Statistics 147: IN-CLASS PRACTICE #1: SAS & R
Summer 2020; 10 pts

NAME: Wesley Chang **ID: (last 4 digits only)** 0996

GOALS:

- ♠ (In class practice) Execution of a simple program in *SAS Version 9 for Windows*, perform some calculations and create a high resolution plot.
- ♠ (In class practice) Read in and print out a data file in *R* and perform some calculations.

Data File needed for this exercise: ages.dat. The file is available on iLearn under **DataFiles**. The actual data starts on Line 2 in the data file.

```
Father  Mother
   23      20
   32      28
   35      33
   19      21
   20      18
   23      22
   23      21
   26      24
   25      26
   25      24
   25      25
   29      27
   24      22
   26      18
   24      21
   39      32
   31      30
   22      27
   25      24
   29      28
```

SAS

1. To invoke SAS, double-click on the **SAS System** icon on the desktop.

   RECALL: You will type your program in the **Program Editor** window!

2. Write a SAS program which, for given values of a variable x, will calculate the values of $y = x^3$ and print both x and y.

   Simply type the following in the **Program Editor** window. (See below.)

```
/* Set up output and graphics options */
options ls = 78 nocenter nodate nonumber ps=55 formdlim='*';
/* Use DM to clear all windows except the editor window */
DM log "odsresults; clear; out; clear; log; clear;";
ods graphics off;


/* Use the title command to generate main titles for your output */
title1 'Statistics 147, Summer 2020';
title2 'Your Name Goes Here';
title3 'SAS Practice Program 1';
```

```
/* Create temporary SAS data set and variable input list*/
data try1;
    /*  Use input statement to tell SAS what variables to read.
        @@ allows entry of more than  one piece of data per line */
    input x @@;

    /* Create new variable, which is equal to x cubed */
    y = x**3;

    /* Use the datalines command to indicate the data is about to follow */
    datalines;
    1 3 5 7 9 -8 -6 -4 -2 0
    ;
run;

/* Print the data as a check */
proc print data = try1;
    title4 'My very first SAS data step <3';
run;

/* See which variables/columns are in the dataset */
proc contents short data = try1;
run;


quit;
```

Reminder note: The two @@ on the end of the input line allow you to enter more than one pair of data values on a line. Without them, SAS would read exactly one value for x. OPTIONAL: Remove the @@ and see for yourself how the output changes.

3. To **save** select **File** → **Save As**. The **Save As** window open. In the box next to *Save as:*, click to get the sub-menu and select *Your name*. In the box next to *File name:*, type in **practice1** and click on **Save**. This will save your file and return you to the **Program Editor** window.

4. To **execute** your program:

   (i) Select **Run** → **Submit**

   (ii) **OR** click on the *Stickman Running Figure* on the main toolbar

   (iii) **OR** highlight the code you want to run (be sure to highlight the ‘run;‘ at the end), right click the highlighted portion, then choose *”Submit Selection”* from the options that appear.

   Upon executing the file in SAS, you will see information being written to the **log** window. This is a listing of all the errors/executions. If everything goes well, you will also get output written to the **Output** window. This window will take up the entire screen.

The output you should see is

```
Statistics 147, Summer 2020
Your Name Goes Here
SAS Practice Program 1
Obs     x       y
  1     1       1
  2     3      27
  3     5     125
  4     7     343
  5     9     729
  6    -8    -512
  7    -6    -216
  8    -4     -64
  9    -2      -8
 10     0       0
```

When your output appears on the screen, have a lab mate, TA or Luke check it, then write their initials here. __________________________

5. Let's create a high resolution plot of x vs y. We can use **proc gplot** for a high resolution plot that can be copied and/or saved. The general format for **proc gplot** is:

```
/*  FORMAT:
    proc gplot data = tempSasDataName;
        plot vertical * horizontal;  */
```

Thus, add the following lines code to your program, right before the **quit** statement:

```
/* Create a plot of y vs x using proc gplot */
proc gplot data = try1;
    /* Update title*/
    title4 'Plot of Y vs. X';

    /* VERTICAL axis comes FIRST in the plot command*/
    plot y * x;
run;
```

When your graph appears on the screen, have a lab mate, TA or Luke check it, then write their initials here. __________________________

6. You do not need to save your output file for this practice session. To **Exit** SAS, select **File** → **Exit** or click on the **X** in the upper corner. You will be asked *Are you sure you want to end the session?*. Click on **OK**.

R

NOTE: Comments in **R** begin with a # sign.

In a recent study, data was recorded concerning the ages at which married couples had their first child. Twenty couples were selected at random and the following data was recorded in the data file, **ages.dat**.

♠ Use the **read.table** command to read the data into **R**. **Be sure to change the path below to the location of your data file.**

Let's use an *R script* to enter our commands. Open **R**. From the main menu select **File** → **New script**. The **R Editor/Source** window will open. (It will say *untitled* until you save the script.)

★ Move the cursor to the **R Editor** window and type in the following:

```
# Statistics 147
# Summer 2020
# In Class R Exercise #1
# Your name goes here

# Use the read.table command to open and read in the data from the ages.dat file
# BE SURE TO CHECK IF YOU NEED TO USE THE ‘header‘ OR ‘skip‘ OPTIONS
#   BY OPENING/INSPECTING THE FILE
ages_df <- read.table("PATH/TO/YOUR/FILE/ages.dat", header= ???)
# Print to check
ages_df


# Use the attach() command to make the columns individually accessible variables
attach(ages_df)

# Use the names() command to obtain the column names
names(ages_df)
# Print the individual vectors to check
_________ # Write the name of the first column here
_________ # Write the name of the second column here
```

Make sure your cursor is in the **R Editor/Source** window.

▲ To **save** your script, from the main menu, select **File** → **Save As**. Select the location where you would like to save your script and type **inclass1_R_su20_XX**, where XX = initials of your name.

▲ Execute your script.

When your output appears on the screen, have a lab mate, TA or Luke check it, then write their initials here. ________________

♠ Find the difference in age between the Father and the Mother.

Make sure your cursor is in the **R Editor** window. Type the following block of **R** code in the **R Editor** window.

```
# Find difference in ages between Father and Mother
# Assign the result to a new variable named ‘age_diff‘
age_diff <- Father - Mother

# Print the results of the newly created variable
age_diff
```

Make sure your cursor is in the **R Editor/Source** window. Save your **R** script (Select **File** → **Save**.) and then

▲ highlight the new text you just typed.

▲ Hit **Command** + **Enter** (Mac) or **Ctrl** + **R** (Windows), **OR** from the main menu, select **Edit** → **Run line or selection**.

When your output appears on the screen, have a lab mate, TA or Luke check it, then write their initials here. ________________

♠ Use the sort function to sort the differences in ages and fill in the resulting values.

Make sure your cursor is in the **R Editor** window. Type the following block of **R** code in the **R Editor** window.

```
# Assign the sorted vector to a new variable called ‘age_diff_sorted‘
age_diff_sorted <- sort(age_diff)

# Print the results of the newly created variable
age_diff_sorted
```

Make sure your cursor is in the **R Editor** window. Save your **R** script (Select **File** → **Save**.) and then

▲ highlight the new text you just typed.

▲ From the main menu, select **Edit** → **Run line or selection**.

Complete the following from the **R Console** window.

```
> age_diff_sorted <- sort(age_diff)
> # Print the results
> age_diff_sorted

 [1] ________________________________________________________________
```

♣ What is the largest difference in age? ___8___

♣ What is the smallest difference in age? (Be careful!) ___0___

NOTE: The above sorted result has *negative* differences (when Mother is older than Father). To check our answer, let's sort the *absolute value* of the age gaps using the `abs()` function.

```
abs_age_diff <- abs(age_diff)
abs_age_diff

abs_age_diff_sorted <- sort(abs_age_diff)
abs_age_diff_sorted
```

You have now successfully completed **IN-CLASS PRACTICE #1: SAS & R**. Please submit your work on iLearn.

GREAT JOB!!!

Luke & Ruihan