

###Welcome to the first assignment of QBS 103!

#Instructions * Write your code in the R code chunks provided. * When you are finished with your assignment, upload this R Markdown file under Assignments tab on our class canvas page. * MAKE SURE TO COMMENT YOUR CODE. * You can work with your classmates, but remember to write their name in the comments. * If you are inspired by code you found online make sure you cite the website, comment your code, and make it your own. No copy and pasting.

- 1) Below I have provided a list of random letters and numbers. Please conduct the following tasks on this list:
 - a) What type of data is each of the “things” in the list? Write these types to a vector called “my_types”. Psssst here is a hint for how you might want to write this code: <https://www.geeksforgeeks.org/adding-elements-in-a-vector-in-r-programming-append-method/#>.
 - b) Count how many of each data type appear in your vector.
 - c) How many numbers in this list are greater than 5?
 - d) Generate a factor of types from your “my_types” vector.
 - e) Get the index (or indices) of the number “1” in the lst list.

```
lst <- list(4, 'j', 'Y', 5, 'f', 'K', 8, 'z', 'T', 'c', 0, 'B', 2, 'm', 'X', 9, 'd', 'V', 3, 'n', 'G', 6, 'h', 'J', 2, 'p', '0', 1, 'q', 'R', 7, 's', 'D', 2, 'l', 'N', 0, 'w', 'E', 5, 'e', 'U', 4, 'i', 'P', 1, 'a', 'H', 8, 'y', 'S', 3, 'g', 'F', 9, 'b', 'M', 6, 'k', 'L', 7)
types = c()
for (i in 1:length(lst)) {
  types = c(types,class(lst[[i]])) #add on to types vec
}
```

```
print(types)
```

```
#b:
numct = length(types[types=='numeric']) #count where type is numeric
print(numct)
charct = length(types[types=='character']) #^ char
print(charct)
```

```
#c
ov5 = 0
for (i in 1:length(lst)){
  if (class(lst[[i]]) == 'numeric' && lst[[i]] >5){
    ov5 = ov5+1 #^ if class num and over 5, increm.
  }
}
print(ov5)
```

```
#d
myfac = factor(types)
```

```

print(myfac)

idxs = c()
for (i in 1:length(lst)){
  if (class(lst[[i]]) == 'numeric' && lst[[i]]==1){
    idxs = c(idxs,i)#get pos of '1', add
  }
}
print(idxs)

```

- 2) In the “data” folder of this repository there is a file called GSE165252_vst_PERFECT.txt. This is real RNA-seq data from a phase II clinical trial investigating the impact of combined impact PD-L1 inhibitor (atezolizumab) and neoadjuvant chemoradiotherapy on resectable esophageal adenocarcinoma patients. You can find a link to the data on GEO here: <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE165252>. You can read about this study here: <https://aacrjournals.org/clincancerres/article/27/12/3351/671431/Neoadjuvant-Chemoradiotherapy-Combined-with>. Download the GSE165252_vst_PERFECT.txt file to your local machine.
 - a) Read in this file. Make sure you take into account that the first row contains Ensembl IDs (gene names)!
 - b) How many samples are in this data set?
 - c) How many genes are in this data set?
 - d) Make a histogram of the expression values of a gene of your choosing from this data set.
 - e) Subset this data to contain only their first three samples and first 2000 genes. Write this subset to a file called “eso_sub.csv”.
- 3) Write a function call “my_first_function”. The function should take the following vector: `my_vector <- c(1,2,3,4,5,6,7,8,9)`. Your function should sort the numbers in `my_vector` into two categories: less than or equal to 5 or greater than 5. Hint: You can take a look at for loops and if/else statements to complete this question!

```

fnam = 'GSE165252_vst_PERFECT.txt'

df1 = read.table(fnam,header=FALSE,sep='\t',stringsAsFactors=FALSE)
#need header false bc dont want 1st row as colnames
print(head(df1))
#now col names from 1st row
colnames(df1) = df1[1,]
#reset df
df1 = df1[-1,]

#b,c:
dim(df1) #62300 samples, 78 genes
head(df1[,1:7]) #first 7 cols to check

```

```

myg = as.numeric(df1$'TE2004-04-atr') #2nd to last col
#hist(myg,breaks=seq(0,2500,by=100))
hist(myg, breaks = seq(0, ceiling(max(myg)), by = 100), xlim = c(0,
ceiling(max(myg))))
hist(myg)
max(myg)
median(myg) #sanity checks cuz this histo looks worthless.
mean(myg)
#help(hist)
print(myg[1:20])

```

```

#e am assuming typo. 2000 samples, 3 genes.
ss1 = df1[1:2000,1:4] #gene ID? + 3 genes' samples
colnames(ss1) = ss1[1,] #subset renaming as df1 above
#reset df
ss1 = ss1[-1,] #remove toprow^^
head(ss1)
write.csv(ss1, file = "eso_sub.csv", row.names = FALSE) #bang.

```

#3: fn

```

my_first_function = function(v){
  c1 = c()#lessthan or eq
  c2= c() #greater
  for (i in (1:length(v))){ #loop thru
    if (v[i] <= 5){
      c1 = c(c1,v[i]) #add
    } else {
      c2 = c(c2,v[i])
    }
  }
  return(list(LEQ5=c1,GT5=c2)) #had mr chat help me w the return here,
was used to python tuple returns
}
my_vector <- c(1,2,3,4,5,6,7,8,9)
print(my_first_function(my_vector))

```