Lab 2 - Data wrangling

Harrison Kane

> **!** Important
>
> This lab is due Friday, Sep 23 at 11:59pm.

Learning goals

In this lab, you will…

- use data wrangling to extract meaning from data
- continue developing a workflow for reproducible data analysis
- continue working with data visualization tools

Getting started

- Go to the sta199-fa22-01 organization on GitHub. Click on the repo with the prefix `lab-02`. It contains the starter documents you need to complete the lab.
- Clone the repo and start a new project in RStudio. See the Lab 0 instructions for details on cloning a repo and starting a new R project.
- First, open the Quarto document `lab-02.qmd` and Render it.
- Make sure it compiles without errors.

Warm up

Before we introduce the data, let's warm up with some simple exercises.

- Update the YAML, changing the author name to your name, and **render** the document.
- Commit your changes with a meaningful commit message.
- Push your changes to GitHub.

- Go to your repo on GitHub and confirm that your changes are visible in your ‘`.qmd` **and** `.pdf` files. If anything is missing, render, commit, and push again.

Packages

We’ll use the **tidyverse** package for much of the data wrangling. This package is already installed for you. You can load it by running the following in your Console:

```r
library(tidyverse)
```

Data

The dataset for this assignment can be found as a CSV (comma separated values) file in the `data` folder of your repository. You can read it in using the following.

```r
nobel <- read_csv("data/nobel.csv")
```

The descriptions of the variables are as follows:

1. `id`: ID number
2. `firstname`: First name of laureate
3. `surname`: Surname
4. `year`: Year prize won
5. `category`: Category of prize
6. `affiliation`: Affiliation of laureate
7. `city`: City of laureate in prize year
8. `country`: Country of laureate in prize year
9. `born_date`: Birth date of laureate
10. `died_date`: Death date of laureate
11. `gender`: Gender of laureate
12. `born_city`: City where laureate was born
13. `born_country`: Country where laureate was born
14. `born_country_code`: Code of country where laureate was born
15. `died_city`: City where laureate died
16. `died_country`: Country where laureate died
17. `died_country_code`: Code of country where laureate died
18. `overall_motivation`: Overall motivation for recognition
19. `share`: Number of other winners award is shared with
20. `motivation`: Motivation for recognition

In a few cases the name of the city/country changed after laureate was given (e.g. in 1975 Bosnia and Herzegovina was called the Socialist Federative Republic of Yugoslavia). In these cases the variables below reflect a different name than their counterparts without the suffix `_original`.

21. `born_country_original`: Original country where laureate was born
22. `born_city_original`: Original city where laureate was born
23. `died_country_original`: Original country where laureate died
24. `died_city_original`: Original city where laureate died
25. `city_original`: Original city where laureate lived at the time of winning the award
26. `country_original`: Original country where laureate lived at the time of winning the award

Get to know your data

1. How many observations and how many variables are in the dataset? Use inline code to answer this question. What does each row represent? There are `nrow(nobel)` observations, and `ncol(nobel)` variables in the dataset.

There are some observations in this dataset that we will exclude from our analysis to match the Buzzfeed results.

2. Create a new data frame called `nobel_living` that filters for

- laureates for whom `country` is available
- laureates who are people as opposed to organizations (organizations are denoted with `"org"` as their `gender`)
- laureates who are still alive (their `died_date` is `NA`)

Confirm that once you have filtered for these characteristics you are left with a data frame with 228 observations, once again using inline code.

Most living Nobel laureates were based in the US when they won their prizes

… says the Buzzfeed article. Let's see if that's true.

First, we'll create a new variable to identify whether the laureate was in the US when they won their prize. We'll use the `mutate()` function for this. The following pipeline mutates the `nobel_living` data frame by adding a new variable called `country_us`. We use an if statement to create this variable. The first argument in the `if_else()` function we're using to write this if statement is the condition we're testing for. If `country` is equal to `"USA"`, we set `country_us` to `"USA"`. If not, we set the `country_us` to `"Other"`.

```
nobel_living <- nobel_living |>
  mutate(
    country_us = if_else(country == "USA", "USA", "Other")
  )
```

Next, we will limit our analysis to only the following categories: Physics, Medicine, Chemistry, and Economics.

```
nobel_living_science <- nobel_living |>
  filter(category %in% c("Physics", "Medicine", "Chemistry", "Economics"))
```

For the following exercises, work with the `nobel_living_science` data frame you created above. This means you'll need to define this data frame in your Quarto document, even though the next exercise doesn't explicitly ask you to do so.

3. Create a faceted bar plot visualizing the relationship between the category of prize and whether the laureate was in the US when they won the nobel prize. Interpret your visualization, and say a few words about whether the Buzzfeed headline is supported by the data.

   - Your visualization should be faceted by category.
   - For each facet you should have two bars, one for winners in the US and one for Other.
   - Flip the coordinates so the bars are horizontal, not vertical.

Now is a good time to render, commit, and push. Make sure that you commit and push all changed documents and your Git pane is completely empty before proceeding.

But of those US-based Nobel laureates, many were born in other countries

4. Create a new variable called `born_country_us` in `nobel_living_science` that has the value `"USA"` if the laureate is born in the US, and `"Other"` otherwise. How many of the winners are born in the US?

> ℹ Note
>
> You should be able to ~~cheat~~ borrow from code you used earlier to create the `country_us` variable.

5. Add a second variable to your visualization from Exercise 3 based on whether the laureate was born in the US or not. Create two visualizations with this new variable added:

   - Plot 1: Segmented frequency bar plot

- Plot 2: Segmented relative frequency bar plot (*Hint:* Add `position = "fill"` to `geom_bar()`.)

Here are some instructions that apply to both of these visualizations:

- Your final visualization should contain a facet for each category.
- Within each facet, there should be two bars for whether the laureate won the award in the US or not.
- Each bar should have segments for whether the laureate was born in the US or not.

Which of these visualizations is a better fit for answering the following question: "Do the data appear to support Buzzfeed's claim that of those US-based Nobel laureates, many were born in other countries?" First, state which plot you're using to answer the question. Then, answer the question, explaining your reasoning in 1-2 sentences.

Now is a good time to render, commit, and push. Make sure that you commit and push all changed documents and your Git pane is completely empty before proceeding.

6. In a single pipeline, filter the `nobel_living_science` data frame for laureates who won their prize in the US, but were born outside of the US, and then create a frequency table (with the `count()` function) for their birth country (`born_country`) and arrange the resulting data frame in descending order of number of observations for each country. Which country is the most common?

Now is a good time to render, commit, and push. Make sure that you commit and push all changed documents and your Git pane is completely empty before proceeding.

Submission

Once you are finished with the lab, you will your final PDF document to Gradescope.

> ⚠ Warning
>
> Before you wrap up the assignment, make sure all documents are updated on your GitHub repo. We will be checking these to make sure you have been practicing how to commit and push changes.
> You must turn in a PDF file to the Gradescope page by the submission deadline to be considered "on time".

To submit your assignment:

- Go to http://www.gradescope.com and click *Log in* in the top right corner.
- Click *School Credentials* → *Duke NetID* and log in using your NetID credentials.
- Click on your *STA 199* course.

- Click on the assignment, and you'll be prompted to submit it.
- Mark all the pages associated with exercise. All the pages of your lab should be associated with at least one question (i.e., should be "checked"). *If you do not do this, you will be subject to lose points on the assignment.*
- Select the first page of your .pdf submission to be associated with the *"Workflow & formatting"* question.

Grading

Component	Points
Ex 1	6
Ex 2	6
Ex 3	8
Ex 4	6
Ex 5	8
Ex 6	8
Workflow & formatting	8
Total	**50**

ℹ Note

The "Workflow & formatting" component assesses the reproducible workflow. This includes having at least 3 informative commit messages, labeling the code chunks, and having readable code that does not exceed 80 characters, i.e., we can read all your code in the rendered PDF.