Assignment 03
Assigned Monday Feb. 1, 2016
Due by the start of lab Monday, Feb. 8, 2015

To be turned it...
1) Script with the functions created is necessary, along with another
2) Script that executes the functions asked for in this lab
3) Written summary of lab.

The objective of this lab is to familiarize yourself with the creation of functions in R, both functions that you have used before and a few that you have not encountered before.

All functions should be contained in a script called "functions.r" and stored in your lab folder, lab_03. You'll create a second script called "execute.r", whose first line will be "source('functions.r')" and the following lines will call the functions as required of you in the questions below.

The “source()” function is used in an R script or at the command line to execute another script, which is the argument of “source()”. Use the commands learned in class, as appropriate, to solve the problems. Look up any terms you do not understand (help in R using ? or help(topic), google, and TA, in that order.) You can also try the function example. Type help(example) to find out more about this function.

All functions should have "reasonable" names.

Scoring: each question has equal weighing.

Task 1
Create a function that sums the squares of all elements of a vector, and returns the sum to the user. The vector should be the sole argument to the function.

Task 2
Create a function that sums the cubes of all the elements of a vector, and returns the sum to the user. The vector should be the sole argument to the function.

Task 3 **Tas□**
The plot command with two arguments draws a scatter plot of y (ordinate) against x (abscissa).

Create a function that first creates a vector of n random numbers, and then a second vector where each random number is squared. Plot a scatter plot of the first vector against the second vector. Do "help(plot)" to learn the basics of the plot command. Use Google to find somcsicW2"d of pq□Do

Extra credit: use a conditional statement (the function if() to check whether the first two vectors have equal length. (worth 10 points).

Task 5

Create a function that returns all the even numbers between *m* and *n*, assuming that *m* is even and *n* is odd with *m* and *n* are two arguments to the function. Make sure that *n* is always greater than *m* when calling the function. Later in the class, we will use "if" statements to do this check. For now, just call the function while paying attention to this yourself. The return value for the function is a vector that contains the even numbers. For example, if *m* = 6 and *n* = 11, your function should return (6, 8, 10). If *m* = 8 and *n* = 13, the function returns (8, 10, 12). Test this function with *m* = 126 and *n* = 255 and provide the results in your report.

Task 6

Create a function that plots the results of the function written in task 5. So, given the two arguments *m* and *n*, generate a graph (using the "plot()" function) showing the integers versus the index. Example: if *m* = 4 and *n* = 11, the plot should contain the points (1, 4); (2, 6), (3, 8), (4, 10). Use the "main=" argument from the plot command to create a title for your plot. If you can figure it out, create names for the x and y axes. Save the plot to a file, and include it in your report along with scripts.