HOMEWORK 7

NUZULUL KURNIANSYAH

November 14, 2016

PROBLEM 1

Cluster analysis on Zyxin gene expression.

A.Produce a scatter plot of the Zyxin gene expression values using di erent symbols for the two groups.

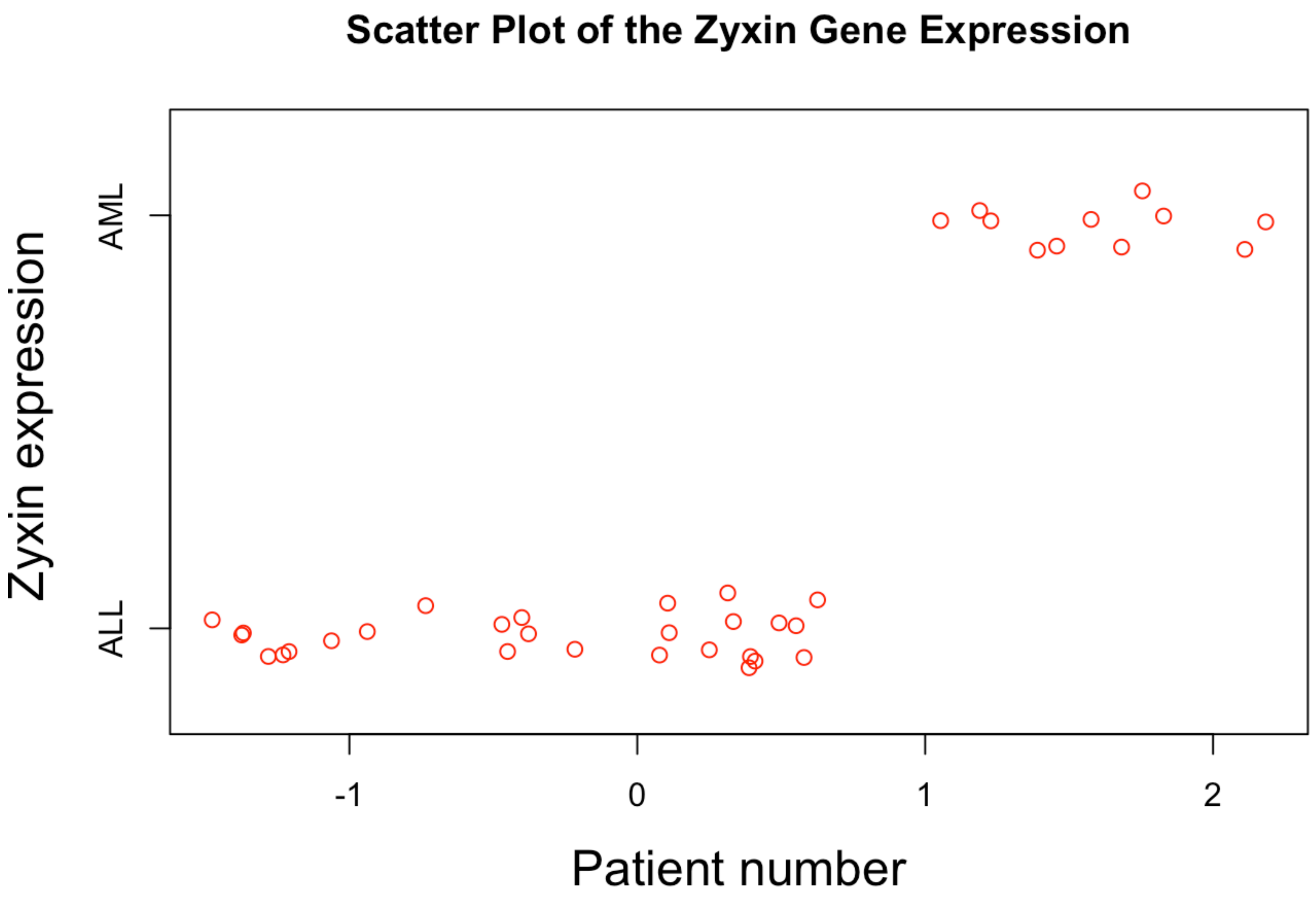
```
#Create Data Frame

classification = factor(golub.cl, labels=c("ALL", "AML")) # Create factor for cancer
classification
golub.df = as.data.frame(t(golub)) # turn transposed Golub data into data frame
colnames(golub.df) = golub.gnames[,2] # gene names as column names for golub.df
golub.df$Classification = classification # add Classification column
dim(golub.df)
```

```
## [1]   38 3052
```

```
zyxin = grep("zyxin",colnames(golub.df), ignore.case = TRUE)
zyxin
```

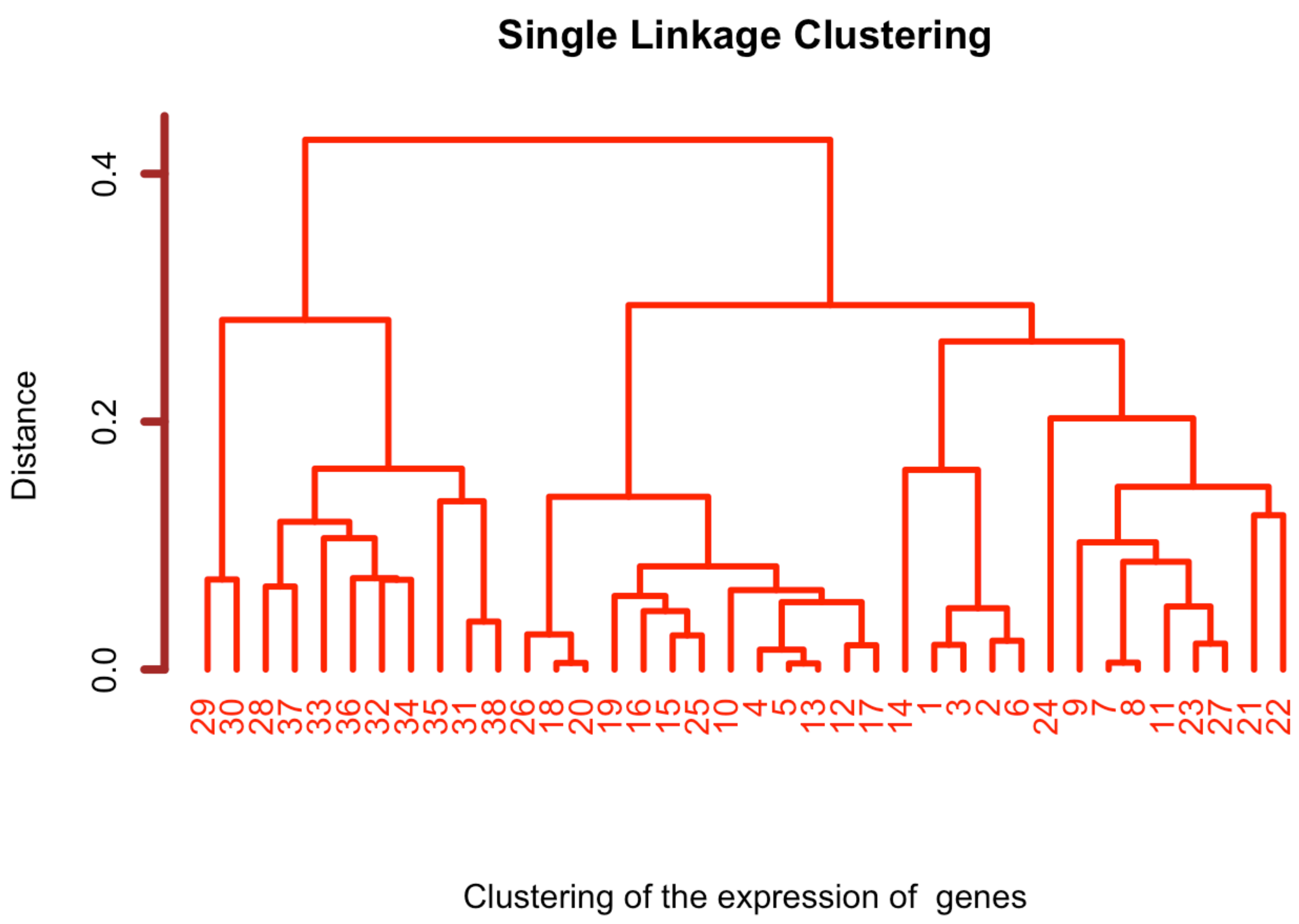
```
## [1] 2124
```

```
stripchart(golub.df[,zyxin]~golub.df$Classification,# values based on classification
       pch=as.numeric(golub.df$Classification), # plot solid circles & change to nume
ric
       method = "jitter",
       cex.lab=1.5, # make axis labels big
       xlab="Patient number",
       ylab=" Zyxin expression",
       main="Scatter Plot of the Zyxin Gene Expression",
       col="red")
```



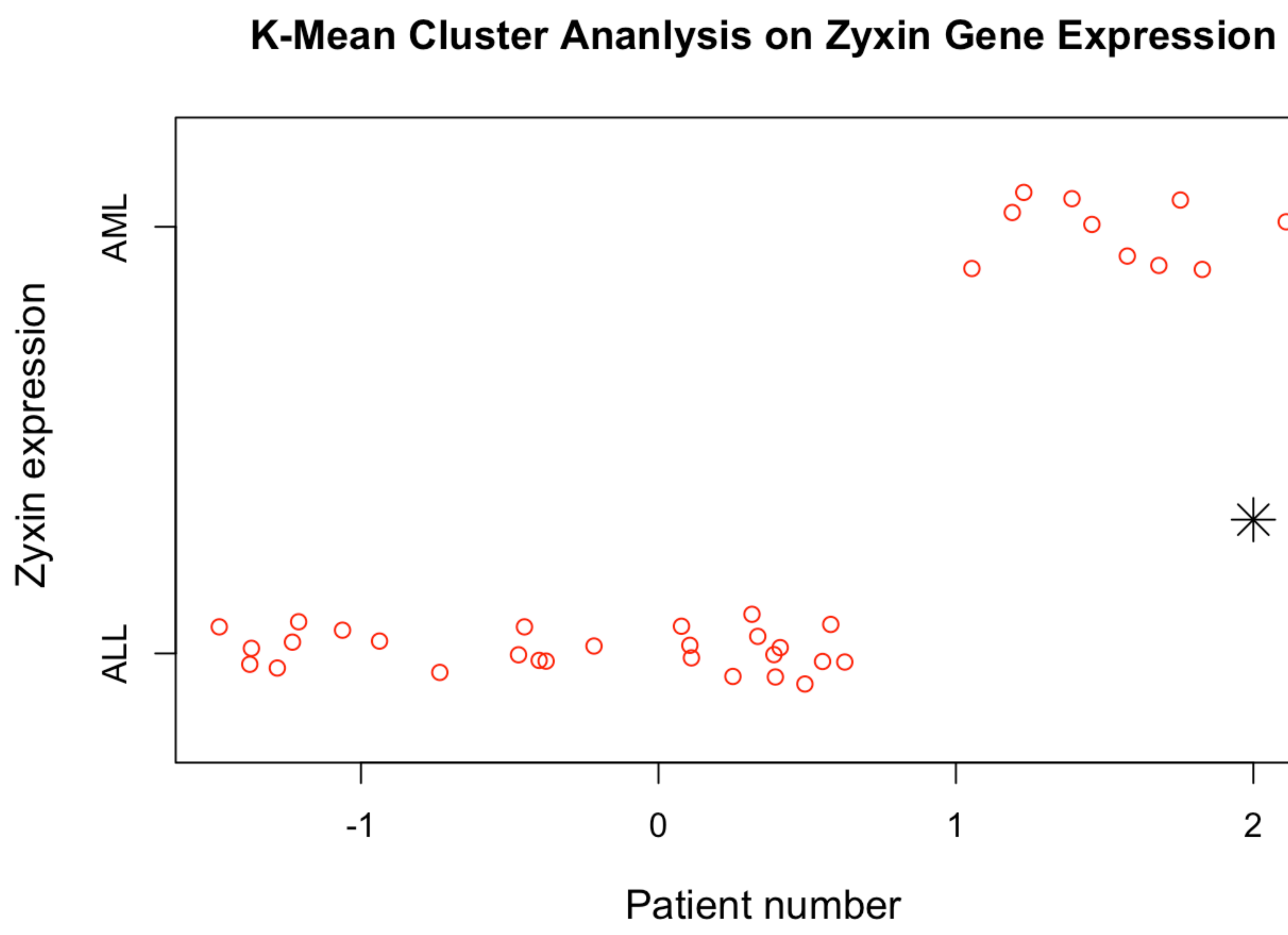
B. Use single linkage cluster analysis to see whether Zyxin gene ex- pression falls into two di erent clusters.

```
df<-data.frame(golub.df[,zyxin]) #create new data framw which contain zyxin
SLinkage<-hclust(dist(df,method="euclidian"),method="single") # get the value of dist
ance & Hierarchical cluster analysis
plot(SLinkage,
     lwd=3,
     col="red",
     ## col.lab = "brown",
     col.axis = "brown",
     ylab="Distance",
     xlab="Clustering of the expression of  genes",
     hang=-1,
     main="Single Linkage Clustering",
     sub=NA,
     axes=FALSE)
axis(side = 2, at = seq(0, 1.2, .2), col = "brown",labels = TRUE, lwd = 4)
```



C.Use k-means cluster analysis on Zyxin gene expression with k = 2. Do the two clusters re ect the diagnosis of the patient groups?

```
initial <- as.matrix(tapply(golub.df[,zyxin],golub.df$Classification,mean), nrow = 2,
ncol=2)
#get initial, change to matrix and  applies a function to each cell of a ragged array
in golub.
cl <- kmeans(df ,1, centers=initial)# K-means clustering with 2 clusters, set initial
in centers
stripchart(golub.df[,zyxin]~ golub.df$Classification,# values based on classification
        pch=as.numeric(golub.df$Classification), # plot solid circles & change to nume
ric
        method = "jitter",
        cex.lab=1.2, # make axis labels big
        xlab="Patient number",
        ylab=" Zyxin expression",
        main="K-Mean Cluster Ananlysis on Zyxin Gene Expression",
        col="red")
points(cl$centers[,1], y=NULL, col = 1, pch = 8, cex=2)
```



```
table(cl$cluster,golub.df$Classification) # create table cl$cluster that contain clus
ter gene Zyxin ALL and AML
```

```
##
##      ALL AML
##    1  23   0
##    2   4  11
```

D. Perform a bootstrap on the cluster means. Do the confidence intervals for the cluster means overlap?

```
n <- nrow(df);
nboot<-1000
boot.cl <- data.frame(matrix(0,nrow=nboot,ncol = 2)) #re-sample with replacement from
the dataset many (> 1000) times

#do looping till 1000 time to get boot cluster
for (i in 1:nboot) {
  dat.star <- df[sample(1:n,replace=TRUE),]
  cl <- kmeans(dat.star, initial, nstart = 10)
  boot.cl[i,] <- c(cl$centers[1,],cl$centers[2,])
}
quantile(boot.cl[,1],c(0.025,0.975))#compute the quantiles for the corresponding conf
idence interval
```

```
##       2.5%      97.5%
## -1.0825551 -0.0339949
```

```
quantile(boot.cl[,2],c(0.025,0.975))
```

```
##      2.5%     97.5%
## 0.6861418 1.8160387
```

PROBLEM 2

Gene expression similar to CCND3 (Cyclin D3). Recall that we did various analysis on the expression data of the CCND3 (Cyclin D3) gene of the Golub (1999) data.

A. Use genefilter() to nd the ten genes with expression patterns most similar to CCND3 (Cyclin D3). Give their probe as well as their biological names.

```
ccnd3=grep("CCND3",colnames(golub.df), ignore.case=TRUE)# Get the index CCND3 genes
closeg <- genefinder(as.matrix(t(golub.df)), ccnd3, 10, method = "euc", scale = "none
") #using gene finder to get 10 genes with expression patterns most similar, genefind
er() work only with matrix
```

```
## Warning in genefinder(as.matrix(t(golub.df)), ccnd3, 10, method = "euc", :
## NAs introduced by coercion
```

```
closeg
```

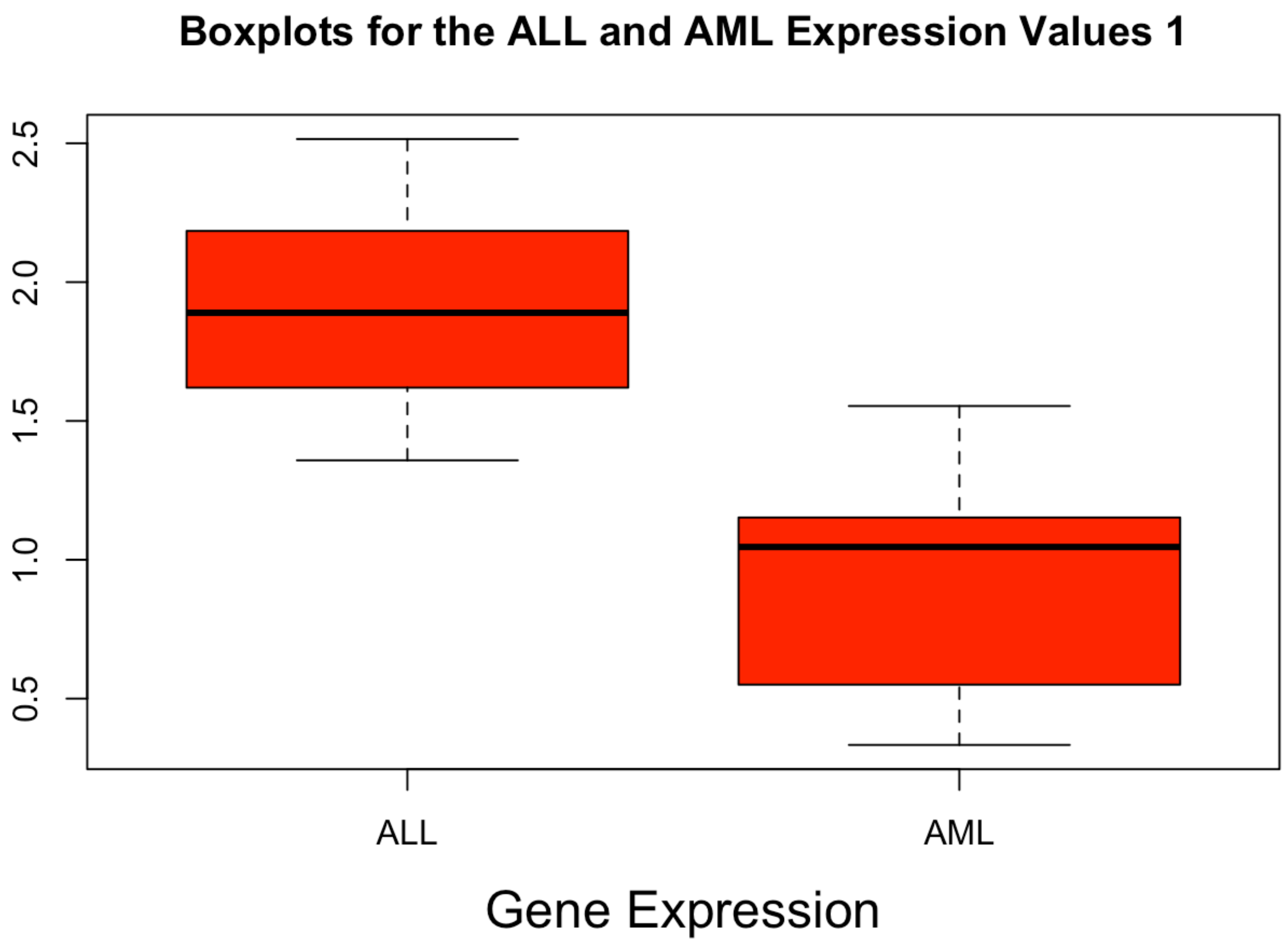
```
## [[1]]
## [[1]]$indices
##  [1]  394 1834  573  849  479  906  560 2365 2447 1723
##
## [[1]]$dists
##  [1] 3.117607 3.623484 3.912971 3.997751 4.089990 4.182336 4.197322
##  [8] 4.228923 4.246708 4.260840
```

```
golub.gnames[closeg[[1]][[1]],2:3] #get genes name and probe set
```

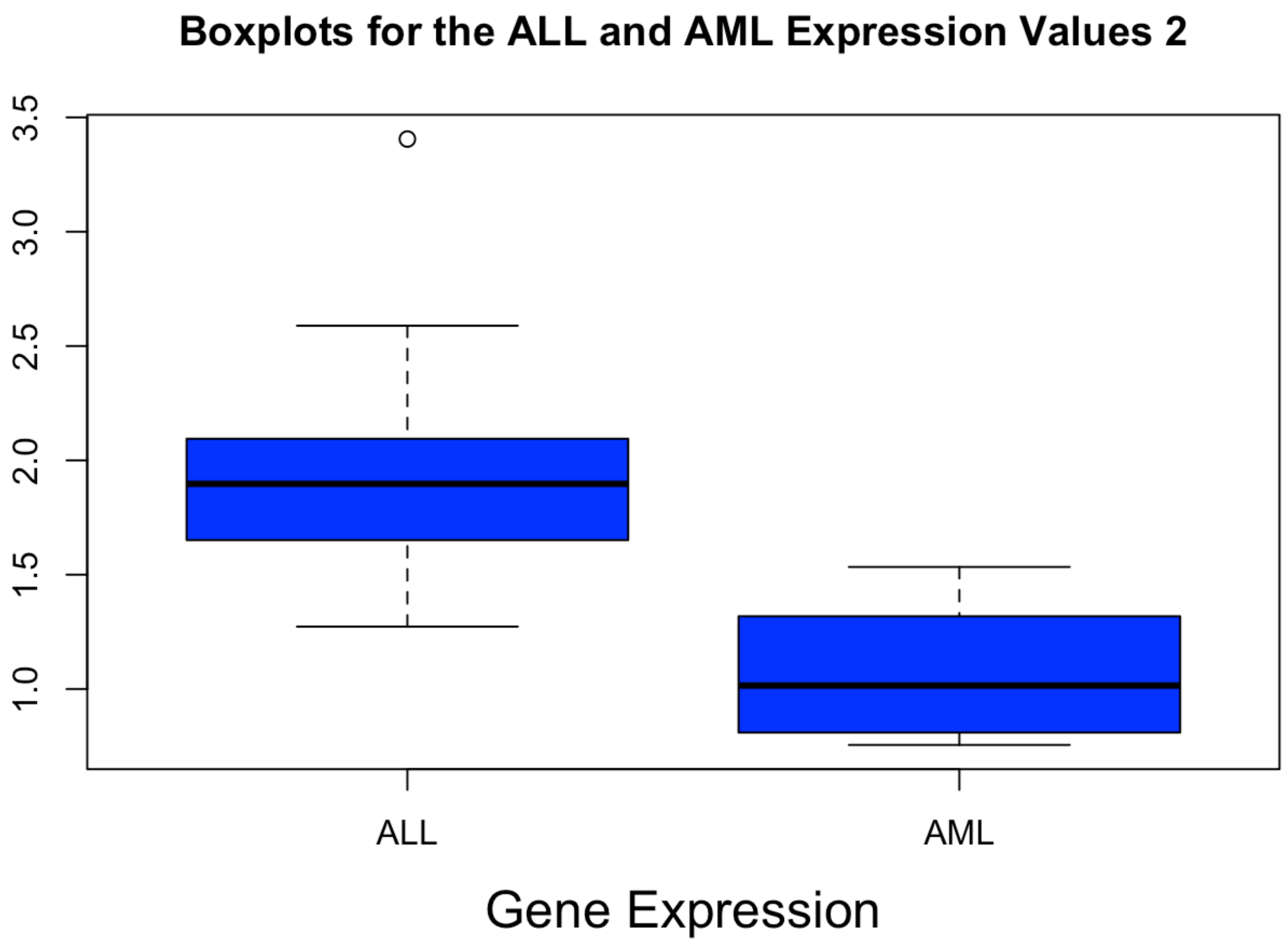
```
##       [,1]
##  [1,] "Macmarcks"
##  [2,] "VIL2 Villin 2 (ezrin)"
##  [3,] "PRE-MRNA SPLICING FACTOR SRP20"
##  [4,] "Oncoprotein 18 (Op18) gene"
##  [5,] "PROTEIN PHOSPHATASE PP2A, 65 KD REGULATORY SUBUNIT, ALPHA ISOFORM"
##  [6,] "UBE1 Ubiquitin activating enzyme E1"
##  [7,] "INTERFERON GAMMA UP-REGULATED I-5111 PROTEIN PRECURSOR"
##  [8,] "GNB1 Guanine nucleotide binding protein (G protein), beta polypeptide 1"
##  [9,] "Suppressor for yeast mutant"
## [10,] "Translation initiation factor 3 47 kDa subunit mRNA"
##       [,2]
##  [1,] "HG1612-HT1612_at"
##  [2,] "X51521_at"
##  [3,] "L10838_at"
##  [4,] "M31303_rna1_at"
##  [5,] "J02902_at"
##  [6,] "M58028_at"
##  [7,] "L07633_at"
##  [8,] "X04526_at"
##  [9,] "Y10807_s_at"
## [10,] "U94855_at"
```

B. Produce 4 separate side-by-side, vertical boxplots for the ALL and AML expression values for the top 4 genes expressed most similarly to CCND3. Compare these side-by-side boxplots to that produced using CCND3 (Cyclin D3) expression and comment on the similarities.

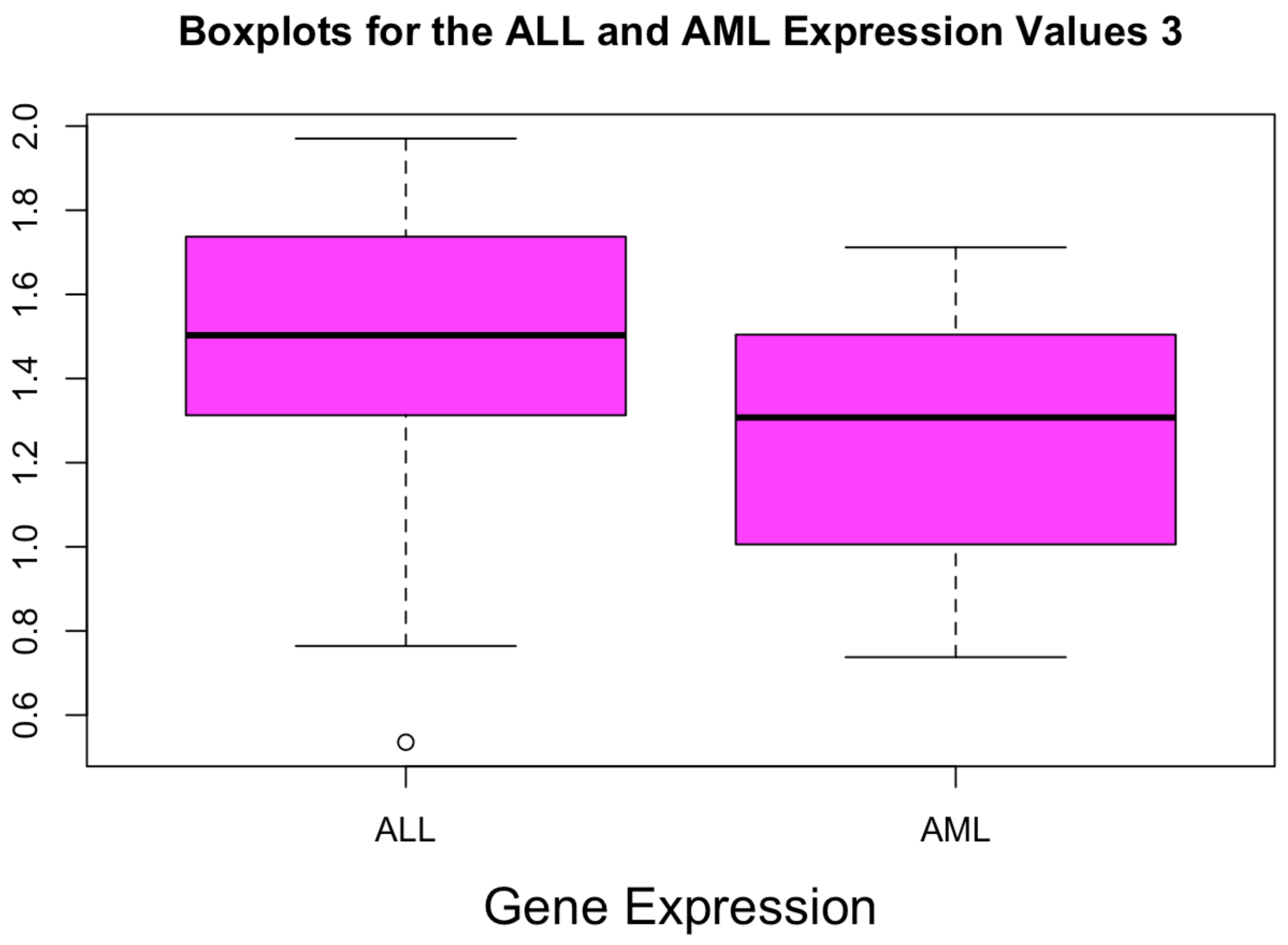
```
gol<- as.data.frame(t(golub)) # recall golub data original matrix and set as data fra
me.
boxplot(gol[,closeg[[1]][[1]][1]] ~ golub.df$Classification, #value 1st gene which ha
s similirity with CCND3
        cex.lab = 1.5,
        main = "Boxplots for the ALL and AML Expression Values 1",
        ylab = NULL,
        xlab = "Gene Expression",
        col=c("red"))
```



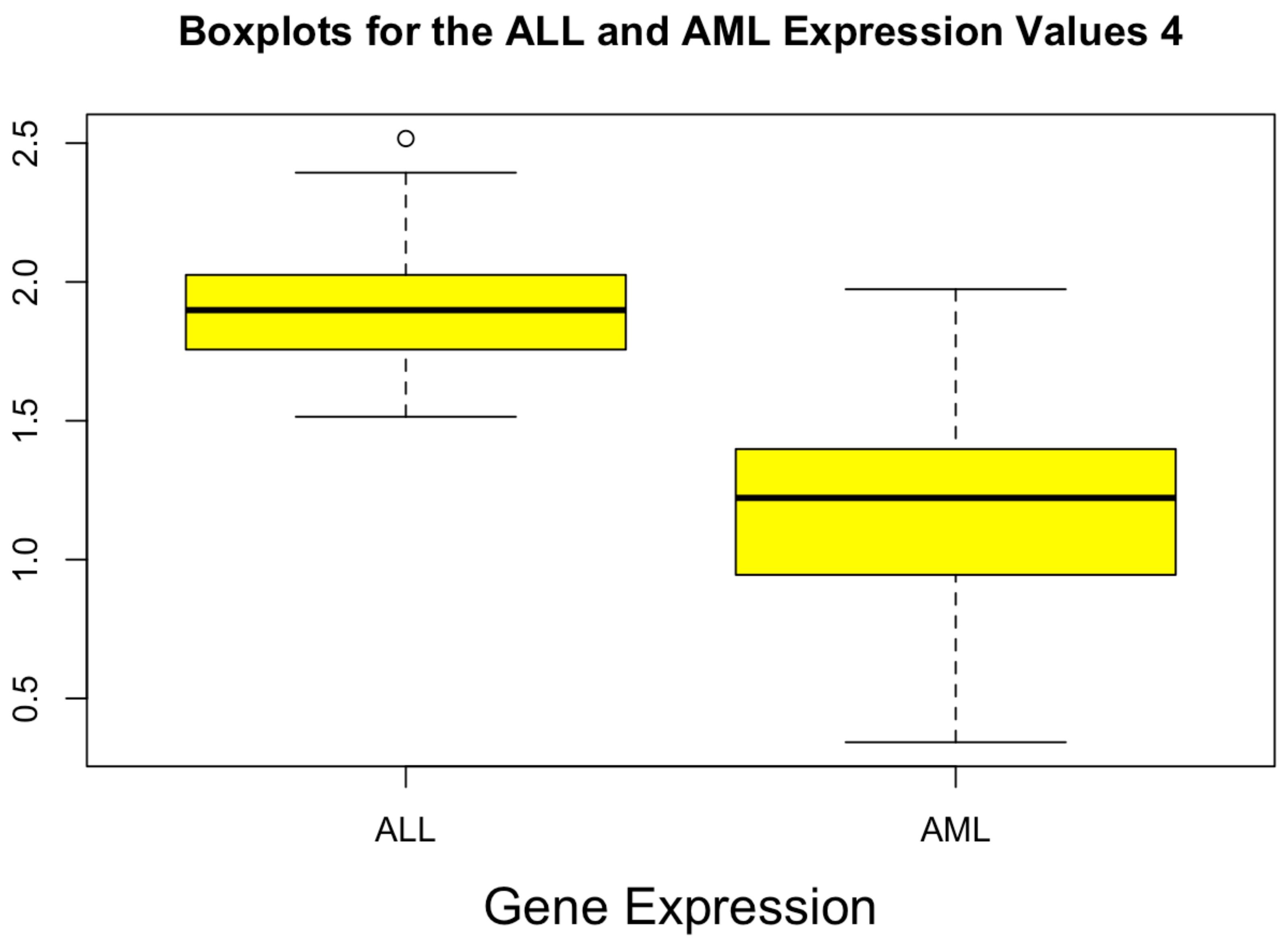
```
boxplot(gol[,closeg[[1]][[1]][2],] ~ golub.df$Classification,
        cex.lab = 1.5,
        main = "Boxplots for the ALL and AML Expression Values 2",
        ylab = NULL,
        xlab = "Gene Expression",
        col=c("Blue"))
```



```
boxplot(gol[,closeg[[1]][[1]][3],] ~golub.df$Classification,
        cex.lab = 1.5,
        main = "Boxplots for the ALL and AML Expression Values 3",
        ylab = NULL,
        xlab = "Gene Expression",
        col=c("magenta"))
```



```
boxplot(gol[,closeg[[1]][[1]][4],] ~golub.df$Classification,
        cex.lab = 1.5,
        main = "Boxplots for the ALL and AML Expression Values 4",
        ylab = NULL,
        xlab = "Gene Expression",
        col=c("yellow"))
```



C. Use grep() to find all the other genes that contain “Cyclin” in their names and compare their smallest distances to the distances found using genefinder() above. How do the distances differ?

```
cyclins=grep("Cyclin",colnames(golub.df), ignore.case=TRUE) # get index
golub.df.index=golub.df[,cyclins] # compute index cyclins to golub.gnames to get gene
s names
colnames(golub.df.index) #Find column names containing cyclins
```

```
##  [1] "CCND2 Cyclin D2"
##  [2] "Prostacyclin synthase"
##  [3] "Calcyclin"
##  [4] "Tetracycline transporter-like protein mRNA"
##  [5] "CDK2 Cyclin-dependent kinase 2"
##  [6] "CCND3 Cyclin D3"
##  [7] "CDKN1A Cyclin-dependent kinase inhibitor 1A (p21, Cip1)"
##  [8] "CCNH Cyclin H"
##  [9] "Cyclin-dependent kinase 4 (CDK4) gene"
## [10] "Cyclin G2 mRNA"
## [11] "Putative cyclin G1 interacting protein mRNA, partial sequence"
## [12] "Cyclin A1 mRNA"
## [13] "Cyclin-selective ubiquitin carrier protein mRNA"
## [14] "HMOX1 Heme oxygenase (decycling) 1"
## [15] "CDK6 Cyclin-dependent kinase 6"
## [16] "Cyclin G1 mRNA"
## [17] "SPHAR gene for cyclin-related protein"
## [18] "CCNF Cyclin F"
```

```
D<-data.frame(matrix(golub)) # create new data frame
dist.cyclin <- dist(D[cyclins,],method="euclidian") # find euclidean distance cyclins
in golub data.
distanceMatrix <- as.matrix(dist.cyclin) # create matrix
rownames(distanceMatrix) <- colnames(distanceMatrix) <-golub.gnames[cyclins ,3]# inse
r value, row name, col names
distanceMatrix[1:5,1:5] # show distance matrix 5x5 including probes set
```

```
##                   D13639_at D38145_at HG2788-HT2896_at L11669_at M68520_at
## D13639_at           0.00000   2.40784          1.24923   1.72804   1.94899
## D38145_at           2.40784   0.00000          1.15861   0.67980   0.45885
## HG2788-HT2896_at    1.24923   1.15861          0.00000   0.47881   0.69976
## L11669_at           1.72804   0.67980          0.47881   0.00000   0.22095
## M68520_at           1.94899   0.45885          0.69976   0.22095   0.00000
```

Based on the table, distance that genefinder produce are larger that cyclin genes.

PROBLEM 3

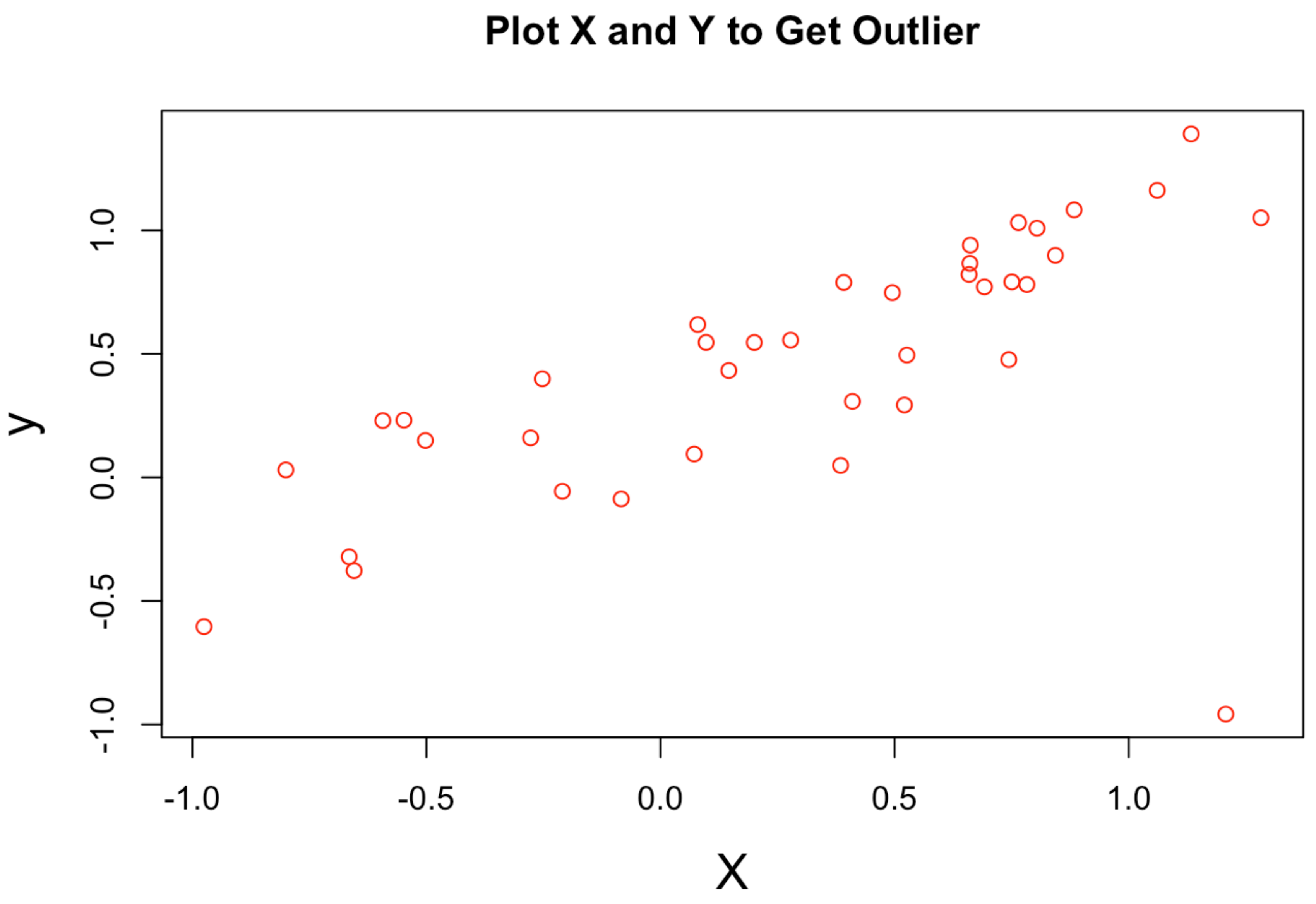
MCM3. In the example for MCM3, a plot shows that there is an outlier

A. Plot the data and invent a manner to find the row number of the outlier.

```
mcm3=grep("MCM3",colnames(golub.df), ignore.case=TRUE) # get index mcm3
df<-data.frame(matrix(golub)) # create new data frame (Matrix value)
x <- golub.df[,mcm3[1]] # set value MCM3 in x
y <- golub.df[,mcm3[2]] # set value MCM3 in y
cor(x,y) # find corelation
```

```
## [1] 0.6376217
```

```
plot(x,y, # create diagram
     cex.lab = 1.5,
        main = "Plot X and Y to Get Outlier",
        ylab = NULL,
        xlab = "X",
        col=c("red"))
```



The value of cor (x,y) is positive which means that larger values of x occur to gether with larger values of y.

B. Remove the outlier and test the correlation coefecient. Compare the results to those above.

```
which.min(y) # smallest outlier = 21
```

```
## [1] 21
```

```
cor.test(x[-21],y[-21]) # test corelation coofecient (we remove outlier)
```

```
##
##  Pearson's product-moment correlation
##
## data:  x[-21] and y[-21]
## t = 10.695, df = 35, p-value = 1.42e-12
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.7690824 0.9341905
## sample estimates:
##      cor
## 0.875043
```

In this case, cor.test(x[-21],y[-21]) > cor.test(x,y)

C. Perform the bootstrap to construct a con dence interval.

```
nboot <- 1000  # set total boot
boot <- data.frame(matrix(0,nrow=nboot,ncol = 1)) # create data frame with matrix for
mat for bootstrap
data <- data.frame(matrix(c(x[-21],y[-21]),ncol=2,byrow=FALSE))#create new data frame
for mcm3 after remove outlier.

for (i in 1:nboot) { # do for loop
  dat.boot <- data[sample(1:nrow(data),replace=TRUE),] # get value for mcm3 after rem
ove outlier.
  boot[i,] <- cor(dat.boot)[2,1] # find corelation coofecient between mcm3 after remo
ve outlier with bootstrap
}

quantile(boot[,1],c(0.025,0.975)) #confidance interval (.25 ane .975)
```

```
##      2.5%     97.5%
## 0.7847832 0.9344559
```

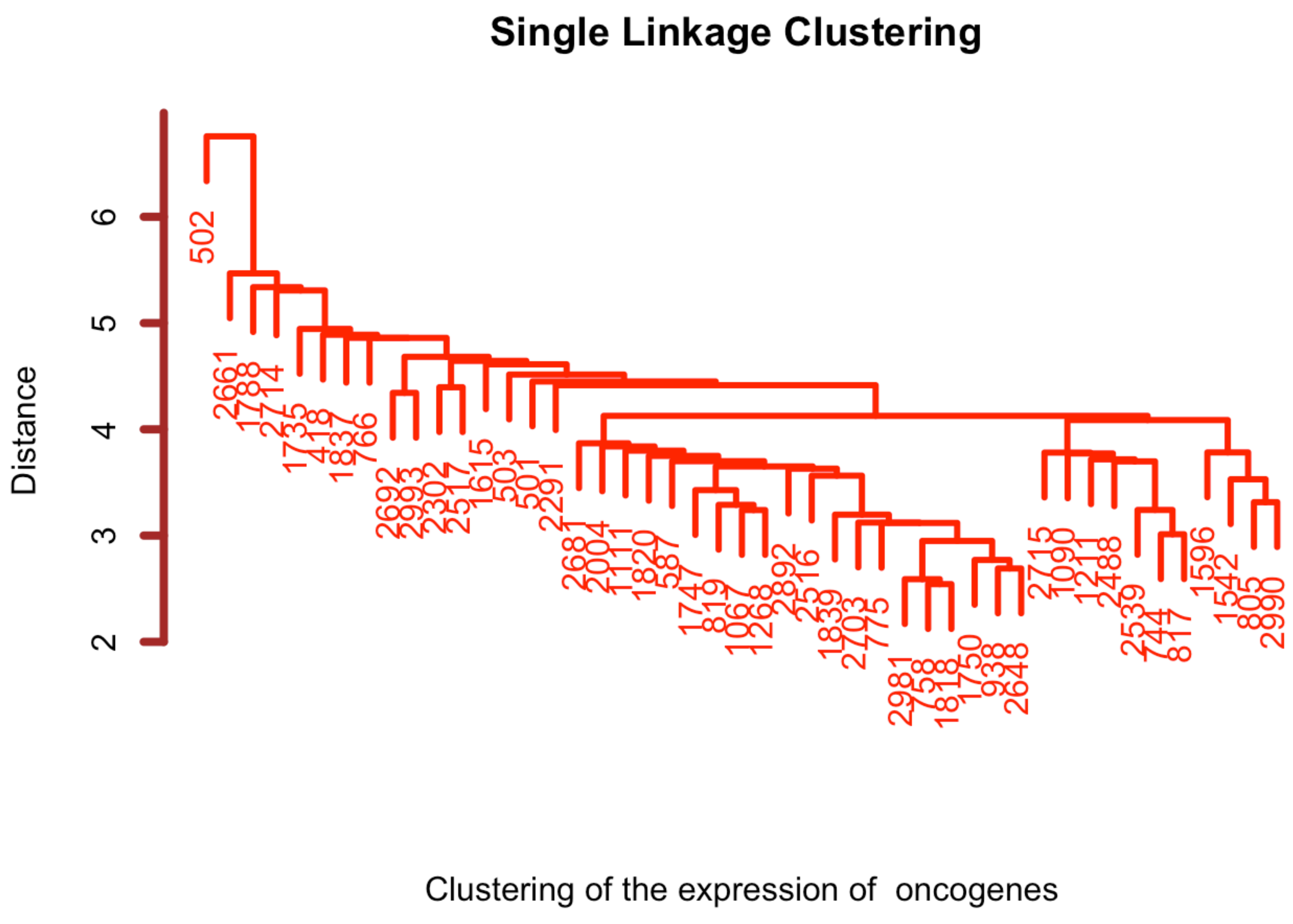
we observed that 97.5% confidence interval is larger than that found by cor.test(x,y) and cor.test(x[-21],y[-21]).

PROBLEM 4

Cluster analysis on a portion of the Golub data

A. Use the grep() function with the string oncogene to select the oncogenes from the Golub data and plot the tree from a single linkage cluster analysis.

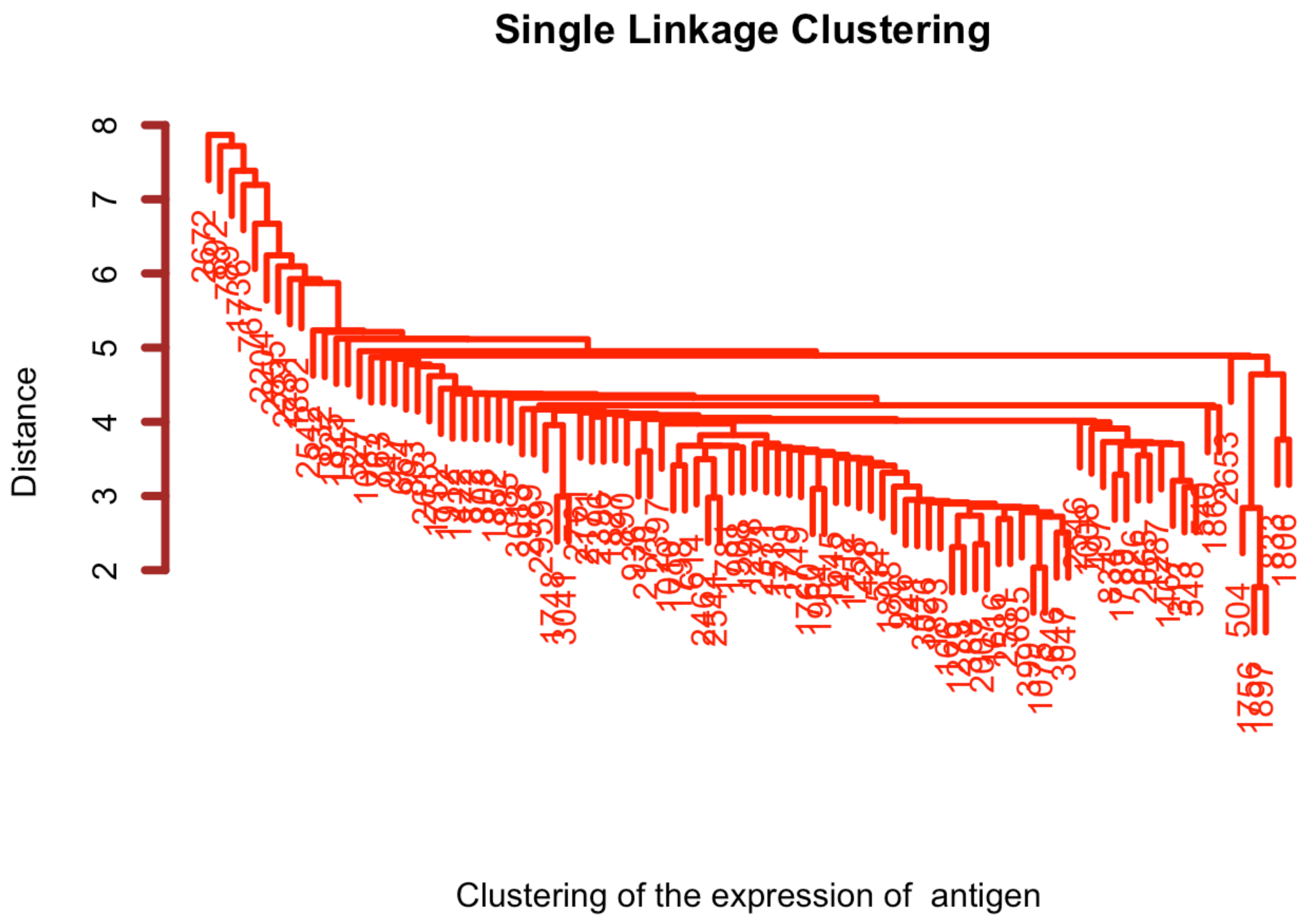
```
gf <-as.data.frame(golub)# re-create golub as data frame
o1=grep("oncogene",colnames(golub.df), ignore.case=TRUE) # get index
golub.df.index= (gf[o1,]) #get position oncogene in golub matrix
F <- hclust(dist(golub.df.index, method="euclidian"),method="single")# get distance a
nd will return value of  golub.df.index in the single linkage cluster.
#plot Dendogram for oncogene
plot(F,
     lwd=3,
     col="red",
     col.axis = "brown",
     ylab="Distance",
     xlab="Clustering of the expression of  oncogenes",
     main="Single Linkage Clustering ",
     sub=NA,
     axes=FALSE)
axis(side = 2, at = seq(2, 8,1 ), col = "brown",labels = TRUE, lwd = 4)
```



B. Do you observe meaningful clusters? Answer: No, the cluster that shown in the graph look not normal.

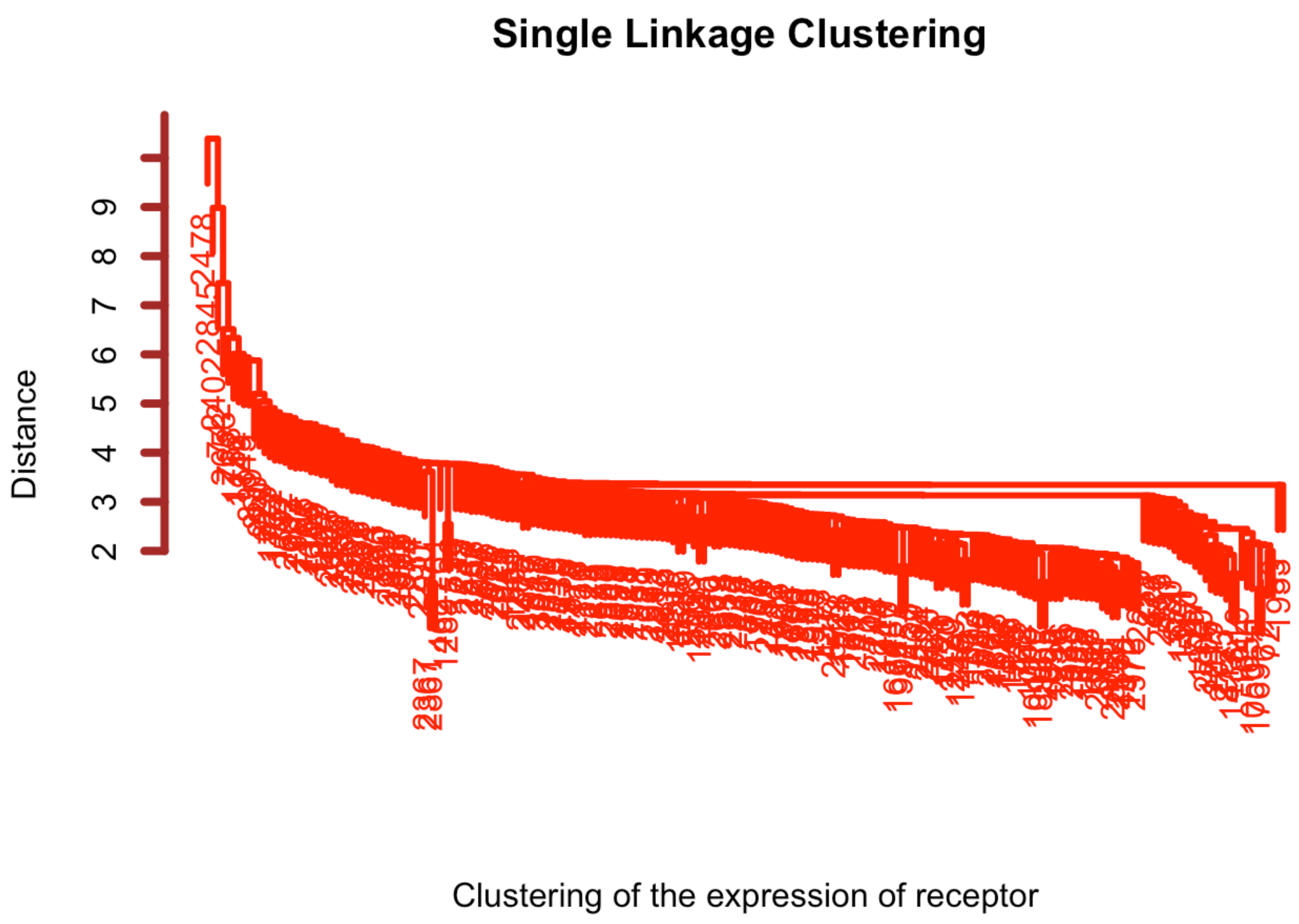
C. Use grep() to select the antigenes and answer the same questions.

```
o2=grep("antigen",colnames(golub.df), ignore.case=TRUE) # get index
golub.df.index1= (gf[o2,]) #get position antigen in golub matrix
F2 <- hclust(dist(golub.df.index1,method="euclidian"),method="single")
plot(F2,
     lwd=3,
     col="red",
     col.axis = "brown",
     ylab="Distance",
     xlab="Clustering of the expression of  antigen",
     main="Single Linkage Clustering ",
     sub=NA,
     axes=FALSE)
axis(side = 2, at = seq(2, 8,1 ), col = "brown",labels = TRUE, lwd = 4)
```



D.Use grep() to select the receptor genes and answer the same questions.

```
o3=grep("receptor",colnames(golub.df), ignore.case=TRUE) # get index
golub.df.index2= (gf[o3,]) #get position receptor in golub matrix
F3 <- hclust(dist(golub.df.index2,method="euclidian"),method="single")
plot(F3,
     lwd=3,
     col="red",
     col.axis = "brown",
     ylab="Distance",
     xlab="Clustering of the expression of receptor",
     main="Single Linkage Clustering ",
     sub=NA,
     axes=FALSE)
axis(side = 2, at = seq(2, 12, 1 ), col = "brown",labels = TRUE, lwd = 4)
```



PROBLEM 5

Principal Components Analysis on part of the ALL data

A. Construct an expression set with the patients with B-cell in stage B1, B2, and B3. Compute the corresponding ANOVA p-values for all the gene expressions. Construct the expression set with the p-values smaller than 0.001. Report the dimensionality of the data matrix with gene expressions that have signi cantly di erent means across the B1, B2, and B3 leukemia subtypes.

```
allb.df = data.frame(ALL[,ALL$BT %in% c("B1","B2","B3")])#creating data frame in ALLB
as Data Frame
allb.df[ allb.df == "NA" ] = NA # Replace "NA" strings with real <NA>
has.expression = grep("_at$|_st$",names(allb.df)) # Get colum that contain expression
data
Fc = factor(allb.df$BT, labels=c("B1", "B2","B3")) # using factor to group data base
on B stage
allb.df$BT=Fc
allb.df$BT
```

```
##  [1] B2 B2 B1 B2 B1 B1 B1 B2 B2 B3 B3 B3 B2 B3 B2 B3 B2 B3 B2 B2 B2 B1 B1
## [24] B2 B1 B2 B1 B2 B2 B2 B2 B1 B2 B2 B2 B2 B2 B2 B2 B2 B2 B1 B2 B2 B3 B3
## [47] B3 B3 B3 B3 B1 B1 B1 B1 B3 B3 B3 B3 B3 B3 B3 B3 B1 B3 B1 B2 B2 B1 B3
## [70] B2 B2 B3 B1 B2 B2 B2 B1 B2
## Levels: B1 B2 B3
```

```
matrix.value = apply(allb.df[has.expression], 2, function(y) {
  m<-as.matrix(y)
  rownames(m)<-y}) # change data frame to matrix format
anova.pValues <- apply(matrix.value, 2, function(x) { anova(lm(x~allb.df$BT ))$Pr[1]
} ) # anova tes and get P Value
head(anova.pValues)
```

```
##   X1000_at   X1001_at X1002_f_at X1003_s_at   X1004_at   X1005_at
## 0.02452274 0.77528321 0.36651993 0.44163026 0.56090606 0.01738907
```

B. Are the correlations between the patients positive

```
ALLBcor <- t(allb.df[,anova.pValues<0.001])# get annova p Value less than 0.001
dim(ALLBcor)# get dimension
```

```
## [1] 499  78
```

```
min(cor(ALLBcor))# find corralation
```

```
## [1] 0.5805595
```

yes, it has corelation between positive, using cor to get corelation of coofecient.

C. Compute the eigenvalues of the correlation matrix. Report the largest ve. Are the first three larger than one?

```
eigen(cor(ALLBcor))$values[1:5] # compute eigen value (largest 5)
```

```
## [1] 65.2016203  2.9652965  2.4781567  0.7556439  0.6040647
```

Ye, all of them larger than 1.

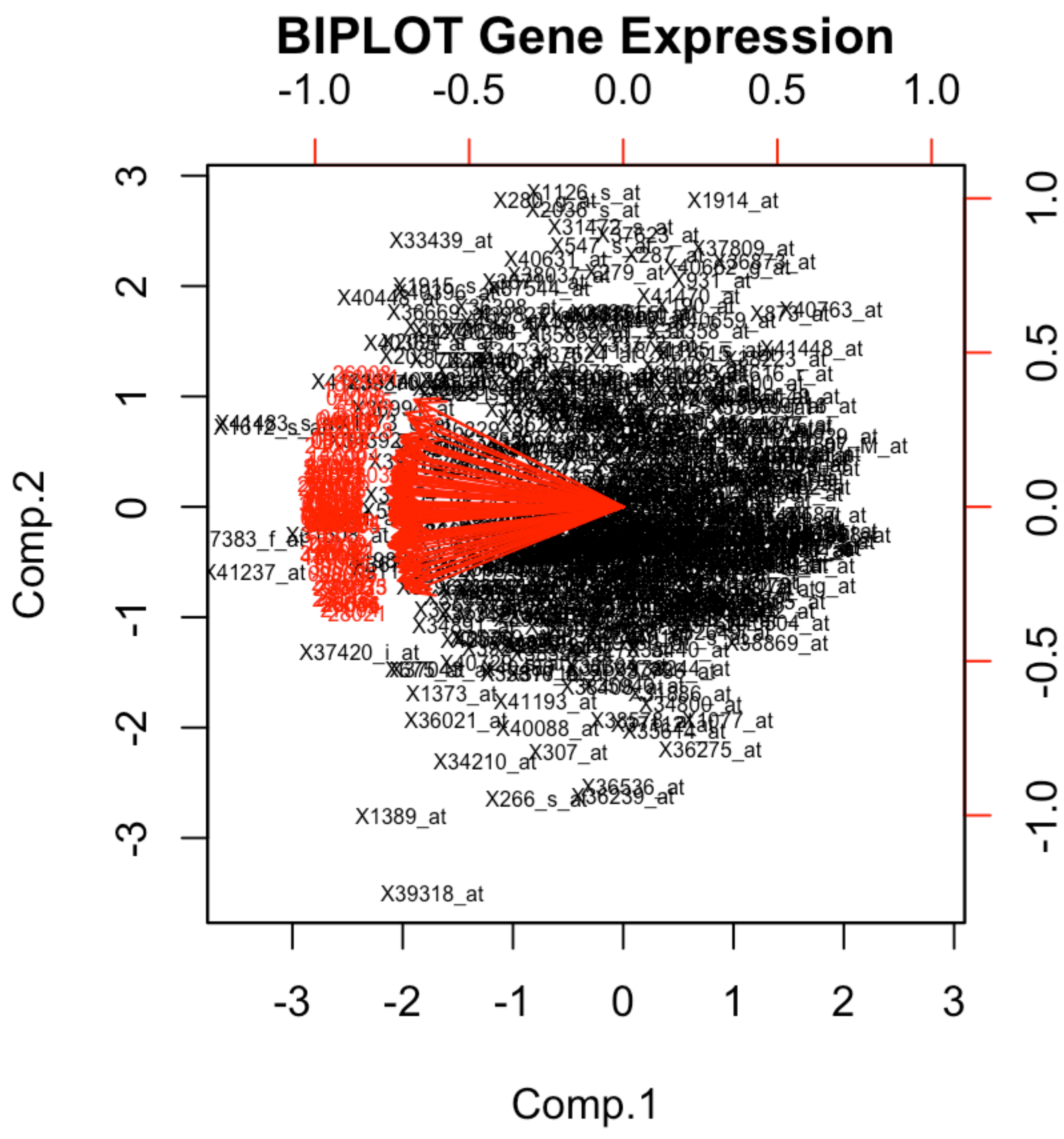
D.Program a bootstrap of the largest five eigenvalues. Report the bootstrap 95% confidence intervals and draw relevant conclusions.

```
data_new <- (ALLBcor) #transpose data ALLBcor
p <- ncol(data_new); n <- nrow(data_new) ; nboot<-1000
eigenvalues <- array(dim=c(nboot,p))# create array to compute value p

for (i in 1:nboot) { # do for loop in bootstrap(re-sampling)
  dat.star <- data_new[sample(1:n,replace=TRUE),] # input p value positive from anova
test
  eigenvalues[i,] <- eigen(cor(dat.star))$values # input 5 eigenvalues in bootstrap
}

for (j in 1:p) {
  print(quantile(eigenvalues[,j],c(0.025,0.95))) # print confidence intervals (2.5 an
d 95 %)
}
```

```
##     2.5%      95% 
## 63.46735 66.53009 
##     2.5%      95% 
## 2.569997 3.446602 
##     2.5%      95% 
## 2.098630 2.820759 
##      2.5%       95% 
## 0.6554786 0.9541170 
##      2.5%       95% 
## 0.5114320 0.7274547 
##      2.5%       95% 
## 0.4104842 0.5716280 
##      2.5%       95% 
## 0.3427172 0.4707076 
##      2.5%       95% 
## 0.2849018 0.3939560 
##      2.5%       95% 
## 0.2568021 0.3439097 
##      2.5%       95% 
## 0.2352953 0.3101343 
##      2.5%       95% 
## 0.2173762 0.2854734 
##      2.5%       95% 
## 0.1944976 0.2628884 
##      2.5%       95% 
## 0.1781400 0.2386698 
##      2.5%       95% 
## 0.1626594 0.2186958 
##      2.5%       95% 
## 0.1519522 0.1995650 
##       2.5%       95% 
## 0.1415362 0.1832185 
##       2.5%       95% 
## 0.1311742 0.1702739 
##       2.5%       95% 
## 0.1224569 0.1585332 
##       2.5%       95% 
## 0.1133810 0.1485926 
##       2.5%       95% 
## 0.1068054 0.1397126 
##       2.5%       95% 
## 0.1004048 0.1304836 
##        2.5%         95% 
## 0.09483436 0.12264172 
##        2.5%         95% 
## 0.08977182 0.11586995 
##        2.5%         95% 
## 0.08462224 0.10993622 
##        2.5%         95% 
## 0.08090152 0.10349304 
##        2.5%         95% 
## 0.07650867 0.09835891 
##        2.5%         95% 
## 0.07295678 0.09350472 
##        2.5%         95% 
## 0.06931715 0.08871260 
##        2.5%         95% 
## 0.06600121 0.08460091 
##        2.5%         95% 
## 0.06216931 0.08013584 
##        2.5%         95% 
## 0.05970115 0.07614264 
##        2.5%         95% 
## 0.05689112 0.07269273 
##        2.5%         95% 
## 0.05450587 0.06918475 
##        2.5%         95% 
## 0.05199530 0.06600342 
##        2.5%         95% 
## 0.04969646 0.06282460 
##        2.5%         95% 
## 0.04730614 0.06033657 
##        2.5%         95% 
## 0.04544341 0.05750841 
##        2.5%         95% 
## 0.04337511 0.05498135 
##        2.5%         95% 
## 0.04157996 0.05272724 
##        2.5%         95% 
## 0.03960476 0.05056544
##       2.5%        95% 
## 0.03791772 0.04856891 
##       2.5%        95% 
## 0.03647594 0.04641690 
##       2.5%        95% 
## 0.03513456 0.04466253 
##      2.5%       95% 
## 0.0333855 0.0428902 
##       2.5%        95% 
## 0.03220346 0.04099237 
##       2.5%        95% 
## 0.03076351 0.03944421 
##       2.5%        95% 
## 0.02963324 0.03763773 
##       2.5%        95% 
## 0.02844526 0.03640434 
##       2.5%        95% 
## 0.02721334 0.03468792 
##       2.5%        95% 
## 0.02610117 0.03344220 
##       2.5%        95% 
## 0.02491942 0.03181502 
##       2.5%        95% 
## 0.02345325 0.03032198 
##       2.5%        95% 
## 0.02267295 0.02904860 
##       2.5%        95% 
## 0.02165964 0.02782985 
##       2.5%        95% 
## 0.02075117 0.02670198 
##       2.5%        95% 
## 0.01977859 0.02560728 
##       2.5%        95% 
## 0.01885887 0.02437784 
##       2.5%        95% 
## 0.01805583 0.02322146 
##       2.5%        95% 
## 0.01720105 0.02226561 
##       2.5%        95% 
## 0.01650597 0.02128239 
##       2.5%        95% 
## 0.01579127 0.02048768 
##       2.5%        95% 
## 0.01495540 0.01936473 
##       2.5%        95% 
## 0.01437228 0.01844721 
##       2.5%        95% 
## 0.01372608 0.01768002 
##       2.5%        95% 
## 0.01303097 0.01692112 
##        2.5%        95% 
## 0.01243689 0.01616288 
##        2.5%        95% 
## 0.01186532 0.01538603 
##        2.5%        95% 
## 0.01129389 0.01467916 
##        2.5%        95% 
## 0.01079552 0.01394969 
##        2.5%        95% 
## 0.01015717 0.01321890 
##         2.5%         95% 
## 0.009621671 0.012554026 
##        2.5%        95% 
## 0.00901668 0.01174581 
##         2.5%         95% 
## 0.008568908 0.011042292 
##         2.5%         95% 
## 0.008043656 0.010491586 
##         2.5%         95% 
## 0.007518252 0.009838128 
##         2.5%         95% 
## 0.006929337 0.009190310 
##         2.5%         95% 
## 0.006361560 0.008461367 
##         2.5%         95% 
## 0.005615072 0.007676405
```

E. Plot the gene expressions while using the first two principal components as the axes.

```
biplot(princomp(data_new,cor=TRUE),pc.biplot=T,cex=0.5,expand=0.8, main="BIPLOT Gene
Expression") # Draw biplot
```


BIPLOT Gene Expression
Comp.1
Comp.2
-1.0
-0.5
0.0
0.5
1.0
-3
-2
-1
0
1
2
3
X1914_at
X33439_at
X1126_s_at
X40763_at
X41448_at
X37420_i_at
X41237_at
X1373_at
X41193_at
X36021_at
X40088_at
X36275_at
X34210_at
X307_at
X266_s_at
X1389_at
X39318_at
X38869_at