Homework 5

Data

As with Homework 4, all the numeric values you need, other than `0.05`, `0`, `1`, `2` and `3` are defined below:

```
Year=c(1936, 1946, 1951, 1963, 1975, 1997, 2006)
CaloriesPerRecipeMean <- c(2123.8, 2122.3, 2089.9, 2250.0, 2234.2, 2249.6, 3051.9)
CaloriesPerRecipeSD <- c(1050.0, 1002.3, 1009.6, 1078.6, 1089.2, 1094.8, 1496.2)
CaloriesPerServingMean <- c(268.1, 271.1, 280.9, 294.7, 285.6, 288.6, 384.4)
CaloriesPerServingSD <- c(124.8, 124.2, 116.2, 117.7, 118.3, 122.0, 168.3)
ServingsPerRecipeMean <- c(12.9, 12.9, 13.0, 12.7, 12.4, 12.4, 12.7)
ServingsPerRecipeSD <- c(13.3, 13.3, 14.5, 14.6, 14.3, 14.3, 13.0)

CookingTooMuch.dat <- data.frame(
  Year=Year,
  CaloriesPerRecipeMean = CaloriesPerRecipeMean,
  CaloriesPerRecipeSD = CaloriesPerRecipeSD,
  CaloriesPerServingMean = CaloriesPerServingMean,
  CaloriesPerServingSD = CaloriesPerServingSD,
  ServingsPerRecipeMean = ServingsPerRecipeMean,
  ServingsPerRecipeSD = ServingsPerRecipeSD
)

sample.size <- 18
tenth.increment <- 0.10
hundredth.increment <- 0.100
idx.1936 <- 1
idx.2006 <- length(CaloriesPerRecipeMean)
idxs36_07 <- c(idx.1936,idx.2006)
alpha=0.05
```

Nearly all the same restrictions apply. Specifically

- There are 6 exercises. Choose 4 to be graded.
- One of the exercises must be completed in both SAS and R. Make sure you document this in the output.
- The other 3 exercises are to be complete in either R or SAS. Make sure you document this in the output.
- You may choose to work the other exercises. If you do, put these *after* the exercises you want graded. Otherwise, we'll grade the first four exercises and stop grading there. Time permitting, we'll provide feedback on the additional exercises.
- You are not required to write additional (other than previous Homework) functions for this exercise, but you may. You will be expected to clearly document additional functions - identify the expected inputs and outputs.
- There are no unit tests for this exercise. Where applicable, you should compare your results to comparable results in previous homework.
- This is an exercise in working with data tables. For three exercises, you will create data tables from sequences; for the last two you are expected to read data from files. One requires you to convert a data table to matrices, without calling `data.frame` directly.

- If you choose SAS, some of the exercises will require you to transfer data between PROC IML and the DATA step. You may need to redefine macros from previous assignments.

Exercise 1

Repeat the analysis from Exercise 1, Homework 4. This time, the results wil be in a data table with 49 rows. There will be 7 columns in the final table, `Year1`, `Year2`, `Mean1`, `SD1`, `Mean2`, `SD2` and `CohenD`. This table will have the same duplications as your matrix in Homework 4 (don't worry, we'll remove those in later exercises).

Part a

Create a data table where each row represents a different combination between years. It should look something like:

Year1	Year2	Mean1	Mean2	SD1	SD2
1936	1936	2123.8	2123.8	1050.0	1050.0
1946	1936	2122.3	2123.8	1002.3	1050.0
1951	1936	2089.9	2123.8	1009.6	1050.0
1963	1936	2250.0	2123.8	1078.6	1050.0
...	...	...	...	...	...
1936	1946	2123.8	2122.3	1050.0	1002.3
1946	1946	2122.3	2122.3	1002.3	1002.3
...	...	...	...	...	...

Start with the vectors defined in Data. You can reuse the matrices from the last homework, if you wish, or you can create new sequences while constructing the data table

If you do this exercise in SAS, use IML to create the vectors, the use `CREATE` to create a data table.

Part b.

Below is a wrapper function that accepts a vector as a parameter and returns Cohen's d for values in that vector. Modify this function so that it selects appropriate elements from the vector and calls **your* Cohen's d function from the previous homework. Assume that `table.row` is a row from the data table you created in Part a.

```
cohen.wrapper <- function(table.row) {
  return(cohen.d(table.row[3],table.row[5],table.row[4],table.row[6]))
}
```

If you choose SAS for this exercise, define a macro to implement Cohen's d formula, using syntax compatible with IML. This macro should have four parameters appropriate for Cohen's d.

Part c

Compute `CohenD` and add this to your table using `apply` and the wrapper function in Part b. Print this table and compare to the matrix you produced in Homework 4.

If you choose SAS, create a second data table, starting with the data table in part a (use SET in the data statement). Insert your macro in the body of this data step. Print your table. Use the names of your data table as parameters to this macro. Your macro will be replaced with the formula; assign the macro invocation a data variable `CohenD`.

Exercise 2.

In this exercise, we reproduce and extend the plot from Exercise 2, Homework 4.

Part a

Create a data table with a sequence from `x3 =` $\mu - 3\sigma \dots \mu + 3\sigma$ using increments defined by `tenth.increment`, as before. Name this column `X`.

If you choose SAS, do this step in IML.

Part b

Compute the likelihood and assign this to three columns, `L1`, `L2`, `L3`. These columns will correspond to the values computed for sequences x1, x2 and x3. To make columns `L1` and `L2` fit in this data table, pad the columns with NA values.

X	L1	L2	L3
$\mu - 3\sigma$	-	-	$L(x, \mu, \sigma)$
$(\mu - 3\sigma) + 0.1$	-	-	$L(x, \mu, \sigma)$
$(\mu - 3\sigma) + 0.2$	-	-	$L(x, \mu, \sigma)$
$\dots$	$\dots$	$\dots$	$\dots$
$\mu - 2\sigma$	-	$L(x, \mu, \sigma)$	$L(x, \mu, \sigma)$
$(\mu - 2\sigma) + 0.1$	-	$L(x, \mu, \sigma)$	$L(x, \mu, \sigma)$
$(\mu - 2\sigma) + 0.2$	-	$L(x, \mu, \sigma)$	$L(x, \mu, \sigma)$
$\dots$	$\dots$	$\dots$	$\dots$
$\mu - \sigma$	$L(x, \mu, \sigma)$	$L(x, \mu, \sigma)$	$L(x, \mu, \sigma)$
$(\mu - \sigma) + 0.1$	$L(x, \mu, \sigma)$	$L(x, \mu, \sigma)$	$L(x, \mu, \sigma)$
$(\mu - \sigma) + 0.2$	$L(x, \mu, \sigma)$	$L(x, \mu, \sigma)$	$L(x, \mu, \sigma)$
$\dots$	$\dots$	$\dots$	$\dots$
$(\mu + 3\sigma) - 0.1$	-	-	$L(x, \mu, \sigma)$
$\mu + 3\sigma$	-	-	$L(x, \mu, \sigma)$

A couple different approaches you might try:

- Use `L3` values for all columns, but change values to `NA` outside the appropriate range (you can use boolean indexes in R).
- Reuse the sequences from Homework 4 to build the data frame, concatenating with sequences of `NA` as needed.

If you choose SAS, costruct this data in IML, then use `CREATE` to create a data table.

Part d

Plot `L3` vs `X`, using formula syntax, with this data table as a parameter to plot, using points as the symbol.

Add points for `L2`, then `L1`, using different colors or symbols. Complete the plot by adding vertical lines at ±1 and ±2, using colors matching `L1` and `L2`, respectively.

If you use SAS, you will only need to call one SGPLOT block, with multiple series or scatter statements.

If you follow the instructions, you should have a graph of a normal probability distribution with different colors for the parts of the curve representing 1, 2 and 3 standard deviations.

Part d

Use apply (or similar function), compute the sum of columns L1, L2 and L3, multiplied by tenth.increment. Compare these values with the sums calculated for the previous exercise.

If you use SAS, you can use PROC SUMMARY for this step.

Exercise 3

Starting with 'CookingToMuch.dat', repeat the analysis from Homework 4, Exercise 5.

Append an appropriate column (named Intercept), with all values of 1, to CookingToMuch.dat, then use column indexes to extract appropriate X and y variables from CookingToMuch.dat. Do not create new matrices or frames for X and y. You may to coerce X or Y to matrices.

If you use SAS for this exercise, use the data table CookingToMuch, and use IML functions (USE/READ) to read from this data table into matrices.

Compute and print beta.hat as before, and compare to

```
lm(CaloriesPerRecipeMean ~ CaloriesPerServingMean,data=CookingTooMuch.dat)
```

```
##
## Call:
## lm(formula = CaloriesPerRecipeMean ~ CaloriesPerServingMean,
##     data = CookingTooMuch.dat)
##
## Coefficients:
##            (Intercept)  CaloriesPerServingMean
##               -166.923                   8.339
```

Change the eval flag in this for an alternative model.

```
lm(CaloriesPerRecipeMean ~ 0 + Intercept +  CaloriesPerServingMean,data=CookingTooMuch.dat)
```

Exercise 4

This exercise will be similar to Exercise 4 in Homework 4.

Part a

First, find the minimum and maximum values for ServingsPerRecipeMean.

Part b

Using your Poisson confidence interval function from Homework 3, Exercise 4, calculate the lower and upper bounds to the minimum and maximum means found in Part a. Set `LB` as the smallest (single value) of these bounds, set `UB` as the largest (single value) of these bounds.

Part c

Create a data frame with a column 'Y' as a sequence from `floor(LB)` to `ceiling(UB)`. Add to this data frame two columns, one the Poisson probability from Homework 4 using $\mu =$ the maximum servings per recipe mean, and the other using the minimum servings per recipe mean.

Part d

Plot both probability series against `Y`, using different colors. Use formula notation for the plots.

Add to this plot two vertical lines, located at the minimum and maximum mean values. Use the same color as the corresponding probablity curves.

Finally, add two pairs of lines corresonding to the upper and lower CI of the minimum and maximum means, calculated in Part b. Use different line types for these lines.

If you do this exercise in SAS, you can do parts a-c in IML, saving the matrices to a data table and use SGPLOT for part d.

Exercise 5

I was shopping for a motorcycle this spring, and in researching models, found a list of the fastest production motorcycles (https://en.wikipedia.org/wiki/List_of_fastest_production_motorcycles) . I edited this page to create a data table in CSV format.

Part a

Download the file `fastest.csv` from D2L and read the file into a data frame or table. Print a summary of the table.

Part b

To show that the data was read correctly, create three plots. Plot

1. Make vs Engine
2. Horsepower vs Engine
3. MPH vs Horsepower

These three plots should reproduce the three types of plots shown in the `RegressionEtcPlots` video, **Categorical vs Categorical**, **Continuous vs Continuous** and **Continuous vs Categorical**. Add these as titles to your plots, as appropriate.

Exercise 6

Part a

Go to http://www.itl.nist.gov/div898/strd/anova/AtmWtAg.html and download the file listed under `Data File in Table Format` (ttps://www.itl.nist.gov/div898/strd/anova/AtmWtAgt.dat)

Part b

Edit this into a file that can be read into R or SAS, or find an appropriate function that can read the file as-is. You will need to upload this file to D2L along with your Rmd/SAS files. Provide a brief comment on changes you make, or assumptions about the file needed for you file to be read into R/SAS. Read the file into a data frame or data table.

Part c

Calculate mean, sd and sample size for the two columns in this data; printing the results. You should store the values in variables. Use function(s) from Homework 3 to answer these two questions:

1. Is the difference between the two columns a small, medium or large effect size?
2. Is the difference between the two columns statistically significant?

Do this by printing function call(s) and results.