Assignment #2 Topic 2 Data Wrangling
CHEM 5450/6450 Spring 2022

**Please save this file as "LAST NAME_Assignment 2.docx"*

Open-Ended Responses

1. **II.1** Which of the following would **not** be an example of data wrangling? (Highlight correct answer)
    a. Filtering a tibble based on a condition
    b. Converting data to long form
    c. Computing a t-test statistic
    d. Adding new variables to a data.frame object
    e. Joining two tibbles together

2. **II.2** We discussed how tidy data is rectangular. Why doesn't R allow you to have data.frames or tibbles that are non-rectangular? For example, the following will generate an error:

```
tibble(x = 1:10, y = 1:9) # error
```

Because tibbles are designed to use list-columns and not rows.

3. **II.3** Imagine my rawdata.csv file lives in a folder like Documents\Work\Dissertation\Survey. When importing this data, what is the advantage to running lines 2-3 as opposed to just line 1?

```
1 read_csv("Documents/Work/Dissertation/Survey/rawdata.csv")
2 setwd("Documents/Work/Dissertation/Survey")
3 read_csv("rawdata.csv")
```

Setting your work directory allows all future read and write lines to not have to include the entire directory every time

4. **II.4** Convert the following code into (a) full length sentence(s) that accurately and completely describe what is happening:

```
copus %>%
  filter(!is.na(Size), Level == "100") %>%
  select(Size, Level, L) %>%
  group_by(Size) %>%
  summarize(Max = max(L), Min = min(L)) %>%
  mutate(Range = Max - Min)
```

Read the data set called copus;
Then filter the data set to remove NA values and just include Level values of 100;
Then select and group by size;
Then summarize within the group and assign a max and min value with the max min functions;
Then make a new Range variable subtracting the max from the min.

5. **II.5** I collected 3 measurements from 3 different participants; the data are shown below. Highlight the word that correctly completes the sentence:

   Participant 1 has explicitly/ implicitly / no missing data; Participant 2 has explicitly / implicitly / no Participant 3 has explicitly / implicitly / no missing data.

Participant	Measurement	Score
1	1	10
1	2	9
1	3	10
2	1	8
2	2	7
3	1	9
3	2	NA
3	3	10

6. **II.6** I have two tibbles, df1 and df2 that I'm looking to join by a common ID. I did this two ways, shown in the code below. Besides the fact that option #2 is more efficient (less code), what is an advantage to option #2?

```
df1 <- tibble(ID = 1:10, Score1 = rnorm(10))
df2 <- tibble(ID = sample(1:10), Score2 = rnorm(10))

# join option #1
df.joined <- df2 %>%
  arrange(ID) %>%
  mutate(Score1 = df1$Score1) %>%
  select(ID, Score1, Score2)

# join option #2
df.joined <- df1 %>%
  left_join(df2, by = "ID")
```

Option 2 is using left join that will merge all the rows from the the left side and any matching rows from the second table for you.

II.9 Some people don't like the default `TukeyHSD()` function in base R; they find the `multcomp::glht()` function more efficient. If I ran a general anova model and then wrapped it in a `glht()` call and stored that in an object called comp, how could I access a table of the covariance matrix?

```
If it is stored in an object called comp you can just double click it from
your environment for a tab to appear with the table.
```

Coding Section

To complete this section, start a new script file with the following layout:

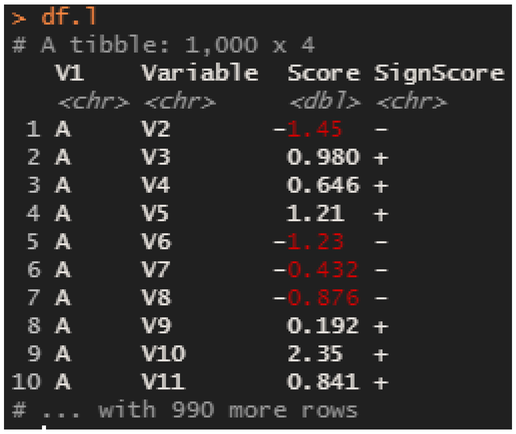
```
# YOUR NAME
# Assignment 1 Introduction to R

# #1 ---------------- (new section: CTRL + SHIFT + R)
here's my code # with adequate commenting

# #2 ---------------- (new section: CTRL + SHIFT + R)
here's my code # with adequate commenting
```

For the following questions, use the data from "df.csv" found in the data director on GitHub.

1. **II.3** Import the df.csv data to an object called "df".

2. **II.4** In one continuous statement connected by piping operators, making the following changes to df and store it as df2:
   a. Rename "X1" as "ID"
   b. All negative values in Variables V2 though V6 should have been positive (all values will be positive for these variables); change this
   c. Any obs with V7 less than -0.9 is an outlier; replace these instances with NA, but do not remove the rest of their data. *Hint: If you are getting an error such as "`false` must be a logical vector, not a double vector," check out the following and replace NA with NA_real_ (https://stackoverflow.com/questions/53636644/r-if-else-assign-na-value/53636776).*
   d. Get rid of all observations where V1 is "D"
   e. Sort the data by V1 (A to C) and within each category, in decreasing V2
   f. Calculate the mean and standard deviation of V2 for each category of V1

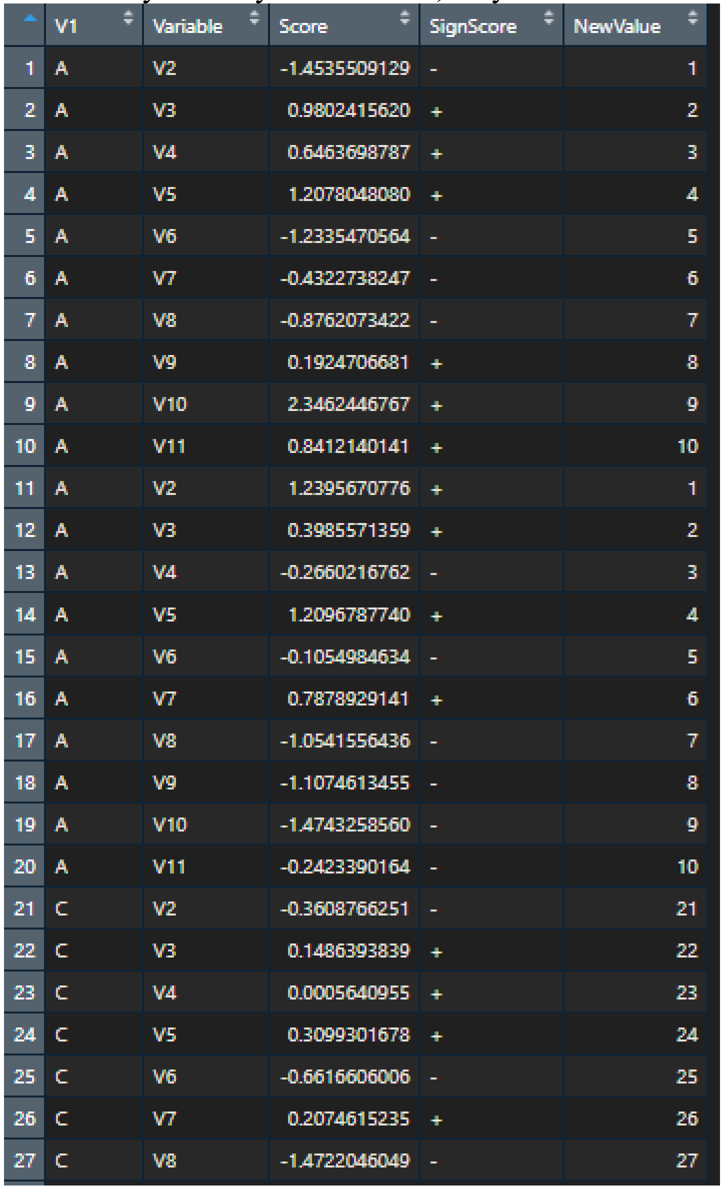
3. **II.5** Ignoring all modifications in #2 and starting from a fresh import of df.csv, modify the data into an object called df.l that matches the screenshot here:



```
> df.l
# A tibble: 1,000 x 4
   V1    Variable  Score SignScore
   <chr> <chr>     <dbl> <chr>
 1 A     V2       -1.45  -
 2 A     V3        0.980 +
 3 A     V4        0.646 +
 4 A     V5        1.21  +
 5 A     V6       -1.23  -
 6 A     V7       -0.432 -
 7 A     V8       -0.876 -
 8 A     V9        0.192 +
 9 A     V10       2.35  +
10 A     V11       0.841 +
# ... with 990 more rows
```

4. **II.6** Create this dataset in R (call it "key") and join it to df.l (from #3) so every obs in df.l that has V1 = "A" and Variable = "V2" should be assigned a NewValue = 1; every obs in df.l that has V1 = "A" and Variable = "V3" should be assigned a New Value = 2… and

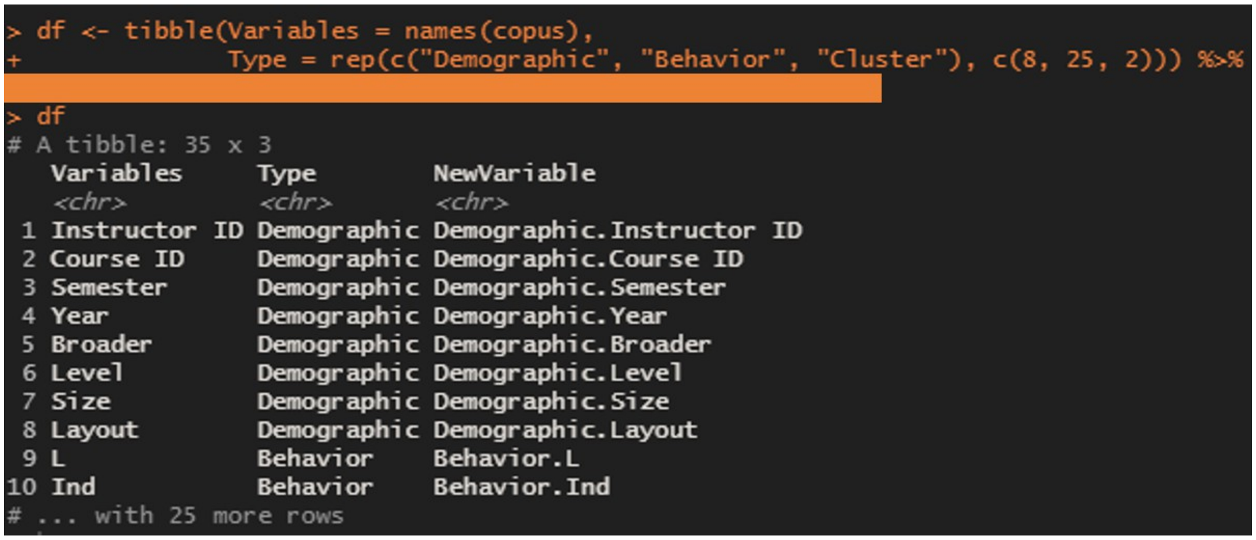
so on. Here is the desired, joined data. While you can do this with a LOT of chained if_else statements… do you really want to? No, no you don't.



	V1	Variable	Score	SignScore	NewValue
1	A	V2	-1.4535509129	-	1
2	A	V3	0.9802415620	+	2
3	A	V4	0.6463698787	+	3
4	A	V5	1.2078048080	+	4
5	A	V6	-1.2335470564	-	5
6	A	V7	-0.4322738247	-	6
7	A	V8	-0.8762073422	-	7
8	A	V9	0.1924706681	+	8
9	A	V10	2.3462446767	+	9
10	A	V11	0.8412140141	+	10
11	A	V2	1.2395670776	+	1
12	A	V3	0.3985571359	+	2
13	A	V4	-0.2660216762	-	3
14	A	V5	1.2096787740	+	4
15	A	V6	-0.1054984634	-	5
16	A	V7	0.7878929141	+	6
17	A	V8	-1.0541556436	-	7
18	A	V9	-1.1074613455	-	8
19	A	V10	-1.4743258560	-	9
20	A	V11	-0.2423390164	-	10
21	C	V2	-0.3608766251	-	21
22	C	V3	0.1486393839	+	22
23	C	V4	0.0005640955	+	23
24	C	V5	0.3099301678	+	24
25	C	V6	-0.6616606006	-	25
26	C	V7	0.2074615235	+	26
27	C	V8	-1.4722046049	-	27

Question 5 is about a **<u>different</u>** df dataset (defined below):

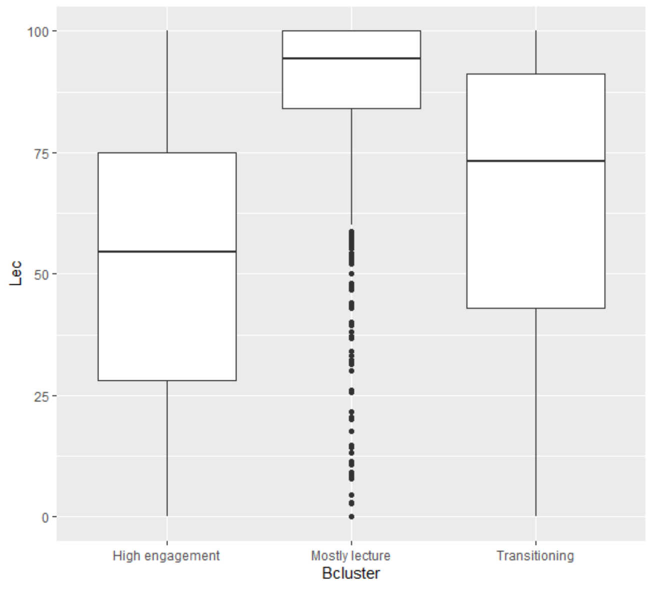
5. **II.7** Study the code section below that has one line that has been redacted. What is the redacted line?



```
> df <- tibble(Variables = names(copus),
+              Type = rep(c("Demographic", "Behavior", "Cluster"), c(8, 25, 2))) %>%

> df
# A tibble: 35 x 3
   Variables     Type        NewVariable
   <chr>         <chr>       <chr>
 1 Instructor ID Demographic Demographic.Instructor ID
 2 Course ID     Demographic Demographic.Course ID
 3 Semester      Demographic Demographic.Semester
 4 Year          Demographic Demographic.Year
 5 Broader       Demographic Demographic.Broader
 6 Level         Demographic Demographic.Level
 7 Size          Demographic Demographic.Size
 8 Layout        Demographic Demographic.Layout
 9 L             Behavior    Behavior.L
10 Ind           Behavior    Behavior.Ind
# ... with 25 more rows
```

Finally, Question 6 pertains back to the copus.csv data:

6. **II.8** One problem (of many) with the following plot is that the x-axis should be presented in the order of “Mostly lecture”, followed by “Transitioning”, and finally “High engagement”. Modify the copus data (Bcluster is on the x-axis) so that this variable will show up correctly in the plot.


Lec
High engagement
Mostly lecture
Transitioning
Bcluster