**Data Science for Social Scientists**

PSYC 546, Spring 2023

Week 2 – In-Class Assignment

**Due Date**: January 26th (by 11:59 PM)

**Reminder**: See the assigned Week 1 readings and the Week 2 Lecture Slides for a tutorial on how to use R to perform the various functions included in the in-class assignment below. **Once completed, you should submit a completed version of this document and your final R script file to the Week 2 – In-Class Assignment – Submission Portal on Canvas**.

**R/RStudio**

Your submitted R script file should contain code to answer all of the questions below. Please use comments (e.g., #Question 1) to label the code for each question.

Questions 1 - 2 will use the survey.csv data set on Canvas.

1. You are interested in the variable **age\_group**. Values on this variable are 0 = younger adults, 1 = middle-aged adults, and 2 = older adults. Check the class type of this variable. You realize you need the variable to be treated categorically (i.e., as a factor) for your analyses. Coerce the variable to be a factor and then make sure that you apply this coercion to the survey data set for the age\_group variable. You should be able to check the class type of the variable again to make sure it worked. [2 points]
2. It is time to enter the *tidyverse*. Load the tidyverse package to your current RStudio environment. Imagine that you need to report the average (mean) and variability (standard deviation) in **age** for those with a child (**child** = 1) and those without a child (**child** = 0). Look about grouping the data based on child status and report the summary statistics of mean and standard deviation. Make sure to use the pipe operator for this one (i.e., if done completely correctly, it should only take a single line of code to answer this question). Report the results below [2 points overall]:
   1. Mean age for those without a child: 34.7
   2. Standard deviation in age for those without a child: 15.2
   3. Mean age for those with a child: 41.3
   4. Standard deviation in age for those with a child: 8.56
3. Load the package **dslabs**. This package contains example data sets for data science practice. Use the data() function to load the **murders** data set. This data set contains FBI 2010 data on the total type of a particular homicide for each of the 50 states (and Washington D.C.). Let us gain practice in adding new columns to data sets both the traditional way and the tidyverse way. In both instances, we want to calculate the raw rate of homicides (total divided by state population).

First, the traditional way. Create a new variable named **murder\_rate\_traditional** in the data set that simply consists of dividing the total murders by population size. Second, use the mutate function in the tidyverse to create a new variable called **murder\_rate\_tidy** that also consists of diving the total murders by population size. After your done, you can check the murders data set to see the two new columns. If done correctly, these two columns should equal each other. [2 points]

1. Imagine you want to create an IF/ELSE statement that will classify a test grade as passing or failing. First, create an object called **test\_grade** and assign it a numerical value between 0 and 100.

Next, create an if/else statement that simply prints out “Pass” if the test grade is greater than or equal to 60, or “Fail” otherwise. After completed, feel free to run all the code to make sure that the appropriate pass/fail status is printed based on the test grade. You can also change the values to be above or below 60 and re-run the code to make sure the appropriate pass/fail status is always printed out. [2 points]

1. Now, let us get some experience creating a custom function. Imagine we want to create a function that simply takes a number and returns the square of the number. Name the function **squaring\_function**. The function will have a single argument (arg) that will end up either being a single number (or a vector of numbers).

In the body of the function, create an object called **output** and simply have the expression for this object be taking the number supplied to the function and squaring it. Then, return this output object.

Finally, after running the squaring\_function and making sure there were no errors, call this custom function with a single number supplied as the argument. If done correctly, the square of that supplied number should be printed out. You can double check with a calculator to make sure the answer is correct. [2 points]