Homework Submission Practice (HW0)

Automated Learning and Data Analysis
Dr. Thomas Price

Spring 2019

Submission Instructions

- **Due Date:** January 22, 2019 at 11:45 PM
- **Total Points**: 5 bonus points on HW1 **if** you submit this assignment and follow all instructions correctly on HW1
- Make sure you clearly list each team member's names and Unity IDs at the top of your submission.
- Your submission should be a single zip file containing your code and a README file with running instructions. Please follow the naming convention for your zip file: *G(hw group no.)_HW(hw no.)*, e.g. *G1_HW0*. **Your code should be named** *hw0.R*.
- Failure to follow naming conventions or programming related instructions specified below may result in your submission not being graded.
- If the instructions are unclear, please post your questions on piazza.

Programming related instructions

- Carefully read what function names have been requested by the instructor. In this homework or the following ones, if your code does not follow the naming format requested by the instructor, you will not receive credit.
- For each function, both the input and output formats, as well as an example will be provided. Please ensure that you follow the correct input and output formats. Once again, if you do not follow the format requested, you will not receive credit.
- Apart from library functions (library/require), seed function, comments and function definitions, your code file should not contain any other function calls/code. If you wish to verify the correctness of your functions, do so in another file which does not need to be submitted. You are welcome to write your own helper sub-functions, but they should be called from within the TA-requested functions only.
- DO NOT set working directory (`setwd` function) or clear memory (`rm(list=ls(all=T))`) in your code. TA(s) will do so in their own auto grader.
- The TA will have an autograder which will first run source(hw0.R), then call each of the functions requested in the homework and compare with the correct solution.
- Your code should be clearly documented.

Software Installation

The TA(s) will be using the following software throughout the course unless otherwise stated.

- **R Version:** Microsoft Revolution R Version 3.5.1.
- **Operating System:** Ubuntu 16.04 or Ubuntu 18.04

- **R IDE:** R Studio Desktop (free version)

If you do not already have the version of R and R Studio specified above, download and install them on your machine. If you already have an older version of R that you do not wish to update, we recommend setting up a virtual machine with the operating system specified above. If you are facing trouble installing R or RStudio, visit the TAs during their office hours. **Ensure that the packages you use are compatible with the given version of R.**

Learning Goals

By the end of this exercise, you should:

- have installed all the necessary softwares required for this course homeworks.
- have explored R data types and data structures.
- be able to create functions in R based on user requirements.
- have explored visualization techniques in R.
- be able to create executable code in R.
- be familiar with submitting programming assignments for this course.

Problems

1. (**Krishna Gadiraju**) In this homework, you will be creating two functions named *intro_to_r* and *intro_to_plotting*. You are already provided *hw*0.*R* template. Using the information provided below, complete the functions in the template. The input, output arguments and their types as well as function requirements have been discussed below. (Note the lower case notation). The variable names and types are given the format [variable name : data type]. Read up on the difference between vectors, lists, named lists and data frames in R before you start this problem.
   Note 1: double vector means a vector of type double.
   Note 2: set seed to 123.

   (a) *intro_to_r*: Create a function called *intro_to_r* which will take an input (*num_values*). Your function should create a random vector of size *num_values* named *new_vector* and should calculate the mean (*new_mean*), median (*new_median*), max (*new_max*) and min (*new_min*) values of this vector. You will then return a list in the following order [*new_vector*, *new_mean*, *new_median*, *new_max*, *new_min*]. Input and output data types are described in further detail below.
   Input: *num_values : integer*
   Output: a list: [*new_vector : double vector*, *new_mean : double*, *new_median : double*, *new_max : double*, *new_min : double*]
   Example: *intro_to_r*(10) should generate a random array of size 10 and of type double, calculate its mean, median, max and min and return these values in the format specified above.

   (b) *intro_to_plotting*: Create a function called *intro_to_plotting* which will take an input (*num_values*). Create a random vector of size *num_values* named *new_vector*. Using *new_vector*, generate three plots as shown below. Use the ggplot2 package for plotting.

      i. **Plot 1:** Scatter plot with *new_vector* on x-axis and *new_vector* on y-axis. Programmatically save this plot as *teamID_plot*01.*pdf*

      ii. **Plot 2:** Scatter plot with *new_vector* on x-axis and new_vector^2 on y-axis. Programmatically save this plot as *teamID_plot*02.*pdf*

   Example: *intro_to_plotting*(100)

   (c) You are allowed to use the predefined functions defined in R Base and ggplot2 for random array generation, max, min, mean and median and plotting operations.