**CSCE 489 Cover Page**

Homework Assignment #01 - Part I (50 points)

First Name: Fengqiao Last Name: Wang UIN:925009578

**Any assignment turned in without a fully completed coverpage will receive ZERO POINTS.**

Please list all below all sources (people, books, webpages, etc.) consulted for this assignment:

CSCE 489 Students Other People Printed Material Web Material (URL) Other

1. 1. 1. 1. 1.

2. 2. 2. 2. 2.

3. 3. 3. 3. 3.

4. 4. 4. 4. 4.

5. 5. 5. 5. 5.

Recall that University Regulations, Section 42, define scholastic dishonesty to include acquiring answers from any unauthorized source, working with another person when not specifically permitted, observing the work of other students during any exam, providing answers when not specifically authorized to do so, informing any person of the contents of an exam prior to the exam, and failing to credit sources used. Disciplinary actions range from grade penalties to expulsion. Please consult the Aggie Honor System Office for additional information regarding academic misconduct – it is your responsibility to understand what constitutes academic misconduct and to ensure that you do not commit it.

**I certify that I have listed above all the sources that I consulted regarding this assignment, and that I have not received nor given any assistance that is contrary to the letter or the spirit of the collaboration guidelines for this assignment**.

Today’s Date:07/05/2019

Printed Name (in lieu of a signature): Fengqiao Wang

(*Q1 to Q5: 02 points each*)

Q1. How is R better than Python? *Write 01 point crisply, clearly and neatly. Make sure it is not a generic point (e.g. Managing and manipulating data)*.

my favorite R’s advantage over Python is the convenience of doing all kinds of statistical data plots. For example the QQplot in R is just one line of code. In python we need to download packages and stuff.

Q2. What will be the output of the following?

u = c(1:3)

v = c(4:0)

*> u = c(1:3)*

*> v = c(4:0)*

*>*

*> u*

*[1] 1 2 3*

*> v*

*[1] 4 3 2 1 0*

Q3. Given the following inputs:

n = c(1, 3, 2)

s = c("Aa", "Bb")

li = list(n, s)

What is the output of

1. li[[2]][1] = "Cc"; li[2]

b) li[2][1] = "Cc"; li[2]

a)

[[1]]

[1] "Cc" “Bb"

b)

[[1]]

[1] "Cc"

Q4. What will be the output of the following:

A = matrix( c(1:6), nrow = 2, ncol = 3, byrow = F)

A [2:1,-1]

[,1] [,2]

[1,] 4 6

[2,] 3 5

Q5. What will be the output of the following:

a) log(-1)

b) x = c(1, 2, 3); x[0]

a) > log(-1)

[1] NaN

Warning message:

In log(-1) : NaNs produced

b) numeric(0)

(*Q6 and Q7: 20 points each*)

Q6. Provide a script that

* 1. creates a vector v with 100,000 random elements
  2. creates a list l with the same elements in v using as.list(v)
  3. build a for loop that simply adds all the values in v and save the result in a variable x
  4. collect the time for that block of code
  5. collect the size (memory) of v
  6. do the same for the list l

Repeat the process for the sizes 100,000, 110,000, 120,000, 130,000, …, 200,000 (11 data points). Plot two graphs using the plot function (and other required function to get the job done, e.g. legend, title, lines):

1. one comparing the vector vs. list memory usage
2. one comparing the vector vs. list running time.

**Provide your comments explaining the differences in memory usage and time efficiency**.

Everything in comments in the vector&listcomparison.R

Q7. Provide a script that

* 1. creates a matrix m with 300x300 random elements
  2. creates a data frame df with the same elements in v using as.data.frame(m)
  3. build two nested for loops that simply adds all the values in m and save the result in a variable x
  4. collect the time for that block of code
  5. collect the size (memory) of m
  6. do the same for the data frame dm

Repeat the process for the sizes 300x300, 320x320, 340x340, 360x360, …, 500x500 (11 data points). Plot two graphs using the plot function (and other required function to get the job done, e.g. legend, title, lines):

1. one comparing the matrix vs. data frame memory usage
2. one comparing the matrix vs. data frame running time.

**Provide your comments explaining the differences in memory usage and time efficiency**.

Everything in comments in the Matrix&DFcomparison.R