Project 4: ggplot2 + Collect Your Own Data

In this project you will first experiment with more complex data visualizations using the R library ggplot2. Then you will collect your own data on a topic of your choice and compare it to a larger pre-existing dataset.

General:

You will work through the project at your own pace. The project will often ask you to do things that you do not know how to do. Do not worry, this is intentional. Any time you feel stuck, ask a TA or the instructor for help. Alternatively, use Google/YouTube/Canvas to find resources to help you.

Instructions:

1.) In project 3, Replit provided us with a very organized space to write and execute R code online. The problem with Replit however is that it does not give us the capability to use R libraries. Libraries are extensions of programming languages that must be downloaded and then imported into individual projects or scripts for use. In this project, we are going to use a very powerful R library called ggplot2 to make even more advanced data visualizations. As a result, we are going to use Posit Cloud for these next two projects.

2. Click on the hyperlink above to go to Posit Cloud. Then click the **Sign-Up** tab in the top right of the screen. From here select the **Cloud Free** Plan. Click the large blue **Sign-Up** button. From here you can sign up with any email you would like (either personal or school). After entering your information and clicking **Sign-Up**, Posit Cloud will send you a confirmation email to whatever email you made the account under. Go to this email to finalize the creation of your account.

3. Once you are in the main page of Posit Cloud, click **New Space** on the right side of the screen. Name your workspace “Data Analytics Summer 2023”. From here click **New Project** on the right side of the screen, and then **New RStudio Project**. Name your project “Project 4”.

4. On the main screen should now be the R console. Click File -> New File -> R Script. Name it “main.r”. This script is where you will write the code and comments for this project.

5. In the R console below (not in the main.r script you just created), copy and run the following line of code: `packages <- c("tidyverse", "ggridges", "patchwork", "viridis", "hrbrthemes", "gapminder")`

Now run this line of code: `install.packages(packages)`

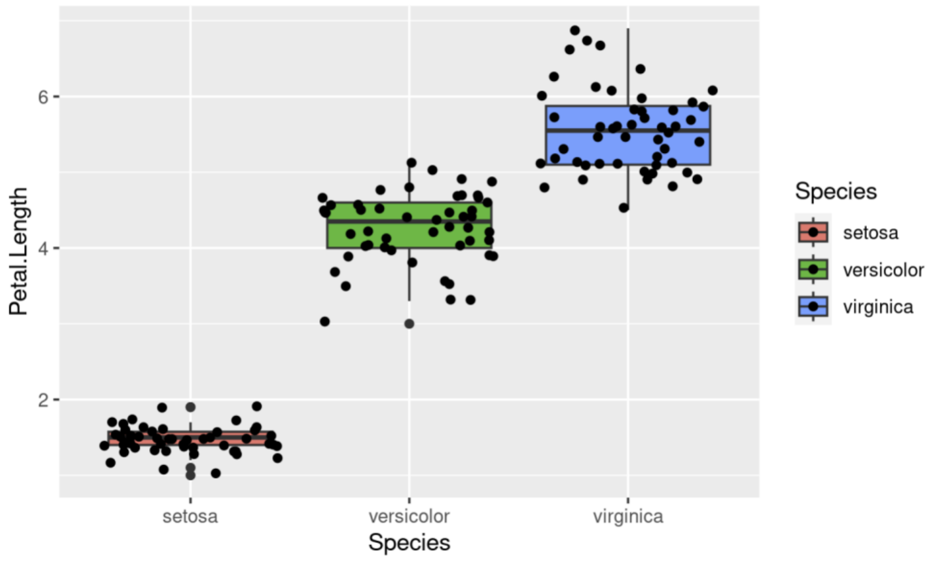
You have now installed all of the packages you need for Project 4!

6. Paste these six lines of code into your main.r script to load the libraries you just installed:

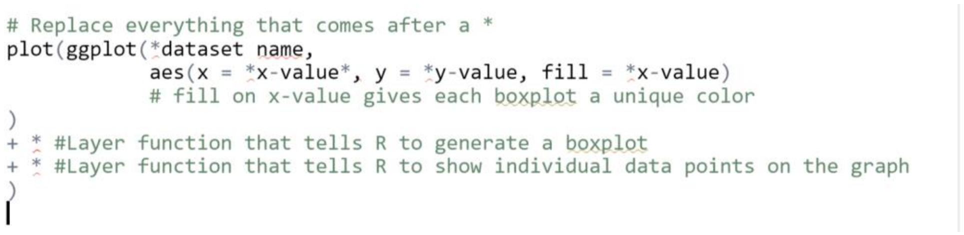
```
library(tidyverse)

library(ggridges)

library(patchwork)

library(viridis)

library(hrbrthemes)

library(gapminder)
```

7. Your first task is to use a pre-loaded iris flower dataset to make three adjacent box plots comparing petal length across the three different species. The data is stored in a data frame called "iris". How many data points from each species of iris flower are in the dataset? (Respond as a comment in main.r). **Very Important Note**: Every time you want to run your main.r script, you need to execute this line in the console: `source("main.r").` You may think this is tedious, but after typing this command once and running it, you can press the up-arrow to restore previously executed commands.

8. Create your boxplots using the ggplot function (Reference: your new best friend). Your graph should look something like this:

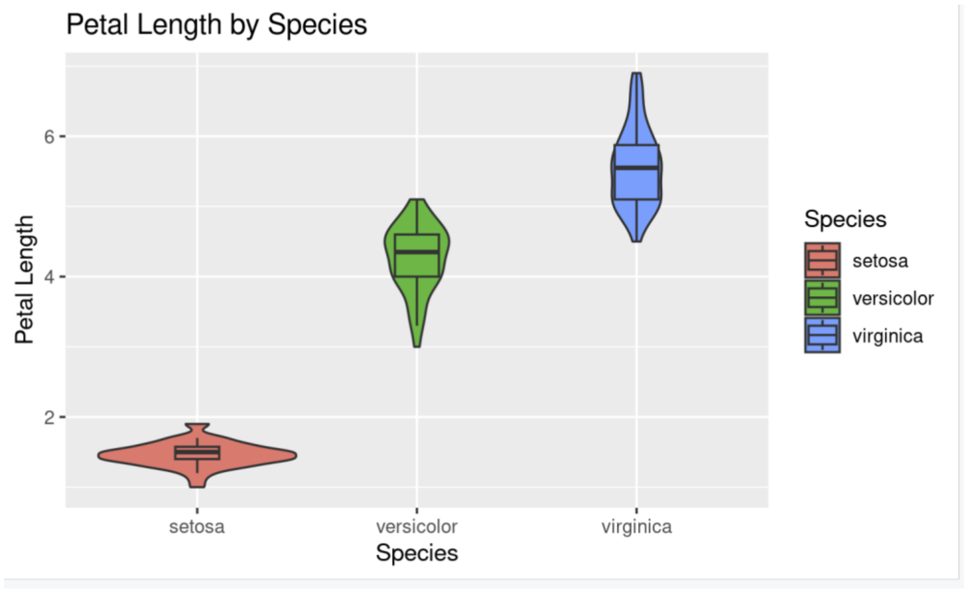


Even though this looks complex, this graph only requires a few lines of code. Here is the pseudocode to generate this graph:



```
# Replace everything that comes after a *
plot(ggplot(*dataset name,
            aes(x = *x-value*, y = *y-value, fill = *x-value)
            # fill on x-value gives each boxplot a unique color
)
+ * #Layer function that tells R to generate a boxplot
+ * #Layer function that tells R to show individual data points on the graph
)
```

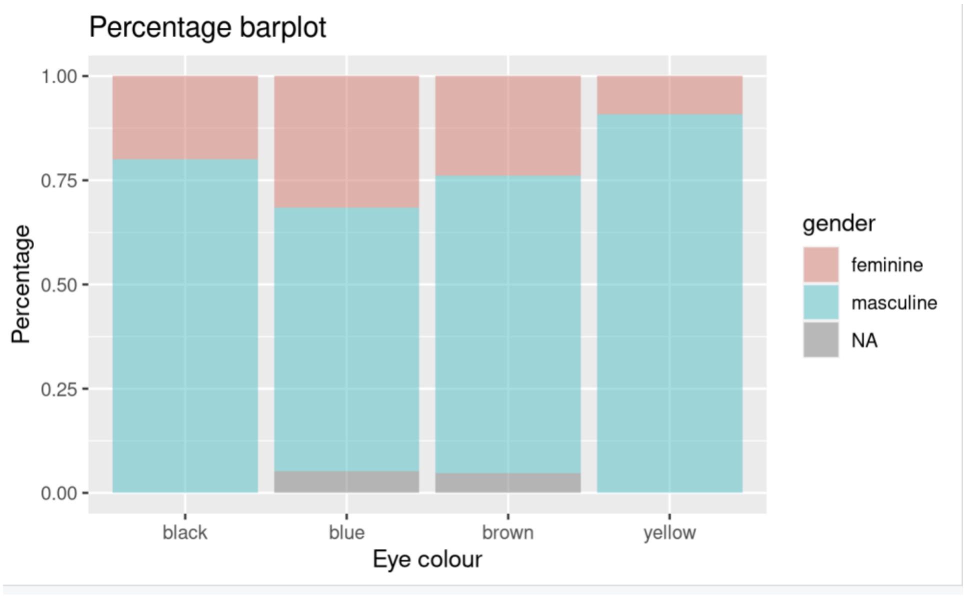
9. Let’s make three improvements to this graph. First let’s add a title. Next, let’s change the name of the y-axis to Petal Length. Both of these can be done with a single scale function. Finally, instead of explicitly showing all of the datapoints, let’s capture the distribution of each plot by adding a violin plot (this can be done with a layer function). Your newly generated graph should look something like this:



Note that only your most recent plot will show up in **Plots**. If you want to go back to view a previous plot for a checkpoint or to modify, just comment out all other plots except the one you want to view. Here’s a helpful tip: If you want to comment out a bunch of highlighted text, press Ctrl + Shift + C.

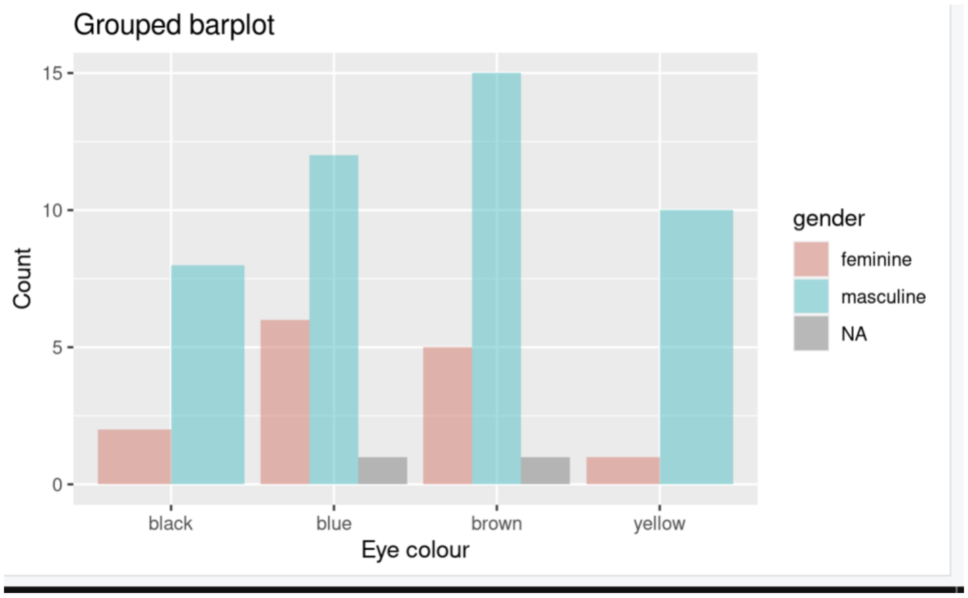
10. For these next few questions we will make some more advanced bar plots using the built-in "starwars" dataset, which contains information about several Star Wars characters. Write a line of code to display all attributes (columns) in the dataset.

11. How many unique eye colors are there in the dataset? Write one line of code to output this value. Then, make a percentage bar chart of four eye colors – black, blue, brown, and yellow. Each bar should show the relative composition of genders of the characters in the dataset having that particular eye color. Your graph should look something like this:



For this problem - as well as many others in this project – you will need to use piping. The pipe operator (%>%) takes the output of one function and uses it as the input to the next function. To use piping in this problem, you would pipe the entire dataset into a filter function that removes all characters that don't have black, blue, brown, or yellow eyes. You would then pipe the results of this filter function into the ggplot() function.

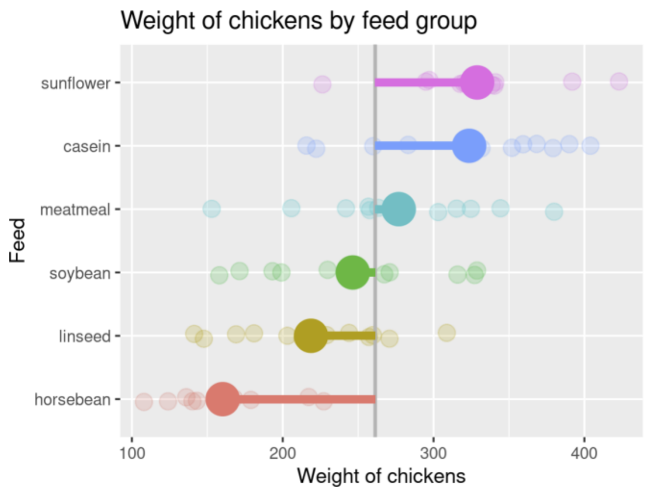
12. Make a bar plot showing the same data as the previous bar plot. This bar graph however will be grouped instead of organized by percentage.



Hint: The code to make this graph is almost exactly the same as the code that makes the previous graph.

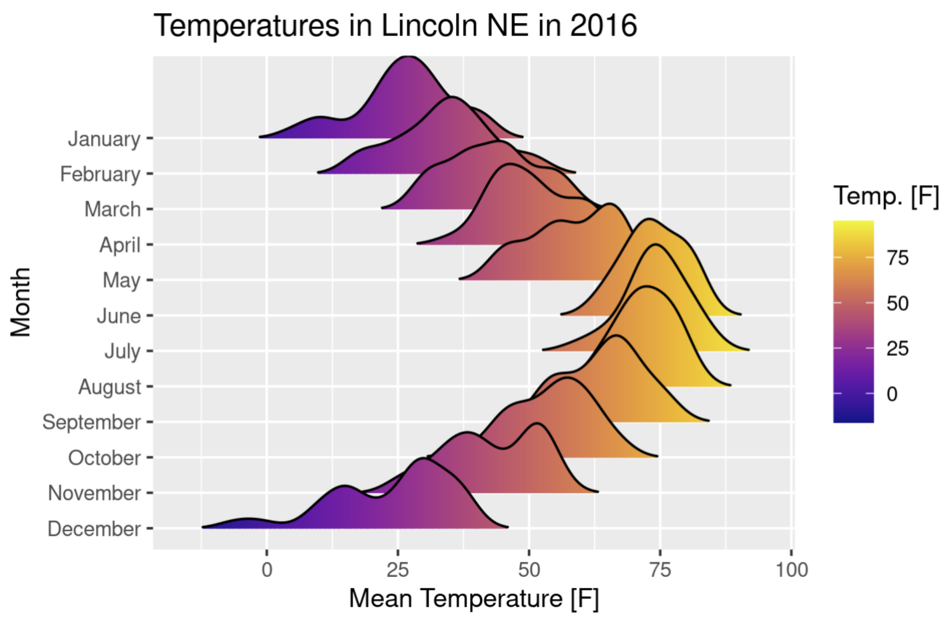
STOP: Check in with a TA or instructor before continuing

13. You will now use a different dataset called "chickwts", which shows the weights and diets of several chickens. Your goal is to make the following graph:



The small blurry dots represent individual datapoints, the large circles represent the mean weight of the chickens on each diet, and the vertical grey line represents the mean weight of all chickens in the dataset. For this problem pipe the dataset into a function that groups the data by diet. Then pipe the results of that function into a function that finds the mean of each group. Then pipe the results of that function into a function that ungroups the data. Then pipe the results of that function into a function that orders the groups from smallest mean to largest mean. Finally, pipe the results of that function into the ggplot() function.

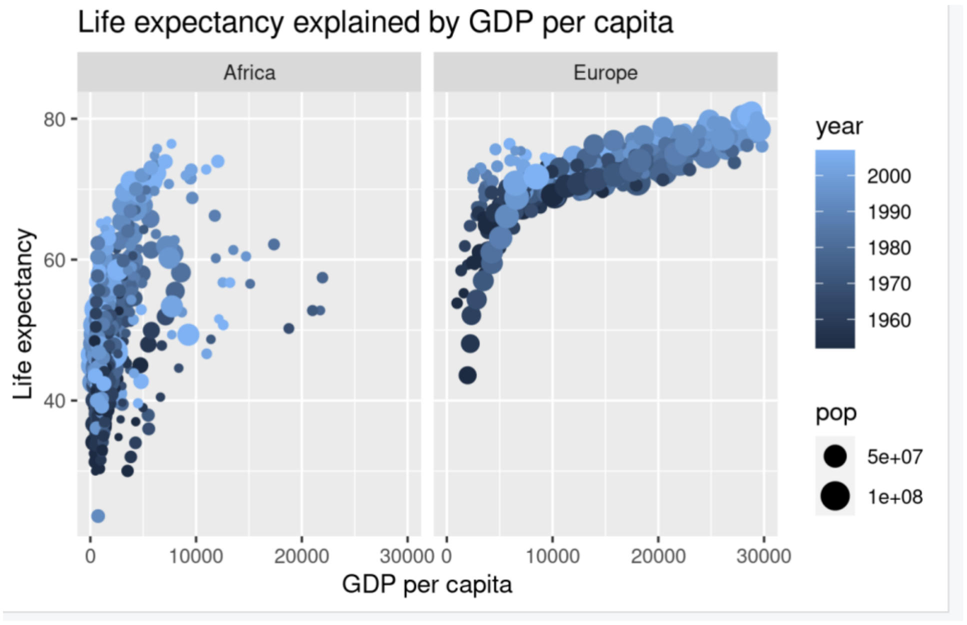
14. This next graph is very visually impressive despite being relatively simple to create. The graph shows monthly temperature distributions in 2016 for the city of Lincoln. There is also a color gradient based on temperature. The most challenging part of making this graph will be finding the right layer function to make the distributions, as it is a function that is not on the function cheat sheet linked in problem 8.



What do the heights of each distribution represent? Respond as a comment in main.r.

15. Let's make one last complex graph using ggplot. This time we will use the "gapminder" dataset. What kind of data is in the "gapminder" dataframe? Respond as a comment in r.main. Your task is to make a graph comparing Life Expectancy vs. GDP per Capita (with GDP per Capita maxing out at 30,000). You will differentiate between European and African countries. Larger data points will represent years with larger population while smaller data points will represent years with smaller population. Finally, you will use a color gradient to distinguish between years. Years closer to the present will be a lighter shade of blue than years further from the present.

Don't be overwhelmed by all this information, ggplot makes this graph relatively simply to code up.



STOP: Check in with a TA or instructor before continuing

16.) So far in this class you have used data that has been given to you. For the next part of this project you are going to collect your own data by designing a survey. Watch this video about avoiding bias when designing a survey.

17.) Think back to the video game survey you filled out in Project 1. Remember how it asked for your name and age? Is this data that we really needed? Make an argument supporting the idea that it was unnecessary to collect data on respondents' names and ages. Respond as a comment in your main.r file for this project.

18.) As data scientists, we have an obligation to protect the privacy of those we are collecting data on, at least as much as possible. One way we can protect individuals' privacy is by not collecting certain data at all. When designing your survey, think hard about whether or not the data you are collecting is truly imperative to your experiment.

19. You will now write your own survey and administer it to ten of your classmates to collect data on a topic of your choice. Note that later in the project you will compare your results to a larger dataset. As a result, you may want to find a dataset first and then choose your topic based on what dataset you find. Any dataset is fine as long as it is a .xlsx or .csv file, though **the dataset should have at least one piece of categorical data!** Here is a decent place to find csv files. You can use the search function to look for specific topics. Alternatively, you can try searching your topic in Google followed by "csv" or "xlsx". For instance, you may search "Dog Ownership Survey Results csv" if you wanted to find data on dog owners.

20. Write your survey as a comment in main.r. **Your survey should collect the same data as the large dataset you found in step 19.** For example, if the large dataset collects data on participant gender, grade, and spending habits, your survey should also collect that data. Think about ways you can reduce bias in your survey. Make sure to avoid leading questions. Additionally, you may want to structure your questions in ways that will allow you to do easier analysis and less cleaning later. For instance, instead of giving an open-ended question, give multiple choice. If you want participants to give a number, specify how they should input it (42 or forty-two?). After you write your survey, explain why you structured it the way you did.

STOP: Check-in with TA or instructor to approve your topic, dataset, and survey

21. Go collect responses from 10 classmates. If we are online you may direct-message individual students to respond to your survey (let us know if this feature is not enabled for you). If in-person, get up and go talk to your classmates! Record these results as a data frame in main.r. You can do this by first creating an empty data frame, and then making a vector for each person you interview. This vector will store that person's survey results. You can use the rbind function to append these vectors as rows into your data frame.

22. Compare your survey results with the results of the larger dataset that you found earlier. Create one or more data visualizations for this comparison. Check out this article if you need inspiration for graphs. Then, as a comment in main.r explain any differences in results. What do you think accounts for this? Respond as a comment in main.r

23. Create a third dataset that is about the same size as your large dataset. However, this dataset will be completely random. It is your job to create a randomly generated dataset in R. How you go about this is up to you, but try to be realistic with your simulation. For instance, if your original dataset contains ages of teens who took a survey, maybe your randomly generated data should only generate ages from 14 to 18. Make another graph to compare this third dataset to both your survey results dataset and your large pre-existing dataset.

STOP: Have a TA or instructor look at 21 – 23 before proceeding

24. Watch these videos (Hypothesis testing, significance tests) and then answer the following questions as a comment in main.r. How can we determine if trends in our data are due to some real-life phenomenon, or are instead the result of random chance? What is a p-value? What is special about the p-value 0.05?

25. Identify two attributes (columns) in your data that might have some sort of relationship. Identify and record the null hypothesis relating these attributes. Then calculate the p-value for this relationship for each of your three datasets. Do any of these datasets have statistically significant relationships between the data? Whatever your results are for this step, try to rationalize them. Respond as a comment in main.r.

STOP: Check in with a TA or instructor before proceeding.

26. Simulate flipping a coin 10,000 times. Then make a line graph showing the percentage of "heads" results after each flip.

27. Estimate pi using a Monte Carlo simulation. Graph the accuracy of the estimation versus the number of simulated points.

STOP: Check in with a TA or instructor before continuing

Congrats! You have finished Project 4! Make sure to submit your main.r script as well as a .png of each plot you made to the Project 4 submission in Canvas.