Day 2, Lecture One: Data Manipulation with Tidyverse

Andie Creel

January, 2023

How I'd recommend following along for this lecture: have the pdf in front of you so you can follow along some of the points, but mostly write the code and comments in a script or R markdown.

Today, we're going to go through a lot of different concepts. I've tried to write this PDF so you can use it as a reference to refer back to repeatedly.

1 Goal

The goal of this lecture is:

1) Learn about packages, the tidyverse
2) Learn how to manipulate and clean data with tidyverse
3) Learn how to set up a r project

2 Why is data manipulation important

Backing up and remembering we're scientists

- We will have a hypothesis of how the world works
- We want to contruct a model that approximates that
- We need data from real work to build that model that approximates the world
- The data we have may not be set up to be plugged into the model we'd like to run
- However, it could be *manipulated* so that we can use the data we have with the model we want

My personal example: American Time Use Survey and Travel Cost Model, Cell Photo Data and green space

3 What are packages and how can I get them?

What are they:

- A package contains a bunch of pre-built functions
- Anyone can load and use them
- Saves you a ton of time because someone already figured out how to do it

Tidyverse

- Collection of R packages
- All are meant for data science
- Have shared syntax
- Makes it easier to import, tidy, transform, visualize, and model data in R
- Shout out to Hadley Wickham and co

How can I install packages (user interface):

1. Go the the files/Plots/Packages quadrant
2. Click on Packages
3. Click Install
4. Search for packages you want ("dplyr", "tidyr", "ggplot2").
    - All of these are tidyverse packages
    - You can install "tidyverse" and have all of them
    - I think it's more valuable to learn one at a time so that you know what function goes with what package

```
# May want to run this as some point, but it takes a long time! so maybe not now
# install.packages("tidyverse")
```

Installing a package vs. loading a pacakge

- You only need to install a package on your local computer once
- You then "load" that package in a script when you want to use it.

Installing a package with r code (one time only)

```
# installing packages using r code
install.packages("dplyr")
install.packages("tidyr")
install.packages("ggplot2")
```

Loading a package

```
library(dplyr)
library(tidyr)
library(ggplot2)
```

4 Manipulating and cleaning with dplyr

- dplyr is my most used package for data cleaning and manipulation
- Let's go through some examples of what we did yesterday, but redo with dplyr code
- Why use dplyr instead of base R?
    - It's faster and more memory efficient (good for large datasets)
    - It's easier to read

4.0.1 Examples from yesterday

```r
# ------------------------------------------------------------------------
# Let's make a dataframe again
# ------------------------------------------------------------------------
myPpl_base <-   data.frame(
   gender = c("Male", "non-binary","Female"),
   male = c(T, F, F),
   height = c(152, 171.5, 165),
   weight = c(81, 93, 78),
   age = c(42,38,26)
)

myPpl_dplyr <- myPpl_base

# ------------------------------------------------------------------------
# manipulating a column (from above: version one of referencing a cell)
# ------------------------------------------------------------------------

# Base R
for (i in 1:length(myPpl_base$age)) {
  myPpl_base$age[i] <- myPpl_base$age[i] + 1 # everyone aged one year
}

# dplyr
myPpl_dplyr <- myPpl_dplyr %>%
  mutate(age = age + 1)

# Check to make sure they're identical
(myPpl_dplyr == myPpl_base)
```

```
##      gender male height weight  age
## [1,]   TRUE TRUE   TRUE   TRUE TRUE
## [2,]   TRUE TRUE   TRUE   TRUE TRUE
## [3,]   TRUE TRUE   TRUE   TRUE TRUE
```

New things introduced

- Pipes %>%: pipes that input (in this case, our dataframe) and pass it onto the next function. You can chain them together.
- `mutate()` mutate is a function that is fed a data frame (from the pipe) and then can change a variable (in this case age)