Module 2 Homework

Problem 1. (30 points)

Computations on gene means of the Golub data set.

```
library("multtest")  # loading required library("multtest")
data(golub)    # loading required dataset
```

(a) Write an R code to compute the mean expression values for every gene among "ALL" patients.

Answer)

```
gol.fac <- factor(golub.cl, levels=0:1, labels = c("ALL", "AML"))
ALLmean <- apply(golub[,gol.fac=="ALL"], 1, mean)
ALLmean
```

(b) Write an R code to compute the mean expression values for every gene among "AML" patients.

Answer)

```
AMLmean<-apply(golub[,gol.fac=="AML"], 1, mean)
AMLmean
```

(c) Give the biological names of the three genes with the largest mean expression value among "ALL" patients.

Answer)

```
arrangedataALL <- order(ALLmean, decreasing = TRUE)
golub.gnames[arrangedataALL[1:3],2]
```

Output:

```
[1] "GB DEF = Chromosome 1q subtelomeric sequence D1S553"
[2] "37 kD laminin receptor precursor/p40 ribosome associated protein
gene"
[3] "RPS14 gene (ribosomal protein S14) extracted from Human ribosomal
 protein S14 gene"
```

(d) Give the biological names of the three genes with the largest mean expression value among "AML" patients.

Answer)

```
arrangedataAML<-order(AMLmean, decreasing = TRUE)
golub.gnames[arrangedataAML[1:3],2]
```

Output:

```
[1] "GB DEF = mRNA fragment for elongation factor TU (N-terminus)"
[2] "GB DEF = HLA-B null allele mRNA"
[3] "Globin, Beta"
```

Submit R commands that does (a)-(d). And answer directly part (c) and (d)

Note: I don't expect to see any output for parts (a) and (b). That will be a lot of numbers which no one needs to see. Answering parts (c) and (d) correctly will be the way to look at the output for (a) and (b). Please don't upload 100 pages of numbers which no one wants to look at.

Problem 2. (30 points)

More work on the Golub data set.

(a) Save the expression values of the first five genes (in the first five rows) for the AML patients in a csv file "AML5.csv".

Answer)

```
gene_datAML <- golub[,gol.fac=="AML"]
gene_AML <- head(gene_datAML, 5)
write.csv(gene_AML,file="AML5.csv")
```

(b) Save the expression values of the first five genes for the ALL patients in a plain text file "ALL5.txt".

Answer)

```
gene_datALL <- golub[,gol.fac=="ALL"]
gene_ALL <- head(gene_datALL,5)
write.table(gene_ALL,file="ALL5.txt")
```

(c) Compute the standard deviation of the expression values on the first patient, of the 100th to 200th genes (total 101 genes).

Answer)

```
std <-golub[100:200,1]
sd(std)
```

Output:

```
> sd(std)
[1] 0.9174976
```

(d) Compute the standard deviation of the expression values of every gene, across all patients. Find the number of genes with standard deviation greater than 1.

Answer)

```
datasd <- apply(golub,1,sd)
sum(datasd>1)
```

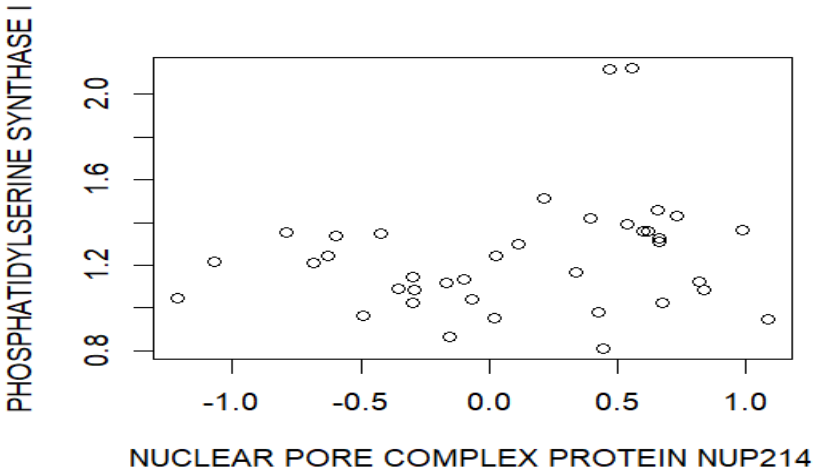
Output:

```
> sum(datasd>1)
[1] 123
```

(e) Do a scatter plot of the 101th gene expressions against the 102th gene expressions, label the x-axis and the y-axis with the genes' biological names using xlab= and ylab= control options.

Answer)

```
gene_x<-(golub[101,])
gene_y<-(golub[102,])
plot(gene_x,gene_y,xlab="NUCLEAR PORE COMPLEX PROTEIN
NUP214",ylab="PHOSPHATIDYLSERINE SYNTHASE I")
```



Submit R commands that does (a)-(e). And the outputs (files for parts (a), (b), numerical answer for part (c) and (d), the figure file for part (e)).

Problem 3. (20 points)

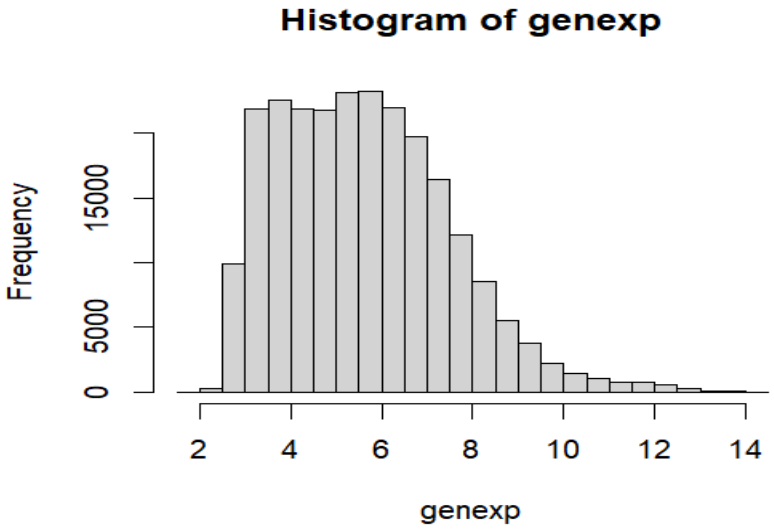
Work with the ALL data set. Load the ALL data from the ALL library and use str and openVignette() for a further orientation.

```
library(ALL) # loading library ALL
data(ALL)  # loading dataset ALL
```

(a) Use exprs(ALL[,ALL$BT=="B1"] to extract the gene expressions from the patients in disease stage B1. Produce one histogram of these gene expressions in this matrix.

Answer)

```
genexp <-c(exprs(ALL[,ALL$BT=="B1"]))
hist(genexp)
```



(b) Compute the mean gene expressions for every gene over these B1 patients.

Answer)

```
meanB1<-apply(exprs(ALL[,ALL$BT=="B1"]),1, mean)
meanB1
```

(c) Give the gene identifiers of the three genes with the largest mean.
Submit R commands that does (a)-(c), and answer part (c) directly.

Answer)

```
ordering<-order(meanB1,decreasing = TRUE)
meanB1[ordering[1:3]]
```

Output:

```
> meanB1[ordering[1:3]]
AFFX-hum_alu_at        31962_at       31957_r_at
       13.41648        13.16671          13.15995
```

Problem 4. (20 points)

We work with the "trees" data set that comes with R. Produce a figure with two overlaid scatterplots: Height versus Girth, Volume versus Girth(The Girth is on the x-axis). Do the Height plot with blue "+" symbols, and do the Volume plot with red "o" symbols. You need to learn to set the ylim= control option so that all points from the two plots can all show up on the merged figure.
Hint: you should use plot() then points() to create the overlaid two scatterplots.

Answer)

```
data(trees)    # loading dataset trees
changedata <-data.frame(trees)   # converting trees dataset to data frame
plot(changedata$Girth, changedata$Height,xlab="Girth", ylab="Volume &
Height", col="blue", pch="+", xlim= c(5.0, 25.0), ylim = c(5.0, 90.0))
points(changedata$Girth, changedata$Volume, col="red", pch="o")
legend("bottomright", c("Height", "Volume"), fill = c("blue","red"),
lty=c(0,0))
```