Exercise 2 - Study on Liver Disease

Exercise 2 - Study on Liver Disease

by Paula Carrio Cordo - 1 October 2016

1. Introduction

We are focus on a study about liver disease. Based on Whole Genome Microarray data of gene expression, we are going to check if there are outliers and/or systematic biases in the 5 samples examined which were taken from sick patients.

2. Loading data

Phenotype information is contained in a file, that we read into a data frame for the study. This file contains sample name, tissue type, patient ID and associated file.

Before starting our analysis: labeling, coloring the subsequent plots, and generate boolean to indicate with which we can access the normal, sick and acute samples only:

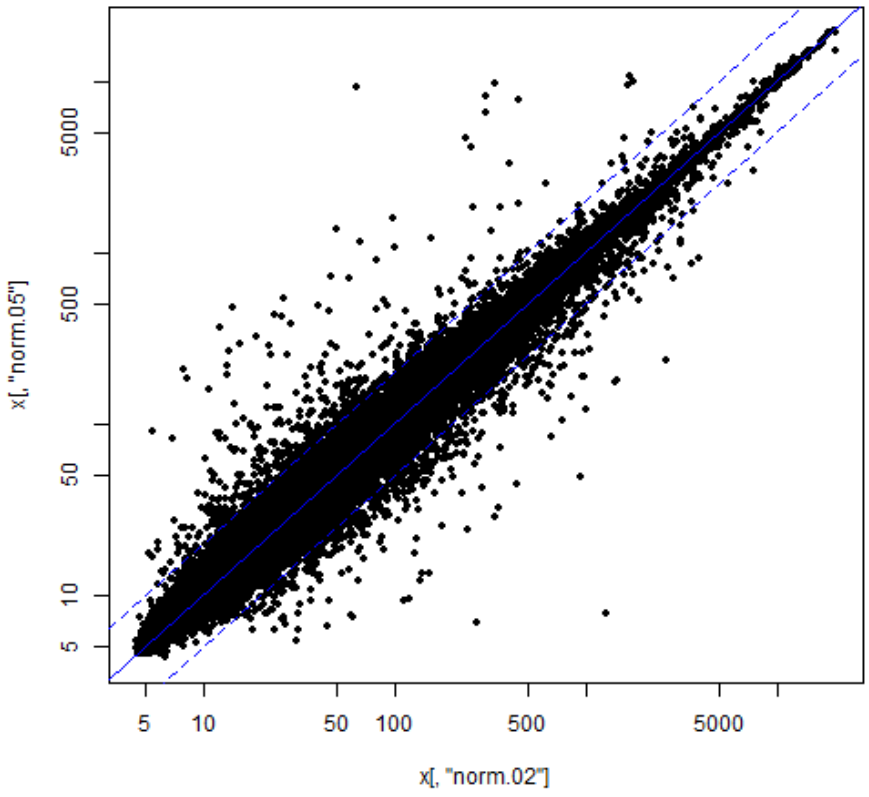
```
samples = rownames(anno)
colors = rainbow(nrow(anno))
isNorm = anno$TissueType == "norm"
isSick = anno$TissueType == "sick"
isAcute = anno$TissueType == "acute"
```

Now we load the expression data:

```
x = read.table("/Users/TOSHIBA/STA426 R/expressionData.txt",
               as.is=TRUE, sep="\t", quote="", row.names=1, header= TRUE)
x = as.matrix(x)
```

With a plot we compare the expression signals from sample 1 and 2. The solid blue line gives the first diagonal and the dashed lines give the boundaries for 2-fold up- or down-regulation.

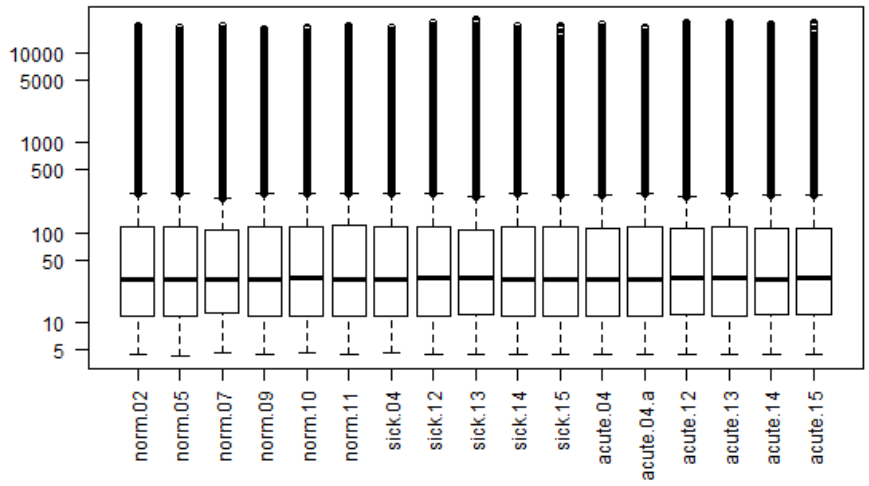
```
plot(x[ , "norm.02"], x[, "norm.05"], log="xy", pch=20)
abline(0, 1, col="blue")
abline(log10(2), 1, col="blue", lty=2)
abline(-log10(2), 1, col="blue", lty=2)
```



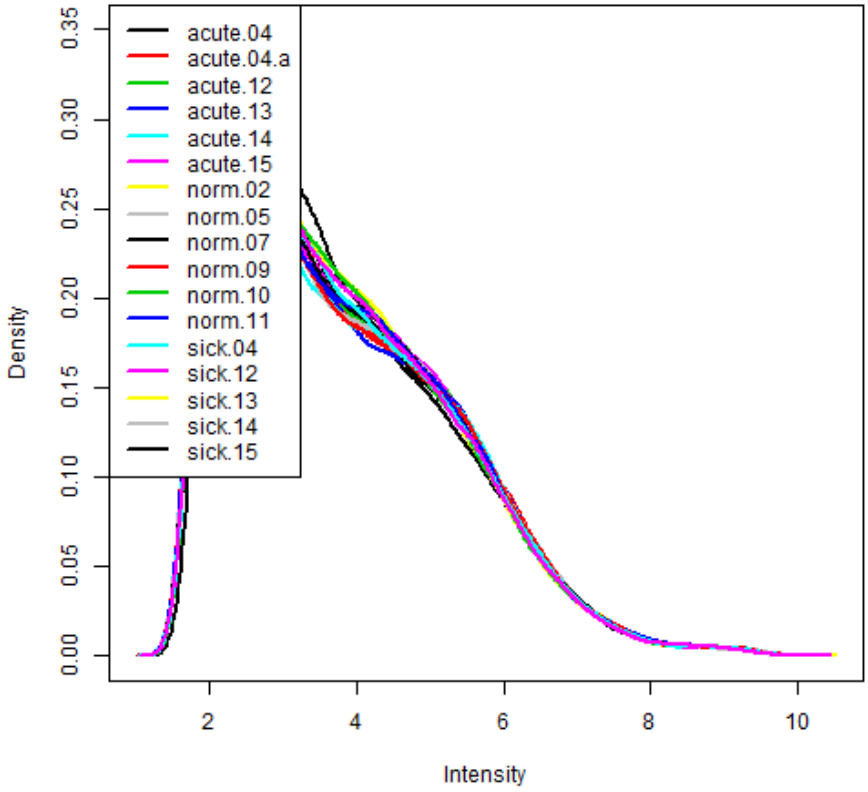
3. Distribution of the intensities

Assuming that the intensiy distribution of the different arrays are similar, we summarize the distribution with a boxplot and a graphic created with function plotDensities.

```
boxplot(x, log="y", cex.lab=0.5, las=2)
```




```
plotDensities(log(x), legend="topright",cex.lab=0.3)
```



(*Legend function does not work)

4. Consistency of the replicates

We need to compute sample correlation on the logarithmic scale.

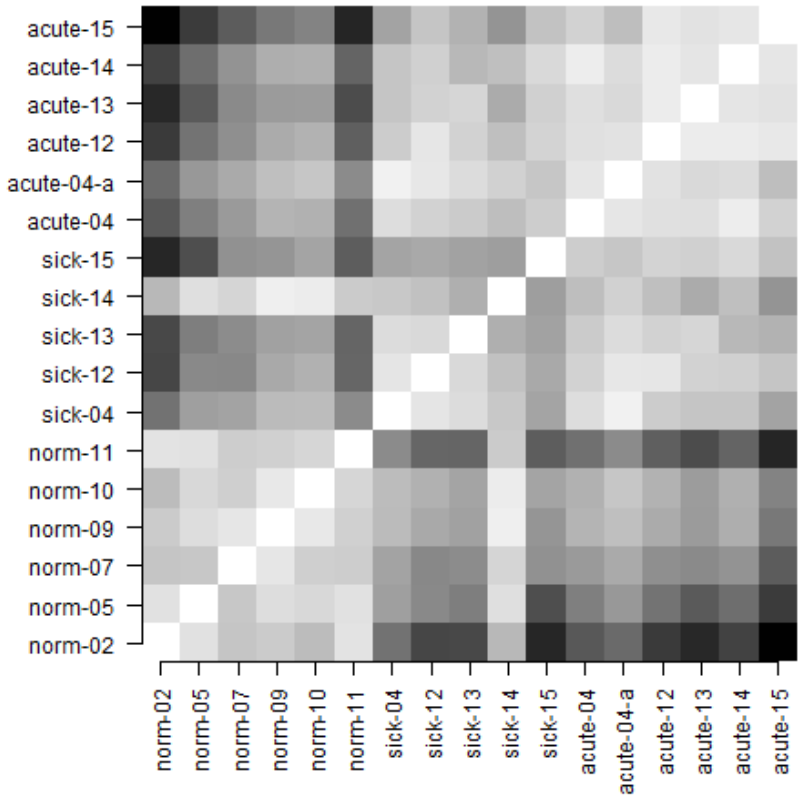
```
corrMatrix <- cor(x)
signif(corrMatrix, digits=3)
```

```
##           norm.02 norm.05 norm.07 norm.09 norm.10 norm.11 sick.04 sick.12
## norm.02     1.000   0.980   0.962   0.965   0.956   0.982   0.907   0.878
## norm.05     0.980   1.000   0.963   0.977   0.974   0.980   0.937   0.923
## norm.07     0.962   0.963   1.000   0.984   0.968   0.967   0.940   0.922
## norm.09     0.965   0.977   0.984   1.000   0.985   0.969   0.955   0.943
## norm.10     0.956   0.974   0.968   0.985   1.000   0.973   0.956   0.949
## norm.11     0.982   0.980   0.967   0.969   0.973   1.000   0.924   0.899
## sick.04     0.907   0.937   0.940   0.955   0.956   0.924   1.000   0.983
```

```
## sick.12      0.878   0.923   0.922   0.943   0.949   0.899   0.983   1.000
## sick.13      0.879   0.915   0.925   0.938   0.940   0.898   0.977   0.975
## sick.14      0.953   0.979   0.972   0.989   0.987   0.966   0.964   0.959
## sick.15      0.857   0.883   0.928   0.931   0.940   0.893   0.940   0.944
## acute.04     0.890   0.915   0.934   0.951   0.949   0.906   0.977   0.970
## acute.04.a   0.901   0.933   0.944   0.958   0.963   0.924   0.991   0.984
## acute.12     0.870   0.908   0.926   0.945   0.950   0.895   0.966   0.983
## acute.13     0.858   0.891   0.923   0.934   0.935   0.882   0.962   0.970
## acute.14     0.876   0.904   0.929   0.947   0.948   0.898   0.962   0.969
## acute.15     0.833   0.871   0.892   0.911   0.919   0.857   0.940   0.962
##            sick.13 sick.14 sick.15 acute.04 acute.04.a acute.12 acute.13
## norm.02      0.879   0.953   0.857    0.890      0.901    0.870    0.858
## norm.05      0.915   0.979   0.883    0.915      0.933    0.908    0.891
## norm.07      0.925   0.972   0.928    0.934      0.944    0.926    0.923
## norm.09      0.938   0.989   0.931    0.951      0.958    0.945    0.934
## norm.10      0.940   0.987   0.940    0.949      0.963    0.950    0.935
## norm.11      0.898   0.966   0.893    0.906      0.924    0.895    0.882
## sick.04      0.977   0.964   0.940    0.977      0.991    0.966    0.962
## sick.12      0.975   0.959   0.944    0.970      0.984    0.983    0.970
## sick.13      1.000   0.948   0.939    0.966      0.977    0.970    0.973
## sick.14      0.948   1.000   0.936    0.957      0.970    0.958    0.945
## sick.15      0.939   0.936   1.000    0.967      0.962    0.971    0.969
## acute.04     0.966   0.957   0.967    1.000      0.984    0.980    0.979
## acute.04.a   0.977   0.970   0.962    0.984      1.000    0.981    0.975
## acute.12     0.970   0.958   0.971    0.980      0.981    1.000    0.987
## acute.13     0.973   0.945   0.969    0.979      0.975    0.987    1.000
## acute.14     0.953   0.958   0.975    0.988      0.977    0.987    0.982
## acute.15     0.949   0.930   0.960    0.970      0.957    0.985    0.981
##            acute.14 acute.15
## norm.02       0.876    0.833
## norm.05       0.904    0.871
## norm.07       0.929    0.892
## norm.09       0.947    0.911
## norm.10       0.948    0.919
## norm.11       0.898    0.857
## sick.04       0.962    0.940
## sick.12       0.969    0.962
## sick.13       0.953    0.949
## sick.14       0.958    0.930
## sick.15       0.975    0.960
## acute.04      0.988    0.970
## acute.04.a    0.977    0.957
## acute.12      0.987    0.985
## acute.13      0.982    0.981
## acute.14      1.000    0.983
## acute.15      0.983    1.000
```

Matrix visualization as an image:

```
par(mar=c(8,8,2,2))
grayScale <- gray((1:256)/256)
image(corrMatrix, col=grayScale,  axes=FALSE)
axis(1, at=seq(from=0, to=1, length.out=length(samples)), labels=samples, las=2)
axis(2, at=seq(from=0, to=1, length.out=length(samples)), labels=samples, las=2)
```

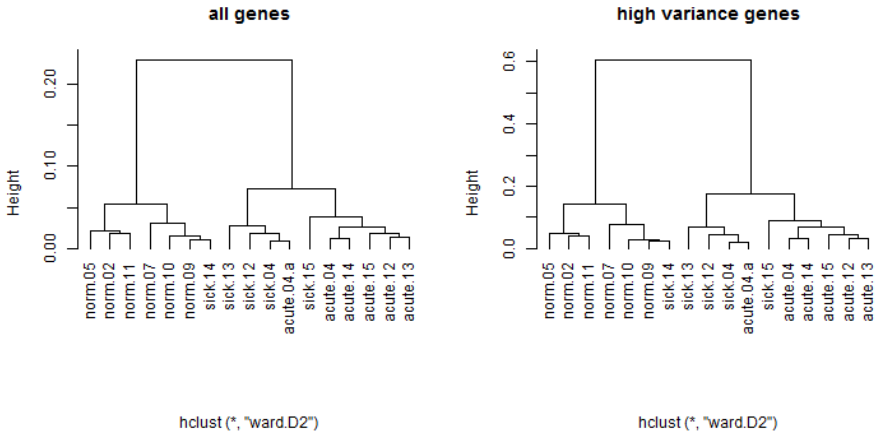


5. Sample Clustering

Clustering to appreciate the similarities of the expression patterns of the samples in a tree.

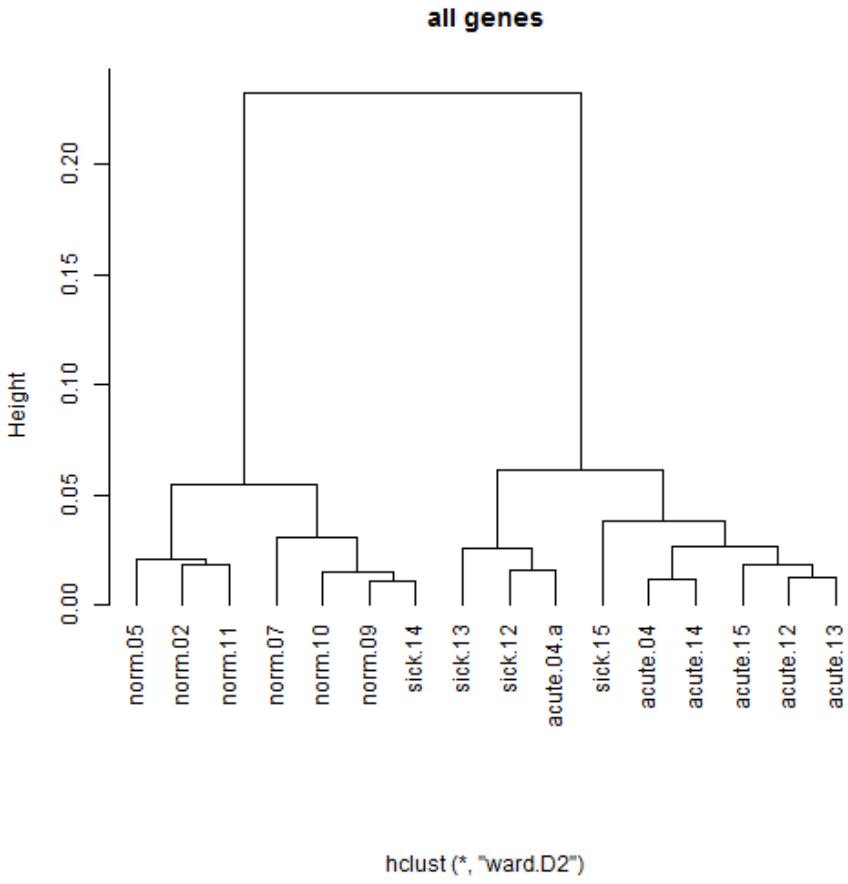
```
x.sd <- apply(x, 1, sd, na.rm=TRUE)
ord <- order(x.sd, decreasing=TRUE)
highVarGenes <- ord[1:500]
```

```
par(mfrow=c(1,2))
d <- as.dist(1-cor(x))
c <- hclust(d, method="ward.D2")
plot(c, hang=-0.1, main="all genes", xlab="")
d <-  as.dist(1-cor(x[highVarGenes, ]))
c <- hclust(d, method="ward.D2")
plot(c, hang=-0.1, main="high variance genes", xlab="")
```



If we run the clustering without sample **sick-04**, the **acute-04** does no longer cluster in the branch with the other sick samples

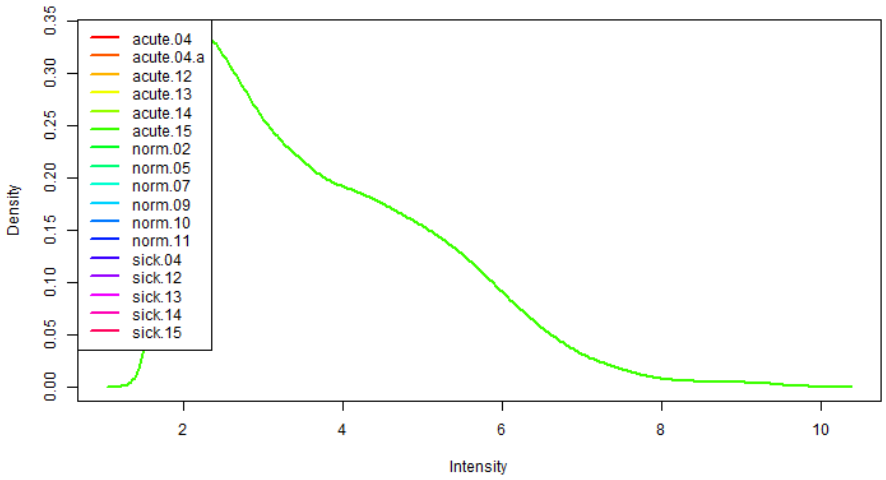
```
par(mfrow=c(1,1))
sub <-  x[ , samples != "sick-04"]
d = as.dist(1-cor(sub))
c=hclust(d, method="ward.D2")
 plot(c, hang=-0.1, main="all genes", xlab="")
```



6. Quantile normalization

We can do a quantile normalization with limma function normalizeQuantiles.

```
x.norm <- normalizeQuantiles(x)
plotDensities(log(x.norm), legend="topright", col=colors)
```

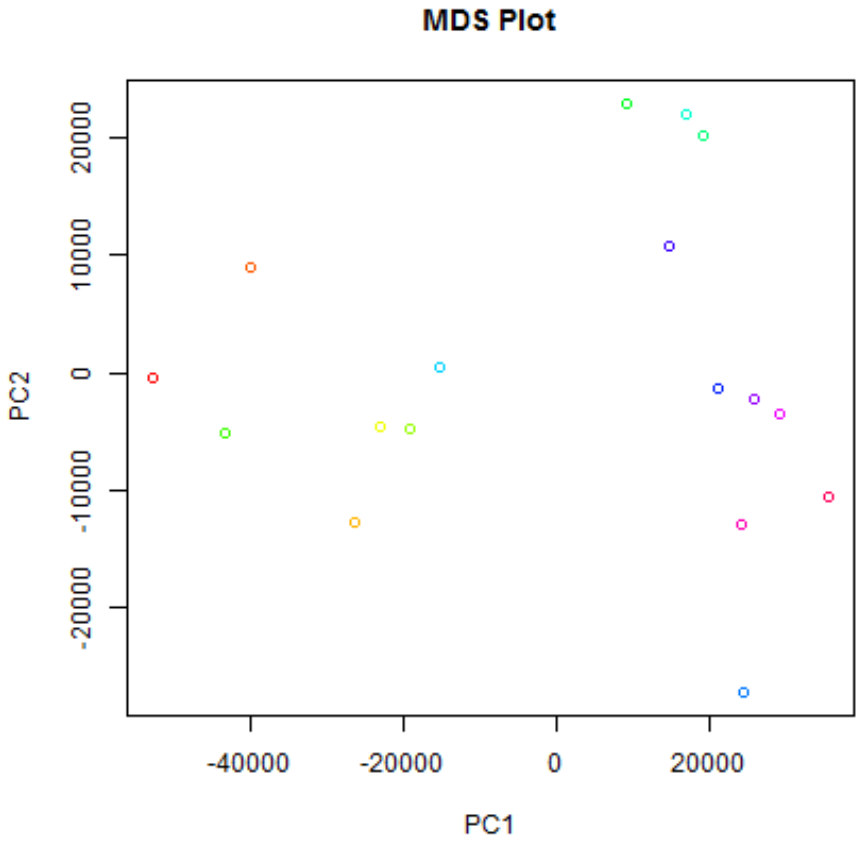


7. Sample Representation in Principal Component Space

Functions cmdscale and prcomp are useful to create a plot that represents the sample disatnaces in a reduced space.

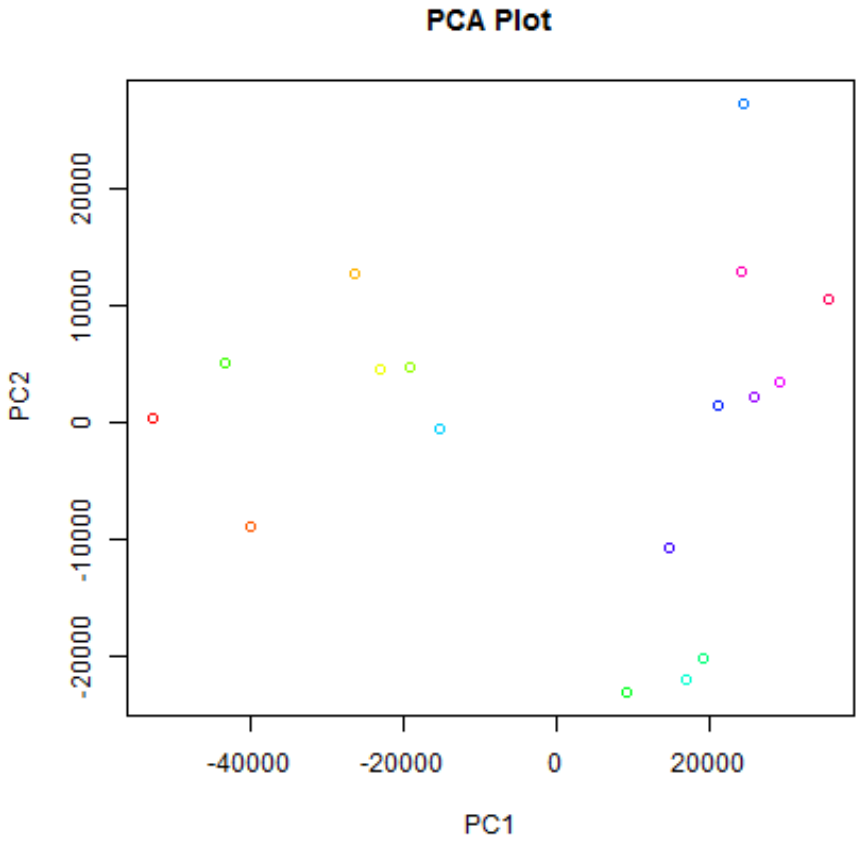
a) Multdimensional scaling based on our data matrix

```
ms <- dist(t(x.norm))
cmds <- cmdscale(ms)
plot(cmds, main="MDS Plot", col=colors,xlab="PC1",ylab="PC2")
```



b) Principal component analysis based on our data matrix

```
prc <- prcomp(t(x.norm))
plot(prc$x[,1], prc$x[,2], main="PCA Plot", col=colors, xlab="PC1", ylab="PC2")
```


PCA Plot
PC2
20000
10000
0
-10000
-20000
-40000
-20000
0
20000
PC1