot before attempting to test for significance. This plot is saved and can be found in the figures folder.

```
exploratory_culmen_mass_plot <- penguins_hypothesis %>%
  plot_exploratory_culmen_mass()
exploratory_culmen_mass_plot
```



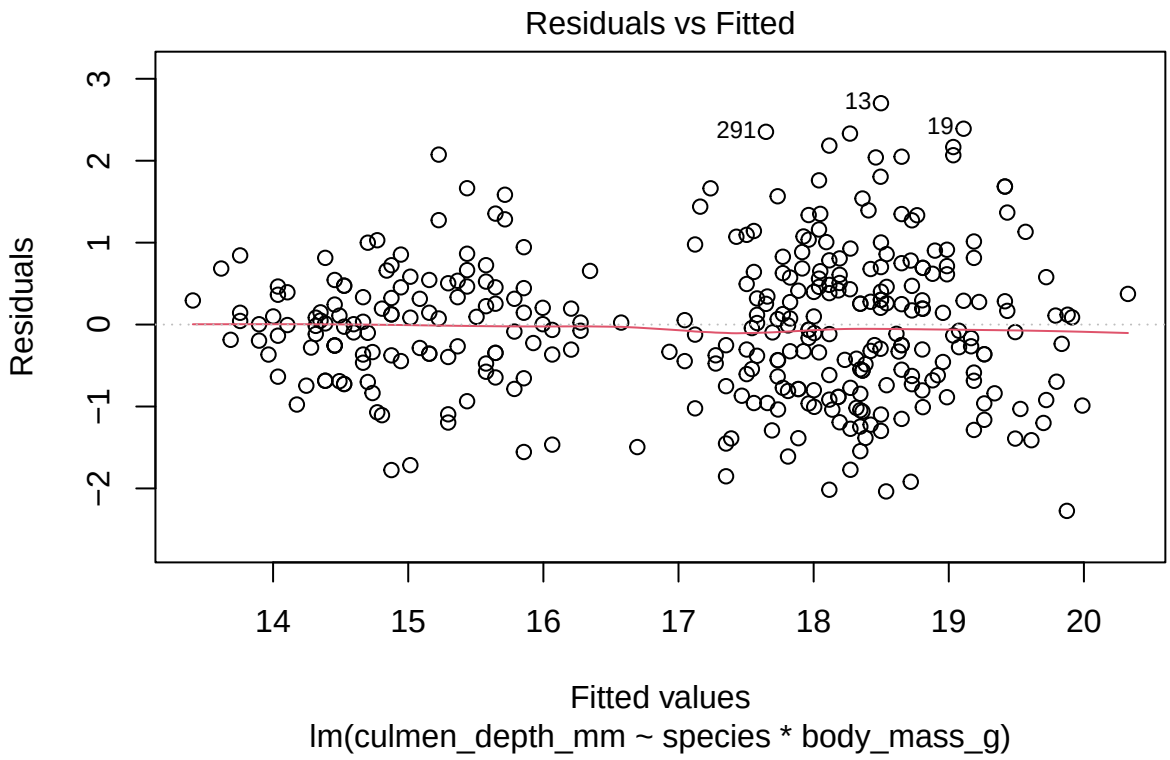

```
save_plot_svg(
  figure = exploratory_culmen_mass_plot,
  filename = "figures/exploratory_culmen_mass_plot.svg",
  size_cm = 15,
  scaling = 1
)
```

```
## pdf
##   2
```

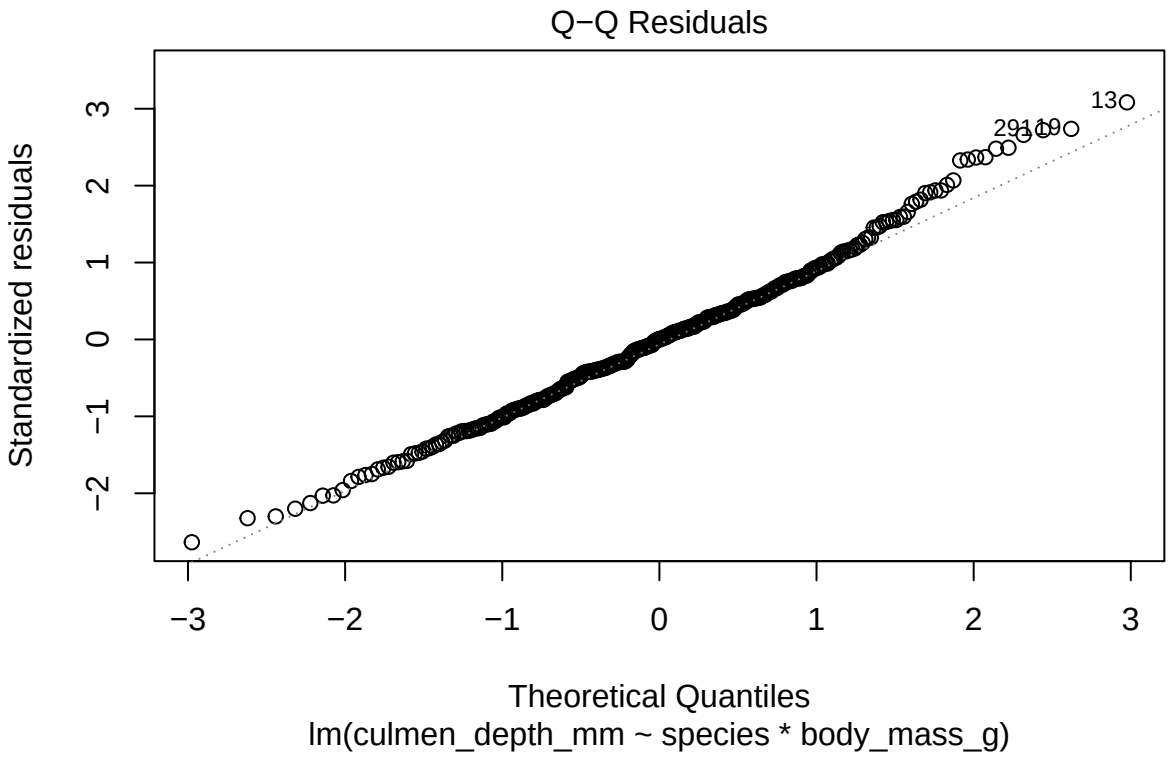
This graph shows that as penguin body mass increases culmen depth decreases. However, the grouping of the species within the graph suggests that the selected trend is an example of Simpson's paradox where the overall trend doesn't match the trend within each species. This graph suggests that within species as body mass increases so does culmen depth. Therefore, I will use an analysis of covarience (ANCOVA) to test if there is any significant differences caused by the species interaction before analysing individually.

To run an ANCOVA a model must be generated and the data checked to make sure it meats the assumptions of an ANCOVA- the data must be sampled randomly and be normally distributed. It is assumed that the dataset was collected correctly with random sampling therefore the code will only check for normality in the model.

```r
culmen_interaction_mod <- lm(culmen_depth_mm ~ species * body_mass_g, penguins_hypothesis)
plot(culmen_interaction_mod, which = 1)
```




```r
plot(culmen_interaction_mod, which = 2)
```



These plots show that the data is normally distributed without any transformations, therefore we can proceed with the analysis

```
anova(culmen_interaction_mod)
```

```
## Analysis of Variance Table
##
## Response: culmen_depth_mm
##                      Df Sum Sq Mean Sq  F value Pr(>F)
## species               2 903.97  451.98 584.3999 <2e-16 ***
## body_mass_g           1 164.86  164.86 213.1570 <2e-16 ***
## species:body_mass_g   2   1.14    0.57   0.7377  0.479
## Residuals           336 259.87    0.77
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The ANCOVA shows that the interaction between species and body mass is not significant- the increase in culmen length due to body mass is not different between species. This means the model can be simplified to not include the interaction. Due to the Simpson's paradox effect shown in the exploratory plot I will only analyse the trend for one species, and as the interaction is not significant as shown by the ANCOVA it can be assumed that this slope is the same for all species.

```
penguins_adelie_hypothesis <- penguins_hypothesis %>%
  filter_by_species("Adelie")

culmen_mod <- lm(culmen_depth_mm ~ body_mass_g, penguins_adelie_hypothesis)
```

Now the simplified model has been generated the significance of it must be tested through an ANOVA

```r
anova(culmen_mod)
```

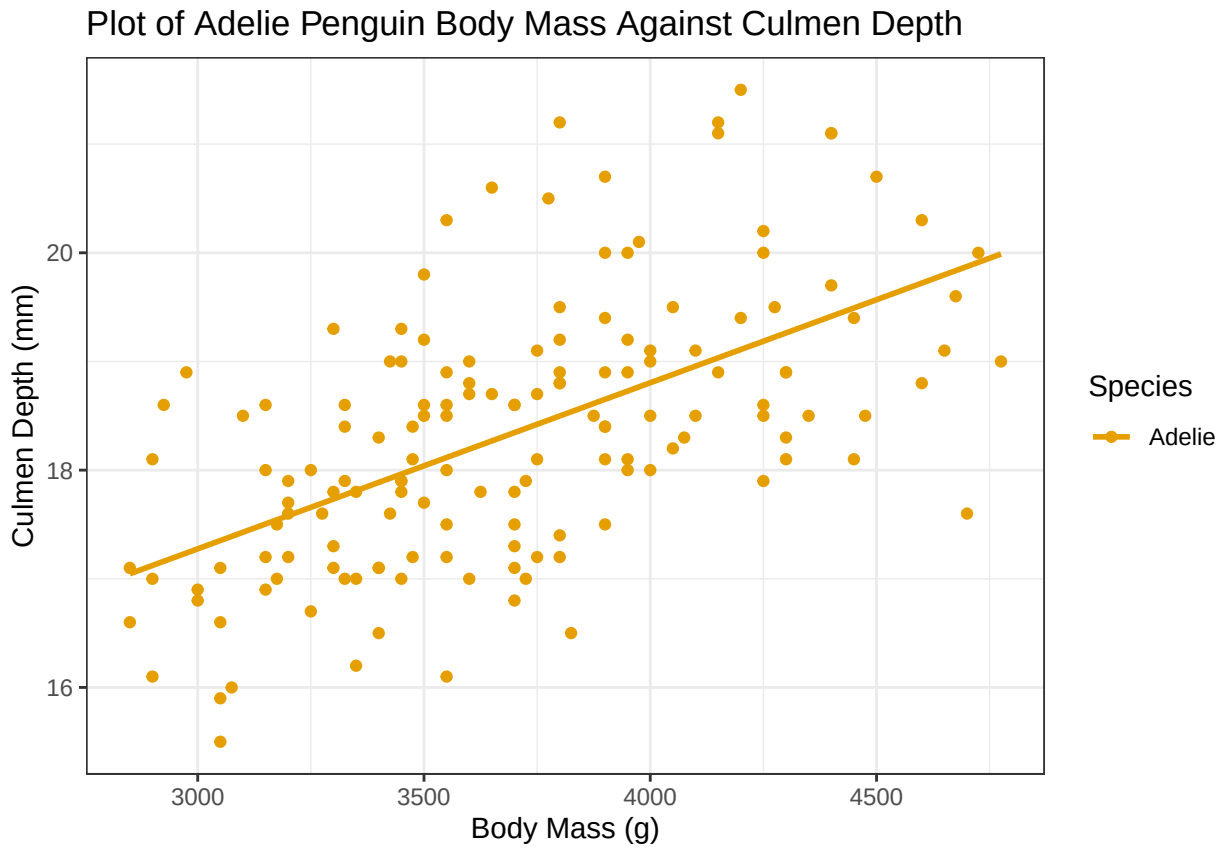
```
## Analysis of Variance Table
##
## Response: culmen_depth_mm
##              Df  Sum Sq Mean Sq F value    Pr(>F)
## body_mass_g   1  73.701  73.701  74.032 9.942e-15 ***
## Residuals   149 148.334   0.996
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Results & Discussion

The plot shows the interaction between body mass and culmen depth for just Adelie penguins as in the ANOVA test above. This plot is saved and can be found in the figures folder.

```r
adelie_plot <- penguins_adelie_hypothesis %>%
  plot_adelie_results()
adelie_plot
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```r
save_plot_svg(
  figure = adelie_plot,
  filename = "figures/adelie_plot.svg",
  size_cm = 15,
  scaling = 1
```

```
)
```

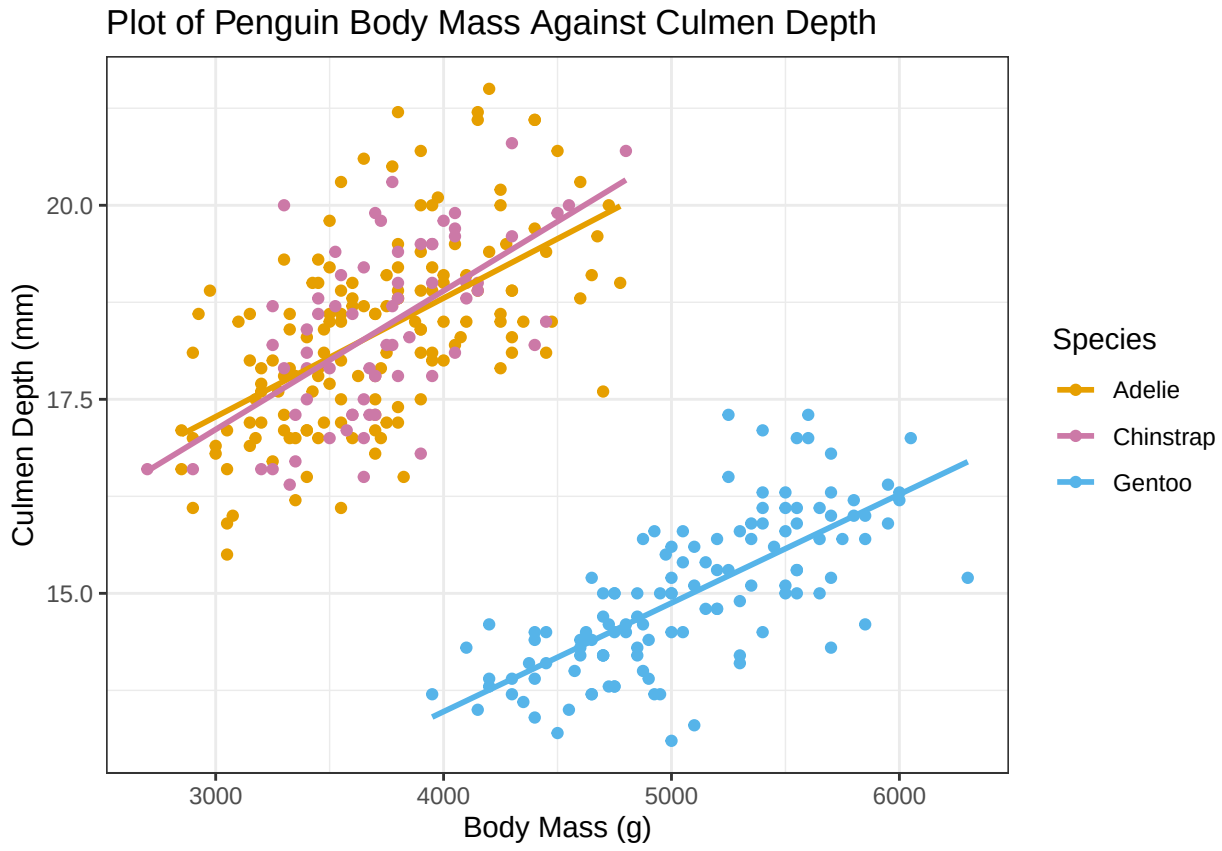
```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## pdf
##   2
```

The final results plot shows the same model method applied to all the penguin species. This plot is saved and can be found in the figures folder.

```
culmen_mass_plot <- penguins_hypothesis %>%
  plot_culmen_mass()
culmen_mass_plot
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
save_plot_svg(
  figure = culmen_mass_plot,
  filename = "figures/culmen_mass_plot.svg",
  size_cm = 15,
  scaling = 1
)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## pdf
##   2
```

Conclusion

The ANOVA shows that there is a significant correlation between penguin body mass and culmen depth. The results graph visibly demonstraights that this is a positive correlation. As the body mass of the penguin increases the culmen depth also increases. This may be due to penguins with more access to food being able to grow deeper culmens. Alternatively the interaction could be the reverse and penguins with deeper culmens are more effective foragers. This may be due to being able to access novel food sources or being more proficient at catching prey. This slope of this interaction not significantly different between the penguins as shown by the ANCOVA.

QUESTION 3: Open Science

a) GitHub

Upload your RProject you created for ***Question 2*** *and any files and subfolders used to GitHub. Do not include any identifiers such as your name. Make sure your GitHub repo is public.*

GitHub link: https://github.com/Elephant34/ReproducibleScience

You will be marked on your repo organisation and readability.

b) Share your repo with a partner, download, and try to run their data pipeline.

Partner's GitHub link: https://github.com/BioBabe2002/Reproducible-Figures-R

You ***must*** *provide this so I can verify there is no plagiarism between you and your partner.*

c) Reflect on your experience running their code. (300-500 words)

- *What elements of your partner's code helped you to understand their data pipeline?*
- *Did it run? Did you need to fix anything?*
- *What suggestions would you make for improving their code to make it more understandable or reproducible, and why?*
- *If you needed to alter your partner's figure using their code, do you think that would be easy or difficult, and why?*

My partners text clearly explains what each code segment is doing and why the step is necessary. The names of variables are well chosen, they are clearly human readable and work to quickly identify what the variable holds. The variable names are consistently formatted lower snake case with the minor exception "Clean_data" which begins with a capital. Unfortunately, I was required to manually install the package "agricolae" before the code would run. All the other packages were installed within the code itself. I also had to create a data folder as it was not included in the GitHub download, however, this was only used for Q1 not as part of the data pipeline itself. Ways the code could be improved include making more consistent use of pipes, for some segments pipes were used whereas in others they were not. Moreover, some sections were split over multiple code segments and then repeated in a pipeline when this was unnecessary. For example, line 98 (removing N/As from the penguin dataset) and line 104 (shortening the island names) are repeated within the cleaning pipeline beginning line 109. A way by which the codes readability could be improved is through separating out some processes into appropriately named functions. For example, it takes a bit of time to understand what line 112 is doing but this could be moved to a function named shorten_island_names() to make it intuitive to understand. The reproducibility of the code could be improved through setting up all the packages used in one code block at the start. This would enable at a glance seeing which packages are required and would avoid repetition of library calls within the code. As the code for the plots is within the markdown file itself, the project could be improved by separating out the plotting calls into separate functions.

There is no common theme between the plots so editing them on mass, for example to change the font, would have to be done for every plot individually. However, the plot code was well laid out over multiple lines so modifying any individual one would be simple. Overall, splitting the code into functions and transferring the functions into separate r script files would improve the projects readability and reproducibility significantly. Moreover, saving the data and graphs used into separate files would reduce the need to run the entire code every time the project is freshly loaded.

d) Reflect on your own code based on your experience with your partner's code and their review of yours. (300-500 words)

- *What improvements did they suggest, and do you agree?*
- *What did you learn about writing code for other people?*

My partner suggested a minor restructure of the order some of my functions run and also to include session and version information within the GitHub respiratory to improve reproducibility. I feel that the restructuring is a matter of personal preference. I strongly agree that I should have included session and version information to maximise reproducibility. The most simple way to complete this would be through use of an renv. However, I did not do this because I was unable to install any packages with a renv active. I did attempt a work around by initialising renv after my packages were already installed. This method appeared to work, but as I was unable to test it, I did not upload the result. From this exercise I realised the importance of saving the data as you modify it so it can be quickly referred back to later. In my code I should have saved my penguin_hypothesis dataset to make it easier to come back later and change the stats without having to rerun the cleaning code. I also noticed the importance of function and variable naming conventions and how they influence your code. I realised for me, the best practise was to structure functions has noun_verb (e.g. clean_data) and the output variable as verb_noun (data_clean). When applied consistently this avoids ambiguity as to which name is a function and which is a variable. However, as I did not explicitly state the conventions I used, future modifications to the project are at risk of not following them. This showed me the importance of properly documenting your code and styling conventions, including file, variable, and function naming. One of the most challenging things about coding for other people is knowing the number and detail of comments to use. Too few can lead to confusion as to the codes purpose but too many clutters the code and can reduce readability. This is why splitting code into appropriately named functions is so important. The pipeline can be explained in a single comment and if a reader wants to understand a specific function, they can look at the comments in that without being overwhelmed in the main file.