Stat 305 Project Template, Your Title Here

Your Name Here

This template was prepared by Professor T. Fiore.

Directions:

In this template you do **your individual coding** that you will submit in steps on Canvas. You will be graded on **your individual coding**. Later, your team will combine the best parts of the individual coding into one Rmd file together and write the report text. In that final report, all the code will be hidden by setting the global options appropriately (explained later), so that only the text and graphics and select output are displayed.

More Directions:

- Rename this file to have your name in it.
- Please remember to comment your codeblocks as you write them! When you are done, go back again and look at your codeblocks and add more comments.
- Use the "return" button to format your code so that there are no line overruns in the codeblocks in the pdf file. So in other words, in your Rmd file, press return after an appropriate comma, after the plus symbol while making a ggplot, and after an appropriate pipe symbol if using dplyr. You cannot put a line break in the middle of a string (e.g. in a filename in read.csv). All of your code should be readable in the knitted pdf file, except when it is physically impossible.
- Delete the remnants of Professor Fiore's code and text after you complete each section. In particular, all these instructions should also be deleted before you submit the first step. This should be a readable file and should not include extraneous set up remarks from Professor Fiore.
- Proofread the entire file from start to finish. Check for cohesion, check that all of Professor Fiore's code and comments are gone, and check that you have a unified document with a few proper English sentences and headings.

Tips:

- As you code your file throughout this project, frequently run your new code block to the console so that you can detect bugs early and fix them! To run a code block to the console, put your cursor in it, and do CTRL+SHIFT+ENTER. To run just the current line to the console, do CTRL+ENTER. The more you code without running to the console, the more difficult it is to find the bugs in the new material! By the way, when you resume work later, you can't run a block in the middle of your file, unless you run the blocks above it upon which it depends: use the button in RStudio "Run: All Chunks Above". Also, knit your file occasionally also to make sure there are no bugs!
- Use a shortcut to create a new code block with CTRL+ALT+I.

Teammates' Names: Write your teammates names here, even though they are not contributing to this file. Professor Fiore will compare the files to make sure each person does different graphs, different confidence intervals, and different hypothesis tests, and different models.

Research Question

Write your team's research question here.

External Requirements: Data Read-In and Package Loading

```
# read in the data in this codeblock

# first make sure this Rmd file and your csv file are in the same folder,
# you need to set the working directory under the Session menu (RStudio top)
# to the source file location

#df = read.csv("filename.csv")
```

```
# load any libraries in this codeblock, not later in the file,
# do not install packages in any Rmd file! Instead install packages
# at your console

#library(packagename)
library(ggplot2)
library(dplyr)
```

Count and Remove Some Missing Values as Appropriate (NAs)

First check how many NAs are in each column, and how many complete rows the data set has. I illustrate this with the built-in `airquality` dataset.

```
dg = airquality
# make a mental note on how many rows and columns we have at the start
dim(dg)
```

```
## [1] 153   6
```

```
# we see in the summary how many missing values in each variable
summary(dg)
```

```
##      Ozone           Solar.R           Wind             Temp
##  Min.   :  1.00   Min.   :  7.0   Min.   : 1.700   Min.   :56.00
##  1st Qu.: 18.00   1st Qu.:115.8   1st Qu.: 7.400   1st Qu.:72.00
##  Median : 31.50   Median :205.0   Median : 9.700   Median :79.00
##  Mean   : 42.13   Mean   :185.9   Mean   : 9.958   Mean   :77.88
##  3rd Qu.: 63.25   3rd Qu.:258.8   3rd Qu.:11.500   3rd Qu.:85.00
##  Max.   :168.00   Max.   :334.0   Max.   :20.700   Max.   :97.00
##  NA's   :37       NA's   :7
##      Month            Day
##  Min.   :5.000   Min.   : 1.0
##  1st Qu.:6.000   1st Qu.: 8.0
##  Median :7.000   Median :16.0
##  Mean   :6.993   Mean   :15.8
```

```
##  3rd Qu.:8.000   3rd Qu.:23.0
##  Max.   :9.000   Max.   :31.0
##
```

```
# more visibly we can see the number of missing values in each variable this way
colSums( is.na(dg)  )
```

```
##   Ozone Solar.R    Wind    Temp   Month     Day
##      37       7       0       0       0       0
```

```
# how many rows have 0 NAs, in other words, how many rows are complete?
sum( complete.cases(dg) )
```

```
## [1] 111
```

```
# how many rows are incomplete? Wow, 27% of the rows are incomplete rows.
sum( !complete.cases(dg)  )
```

```
## [1] 42
```

It is not a good practice to simply remove all rows with a missing value because that would unnecessarily throw out too much data! In this case 27%!

Let's say I am interested in just the relationship between `Ozone` and `Temp`. Then I would only want to throw away rows that are missing those, not rows that are missing other variables.

```
# take just Ozone and Temp columns
ozonetemp = select(dg, Ozone, Temp)
# only 37 rows are missing, instead of 42!
sum( !complete.cases(ozonetemp))
```

```
## [1] 37
```

```
# take only those rows that have both ozone and temp
ozonetemp = ozonetemp[ complete.cases(ozonetemp), ]
# confirm there are no missing values
colSums(  is.na(ozonetemp))
```

```
## Ozone  Temp
##     0     0
```

Cool, so we threw out only 37 rows instead of throwing out 42 rows.

Fortunately, a lot of R commands work even when there are NAs in the data frame, so sometimes we don't even need to remove the NAs. Keep them in mind as you work and deal with them appropriately according to whatever you are doing!

Dataset Description

Describe the dataset:

- the real source (don't say from class Canvas site or Kaggle)
- number of rows after removing the NAs, how many rows did you remove
- describe the variables of interest (you can refer to them by columnname using backticks), say their units. Probably you would have 5 to 10 variables.
- state which variables are categorical or numerical
- is this observational data, or a randomized experiment? Remember, from observational data you can infer association and correlation, but not causation. From a randomized experiment, you can infer causation. It is OK to use observational data, probably most data provided in class (or on Kaggle) is observational data.

Data Transformation

You will likely want to use `dplyr`'s `groupby` and `summarize` operations to create secondary data frames that you want to investigate or plot. Do that here in a code block or two or three or more.

Sometimes we need to reshape data between wide and long format, e.g. for specialized commands. Hopefully you wouldn't need that here, and hopefully your data is already in long format. If you do need to reshape, this is the section to do it.

Another thing that you may need to do is recode certain variables as factor variables!

```
# check that variables you think should be factors are factors!
str(dg)
```

```
## 'data.frame':    153 obs. of  6 variables:
##  $ Ozone  : int  41 36 12 18 NA 28 23 19 8 NA ...
##  $ Solar.R: int  190 118 149 313 NA NA 299 99 19 194 ...
##  $ Wind   : num  7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...
##  $ Temp   : int  67 72 74 62 56 66 65 59 61 69 ...
##  $ Month  : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ Day    : int  1 2 3 4 5 6 7 8 9 10 ...
```

```
# darn, it tells me the month is an integer not a factor, which
# can't be right, since I can't average months.
# we might also consider recoding day

# so let's recode it and check it is now a factor.
dg$Month = as.factor(dg$Month)
dg$Day = as.factor(dg$Day)
str(dg)
```

```
## 'data.frame':    153 obs. of  6 variables:
##  $ Ozone  : int  41 36 12 18 NA 28 23 19 8 NA ...
##  $ Solar.R: int  190 118 149 313 NA NA 299 99 19 194 ...
##  $ Wind   : num  7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...
##  $ Temp   : int  67 72 74 62 56 66 65 59 61 69 ...
##  $ Month  : Factor w/ 5 levels "5","6","7","8",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Day    : Factor w/ 31 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
```

Exploratory Data Analysis: Descriptive Statistics and Visualizations

1) **With your team create a table of summary statistics of the numerical variables** as sketched in the guidelines. Each person should do that here in his/her file, but it can be the same as the rest of the team because you work it out together. Here is how to do it for two numerical variables (you should generalize this to a loop using the aggregation pattern for more variables). Apply `favstats()` from the `mosaic` package to each numerical variable, stack the resulting one-row data frames using `rbind()` and then adjoin a column of variable names to the front using `cbind()`. (You can look back at the HW about weather stations if you forgot how `rbind()` works for stacking data frames with the same column names).

2) **Each individual** person makes here at least **3 different graphs** on your own. Discuss with the team ideas of who makes what so you don't make the same graphs. Make a variety of graphs, such as scatterplots, histograms, side-by-side boxplots, facets, etc. These graphs should be somehow related to your research question. Some plots will require you to do some data transformation first before making the plot (e.g. your plot shows the result of a `group_by` and `summarize` pipeline).
Very briefly describe notable patterns in the plots here in your file. The notable patterns should be related to your research question and hypothesis tests done in the next section. In your description avoid saying things like "this causes that" because a graph does not show causation. Instead, weaken your language to say things like "these two variables are associated" etc.
Do not just make random plots and overwhelm the reader. Instead, you should explore and make at least 3 plots and then include in the final report only the plots from here that show something. Be selective, do pick and choose.
Remember to give every plot a title and axes labels with units (if applicable). You may need to adjust the range on axes, but be careful you don't leave data out of the picture (`ggplot` will give a warning in that case).

3) Also do another numerical summary and comment about it, e.g. use the `summary()` command and `table()` command, and briefly comment about it. You may need to do a `group_by` and `summarize` first.

Statistical Analysis: Confidence Interval, Hypothesis Test, and Model or Machine Learning

1) **On your own,** create at least one bootstrap confidence interval. Discuss with your team so that you do NOT do the same confidence interval.

2) **On your own,** do at least one hypothesis test. Discuss with your team so that you do NOT do the same hypothesis test. Clearly state what the population is (remember, the sample is a subset of the population). You want to infer from the sample to the population, so select the appropriate population based on your dataset. Clearly state the hypotheses in terms of population parameters:
H_0 : blah blah blah
H_A : blah blah blah
State your conclusion for each test in a phrase like:
"We reject the null hypothesis. The data provide convincing evidence at the .05 significance level that. . . ."
or
"We fail to reject the null hypothesis. The data *do not* provide convincing evidence at the .05 significance level that. . . ."

3) **On your own,** fit at least one model, or use at least one machine learning technique. Discuss with your team so that you do not do the same thing.

One more thing

Do one more thing that interests you. Perhaps another graph, or another hypothesis test, or try to fit another model, or do a simulation to make a sampling distribution of an estimator, or get another data set from Kaggle and somehow link it to this one. **Be creative and push your boundaries!**

Conclusion

Write one paragraph that summarizes what you did in this Rmd file and reiterate your main finding(s).