Assignment #1

MSAN 593

***DUE**: Friday, July 22, 23:45*

Be sure to upload **both** an `*.Rmd` file as well as the compiled `pdf` to Canvas by the due date and time. Late submissions will receive a grade of zero.

Your `*.Rmd` file will be run on local machines by graders. If you file does not run, you will automatically lose 30% of the grade. If you resubmit your corrected homework by the last class of the module, it will be graded out of the remaining 70%. failure to resubmit will result in a grade of 0.

Always use `echo = TRUE` so that I can see all code, and include relevant results. You can also assume that the data file(s) being read into your `*.Rmd` file are in the current local directory, e.g., `read.csv('myFile.csv')` will work. Do **not** hardcode a specific directory structure.

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.

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 values in `num`, excluding the fourth value
   - 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	Descriptpion
`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 exrecise 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 dsitribtion 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.