

# Stat 604

## Assignment 07 R

### Scope:

This assignment reinforces the material covered in all R Lectures, concentrating primarily on the last three lectures.

### Specific Instructions for this Assignment:

Perform each of the exercises listed below. Your output should match the sample output file posted on eCampus including all titles and labels. As in the previous assignment, your script must include code to send the graphics directly to a PDF document and close out the PDF graphics device when you are finished writing to it. You will also be required to upload to WebAssign your R script and console output in addition to the PDF document that you create. Your R script will need to include the header lines with your name, housekeeping functions, etc. as required in the previous assignments.

In this assignment, you will create a function. The creation of a function is an iterative process. The first step of that process (explained in #3 below) is to get the code working as standard R code with all of the values hard coded. After the process has been successfully tested with the hard coded values, convert the process to a function and replace the desired hard coded values with function parameters. You may want to delay sending output to PDF until you have your manual process working as desired.

An exponentially weighted moving average (EMA) is frequently used in the finance industry to form a stock price trend indicator while smoothing price action and filtering out noise. EMA for a given time period is calculated using the following formula:

$$EMA_i = (P_i \times \alpha) + (EMA_{i-1} \times (1 - \alpha)) \text{ where } \alpha = \frac{2}{1 + N}$$

**P** is the closing price at the end of that time period and **N** is the number of periods to include in the moving average. For those who may not know, the “a” looking character is the Greek letter alpha.

Along with the header, your R script will include two major sections. The first section will graph the EMA of stock prices (adjusted closing price) using hard coded values. The second section will contain the code to create the function, the appropriate graphics setup, and two calls to the function. The console and PDF output should represent only the final submission of your R script after restarting R.

1. The file cisco.csv contains stock price information for every day since Cisco stock started trading publicly. Download the file from eCampus and become familiar with its contents. Write a line of code that reads this file into a data frame in your R session for use in this assignment.
2. When you define your PDF document and open the device, set the size to 11 X 8.5 inches to match the size of a normal letter size sheet of paper turned so that it is wider than tall. (Research the help documentation on pdf to determine the correct options to use.)
3. Create and plot a 30 Day Exponential Moving Average with these steps:
  - a. Go ahead and assign a value for the alpha component of the formula so the EMA code does not get too long later on. If you wish you may also assign a scalar the value of 30 for our N value and use it in your code since this will be an argument to the function later in the assignment.

- b. Create a vector populated with zeroes that will be used to contain the EMA values. The length of this vector needs to be the same as the length of the vector of adjusted closing prices from the Cisco data.
  - c. One of the problems with moving averages is that they rely on a previous value and, at some point in the beginning, there is no previous value. We are going to solve this problem in this instance by populating the EMA on day 30 with the simple average adjusted closing prices of the first 30 days in which Cisco stock was traded. Create an assignment statement that puts the 30 day average in the 30<sup>th</sup> member of the EMA vector.
  - d. Use a loop to run the EMA formula repetitively from day 31 to the last day for which Cisco stock prices are available. Each iteration of the loop will use the corresponding day's adjusted closing price to calculate the EMA value for that day and store the calculated value in the appropriate place in the EMA vector.
  - e. There are roughly 260 weekdays in a year. Plot a line showing the last 260 EMA values. You may simply use the numbers 1 through 260 as the x values and then use the EMA values for y. This line should be purple. The range of the Y axis is to be 0 to 35. To make our program fully adaptive, we would need to use an algorithm to base the upper limit on the maximum stock price but this assignment is going to take a long time without adding this requirement so hard code the values of 0 and 35 for now. We will provide an argument to adjust this in the function. Set the title and axis labels as shown.
  - f. At the bottom of the graph, add the EMA formula as shown. Use the point 0,2 as the reference point for the formula line.
  - g. Add to the graph a green2 line showing the actual adjusted closing prices for the last 260 days available in the Cisco data.
4. Convert the completed code to a function including the code that will plot the graph. The first argument to the function is to be the name of the vector containing the adjusted closing prices so that we can analyze data from any stock without having to rewrite our function. The second argument will be the number of days over which the EMA is computed. The default value will be 30 like we used above. Be sure to replace all references to the price vector and the number of days with the appropriate function argument. Note that any value in your original code that was based on the 30 day average will use the N argument or an algebraic expression based on N. The third argument will set the upper limit for the Y axis. Make the default 35 and make sure the graphing expressions use this argument. The graph title and formula line must be constructed so they automatically change according to the value of N used in the function call.
5. Set up your graphics for this section of the script so that there will be two columns of graphics on the output page. We want to set the overall margins of the graphics within the page to be 1.5 inches on the top to allow space for punched holes and 0.5 inches on the bottom, right, and left. Set the margins of the individual graph to be 2 lines on top, 4 lines left, 0 lines right and 4 lines bottom.
6. Call your function twice. The first call should have the reference to the adjusted closing price column as its only argument. For the second call, add a value of 100 for the N argument. (When you are testing your program, you should experiment with an N of 10 and a few other values to see how the size of the EMA affects the graph. But, do not include the results of these tests in your homework submission.)
7. Write the system time in the bottom outer margin of the page. Left justify the time as shown in the sample.
8. After you have finished writing and testing your code close R. Then reopen R and run your script. Submit to WebAssign the script, console, and output from this final run.