**IE-507 Lab 5**
**23/08/2017**
**Instructions:** In this lab, we will practice more programming with Python. You will also be expected to make some histograms. In this lab, you will be required to save and upload any plots or figures that you generate from Python. Remember to include proper labels of axis, legends and plot titles in each figure.

You are required to work on your own and submit a report on paper.

*All questions marked with [R] should be answered in the report. Upload all your Python files and image files (png or eps or pdf ) on Moodle. Question 1 and 2 each carries 10 marks. Question 3 is optional. If anyone correctly completes all the questions(including 3), then he/ she will get a bonus 2 marks which will be adjusted in other labs.*

**We are following a strict deadline. So, any kind of excuse for late submission will not be considered at any level. Also, your report must be in sync with the code files and image files you have submitted on moodle.**

**Exercise 1**

(a) Write a function in Python ex1fun(a,m) which takes as input a column vector $a$ and a positive integer $m$. This function should create a vector $b$ such that the first element of $b$ is the average value of the first $m$ values of $a$, the second element is the average of the next $m$ values of $a$ and so forth. If less than $m$ values remain at the end, they may be discarded.The function should display the following:

(i) The number of elements in $b$.

(ii) The minimum value amongst all elements of the matrix $b$.

(iii) The maximum value amongst all elements of the matrix $b$.

(iv) The average value of elements of $b$.

(v) The standard deviation of elements of $b$.

(b) **[R]** Use this function to report the above five values for ex1a available on Moodle, and $m = 4$.

(c) **[R]** What is the likely distribution of the elements of array $b$ ? You may use the histplot function is Python to look at its histogram.

(d) **[R]** By appropriately choosing different values of $m$ and running the above function, comment how do each of the above five values change with $m$.

**Exercise 2**

Write a function which can simulate rolling a die with a provided number of sides. You will then use this function to simulate rolling multiple dice combinations, while tracking the number of times each possible number appears.

First, simulate rolling a single 6-sided die (possible die roll values are 1-6) 5,000 times. How many times does each number appear? Make sure your program reports these statistics.

Next, perform the same experiment, but with a single 12-sided die (possible die roll values are 1-12). Make sure your program reports how many times each number appears.

Finally, simulate rolling two 6-sided dice. Add the resulting roll together (possible values are 2-12), and track how many times each value appears. Again, make sure your program reports how many times each number appears.

[**R**] Graph your results, and submit a short write up (2 paragraphs) about why you see the results you found. Are they different or similar? Why?

**Exercise 3 : Fun with functions**

(a) Write a Python program to reverse a string.
Sample String : "1234abcd"
Expected Output : "dcba4321"

(b) Write a Python function to check whether a number is perfect or not.
According to Wikipedia : In number theory, a perfect number is a positive integer that is equal to the sum of its proper positive divisors, that is, the sum of its positive divisors excluding the number itself (also known as its aliquot sum).
Equivalently, a perfect number is a number that is half the sum of all of its positive divisors (including itself).
Example : The first perfect number is 6, because 1, 2, and 3 are its proper positive divisors, and 1 + 2 + 3 = 6. Equivalently, the number 6 is equal to half the sum of all its positive divisors: ( 1 + 2 + 3 + 6 ) / 2 = 6. The next perfect number is 28 = 1 + 2 + 4 + 7 + 14. This is followed by the perfect numbers 496 and 8128.