Week 7th Homework # 6 *Biostatistics 203A* *Yenlin Lai*

Due date: Thursday November 18th @ 6PM

1. Using TED talk data from Week 7, replace the ted$speaker of "Hans Rosling" to your name:

```
ted$headline[ted$speaker == "Hans Rosling"]
```

```
> ted <- read.csv("D:/UCLA Biostat/Fall 2021/Biostat 203A/Assignment6/TED_Talks.csv", header = T)

> ted$headline[ted$speaker == "Hans Rosling"]

[1] "The best stats you've ever seen"

[2] "New insights on poverty"

[3] "Insights on HIV, in stunning data visuals"

[4] "Let my dataset change your mindset"

[5] "Asia's rise -- how and when"

[6] "Global population growth, box by box"

[7] "The good news of the decade? We're winning the war against child mortality"

[8] "The magic washing machine"

[9] "Religions and babies"
```

The above list represents the headlines for the TED Talks delivered by Hans Rosling.

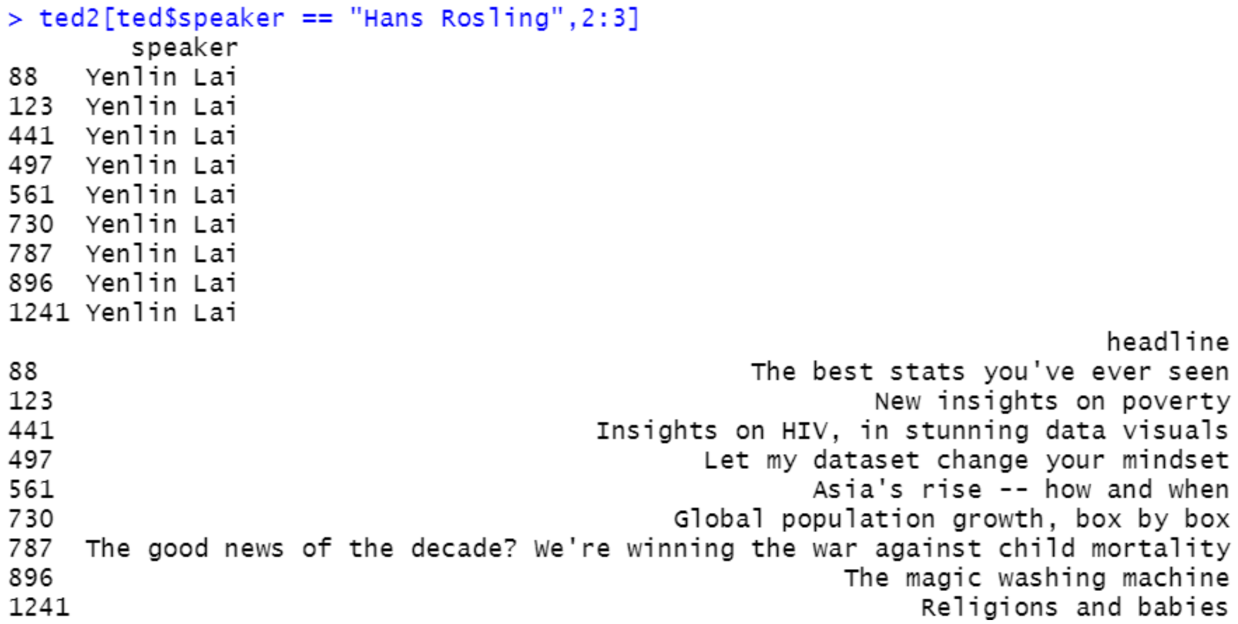
We now have all the tools we need to modify the ted2 data frame by replacing each instance of *Hans Rosling* in the speaker column with our own names. Remember to use the ted2 data frame (instead of the ted dataframe). Also, remember R's recycling rules. You should only have to type your name once in order to replace *Hans Rosling* with your own name all 9 times.

```
> ted2 <- ted

> ted2$speaker[ted$speaker == "Hans Rosling"] <- "Yenlin Lai"
```

After having made this modification, you can check the effectiveness of your code using the following:

```
ted2[ted$speaker == "Hans Rosling",2:3]
```



```
> ted2[ted$speaker == "Hans Rosling",2:3]
        speaker
88   Yenlin Lai
123  Yenlin Lai
441  Yenlin Lai
497  Yenlin Lai
561  Yenlin Lai
730  Yenlin Lai
787  Yenlin Lai
896  Yenlin Lai
1241 Yenlin Lai
                                                                       headline
88                                              The best stats you've ever seen
123                                                     New insights on poverty
441                                   Insights on HIV, in stunning data visuals
497                                          Let my dataset change your mindset
561                                                 Asia's rise -- how and when
730                                        Global population growth, box by box
787  The good news of the decade? We're winning the war against child mortality
896                                                   The magic washing machine
1241                                                       Religions and babies
```

This code should produce a data frame with 9 rows and 2 columns. The first column should contain 9 instances of your name. The second column should contain the 9 headlines for Hans Rosling’s TED Talks. Can you explain why the above command works?

We want to check if the modified data ted2 has replaced each instance of Hans Rosling in the speaker column with our own names, and contains 9 of instances. The command ted2[ted$speaker == "Hans Rosling",2:3] shows that the second and third columns of ted2 data (the variables of speaker and headline) where the observations of rows are “Hans Rosling” in the original data ted. This command can show if we successfully replace Hans Rosling with our own names.

Logical subsetting is very powerful because it allows you to quickly and easily identify, extract, and modify individual values in your data set.

Let’s try to exercise our logical subsetting skills a bit:

- Create a new data set called ted.17 that contains only the TED Talks that occurred in 2017. How many talks does this represent?

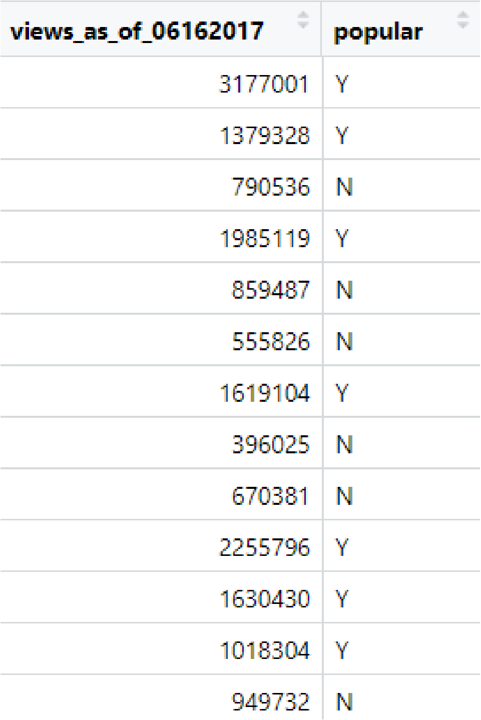
  ```
  > ted.17 <- ted[ted$year_filmed == 2017,]
  > nrow(ted.17)
  [1] 39
  ```

  There are 39 TED talks that occurred in 2017.

- In the ted2 data frame, create a new variable called popular that contains a Y if the talk exceeded a million views as of 6/16/17 and N if the talk did not. (Hint: you may want to start by creating a new column and setting all values to N.

You can then update the column in a second command to place the values of Y where appropriate).

```
> ted2$popular <- "N"
> ted2$popular[ted2$views_as_of_06162017 >= 1000000] <- "Y"
> ted2$popular[1:10]
 [1] "Y" "Y" "N" "Y" "N" "N" "Y" "N" "N" "Y"
```



views_as_of_06162017	popular
3177001	Y
1379328	Y
790536	N
1985119	Y
859487	N
555826	N
1619104	Y
396025	N
670381	N
2255796	Y
1630430	Y
1018304	Y
949732	N

2. PHQ-9 Example

Let’s test out our understanding of if and else statements.

The 9-Item Patient Health Questionnaire (PHQ-9) is a questionnaire administered to patients in a primary care setting to assess for depression. Total scores on the PHQ-9 range from 0-27 and a score of 10 or higher indicates moderate depression severity and typically necessitates development of a treatment plan, including follow-up. Patients who score 10 or higher on the PHQ-9 at one assessment time point are typically assessed 6 months later and considered to have met the criteria for meaningful improvement if the following criterion is satisfied:

The score at 6-month follow-up is below 10 **OR**

The score at 6-month follow-up reflects a 50% or greater reduction relative to the score at baseline

How would we design a set of if and else statements to determine whether a patient met the criteria for meaningful improvement? You may want to start by calculating percent change and creating an object to store the result (meaningful improvement, yes or no).

For example:

```
phq9.1 <- 23

phq9.2 <- 12

phq9.pctchange <- (phq9.1 - phq9.2)/phq9.1

improve <- "N"
```

Test out the code you have written to ensure it provides the correct result for a variety of different phq9.1 and phq9.2 inputs.

```
> library(sas7bdat)
Warning message:
package ‘sas7bdat’ was built under R version 4.1.1
> phq9 <- read.sas7bdat("D:/UCLA Biostat/Fall 2021/Biostat 203A/Assignment6/phq9.sas7bdat")
> phq9.b <- phq9[phq9$BHC_TYPE_FFU == "BASELINE",]
> phq9.f <- phq9[phq9$BHC_TYPE_FFU == "FIRSTFOLLOWUP",]
> phq9.bf <- merge(phq9.b, phq9.f, by="IDNUM")
```

We first use the package *sas7bdat* to read the file ph9.sas7bdat. Then split the data phq9 to two data containing only BASELINE and FIRSTFOLLOWUP. Merge it again by IDNUM.

```
> phq9.bf$improve <- "N"
```

Create an object improve to store the result.

```
> phq9.bfnew <- phq9.bf[phq9.bf$PHQ9_TS.x >= 10,]
```

Patients who score 10 or higher on the PHQ-9 at one assessment time point are typically assessed 6 months later and considered to have met the criteria for meaningful improvement if the following criterion is satisfied:

The score at 6-month follow-up is below 10 OR

The score at 6-month follow-up reflects a 50% or greater reduction relative to the score at baseline.

Therefore, we ignore the patients whose score of BASELINE below 10, and create a new data set phq.9bfnew.

```
> phq9.bfnew$percentage = (phq9.bfnew$PHQ9_TS.x-phq9.bfnew$PHQ9_TS.y)/phq9.bfnew$PHQ9_TS.x
```

Compute the reduction relative to the score at baseline, and create an object percentage to store the result.

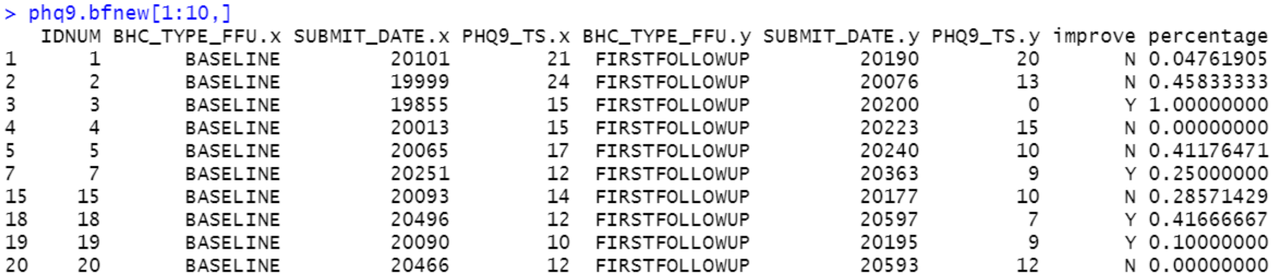
```
> for (n in 1:nrow(phq9.bfnew)) {
+   if(phq9.bfnew$PHQ9_TS.y[n] < 10 | phq9.bfnew$percentage[n] >= 0.5) {
+   phq9.bfnew$improve[n] <- "Y"
+   }
+   else {phq9.bfnew$improve[n] <- "N"
+   }
+ }
```

Write *if* and *else* statements in *for* loop, by the conditions: the score at 6-month follow-up is below 10 OR the score at 6-month follow-up reflects a 50% or greater reduction relative to the score at baseline. We give “Y” to improve variable meaning that these patients improved after 6-month follow-up.

```
> table(phq9.bfnew$improve)
 N  Y
39 30
```

We finally get 30 patients who got improvement after 6-month follow-up and 39 patients who did not.

Part of the result is shown below.



```
> phq9.bfnew[1:10,]
   IDNUM BHC_TYPE_FFU.x SUBMIT_DATE.x PHQ9_TS.x BHC_TYPE_FFU.y SUBMIT_DATE.y PHQ9_TS.y improve percentage
1      1       BASELINE         20101        21  FIRSTFOLLOWUP         20190        20       N 0.04761905
2      2       BASELINE         19999        24  FIRSTFOLLOWUP         20076        13       N 0.45833333
3      3       BASELINE         19855        15  FIRSTFOLLOWUP         20200         0       Y 1.00000000
4      4       BASELINE         20013        15  FIRSTFOLLOWUP         20223        15       N 0.00000000
5      5       BASELINE         20065        17  FIRSTFOLLOWUP         20240        10       N 0.41176471
7      7       BASELINE         20251        12  FIRSTFOLLOWUP         20363         9       Y 0.25000000
15    15       BASELINE         20093        14  FIRSTFOLLOWUP         20177        10       N 0.28571429
18    18       BASELINE         20496        12  FIRSTFOLLOWUP         20597         7       Y 0.41666667
19    19       BASELINE         20090        10  FIRSTFOLLOWUP         20195         9       Y 0.10000000
20    20       BASELINE         20466        12  FIRSTFOLLOWUP         20593        12       N 0.00000000
```