DATA 505: Lab 2

Getting to know data frames

YOUR NAME HERE

2025-09-02

Table of contents

1 Setup & Goals

Just like in Lab 1, you will edit a `.qmd` Quarto script, render the result to a PDF, and submit both on Moodle (if you do not succeed at rendering the PDF, still submit the `.qmd` script: you may still get full "satisfactory" credit if you have made a reasonable effort).

Goals for Lab 2:

- Gain more fluency with writing R code in the context of a Quarto script in RStudio
- Learn to read data into R from a text (comma-separated values, `.csv`) file
- Reinforce concepts from this week's lecture about types ond classes of variables
- Get some practice taking subsets of data frame objects

2 The data file

To do this, we will use the same dataset as in Section 7 of our primary textbook

- The following code will download the `students.csv` dataset to your current working directory. You should leave this code unmodified:

```
download.file("https://andrew-bean.github.io/data505/labs/week_2/students.csv", "students.csv
stopifnot(file.exists("students.csv"))
```

- I recommend running this code above first before writing any more code (with your cursor in the code chunk, from the "Run" menu, you can select "Run current chunk")

3 Reading data into R

- Take a look at the file you have just downloaded. It should look like this.
- The base R function for reading `.csv` files into data frames is `read.csv()`. An example usage would be

```
# just an example: no need to edit
my_data <- read.csv("my_file.csv")
```

- Try using `read.csv` to read the `students.csv` file and store the result in an object named `students`:

```
## YOUR CODE HERE
```

- Write a line of code to determine the **class** of the resulting `students` object

```
## YOUR CODE HERE
```

- Write another line of code to determine the **basic type** of the `students` object

```
## YOUR CODE HERE
```

- Apply the `str()` function to the `students` object to learn more about its structure

```
## YOUR CODE HERE
```

- Now use the `names()` function to get a list of the column names

```
## YOUR CODE HERE
```

4 Subsetting the data

- Use `$` (as discussed here in today's lecture) to select the `AGE` column and store it as another object named `ages`

```
## YOUR CODE HERE
```

- Now repeat using `[["{column name}"]]` syntax illustrated at the same link

```
## YOUR CODE HERE
```

- Use the `length()` function to determine the length of the `ages` vector

```
## YOUR CODE HERE
```

- Define another variable `age` which is consists only of the fist element of the `ages` vector. Recall we discussed vector subsetting here in today's lecture.

```
## YOUR CODE HERE
```

- Now something a bit more challenging. Use data frame subsetting, a relational operator and vector subsetting together to do the following: return a vector containing the ages of all students on the "Lunch only" meal plan

```
## YOUR CODE HERE
```

5 Exploring column types and coercion

- Write a line of code to determine the type of the `ages` vector you created earlier (the `AGE` column from the `students` data frame)

```
## YOUR CODE HERE
```

- This variable ought to be numeric. Use explicit coercion to convert it.

```
## YOUR CODE HERE
```

- Modify the `students` dataset to ensure the `AGE` column is numeric

```
## YOUR CODE HERE
```

- What is the type of the `mealPlan` column? Write a line of code to determine this.

```
## YOUR CODE HERE
```

- Modify the `students` dataset to ensure the `mealPlan` column is a factor

```
## YOUR CODE HERE
```

6 Bonus

What we have done here is some cumbersome "data cleaning". We read data in from a file, but `read.csv` did not recognize the intended types of the csv columns immediately. So we coerced some columns from characters to factors or numerics.

- Try following some of Section 7.2.1 of the R for Data Science book to try to do this more concisely with the `read_csv()` function in the `readr` package.

```
# install the readr package if needed
if(!"readr" %in% installed.packages()){
  install.packages("readr")
}
# load it
library(readr)

## YOUR CODE HERE
```