**Data Science for Social Scientists**

PSYC 546, Spring 2023

Week 3 – In-Class Assignment

**Due Date**: February 2nd (by 11:59 PM)

**Reminder**: See the assigned Week 2 readings and the Week 3 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 3 – 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 - 4 will use the contacts.sav data set on Canvas.

1. Using the read\_sav() function within the haven package, import the contacts.sav data file into your R/RStudio environment. If you use the drop-down menu method in RStudio, please still paste the printed code from your console to your saved script file. [0.5 points]
2. Using the base-R traditional method of recoding variables, create a new variable named **dept\_recoded**. Recode the dept variable. In the new variable, “Development” and “Finance” should be coded as 0, “Computer Services” should be coded as 1, and “Other” should be added to missingness (NA). Report the frequencies for the two groups [2 points overall]:
   1. Development and Finance: 29
   2. Computer Services: 30
3. Using a tidyverse method of your choosing, recode the rank variable into a new variable called **rank\_recoded**. Specifically, “Employees” should be coded as 0 on the new variable, Jr. and Sr. managers should be coded as 1, and VPs and Presidents/CEO/CFO should be coded as 2. Report the frequencies for these three groups [2 points overall]:
   1. Employees (Group 0): 11
   2. Managers (Group 1): 29
   3. VPs and Presidents (Group 2): 19
4. Using any recoded method of your choice, recode the numerical sales variable into a new variable named **sales\_recoded**. Sales greater than or equal to 65 are considered high and should be assigned the value 1; sales less than 65 are considered low and should be assigned the value 0. Report the frequencies for the two groups [1 point overall]:
   1. Low (0): 56
   2. High (1): 14
5. Manually create a data frame called **swl** that contains responses on a satisfaction with life self-report scale. The data frame should consist of the participant ID and their responses on the three items (reported below) [2 points overall]:

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **item1** | **item2** | **item3** |
| 1 | 4 | 5 | 4 |
| 2 | 2 | 4 | 3 |
| 3 | 3 | 2 | 2 |
| 4 | 4 | 4 | NA |
| 5 | 5 | 3 | 4 |

1. Create a new column in the swl data frame called **swl\_mean** that consists of the row mean for each participant on the three satisfaction with life items. In this example, all participants should get a mean scale score, including any participant with missing data. [1 point]
2. Sort the data frame ascending based on the swl\_mean column and then apply this sorting to the swl data frame object (i.e., overwrite the original swl data frame with the sorted one). [0.5 points]
3. Slice the swl data frame so that you print out the swl\_mean value for the participant with an ID of 5. Yes, with a small data frame like this, you could just look at the data frame to quickly see this value. But let us imagine you manage a data base with thousands of individuals and had to carry out a similar task. In this example, you do not need to save this value into a new object, simply print it to your console. [1 point]