## HU Extension Assignment 01 E-63 Big Data Analytics

## 

### Handed out: 01/27/2017 Due by 11:30PM on Friday, 02/03/2017

This assignment is written using some function names that are present in R. However, you can write your solution in any language of your choice. I believe that Python and R are the most convenient. Python’s pandas are somewhat enhanced data frames. Scala and Java also have data frames. Use a language of convenience.

**Problem 1.** Plot the binomial distribution for p = 0.3, p = 05 and p = 0.8 and the total number of trials n = 60 as a function of k the number of successful trials. For each value of p, determine 1st Quartile, median, mean, standard deviation and the 3rd Quartile. Present those values as a vertical box plot with the probability p on the horizontal axis.

**Problem 2**. Let us finish the plot of the correlation between waiting times and durations of Old Faithful data. Recreate the scatter plot of waiting vs. duration times. As we mentioned in class, the best linear assessment in the sense of the least squares fit of a relationship (proportionality) between two or many variables can be achieved with function lm() in R, where lm stands for the linear model. The first argument of lm() is called formula accepts a model which starts with the response variable, waiting in our case, followed by a tilde (symbol ~, read as “is modeled as”) followed by the (so called Wilkinson-Rogers) model on the right. In our case we simply assume that waiting time is proportional to the duration time and that “model” reads: formula = waiting ~ duration. The second argument of function lm() is called data and, in our case, will take value faithful, the data set containing our data. Store the result of function lm() in a variable. The name of that variable is not essential. Call it model. Print the variable. The first component of that variable is the intercept of calculated line with the vertical axis (waiting, here) and the second if the slope of the line. Convince yourself that line with those parameters will truly lie on your graph. Function abline() adds a line to the previously created graph. Next, pass the variable model to the function abline(). Make that line somewhat thicker and red. Use help(functionName) to find details about invocations of both lm() and abline() functions.

**Problem 3.** You noticed that eruptions clearly fall into two categories, short and long. Let us say that short eruptions are all which have duration shorter than 3.1 minute. Add a new column to data frame faithful called type, which would have value ‘short’ for all short eruptions and value ‘long’ for all long eruptions. Next use boxplot() function to provide your readers with some basic statistical measures for waiting and then in a separate plot for duration times. Please note that boxplot() function also accepts as its first argument a formula such as waiting ~ type, where waiting is the numeric vector of data values to be split in groups according to the grouping variable type. The second argument of function boxplot() is called data, which in our case will take the name of our dataset, i.e. faithful. Find a way to add meaningful legends to your graphs.

**Problem 4**. Find a way to generate a random variable with a uniform probability between -1 and 2. Create a histograms with 20 bars to convince yourself that generated values truly fall under a uniform distribution. Create a histogram presenting the relative cumulative distribution of generated data.

**Problem 5.** Create a matrix with 40 columns and 100 rows. Populate each column with random variable of the type created in problem 4. Do not create each vector manually. Try to find a way to present two distributions contained in any two of the columns of your matrix on a single plot. To do that you might want to export the distribution data from two columns into two stand-alone vectors of equal length, e.g. y1 and y2. Plot one distribution first using a call to plot(x,y1), where vector x contains the parameter vector with values between -1 and 2 you selected above. To add the next curve (distribution y2) try invoking function lines(x,y2). To improve your diagram, present two curves in different colors and add labels on x and y axis, as well as the title to your graph.

**Problem 6**. Start with your matrix from problem 5. Add yet another column to that matrix and populate that column with the sum of original 40 columns. Create a histogram of values in the new column showing that the distribution starts to resemble the Gaussian curve. Add a true, calculated, Gaussian curve to that diagram with the parameters you expect from the sum of 40 random variables of uniform distribution with values between -1 and 2.

SUBMISSION INSTRUCTIONS:

Your main submission should be an MS Word document containing your code, results produced by that code and brief textual descriptions of what you did and why. Typically, you just copy your code and results and past them into this Word document. In other words start with text of this homework assignment as the template. Please add any other files that you might have used or generated. Please do not provide ZIP or RAR or any other archives. Canvas cannot open them and they turn into a nuisance for us.