Section 3 Activity (w/ suggested answers)

Purpose

Today is a practice day. Your mission, should you choose to accept it[1], is to work in groups to examine a dataset, extract some interesting information about it, and then communicate that to others. For administrative purposes, at the end of section reply to the **Section 3 Activity** thread with your name, your group members and what initial dataset you chose to use.

Datasets

There are six different example datasets on bCourses as well as a a brief data explainer. Pick one of them for today and download it to your machine. You can switch at some point later if you so choose.

I tried to pick datasets based on what the class discussed last section. Each dataset is purposely limited to a small number of columns.

A Brief Digression on File Paths

For the better of the last thirty years, computer operating systems manufacturers have been trying to hide how the computer works from you. That creates problems when we try to read in data.

1. Find out where you are

Use the `getwd()` function to find your current directory.

[1] Frankly, it'd be a bit weird if you didn't, but to each their own.

2. Go somewhere else

Use the `setwd()` function to change to a different directory. Ideally, we'd like to change to the directory (think folder) where we have saved all of our data.

3. See what's in where you are

Use the `list.files()` function to see what is in your working directory.

4. Test if the name of your dataset is in files

For example, suppose I downloaded the `squirrels.csv` dataset to my desktop.

```r
## for example, I saved this file to my desktop.
"squirrels.csv" %in% list.files("~/Desktop")
```

```
[1] TRUE
```

```r
## running the code without the argument produces FALSE
## if my working directory isn't my Desktop
"squirrels.csv" %in% list.files()
```

```
[1] FALSE
```

Note that I have typed the file name in double quotes. You could also use single quotes, but the following will produce an error. What error do you get?

```r
squirrels.csv %in% list.files()
```

Conceptual Question 1

Suppose you had a large number of datasets that you wanted to read in at once. Talk with your group about the steps that you would like the computer to run.

Bonus: If you feel comfortable with the R programming language or control flow, write out some `pseudocode` for this procedure.

Basic R Operations

1. Read in your data frame to R and save it as an object in your environment with an appropriate variable name.

```r
squirrels = read.csv("./exampleData/squirrels.csv")
head(squirrels)
```

```
  id year   animal          place             state numAttacks Duration
1  1 2020 Squirrel       Columbia                SC          1        0
2  2 2020 Squirrel          Omaha                NE          1      120
3  3 2020      Cow      Chapelton South Lanarkshire          1      300
4  4 2019 Squirrel         Topeka                KS          1       60
5  5 2019    Mouse           Cona              <NA>          1        0
6  6 2019 Squirrel West Lafayette                IN          1      450
  Affected
1     1800
2        0
3      800
4     4000
5        0
6     4400
```

2. Find the dimensions of the dataset and assign each dimension to an appropriate variable name.

```r
rows = nrow(squirrels)
cols = ncol(squirrels)

## or
dims = dim(squirrels)
rows2 = dims[1]
cols2 = dims[2]
```

3. Perform the basic mathematical operations on these two variables.

For example, What is their sum? Their division? What if you take the number of rows to the power of the number of dimensions? Do the number of columns divide evenly into the rows?

```
rows + cols
```

```
[1] 2586
```

```
rows / cols
```

```
[1] 322.25
```

```
rows ^ cols
```

```
[1] 1.951027e+27
```

```
rows %% cols == 0
```

```
[1] FALSE
```

4. Take a numeric column from your data frame and assign it to a separate variable.

Add the number of rows to each value in this new vector. What is its length? How about its average? Are there any missing values?

```
## get information about the data frame
str(squirrels)
```

```
'data.frame':	2578 obs. of  8 variables:
 $ id        : int  1 2 3 4 5 6 7 8 9 10 ...
 $ year      : int  2020 2020 2020 2019 2019 2019 2019 2019 2019 2019 ...
 $ animal    : chr  "Squirrel" "Squirrel" "Cow" "Squirrel" ...
 $ place     : chr  "Columbia" "Omaha" "Chapelton" "Topeka" ...
 $ state     : chr  "SC" "NE" "South Lanarkshire" "KS" ...
 $ numAttacks: int  1 1 1 1 1 1 1 1 1 1 ...
 $ Duration  : int  0 120 300 60 0 450 50 0 0 0 ...
 $ Affected  : num  1800 0 800 4000 0 4400 21 0 0 0 ...
```

```r
numericCol = squirrels$Affected
numericCol = numericCol + rows
length(numericCol)
```

```
[1] 2578
```

```r
mean(numericCol, na.rm = T)
```

```
[1] 4964.449
```

```r
sum(is.na(numericCol))
```

```
[1] 0
```

Data Frames

Return to your original data frame for the next section.

1. Using bracket notation, make three new R objects (not in your data frame) with different slices of any variable from your data set.

```r
a = sample(squirrels$Affected, 200)
b = squirrels$Affected[seq(400, nrow(squirrels), 4)]
c = squirrels$Affected[c(1:5)]
```

2. Using bracket notation, make a new variable (not in your data frame) that is a logical vector indexing a numeric variable by a condition in a different variable

```r
logicalVec = ifelse(squirrels$animal != "Squirrel", TRUE, FALSE)
```

Compute the sum of two different variables indexed in this way. What is their difference? Imagine you were explaining why this difference was meaningful. What would you say?

```r
logicalVec2 = ifelse(squirrels$year == 2019, TRUE, FALSE)
```

```
sum(logicalVec, logicalVec2)
```

```
[1] 1336
```

3. Subset your data set by something that you find meaningful in the data. Justify your choice.

```
## Get all squirrel attacks in 2019 only
squirrels2019 = which(squirrels$animal == "Squirrel" & squirrels$year == 2019)

squirrels[squirrels2019,]
```

```
   id year   animal          place state numAttacks Duration Affected
4   4 2019 Squirrel         Topeka    KS          1       60     4000
6   6 2019 Squirrel West Lafayette    IN          1      450     4400
7   7 2019 Squirrel     Youngstown    OH          1       50       21
13 13 2019 Squirrel        Toronto    ON          1        0     4000
14 14 2019 Squirrel          Bewer    ME          1       30     2400
15 15 2019 Squirrel       Weymouth    MA          1       43     3400
17 17 2019 Squirrel        Norfolk    VA          1        0        0
18 18 2019 Squirrel         Bangor    ME          1       60     1400
19 19 2019 Squirrel        Trinity    CA          1        0        0
20 20 2019 Squirrel         Bangor    MN          1       60     1400
21 21 2019 Squirrel      Red Bluff    CA          1        0     1700
```

Make sure to have at least one condition (though you can have more if you'd like) in your call.

4. Use two of the three different methods of subsetting() shown in lecture to get the same result as part 4. Do not repeate your first method.

```
method1 = subset(squirrels, animal == "Squirrel" & year == 2019)
head(method1)
```

```
   id year   animal          place state numAttacks Duration Affected
4   4 2019 Squirrel         Topeka    KS          1       60     4000
6   6 2019 Squirrel West Lafayette    IN          1      450     4400
```

```
7   7 2019 Squirrel     Youngstown    OH          1       50       21
13 13 2019 Squirrel        Toronto    ON          1        0     4000
14 14 2019 Squirrel          Bewer    ME          1       30     2400
15 15 2019 Squirrel       Weymouth    MA          1       43     3400
```

```r
method2 = squirrels[squirrels$animal == "Squirrel" & squirrels$year == 2019,]

library(dplyr)
method3 = squirrels |>
  filter(animal == "Squirrel", year == 2019)

## Note |> is the base R pipe. If we've loaded dplyr we can also
## do
method3b = squirrels %>%
  filter(animal == "Squirrel", year == 2019)
```

5. Come up with a way to sample rows of your data set to make a data frame that has just 1/4 of the rows. Repeat steps 4 and 5 on this new data frame with a different condition.

Nothing changes with the methods, but the new part to include is sampling rows.

```r
set.seed(123) # for reproducibility
idx = sample(nrow(squirrels), nrow(squirrels)/4)
squirrel_subset = squirrels[idx,]
summary(squirrel_subset)
```

```
       id              year         animal             place          
 Min.   :   8.0   Min.   :1984   Length:644         Length:644        
 1st Qu.: 669.5   1st Qu.:2014   Class :character   Class :character  
 Median :1340.0   Median :2015   Mode  :character   Mode  :character  
 Mean   :1321.4   Mean   :2015                                        
 3rd Qu.:1962.0   3rd Qu.:2017                                        
 Max.   :2574.0   Max.   :2019                                        
    state            numAttacks     Duration         Affected   
 Length:644         Min.   :1    Min.   :   0.0   Min.   :     0  
 Class :character   1st Qu.:1    1st Qu.:   0.0   1st Qu.:     0  
 Mode  :character   Median :1    Median :   0.0   Median :     2  
                    Mean   :1    Mean   :  51.3   Mean   :  2223  
                    3rd Qu.:1    3rd Qu.:  60.0   3rd Qu.:  2500  
                    Max.   :1    Max.   :3600.0   Max.   :100000
```

You may also want to consider summarizing this smaller data frame.

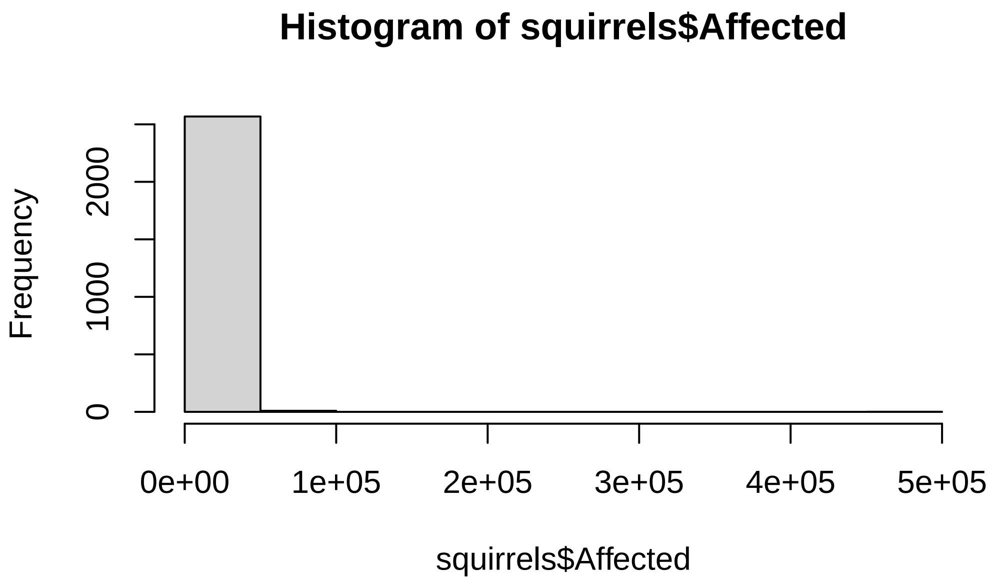
6. Run a linear regression on your original data set. The regression should predict some value.

```
summary(lm(Affected ~ Duration, data = squirrels))
```

```

Call:
lm(formula = Affected ~ Duration, data = squirrels)

Residuals:
   Min     1Q Median     3Q    Max 
-27175  -2072  -2072    -89 497026 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 2071.727    238.080   8.702  < 2e-16 ***
Duration       5.857      1.109   5.284 1.37e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 11700 on 2576 degrees of freedom
Multiple R-squared:  0.01072,	Adjusted R-squared:  0.01034 
F-statistic: 27.92 on 1 and 2576 DF,  p-value: 1.373e-07
```

7. Consider visualizing the dataset. You can use whatever visualization package you prefer.

```
hist(squirrels$Affected)
```



Find New Friends

When you have completed the activity, find someone that you have not spoken to and discuss what you came up with. Did you make the same choices? Is there something about their code that you like that you can use going forward? Is there something about your code that would be helpful for them going forward?

If you find that you have lots of extra time, pick another data set and try out some different operations than the ones you did previously. We are going for reps here.