

Exercise: Spatial Position and Adjustment

Overview

This exercise provides some practice manipulating the spatial position of points and bars in visualizations and is designed to be completed in 60 minutes or fewer. Although you should have time to complete the parts, if you are unable, *as with all exercises, you are encouraged to complete it outside of class time* so that you are able to incorporate your experiences and knowledge into future exercises. You may need to consult course reading materials located at the course site as some elements may not have been covered in the basic content contained in associated videos.

This exercise assumes you have an understanding of `{dplyr}` functions like `select()`, `filter()`, `mutate()`, and `summarize()` and `{ggplot2}` functions like `ggplot()`, `geom_point()`, `geom_bar()`, and `aes()`. Your ability to work through the exercise will also be influenced in part by your prior practice using these functions as part of your *course allocation time outside of class*.

This exercise focuses on:

- Intentional File Creation and File Management
- Creating point plots and bar plots
- Adjusting spatial position of geoms
- Mapping variables to aesthetics
- Combining plots into Grobs

This exercise uses:

- Your RStudio Git version-control project for your team project (presumably completed)
- `{here}`, `{dplyr}`, `{ggplot2}` functions and functions from relevant Base R libraries (e.g., `read.csv()`, `readRDS()`, `saveRDS()`, etc.)
- Your knowledge from past in-class exercises, videos, homework, etc. and corresponding modules from the course site.

Note: If your RStudio project for your team is not created, you will work in your `dataviz-exercises` project and use a different data set, for example `ggplot2::mpg` or `ggplot2::diamonds`. Do not use class time to set up your team project.

Part 1: Creating a Point Plot: Default Overplotting

Using a data set, identify some relevant numeric and non-numeric variables. Create a point plot with a meaningful numeric variable along the y-axis. Either a numeric or non-numeric variable can be used along the x-axis. Consider variables of any type to map to an aesthetic other than x or y.

1. Create an `.R` script file (Recommendation: Start with `starter_script_file.R`)
2. Name the file appropriately based on the goal of the three parts (review briefly). Add either your name or initials as a prefix to the file name. Do not use spaces.
3. Load your libraries and functions you may need.
4. Write code to read your raw data file using `{here}` paths. Assign the data frame an object name.
5. Write code to pipe the data frame to `{ggplot2}` functions in order to create a simple point plot using `geom_point()`. Map a numeric variable to an aesthetic other than x or y. Set an aesthetic as well so that you have practice doing so.
6. Whether there are issues with *overplotting* for your plot, or not, adjust the position of the points so that they *vary freely along both the x and y axes*. Feel free to modify other aesthetics to address overplotting as well.
7. Save your plot script file.

Make sure that your script file is saved in the appropriate project directory.

Part 2: Creating a Point Plot: Overplotting with Constraints

Using the same data set, create a new version of your plot.

1. Add a new code section to your script that recreates the point plot, now adjusting the position of the points so that they *do not vary freely along either the x or y axis*. Feel free to modify other aesthetics to address overplotting as well.
2. Save your plot script file.

Part 3: Creating Bar Plots: Stacked and Grouped

Stacked and grouped bar plots or column plots serve different purposes. Each will influence the user's ability to extracting meaningful information more or less effortfully. You will create both visualizations.

1. Add a new code section to your script that creates a stacked bar plot to understand your data.
2. Add a new code section to your script that creates a grouped bar plot to understand your data.
3. Save your plot script file.
4. Each data visualization has a goal and should be chosen intentionally to facilitate your story telling. Consider what information you might wish to communicate with the two plot types. Consider how the user experience will be facilitated by using one plot over another. Keep a record of your thoughts to inform you future plots.
5. Consider also how you would walk your audience through the visualization. With which plot element will you start? What will you say? What are you communicating? What is the story? For in-class and liaison presentations, do not assume that your audience is skilled at examining and interpreting data visualizations.

Bonus Practice: Creating Grobs

1. Take two of your plots and combine them into a **Grob** with 1 column.
2. Take two of your plots and combine them into a **Grob** with 2 columns.