Week 4, Day 1

9/27/2020

Let's continue working with enrollment data. I have downloaded 5 years of fall enrollment data in to the `raw_data` directory from the official source.

Scene 1

Prompt: Write a pipe which creates an object named `d_2019` by reading in and cleaning up the data from Fall 2019. (You will need to examine the file names to determine which file this is.) You may consult and re-use the code from last week. The variable names in the tibble should be `id`, `title`, `name`, `department` and `u_grad`. Keep only classes with more than 10 undergrads enrolled.

```
## Rows: 628
## Columns: 4
## $ course_id         <chr> "207805", "123435", "109427", "205832", "203313",...
## $ course_name       <chr> "Black Womens Voices in the #Me", "Poverty, Race,...
## $ course_department <chr> "African & African Amer Studies", "African & Afri...
## $ u_grad            <dbl> 25, 165, 20, 14, 16, 16, 12, 15, 12, 12, 17, 11, ...
```

Scene 2

Prompt: We could copy/paste this code 5 times, adjust the files names, and then read in each file. But, as you know from Chapter 4, that is a bad idea. It also scales very poorly. Create a function called `read_enrollent` which takes one argument, `file`. Use that function to read in the data from fall 2019 and assign it to an object called `d_2019_take_2`. Do you get the same answer as you did in Scene 1?

```
## Rows: 628
## Columns: 4
## $ course_id         <chr> "207805", "123435", "109427", "205832", "203313",...
## $ course_name       <chr> "Black Womens Voices in the #Me", "Poverty, Race,...
## $ course_department <chr> "African & African Amer Studies", "African & Afri...
## $ u_grad            <dbl> 25, 165, 20, 14, 16, 16, 12, 15, 12, 12, 17, 11, ...
```

Scene 3

Prompt: Call `read_enrollent()` five times, once for each of our data sets. Note how different the file names are. Real data is messy! Assign the result of each call to an object, `d_2019`, `d_2018` and so on. Should be easy . . .

Arrg! Depending in how you wrote `read_enrollment()`, you will probably be getting an error, for at least some of the years. How annoying that Harvard changes the format! Make your function flexible enough to deal with all these files. Hint: You need to add at least one argument in addition to `file` so that you can change the behavior of the function when you call it. Give that new argument a sensible default.

Scene 4

Prompt: Combine the five tibbles which you have into a single tibble which can then be used for analysis and graphics. There are many ways to do this, but we recommend `bind_rows()`. Hint: make use of the `.id` argument, which may be aided by placing the tibbles in a list.

Scene 5

Prompt: Make an interesting plot with this data. Take that plot and publish it on Rpubs. Add a link to the Rpubs in the #general Slack channel.