Assignment #1

MSDS 593 - Summer 2018

***DUE**: Thursday, July 19, 2018, 09.15*

Instructions

Be sure to hand in a paper copy of the knitted `*.Rmd` file in class before quiz (printed double-sided, stapled in top-left corner), as well as upload **both** your `*.Rmd` file as well as the knitted `pdf` to Canvas by the due date and time. Late submissions will receive a grade of zero.

1. This homework is intended to be completed and submitted individually.
2. All code should be commented in a neat, concise fashion, explaining the objective(s) of individual lines of code.
3. When making reference(s) to *summary* results, include all relevant output in text of the deliverable where it is being discussed, not in an appendix at the back of the deliverable.
4. Do not include a copy of the raw data in the body of the deliverable unless there is a compelling reason.
5. `R` can generate hundreds of graphs and statistical output extremely easily. Only include *relevant* graphs and output in the deliverable. All graphs and statistical output included in the deliverable should be referenced in the text of the deliverable.
6. **There should be no orphaned figures or graphs**. Everything should be orderly and easy for a grader to read.
7. All code should be visible in the submitted, paper-version of the homework and `pdf` versions of the homework, i.e., for each code chunk, be sure to set `echo = TRUE`
8. Homework **may not be emailed to the instructor**. All homework should be submitted in class *and* uploaded to Canvas.

Question 1

1. Create the following vectors, populated with information about the four MSAN boot-camp classes
    - `courseNum` with all course numbers
    - `coursename` with all course names
    - `courseProf` with the names of the instructor for each course
    - `enrolled`, a logical vector indicating which courses you are formally enrolled in
    - `anticipatedGrade` with your anticipated letter grade in each course, with an `NA` indicating the course you are **not** enrolled in
    - `anticipatedHours` with your anticipated hours spent on each class per week based on on your experience during the first week, with an `NA` indicating the course you are **not** enrolled in

Create a **table** summarizing the `type` and `class` for each vector. The table should be generated using code, i.e., dynamically generated via code, **NOT** hard-coded.

2. Create a data frame called `bootcampDataFrame` by combining all of the above vectors and create another **table** summarizing the `type` and `class` for the data frame. Do the data frame variables retain their original types/classes?

3. Combine the vectors from 1.1 into a list called `bootcampDataList`, where each vector is an element of the list. Assign the names of each element to be the names of the original vectors. Do the elements of the list maintain their original types/classes?

4. Write code that returns the following values in code chunks using `echo = TRUE` so that your code as well as your output is displayed after each calculation:

- The total number of hours you anticipate spending on coursework, both per week, and over all of boot camp
- A data frame with only the third row and first two columns of `bootcampDataFrame`
- The first value in the second element of `bootcampDataList`

5. If you haven't already, convert the `anticipatedGrade` variable in `bootcampDataFrame` into an ordinal factor

   - What is the maximum letter grade you anticipate receiving in boot-camp?
   - What is the name and course number of that class? **n.b.** I want to see a single textual output with **both** course number and course name separated by a colon, e.g. `MSAN 593: Exploratory Data Analysis`

Question 2

1. Read in the file `titanic.csv` and store the data in the data frame `titanicData`.

Variable Name	Description
`survival`	Survival (0 = No; 1 = Yes)
`pclass`	Passenger Class (1 = 1st; 2 = 2nd; 3 = 3rd)
`name`	Name
`sex`	Sex
`age`	Age
`sibsp`	Number of Siblings/Spouses Aboard
`parch`	Number of Parents/Children Aboard
`ticket`	Ticket Number
`fare`	Passenger Fare
`cabin`	Cabin
`embarked`	Port of Embarkation (C = Cherbourg; Q = Queenstown; S = Southampton)

2. How many rows are in this data frame?

3. How many columns are in this data frame?

4. Which variable has the most `NA` entries?

5. Which variables, if any, should be converted to a different type than the default type they were imported as? Include of list of those you wish to change, what type they were previously, and what type you changed them to.

6. If you haven't already, coerce the `survived` variable into type `logical`.

   - What is the mean age of survivors?
   - What is the mean age of those who did not survive?
   - Plot side-by-side histograms of the ages of survivors and non-survivors.

7. Include the first 10 value of the `cabin` variable in this deliverable, observing that many are blank. Write and run a script that replaces all blanks in the **entire** data frame `titanicData` with `NA`s.

8. What percent of the observations for `age` are `NA`s? Replace all `NAs` with the mean age. This technique is called *imputation*. Google this term and list one downside for this particular method of imputation (you don't need write a thesis, just an intelligent sentence or two will suffice).

Question 3

1. The mean of a random variable $\sim \mathcal{U}\{a, b\}$ is $\frac{a+b}{2}$ and the variance is $\frac{(b-a)^2}{12}$
   - Generate 100 random variables $\sim \mathcal{U}\{-1, 1\}$ and compute the mean and variance (no need to set the seed for this exercise).
   - Repeat the previous step for sample sizes of 1,000, 10,000, 100,000 and 1,000,000, computing the mean and variance for each sample size.
   - Create a data frame called `unifDataFrame` with seven variables: `sampleSize`, `theoreticalMean`, `sampleMean`, `deltaMean`, `theoreticalVariance`, `sampleVariance`, `deltaVariance`, `deltaMean` and `deltaVariance` are the differences between the sample abd theoretical mean and variances respectively for each sample size. Be sure to popuate the daat frame using a loop, **not** manually.
   - Create a plot with `sampleSize` on the x-axis and `deltaMean` on the y-axis.
   - Create a plot with `sampleSize` on the x-axis and `deltaVariance` on the y-axis.
2. Create a vector of 10,000,000 random variables $\sim \mathcal{U}\{0, 1\}$ and store them in the vector called `myRunifVec`. Randomly sample and create a histogram 100,000 values from this vector. What is the distribution of the sample? Repeat this exercise a few more times to convince yourself that when randomly sampling from a $\mathcal{U}\{a, b\}$ distribution, the sample is also $\sim \mathcal{U}\{a, b\}$.
3. Create the data frame `myRunifDataFrame` with two variables, `col1` and `col2`. In each variable, store two different samples of 10,000,000 random variables sampled from a $\sim \mathcal{U}\{0, 1\}$ distribution. Create a third variable in `myRunifDataFrame` called `runifSum`, which is the sum of `col1` and `col2` and create a histogram. This is called a convolution. Notice how the shape of the distribution of the sum of two uniform variables looks nothing like the distribution of a uniform random variable.
4. Repeat 3, this time sampling from an exponential distribution with $\lambda = 1$. The convolution of two independent exponentially distributed random variables results in a Gamma distribution. Be sure to include a histogram of the distribution of the convoluted exponentially distributed random variables.

Question 4

n.b. Some (many?) of you may not be sufficiently familiar with regression at this stage of the prorgam. I have provided you with some—but not all—formulae below. Look up any formulae you are unfamiliar with on the web.

DO NOT USE ANY BASE R REGRESSION FUNCTIONS OR REGRESSION PACKAGES FOR THIS PROBLEM

Using the following code:

```r
set.seed(100)
x_1 <- runif(100000, -100, 100)
y_1 <- rexp(100000, rate = 0.5)
```

1. Manually compute the coefficients for the simple linear regression.

$$b_0 = \frac{1}{n}\left(\sum Y_i - b_1 \sum X_i\right)$$

$$b_1 = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{\sum (X_i - \bar{X})^2}$$

2. **Manually** compute SSE, SSR, $SSTO$ and compute the simple coefficient of determination, R^2.
3. **Manually** generate a scatter plot of y on x, and draw the fitted regression line on the same plot.
4. **Manually** compute the residuals for the fitted values.

5. **Manually** generate a residual plots of e_i on x; be sure to inculde a horizontal line at $e_i = 0$.

Repeat all of Question 4 using the following two code chunks:

```r
set.seed(999)
x_2 <- rnorm(100000, -100, 100)
y_2 <- rexp(100000, rate = 0.5)
```

```r
set.seed(543)
x_3 <- rnorm(100000, -100, 100)
y_3 <- rnorm(100000, -100, 100)
```

Lastly, create a table comparing the models and comment briefly on the comparison.

	Model 1	Model 2	Model 3
b_0			
b_1			
SSE			
SSR			
$SSTO$			
R^2			