HW 04: Data Visualization

Graphical Analysis of Biological Data

By the end of this assignment, you should demonstrate an ability to

- explain the principles of the layered grammar of graphics;
- produce graphs in `ggplot2`, including boxplots, scatterplots, and line graphs.

Click on any blue text to visit the external website.

This assignment has two parts. The first part in particular is long so allow plenty of time.

Note: If you contact me for help or (better yet) open an issue in the public discussion forum, please include the code that is not working and also tell me what you have tried.

Preparation

- Open your `.Rproj` project file in RStudio.

  This was a common problem early on in the course. The R Project file tells RStudio that you are operating under Git version control. **Open your project file every time you start to do an assignment for this course.**

  Project files do much more than link RStudio to Git. You only need one `.Rproj` file for this course but, in general, you make a new project file for each study that you do.

- Create an `hw04` folder inside the same folder as your project file.

  This was a less common error but one that did pop up. The folder you created for this course is tracked by Git. It is your local repository, or repo. Git will be aware of any files or folders you place inside your local repo but will be oblivious to anything outside your repo. If you put the `hw04` folder inside your local repo, and your homework files inside your `hw04` folder, then Git will be aware of all of them.

Part 1: Intro to ggplot2

- Download this R Notebook file. Right-click to save the file as `<lastname>_hw04a.Rmd` inside the "hw04" folder you just made. Do I need to tell you to use your actual last name?
- Read R4ds Chapter 3: Data Visualisation.
- As you read the chapter, insert a code chunk in your notebook and recreate *every* example in the chapter, except for those I tell you in the notebook to skip. *Type the code!* In part 2, you will have to type the code from scratch so typing the code will help you remember it better. The point here is *learn* how to do this!
- Answer the questions in each section, except where I tell you to skip them. You will have to enter code chunks to answer the quwstions and also to write text outside of the code chunks. In other words, you will take advantage of R Notebook features.
- For each code chunk you add, write a brief description of what that chunk does. That includes examples and questions. Writing these descriptions will help you learn and understand the code.
- Some answers will require you to run code and enter text. Other questions will require only one or the other. The questions should give you the context you need.

- **Stage and commit regularly!** A good habit would be to stage and commit after you complete each section. You should also push each time but at least push your completed notebook to your remote repo.

Part 2: Graph some biological data

Do this *only* after you have completed Part 1, or it will not make sense.

- If you have taken a break, make sure your Rproj file is open.
- Create a new R Notebook called `<lastname>_hw04b.Rmd` and save it in your `hw04` folder.
- Copy and paste the YAML header from Part 1 and replace the default header in your new document. Change the title as appopriate.
- Load the `tidyverse` package in your first code chunk.

Develop this habit for the remaining assignments: Open your Rproj file, download or create your new notebook, edit the YAML file, and then insert your first code chunk where you will load any packages needed for the assignment.

For Part 2, you will use `ggplot2` to make plots from some of the datasets that come with R and the `tidyverse` packages. I will give you the dataset to use, and other information to use for mapping, etc. I expect you will write and execute the code.

- The first time you use a dataset, load it with the command `data(dataset name)` in your code chunk. For example, `data(faithful)` loads the Old Faithful dataset. Technically, you do not have to do this but it is good coding practice. **I expect that you will do this.**
- After you load the dataset, and only for the first time, enter `?<dataset name>` in your code chunk to see the format of the data. For example, `?faithful` will give you information about the Old Faithful dataset.

  Note: You should *always* inspect your data visually. That is why I am asking you to do this step.

  You only need to do these two steps the first time you use a dataset.
- Run each code chunk, and write 1-2 sentences that describes any trends or patterns that you observe in the plot. In other words, think like a scientist!
- Include the `#### Plot <no.>` header above each plot.

Plot 1

- Plot type: scatterplot
- Dataset: trees
- x-axis: height
- y-axis: girth

Plot 2

Apply some of your skills that you learned during Assignment 02. You will make two vectors, then combine them into a data frame for plotting. Review the assignments if necessary.

- Make a vector called `year` for 1821 to 1934. Rememeber how to use `:` to make a sequence of numbers?

- Look at the `class()` of the `lynx` dataset. The `lynx` dataset is a "time series" class (`ts`). You can convert the time series data to a vector by using the `as.vector()` function. Just put the dataset name inside the parentheses. Assign this to the variable `pelts`.
- Make a dataframe called `lynx_pelts` from these two vectors.
- Plot type: linegraph
- Dataset: lynx_pelts (see above)
- x-axis: year
- y-axis: pelts
- Make the line color maroon. Maroon is one of the default R colors.

Plot 3

- Plot type: scatterplot
- Dataset: iris
- x-axis: petal length
- y-axis: petal width
- Point color and shape by species. You do not have to use fillable shapes.
- Point size of 2
- Add a `labs` layer to change the x- and y-axis labels so that they do not have periods in the names (i.e., `Petal Length`, `Petal Width`).

Plots 4 and 5

- This requires two code chunks, which will be nearly identical
- Plot type: Violin plot (look up `geom_violin`)
- Dataset: msleep
- x-axis: vore
- y-axis: sleep_rem
- fill = grayXX where XX is either 30 or 70.
- In your description, describe in your own words what violin plots display (you can search the interwebs), and what is the difference among the two versions of gray shading. *Hint:* the grays extend from `gray0` to `gray100`. You can learn more about colors in R from this PDF file.

Plot 7

- Plot type: boxplot
- Dataset: msleep
- x-axis: order
- y-axis: sleep_total
- use `coord_flip()`

Plot 8

- Plot type: boxplot with points
- Dataset: msleep
- x-axis: conservation
- y-axis: awake

- You must have a boxplot layer, a point layer, and a jitter layer.
- color by conservation status.
- You *may* use `coord_flip` but it is not required. Try both and choose the one you think looks best.
- Add a `lab` layer to change both axis labels so each starts with an upper-case letter.
- Search the internet to find out how to change the legend title to `Conservation`. Make that change. (Do not try to change the actual legend entries like "cd" and "vu"). **Note:** This can be done a couple of different ways but `scale_color_discrete()` is one good way.

Plots 9 and 10.

- Make two scatterplots of your choice, with the following constraints.
- One should plot any one of the "sleep" or "awake" variables against body weight. The other should plot any one of the "sleep" or "awake" variables against brain weight.
- In each plot, color the points by one of the nominal data categories. Use a different category for each plot.
- Apply `facet_wrap()` to at least one of the plots using one of the nominal variables. You decide whether you use 2 or 3 columns. *Hint:* use one of the nominal variables with relatively few different types for wrapping. *Explore:* What happens if you use a nominal variable like `genus`, with lots of types?
- You should try a few versions of each graph until you find combinations that you think show some clear trends.
- Describe the patterns or trends you see in each graph.