Midterm Exam (STAT 5870 Big Data Analysis Using Python)

2:00 pm – 4:30 pm, Jun. 8

Instructions:

1. Write your code to each question in .py file.
2. After you finish all the problems, upload your .py file to Midterm Exam Dropbox in the course Elearning. In addition, upload a word file containing all the plots or upload every individual plot file.

Problem 1. (32 pts) Dr. Lee is planning to design a coin-tossing and a dice-rolling simulation to help undergraduate students to understand the concept of relative frequency interpretation of probability. Help Dr. Lee's teaching via solving following problems.

- Coin-tossing Simulation

(a) (2 pts) Create an one-dimensional array containing "H" and "T" and assign it to `S_CoinTossing`.

(b) (3 pts) Generate 10 random samples with replacement from the `S_CoinTossing` and assign it to `CoinTossing10`. Run `np.random.seed(seed=0)` before you generate random samples. Calculate the proportion of "H".

(c) (3 pts) Generate 100 random samples with replacement from the `S_CoinTossing` and assign it to `CoinTossing100`. Run `np.random.seed(seed=0)` before you generate random samples. Calculate the proportion of "H".

(d) (3 pts) Generate 10000 random samples with replacement from the `S_CoinTossing` and assign it to `CoinTossing10000`. Run `np.random.seed(seed=0)` before you generate random samples. Calculate the proportion of "H".

(e) (3 pts) Explain the trend you see from the above three proportions.

- Dice-rolling Simulation

(f) (6 pts) Write your own function to simulate the result of rolling a dice n times.

(g) (3 pts) Use your function to generate the simulation result of rolling a dice 10 times and draw a bar plot to visualize the number of counts for all six numbers. Run `np.random.seed(seed=0)` before you generate the simulation result.

(h) (3 pts) Use your function to generate the simulation result of rolling a dice 100 times and draw a bar plot to visualize the number of counts for all six numbers. Run `np.random.seed(seed=0)` before you generate the simulation result.

(i) (3 pts) Use your function to generate the simulation result of rolling a dice 10000 times and draw a bar plot to visualize the number of counts for all six numbers. Run `np.random.seed(seed=0)` before you generate the simulation result.

(j) (3 pts) Explain how does the result change when we increase the number of times of rolling a dice.

Problem 2. (28 pts) Dr. Lee is interested in doing a research on iris flowers. Help Dr. Lee's research via solving following problems. The csv file "`iris.csv`" is a data set containing measurements of iris flowers. The data set consists of 5 variables: `sepal_length`, `sepal_width`, `petal_length`, `petal_width`, and `species`.

(a) (1 pts) Import "`iris.csv`" to Python and assign it to `iris`.

(b) (3 pts) Dr. Lee is interested in finding linear correlation between two variables X and Y. You can use the below code to find Pearson correlation coefficient between two variables X and Y.

```
from scipy.stats import pearsonr
pearsonr(X, Y)[0]
```

Find the Pearson correlation coefficient between `sepal_width` and `sepal_length`. Is there a positive linear relation? Or a negative linear relation?

(c) (6 pts) Dr. Lee thinks the result from the previous problem is weird and wants to visualize the data. Draw a scatter plot with a fitted regression line. Put `sepal_width` on the x-axis and `sepal_length` on the y-axis.

(d) (6 pts) After checking the scatter plot, Dr. Lee thinks maybe Simpson's paradox occurred here. Simpson's paradox is a phenomenon in statistics, in which a trend appears in several different groups of data but disappears or reverses when these groups are combined. Draw a scatter plot with fitted regression lines. Put `sepal_width` on the x-axis and `sepal_length` on the y-axis and use different colors for different `species`.

(e) (6 pts) Find the Pearson correlation coefficient between `sepal_width` and `sepal_length` for each `species` separately.

(f) (6 pts) Use the results from (d) and (e) to make final conclusions on the relation between `sepal_width` and `sepal_length`.

Problem 3. (40 pts) Dr. Lee is interested in analyzing unemployment rate in United States. Help Dr. Lee's project via solving following problems. The csv file "`unemp.csv`" is a data set containing US unemployment rates by county from 1990 to 2016. The data set consists of 5 variables: `Year`, `Month`, `State`, `County`, and `Rate`.

(a) (1 pts) Import "`unemp.csv`" to Python and assign it to `unemp`.

(b) (3 pts) How many observations are in the `umemp` data set?

(c) (3 pts) Print the first 10 rows of the `unemp` data set.

(d) (3 pts) Check whether there are any missing values in the `unemp` data set.

(e) (4 pts) Find the mean unemployment rate for every year in the `unemp` data set.

(f) (6 pts) Draw a line plot with year on the x-axis and mean unemployment rate on the y-axis.

(g) (4 pts) Find a subset of `unemp` which contains all counties in Michigan in January 2016 and assign it to `unemp_MI_Jan_2016`.

(h) (6 pts) Use the subset `unemp_MI_Jan_2016` to find the top five counties in Michigan with lowest unemployment rate in January 2016.

(i) (4 pts) Find a subset of `unemp` which contains Kalamazoo County in Michigan from 2010 to 2015 and assign it to `unemp_Kzoo_MI_2010_2015`.

(j) (6 pts) Use the subset `unemp_Kzoo_MI_2010_2015` to draw a side-by-side boxplot to compare the unemployment rate from 2010 to 2015 in Kalamazoo county, Michigan.