Data 200: Lab 1b

Introduction to toolkit

Coding with R

In DATA 200 we will use the programming language **R** to help us with our tasks such as calculations, exploring data, and producing visuals.

Calculation basics

```
2 + 5
```

```
[1] 7
```

Mark above which part is the **code** and the **output** of the code.

Note the `[1]` in front of the 7. This marks the 7 as _____.

We can do the standard operations of `+`, `-`, `*`, `/`, `sqrt()`, `exp()`, and many more!

What if I wanted to save my calculation? I can use the **assignment operator** `<-`, to save my calculation into an **object** with a name I choose.

```
my_calculation <- 3 * 12
```

Now `my_calculation` has the value 36 stored. I can also print the stored value.

```
my_calculation
```

```
[1] 36
```

What value do you think would be printed below?

```
my_calculation / 2
```

Vectors and functions

Say I went to 5 local coffee shops and recorded the price of a cup of coffee. I can make a **vector** of these prices, an object which store values of a similar type.

```
c(3, 2.50, 4.25, 3.5, 5)
```

```
[1] 3.00 2.50 4.25 3.50 5.00
```

The commas separated the values and the `c()` combined the values into a vector. Write code below to store the vector above into an object called `coffee_prices`.

The `c()` used above is an example of a **function**, functions do actions for us in R. What do you think the output of the following code using the `max()` function will be?

```
max(coffee_prices)
```

Data frames

I was a little inconsistent when recording prices. The prices actually correspond to different sizes of coffees: small, small, large, medium, and large respectively.

```
coffee_sizes <- c("small", "small", "large", "medium", "large")
```

Instead of storing numbers this time we stored what are called **characters**, which have to have quotes surrounding.

I can store the two vectors together into a **data frame** so all of my information is organized.

```
coffee_data <- data.frame(coffee_prices, coffee_sizes)

coffee_data
```

```
  coffee_prices coffee_sizes
1          3.00        small
2          2.50        small
3          4.25        large
4          3.50       medium
5          5.00        large
```

Quarto documents

In this class we will use **Quarto documents**, which allow us to

- combine code, output, regular text, and a lot more,
- code reproducibly because the code is saved, and
- produce reports, with or without the code showing.

Fill in the blanks using the following vocabulary: **code chunk**, **yaml**, **Render**, **html**.

The _____ of a Quarto document is where we specify things for the document such as title, author, etc.

We can type regular text in the document just like a Word document, and when we want to include code it has to be in a _____.

You can _____ a Quarto document into an html, pdf, or Word doc. In this class we will mainly produce html documents, which are webpages just like our lecture slides!

We will learn much more about Quarto documents as we use them.

Version control with GitHub

Fill in the blanks using the following vocabulary: **clone**, **push**, **Git**, **GitHub**, **commit**.

We will use _____ on our computer to help track files.

When I _____ a **repository** from GitHub, it downloads the folder to my computer and allows me to track changes I make.

When I edit and save a file tracked with Git, I can make a _____ to save a message explaining what I changed in the file.

I can _____ my changes to GitHub online, so my changes aren't only saved locally on my computer.

Let's get working

1. Make GitHub account and provide me with your username in Google Form.
2. Run the following with your GitHub username and email (you will have to install the `usethis` R package first:

```
library(usethis)
use_git_config(user.name = "Jane Doe", user.email = "jane@example.org")
```

3. Follow the last slide video to connect RStudio and Github.
4. Clone the week-01 repo to your computer, make a change to a file and save, commit the change, push to GitHub, and see if you can see the change on GitHub.