Tutorial 10 - Interacting with data structures

Reminder - Don't forget the handouts

`R` is heavy on the use of variables and functions!

Remember that we gave you two handouts the first week of the R unit:

1) data structures and frequently used functions
2) how to navigate the file system from within R and read/write files

These are great resources as you work through challenges and exercises!

Reminder - Github with R

The Terminal tab in RStudio allows you to interact with `bash`.

This is helpful because you can use `git` and interact with github this way.

- In `OSX` you should be all set by clicking on the *Terminal* tab.
- In `Windows` you'll need to run `bash` (type bash and press Enter) to access the file system and git tools.

If working with the terminal tab in RStudio doesn't work for you, you can always use a bash terminal outside of R. The files you read and write, as well as the R scripts you create are just text files that exist in your file system. Therefore, they can be viewed and version controlled from bash too.

Why subset data?

Stuart may have neglected to motivate our discussion of subsetting data structures this week. . .

When analyzing biological data, we often want to summarize or analyze data similarly across categories.

- maybe we ran three separate experiments, but the lab that generated the data provided all of the data in a single data file.
- maybe we counted the number of fish in three different lakes over many years and we want to know the average and range of fish abundances for each lake.

To summarize or analyze data across categories (experiments and lakes in the example above) we need to be able to access a portion or **subset** of all of our data.

Challenge - quick challenges on indexing

- Create a list containing a vector of 5 names and a 2x2 matrix containing the numbers 1 to 4
  - access the 4th and 5th name from the vector in your list
  - access the number in the 1st row and 2nd column of the matrix
- Load the `wages.csv` file as a dataframe
  - access the 15th row of that dataframe with square brackets
  - find the minimum wage in the entire dataframe
- Create 2 new `.csv` files to send to different co-workers
  - The first file should be called `femaleWages.csv` and have all of the same columns as `wages.csv`, but only contain data for females.
  - The second file should only have gender and wages columns, but include only individuals that have 12 or more years of school.

Challenge - revisiting the challenges from lecture

1. Write a file containing the unique gender-yearsExperience combinations contained in the file "wages.csv". The file you create should contain gender in the first column and yearsExperience in a second column with a space separating the two columns. The rows should be sorted first by gender and then by yearsExperience, but remember to keep the pairings in a given row intact. Don't worry about column names in the output file.

2. Return the following information to the R console when the script is executed: the gender, yearsExperience, and wage for the highest earner, the gender, yearsExperience, and wage for the lowest earner, and the number of females in the top ten earners in this data set. Be sure to indicate, which output is which when returning them to the console.

3. Return one more piece of information to the console: the effect of graduating college (12 vs. 16 years of school) on the minimum wage for earners in this dataset.

Looking ahead to next week

As always, Exercise 8 is due next Friday (11/11). Be sure to fork and clone this from your TA's github repository.

We will be covering plotting in R next week. You have a quiz due Monday (11/7) that asks you to find and upload a scientific figure.