**ASSIGNMENT 2 (Summer Analytics 2019)**

*Make a Jupyter notebook including your code, plots, and comments and follow the same procedure as previous assignment to submit the assignment by giving github link in the google form provided.*

**Part 1: Real estate prices**

Your first task is to carry out a little bit of data exploration using Python tools.

*Houses.csv* is a CSV (comma-separated values) file listing real estate sales in England between 1995 and 2016. (Actually, to make things a bit faster it's only a subset.)

* Load the CSV file into Python.
* The second column in the CSV file represents the price of the property. Compute basic descriptive statistics about the prices in the whole dataset: mean, median, standard deviation, minimum, and maximum.
* Plot a histogram that shows the distribution of the prices. Why is it so ugly? What can you do to make it more informative?
* Is real estate more expensive in London? Plot histograms for the two subsets of properties inside and outside London, respectively. For practical purposes, we can define "inside London" to mean that the string in the 13th column includes the string LONDON.
* Make a plot that shows the average price per year.

**Part 2: Generating random numbers (quick detour)**

Consider the random number generation functions in NumPy

Use this link for syntax help:

<https://docs.scipy.org/doc/numpy/reference/routines.random.html>

* Generate a set of random numbers using the function rand and plot its histogram. What is the shape of this histogram and why?
* Investigate how the shape of the histogram is affected by the number of random numbers you have generated.
* Instead of using rand (which corresponds to a *uniform* distribution), generate numbers using some other distribution and plot a histogram. What is the shape now? For instance, with normal, the normal (or Gaussian) distribution, you should get the familiar bell shape,

## Part 3: Simulating probabilistic models (Optional)

### (a) The persistent student

We will now simulate a scenario where a student takes an exam repeatedly, until passing.

If a student does not pass an exam, IIT Guwahati allows the student to go to an unlimited number of re-sit exams. Let's assume that students never give up, so that they will go to the exam again and again until they finally pass. Write a function that simulates a student going to exams until passing, and returns the number of attempts the student needed before passing. You can assume that the probability of passing a single exam is a constant p\_pass. If you want, you can reuse your function success from the previous task: in this case, this would mean a passed exam, not just a correctly answered question.

def number\_of\_attempts(p\_pass):

... YOUR CODE HERE ...

**Investigating the distribution**

Simulate this model multiple times, as in (a). For instance, let p\_pass be 0.4. Plot the result using a histogram.

### (b) An unusual village

The inhabitants of Normlösa, a small village in the fertile plains of eastern Sweden, are infamous not only for their unscrupulous behavior but also because the males in the village are exceptionally short and stocky, while the female villagers tend to be tall and lean. Geneticists from nearby Linköping University have so far failed to come up with a credible explanation of this remarkable tendency.

Write a Python function to generate the height and weight of a random inhabitant of Normlösa. Use the following process:

* first, randomly select the gender of the villager; the proportion of males in this village is about 40%.
* then draw random numbers from a Gaussian distribution (normal distribution) for the height and weight of the person; for this, you might use the NumPy function np.random.normal(loc, scale), where loc is the mean and scale the standard deviation.
  + for males, the mean height is 140 and the height standard deviation is 15; the mean weight is 90 and the weight standard deviation is 10;
  + for females, the mean height is 195 and the height standard deviation is 10; the mean weight is 60 and the weight standard deviation is 5.

Generate a dataset consisting of height–weight pairs for 50 Normlösa inhabitants. Make a scatterplot of the height–weight data.

Let's pretend for a moment that you have been given the datapoints (the list of height–weight pairs) but you have no information about how they were generated. Could you think of a way to reconstruct the parameters you used in the code previously? For example, that the proportion of males is 40%, that the mean weight of a female is 60 kilograms, etc.