Assignment 5: RNA-seq Assignment Date: Monday October 24, 2022 Due Date: Monday October 31 2022 @ 11:59pm

Assignment Overview In this assignment you will explore a couple of aspects of RNA-seq (with a small introduction to clustering). For this assignment, you will have to generate some visualizations - we recommend R or Python, but use a language you are comfortable with!

Make sure to show your work/code in your writeup!

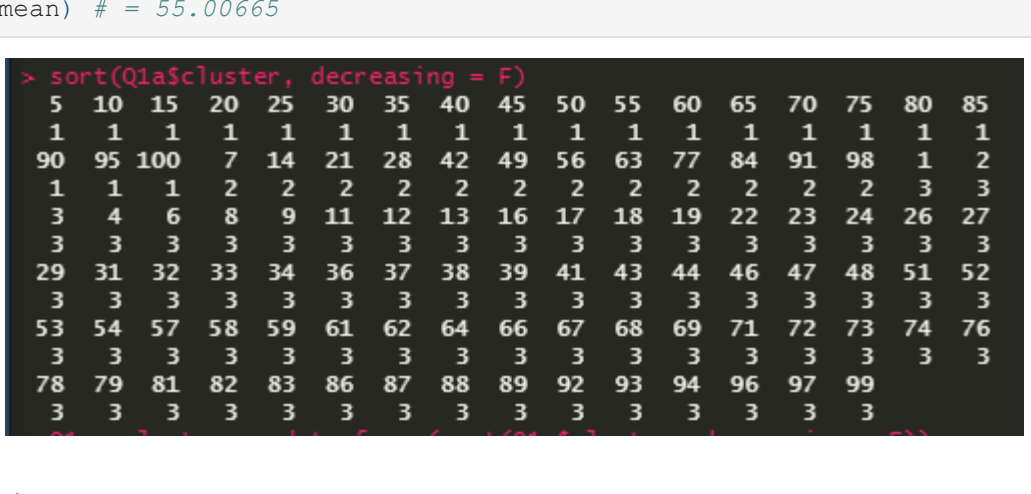
As a reminder, any questions about the assignment should be posted to Piazza.

Question 1. Time Series [20 pts] This file contains normalized expression values for 100 genes over 10 time points. Most genes have a stable background expression level, but some special genes show increased expression over the timecourse and some show decreased expression.

Question 1a. Cluster the genes using an algorithm of your choice. Which genes show increasing expression and which genes show decreasing expression, and how did you determine this? What is the background expression level (numerical value) and how did you determine this? [Hint: K-means and hierarchical clustering are common clustering algorithms you could try.]

```
Ans:
There are 12 genes that show decrease in expression over time and 20 genes that show increase
in expression over time.

The genes that show decreasing in expression are:
gene_7, gene_14, gene_21, gene_28, gene_42, gene_49, gene_56, gene_63, gene_77, gene_84, gene_91, gene_98

The genes that show increasing in expression are:
gene_5, gene_10, gene_15, gene_20, gene_25, gene_30, gene_35, gene_40, gene_45,
gene_50, gene_55, gene_60, gene_65, gene_70, gene_75, gene_80, gene_85, gene_90, gene_95, gene_100


The genes were seperated into clusters using kmeans clustering and also by hierarchial clustering.
The increasing and decreasing clusters were then identified using time point regression for each gene
in each cluster and then calculating the average slope.


Here cluster 3 corresponds to the background genes because it has an averaged slope of around -0.011
This cluster consists of 68 genes and the background expression level of these genes
"mean(background$Q1_mean) = 55.00665 "
```

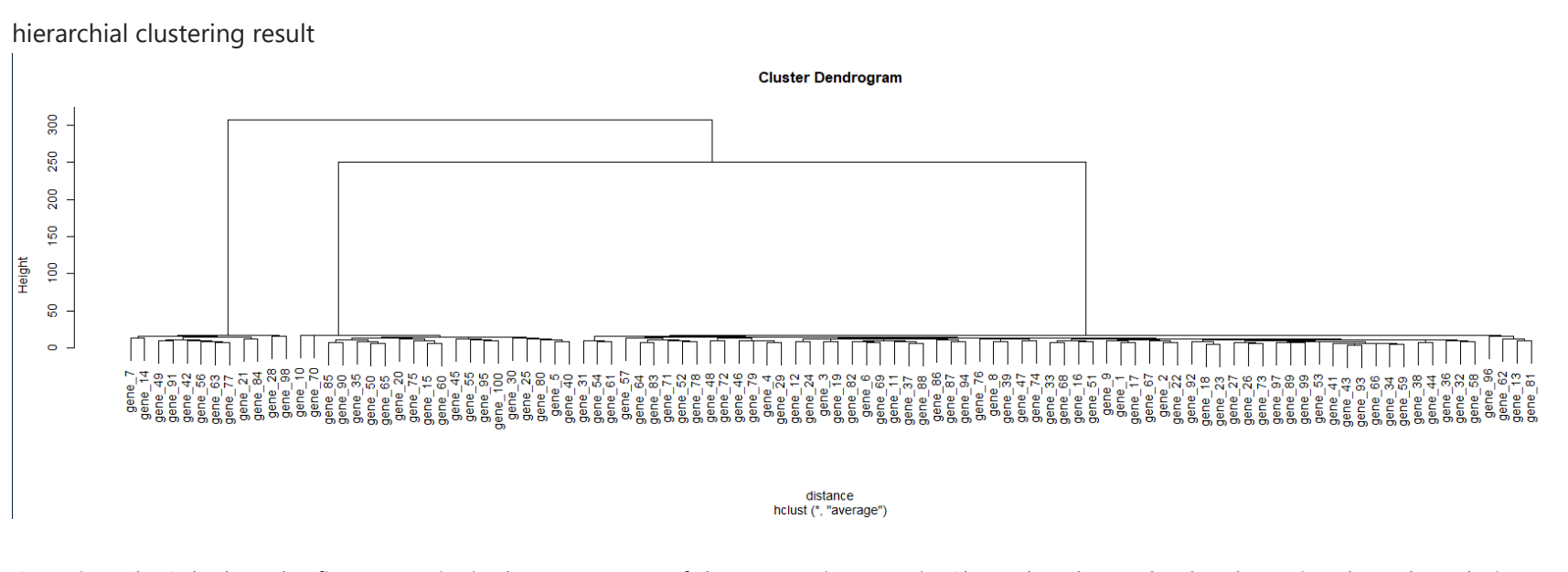
```
getwd()
setwd("D:/Comparative_Genomics/Assignment5/")

Expression_Data1 = read.csv("Expression1.txt", header = T, sep = "\t", row.names = 1)

#kmeans clustering
kmeans = kmeans(Expression_data1, centers = 3)
kmeans$cluster

#hierarchical clustering
distance = dist(Expression_Data[,c(1:10)], method = "manhattan")
clusters = hclust(distance, method = "average")
plot(clusters)


#time point regression

library(dplyr)
background = filter(Expression_Data1 , Expression_Data1$clusters == 3)
increasing = filter(Expression_Data1, Expression_Data1$clusters == 1)
decreasing = filter(Expression_Data1, Expression_Data1$clusters == 2)

background_expres = t(background[,c(1:10)])
increasing_expres = t(increasing[,c(1:10)])
decreasing_expres = t(decreasing[,c(1:10)])

background_slopes = c()
increasing_slopes = c()
decreasing_slopes = c()

time_point = c(1:10)
my_rownames = c(1:12)
for (i in my_rownames){
  x = decreasing_expres[,i]
  a = lm(time_point ~ x )
  required = a$coefficients[2]
  decreasing_slopes = append(decreasing_slopes, required)
}
mean(background_slopes) # = -0.0112971
mean(increasing_slopes) # =  0.1001658
mean(decreasing_slopes) # = -0.09934661


mean(background$Q1_mean) # = 55.00665
```



```
> sort(Q1a$cluster, decreasing = F)
  5  10  15  20  25  30  35  40  45  50  55  60  65  70  75  80  85
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
 90  95 100   7  14  21  28  42  49  56  63  77  84  91  98   1   2
  1   1   1   2   2   2   2   2   2   2   2   2   2   2   2   3   3
  3   4   6   8   9  11  12  13  16  17  18  19  22  23  24  26  27
  3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3
 29  31  32  33  34  36  37  38  39  41  43  44  46  47  48  51  52
  3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3
 53  54  57  58  59  61  62  64  66  67  68  69  71  72  73  74  76
  3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3
 78  79  81  82  83  86  87  88  89  92  93  94  96  97  99
  3   3   3   3   3   3   3   3   3   3   3   3   3   3   3
```

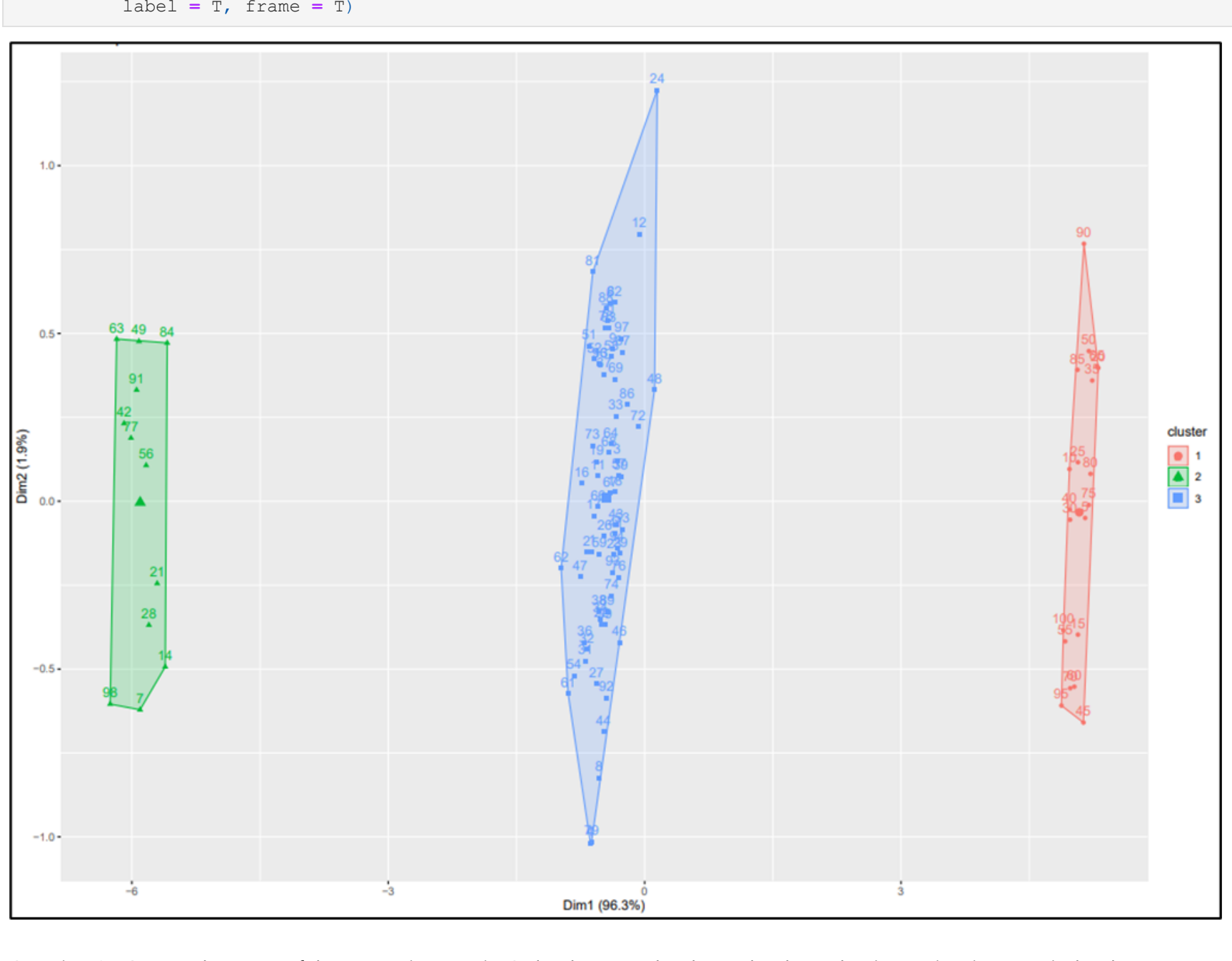
kmeans clustering result

hierarchial clustering result



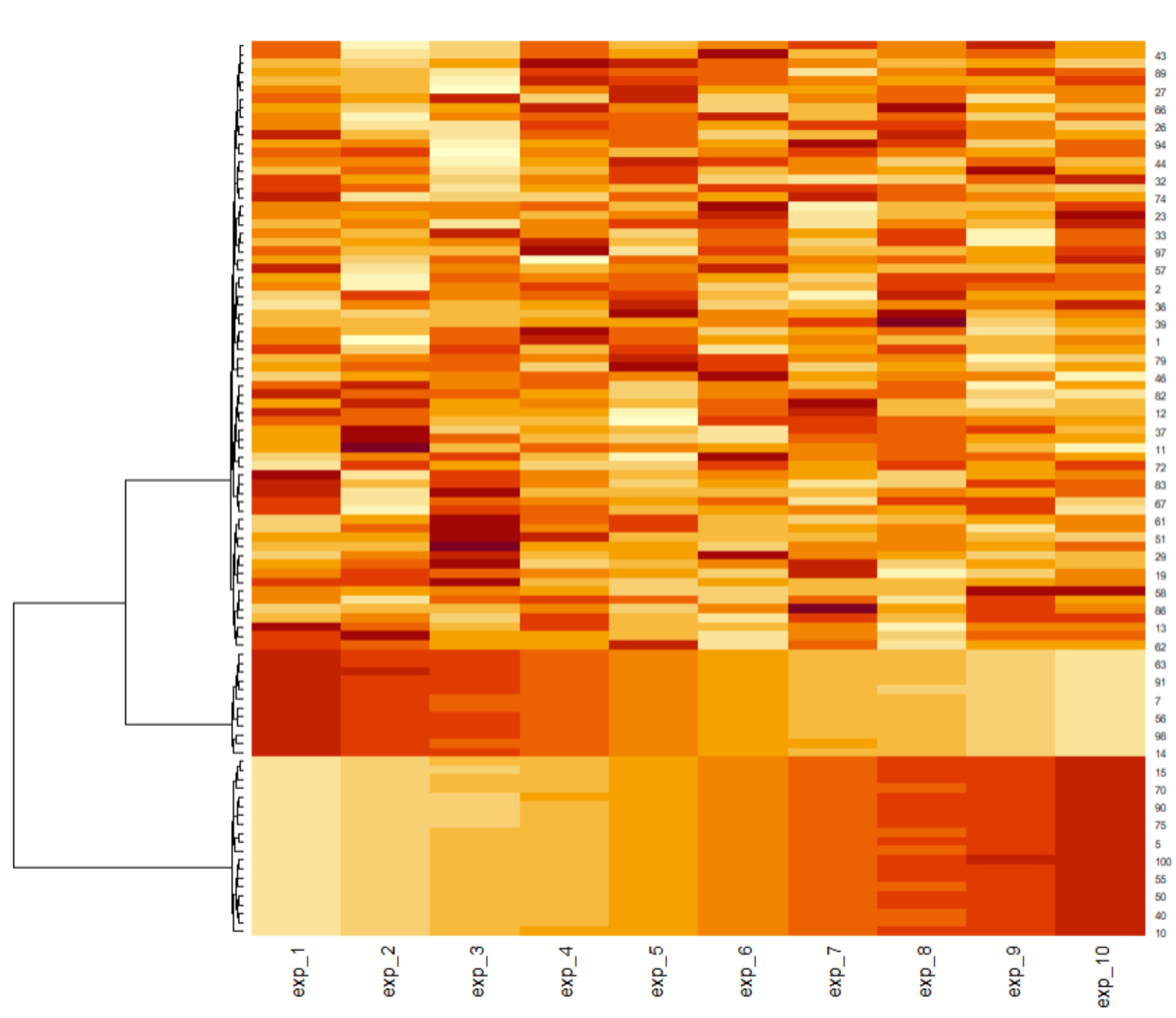
Question 1b. Calculate the first two principal components of the expression matrix. Show the plot and color the points based on their cluster from part (a). Does the PC1 axis, PC2 axis, neither, or both correspond to the clustering?

```
Ans:
Yes both the PC1 and PC2 axis correspond to the clustering results.
The same genes with increasing and decreasing expression were also calculated by
both kmeans and hierarchial clustering.


Script:
library(ggfortify)

head(Expression_Data1)
pca.res <- prcomp(Expression_Data1[,1:10] ,
                  center = T, retx=T, scale. = T)

pca.res <- prcomp(Expression_data1 , center = T, retx=T, scale = T)

autoplot(pca.res, data = Expression_Data1[,1:10],
         colour = Expression_Data1$clusters,
         label = T, frame = T)
```



Question 1c. Create a heatmap of the expression matrix. Order the genes by cluster, but keep the time points in numerical order.

```
Ans:
Expression_data2 = Expression_Data1[order(Expression_Data1$clusters),]
heatmap(as.matrix(Expression_data2[,1:10]), Colv = NA)
```

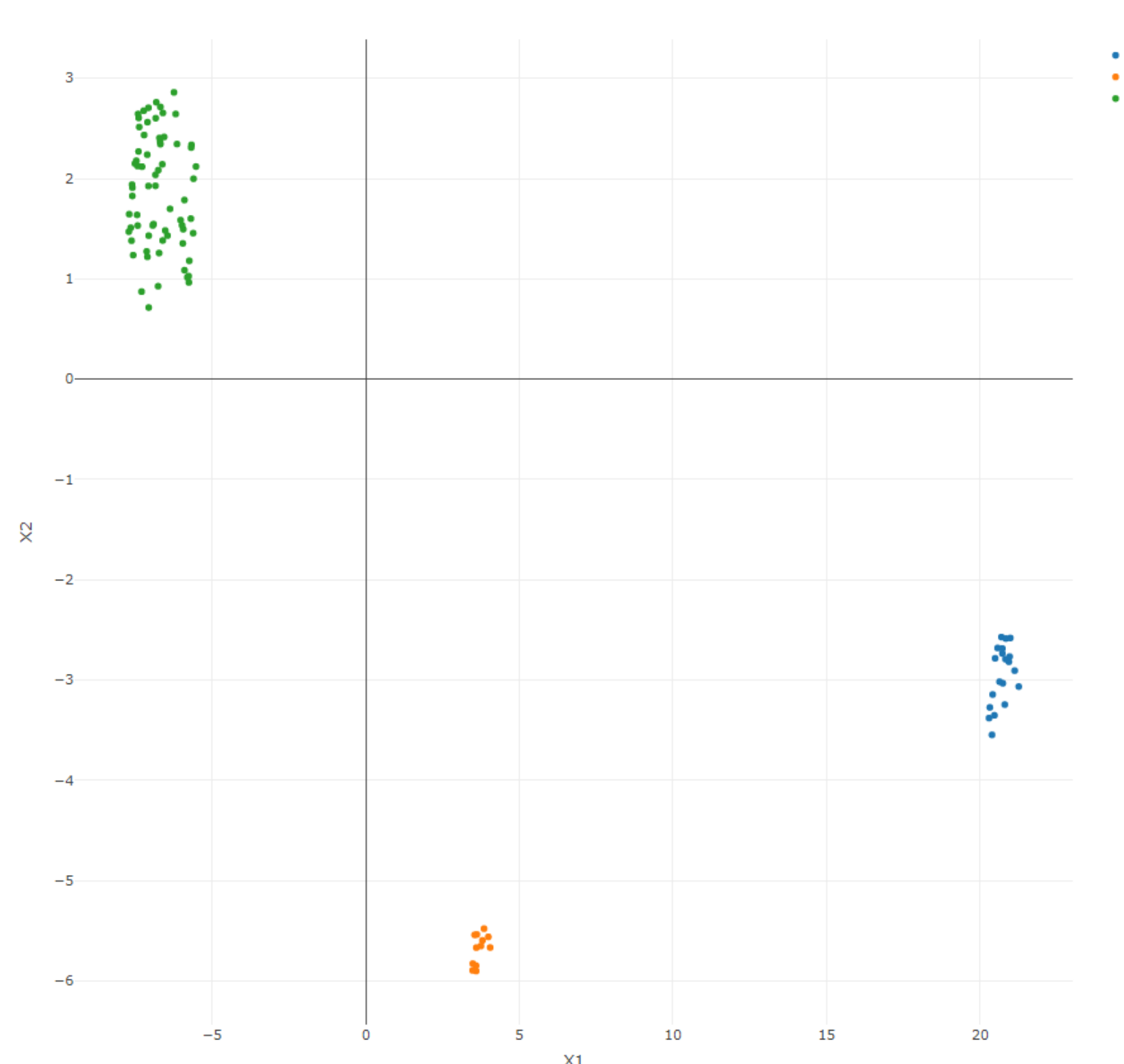


Question 1d. Visualize the expression data using t-SNE.

```
library(tsne)
library(plotly)
set.seed(0)
tsne <- tsne(Expression_data2[,1:10], initial_dims = 2)
tsne = data.frame(tsne)
pdb = cbind(tsne, Expression_data2$clusters)
fig = plot_ly(data = pdb, x = ~X1, y = ~X2, type = "scatter",
             mode = 'markers', split = ~ Expression_data2$clusters)
fig
```

Question 1e. Using the same data, visualize the expression data using UMAP.

```
library(umap)

data.umap = umap(Expression_Data1[,1:10], n_components = 3, random_state = 3)
layout = data.umap[["layout"]]
layout = data.frame(layout)
final = cbind(layout, Expression_data2$clusters)
head(final)
fig3 = plot_ly(final, x = ~X1, y = ~X2,
               mode = 'markers', split = ~ Expression_Data1$clusters )

fig3
```



Question 1f. In a few sentences, compare the (1) heatmap, (2) PCA, (3) t-SNE and (4) UMAP results. Be sure to comment on understandability, relative positioning of clusters, runtime, and any other significant factors that you see.

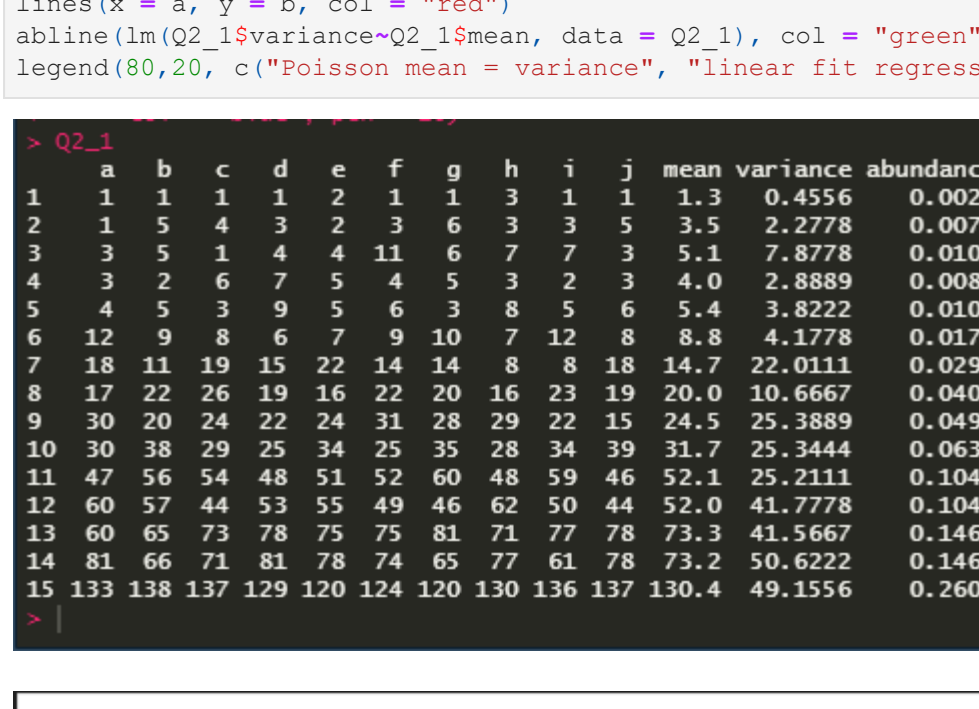
```
Ans:
Heatmap: This plot clearly bifurcates the increasing expression of genes over time (for the 20 genes)
    and decreasing expression of genes  over time ( for the 12 genes).
    i.e heat map bifurcates the three clusters along Y-axis and shows the steady increase and decrease
    over time of the respective gene expression along X-axis.
    This is the only method that clusters both the rows and columns and is able to present
    it on a single graph.

PCA: This plot divides the genes into three parts.
    Genes with an overall increase in expression are on the positive X-axis and
    the gene with an overall decrease in gene expression are on the negative X-axis.
    Like the heatmap, PCA does not show an increase in expression of each gene over time.
    PCA and clustering give the same results. The data points though bifurcated are all cluttered together.

t-SNE: This method uses probability theory.
    Compared to the PCA method, it takes significantly more time to converge,
    but presents significantly better insights when visualized
    (Like PCA the data points are not cluttered. They are bifurcated and clear).
    Here I project the gene expression data to two dimensions.
    tSNE result gives one false-positive value
    (i.e one low expressed gene as predicted by kmeans is cluttered with the normal values.)
    This method is less accurate than UMAP.

UMAP: This method is also used to visualize complex data into low dimensions.
    As the number of datapoints increase, UMAP becomes more time efficient as compared to the tSNE.
    This method did not give false positive value like the tSNE.
    Overall, I found UMAP to be the most convenient tool.
    It was faster than all the methods and gave a clear bifurcation between clusters.
```

Question 2. Sampling Simulation [10 pts] A typical human cell has ~250,000 transcripts, and a typical bulk RNA-seq experiment may involve millions of cells. Consequently, in an RNAseq experiment you may start with trillions of RNA molecules, although your sequencer will only give a few tens of millions of reads. Therefore your RNAseq experiment will be a small sampling of the full composition. We hope the sequences will be a representative sample of the total population, but if your sample is very unlucky or biased it may not represent the true distribution. We will explore this concept by sampling a small subset of transcripts (500 to 50000) out of a much larger set (1M) so that you can evaluate this bias.

In data1.txt with 100,000 lines we provide an abstraction of RNA-seq data where normalization has been performed and the number of times a gene name occurs corresponds to the number of transcripts in the sample.

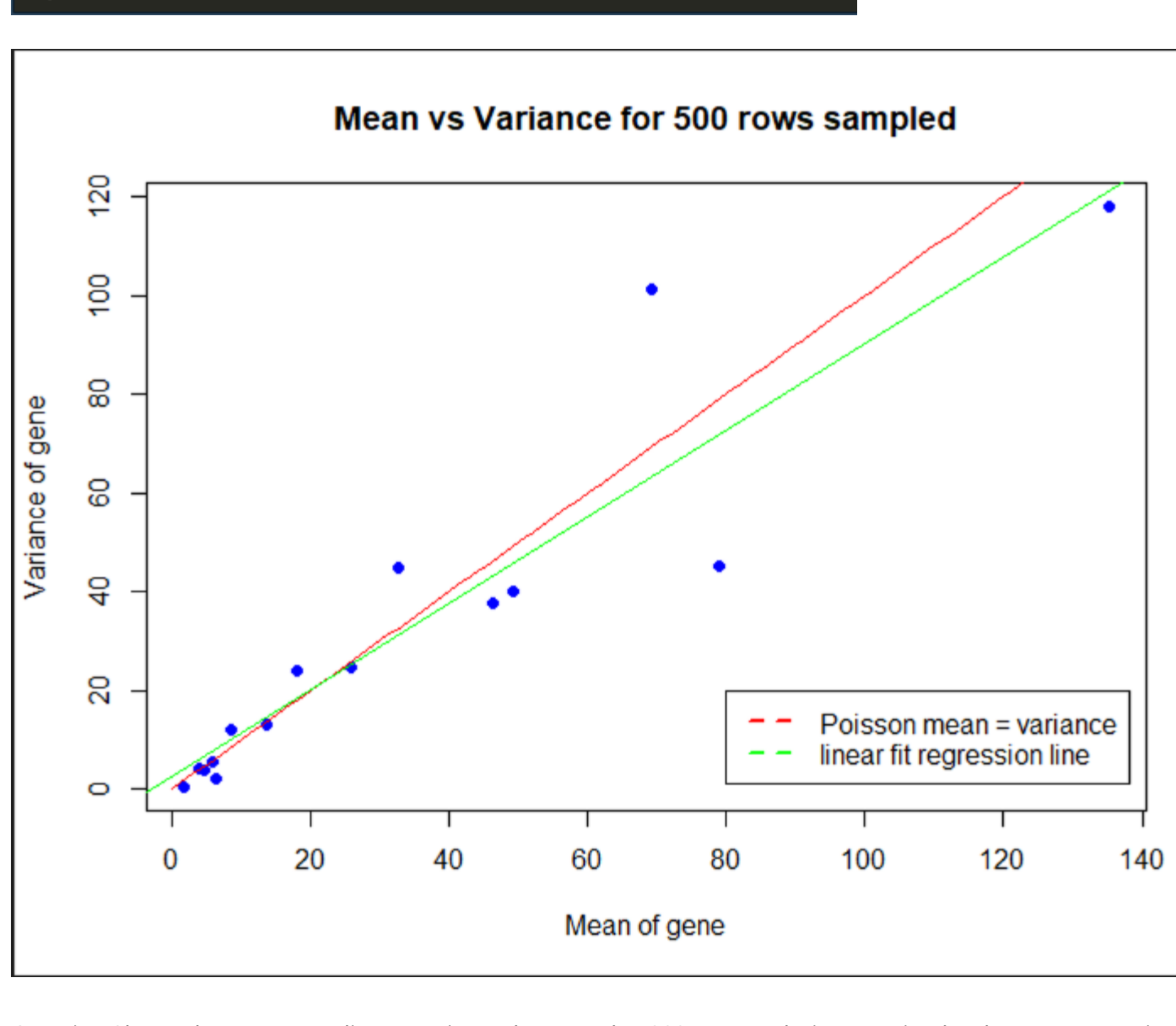
Question 2a. Randomly sample 500 rows. Do this simulation 10 times and record the relative abundance of each of the 15 genes. Make a scatterplot the mean vs. variance of each gene (x-axis=mean of gene_i, y-axis=variance of gene_i)

```
##Q2
library(matrixStats)
getwd()
setwd("D:/Comparative_Genomics/Assignment5/")


data0 = read.table("data1.txt", header = F)
data1 = as.vector(data0$V1)
table(data1)

##Q2a
a = c()
b = c()
c = c()
d = c()
e = c()
f = c()
g = c()
h = c()
i = c()
j = c()

#replaced with 5000 and 50000 for the next questions
a = as.vector(table(append(a , sample(data1, 500))))
b = as.vector(table(append(b , sample(data1, 500))))
c = as.vector(table(append(c , sample(data1, 500))))
d = as.vector(table(append(d , sample(data1, 500))))
e = as.vector(table(append(e , sample(data1, 500))))
f = as.vector(table(append(f , sample(data1, 500))))
g = as.vector(table(append(g , sample(data1, 500))))
h = as.vector(table(append(h , sample(data1, 500))))
i = as.vector(table(append(i , sample(data1, 500))))
j = as.vector(table(append(j , sample(data1, 500))))

total_sum = sum(a + b + c + d + e + f + g + h + i + j)

Q2_1 = data.frame(a,b,c,d,e,f,g,h,i,j)
Q2_1$mean = rowMeans(Q2_1)
Q2_1$variance = round(rowVars(as.matrix(Q2_1[,c(1:10)])),4)
Q2_1$abundance = rowSums(Q2_1[,c(1:10)])/total_sum

plot(x = Q2_1$mean, y = Q2_1$variance, xlab="Mean of gene",
     ylab="Variance of gene", main="Mean vs Variance for 500 rows sampled",
     col = "blue", pch = 16)

a  = c(0:150)
b = c(0:150)
lines(x = a, y = b, col = "red")
abline(lm(Q2_1$variance~Q2_1$mean, data = Q2_1), col = "green")
legend(80,20, c("Poisson mean = variance", "linear fit regression line"), col=c("red","green"), lwd=2, lty=2)
```

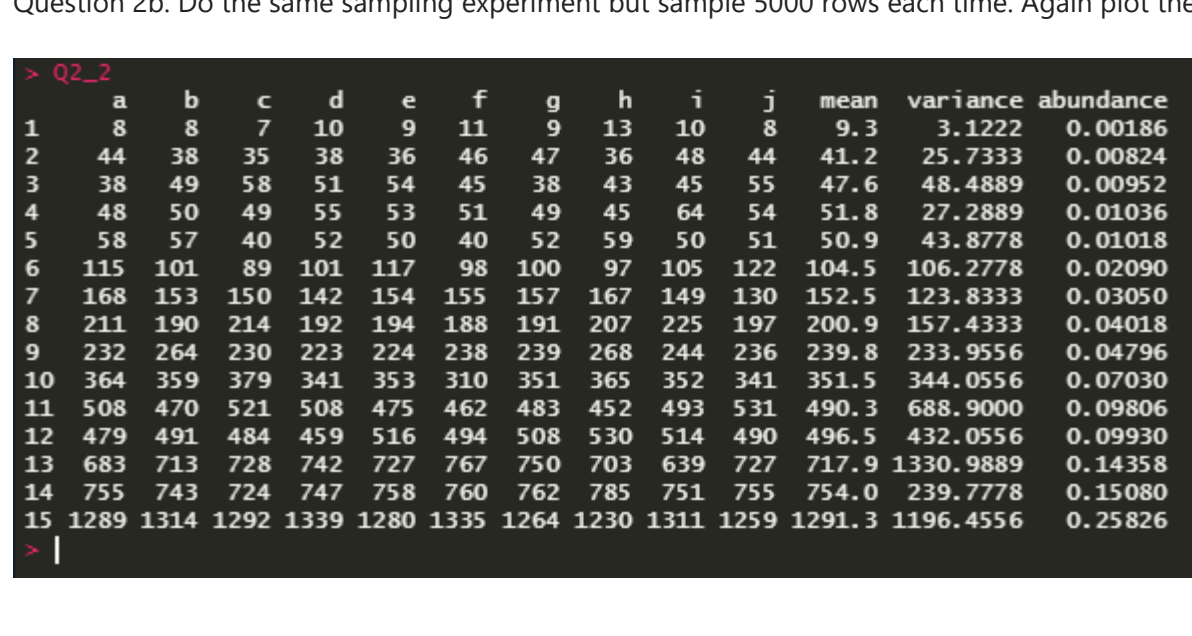


> Q2_1

	a	b	c	d	e	f	g	h	i	j	mean	variance	abundance
1	1	1	1	1	2	1	1	3	1	1	1.3	0.4556	0.0026
2	1	5	4	3	2	3	6	3	3	5	3.5	2.2778	0.0070
3	3	5	1	4	4	11	6	7	7	3	5.1	7.8778	0.0102
4	3	2	6	7	5	4	5	3	2	3	4.0	2.8889	0.0080
5	4	5	3	9	5	6	3	8	5	6	5.4	3.8222	0.0108
6	12	9	8	6	7	9	10	7	12	8	8.8	4.1778	0.0176
7	18	11	19	15	22	14	14	8	8	18	14.7	22.0111	0.0294
8	17	22	26	19	16	22	20	16	23	19	20.0	10.6667	0.0400
9	30	20	24	22	24	31	28	29	22	15	24.5	25.3889	0.0490
10	30	38	29	25	34	25	35	28	34	39	31.7	25.3444	0.0634
11	47	56	54	48	51	52	60	48	59	46	52.1	25.2111	0.1042
12	60	57	44	53	55	49	46	62	50	44	52.0	41.7778	0.1040
13	60	65	73	78	75	75	81	71	77	78	73.3	41.5667	0.1466
14	81	66	71	81	78	74	65	77	61	78	73.2	50.6222	0.1464
15	133	138	137	129	120	124	120	130	136	137	130.4	49.1556	0.2608

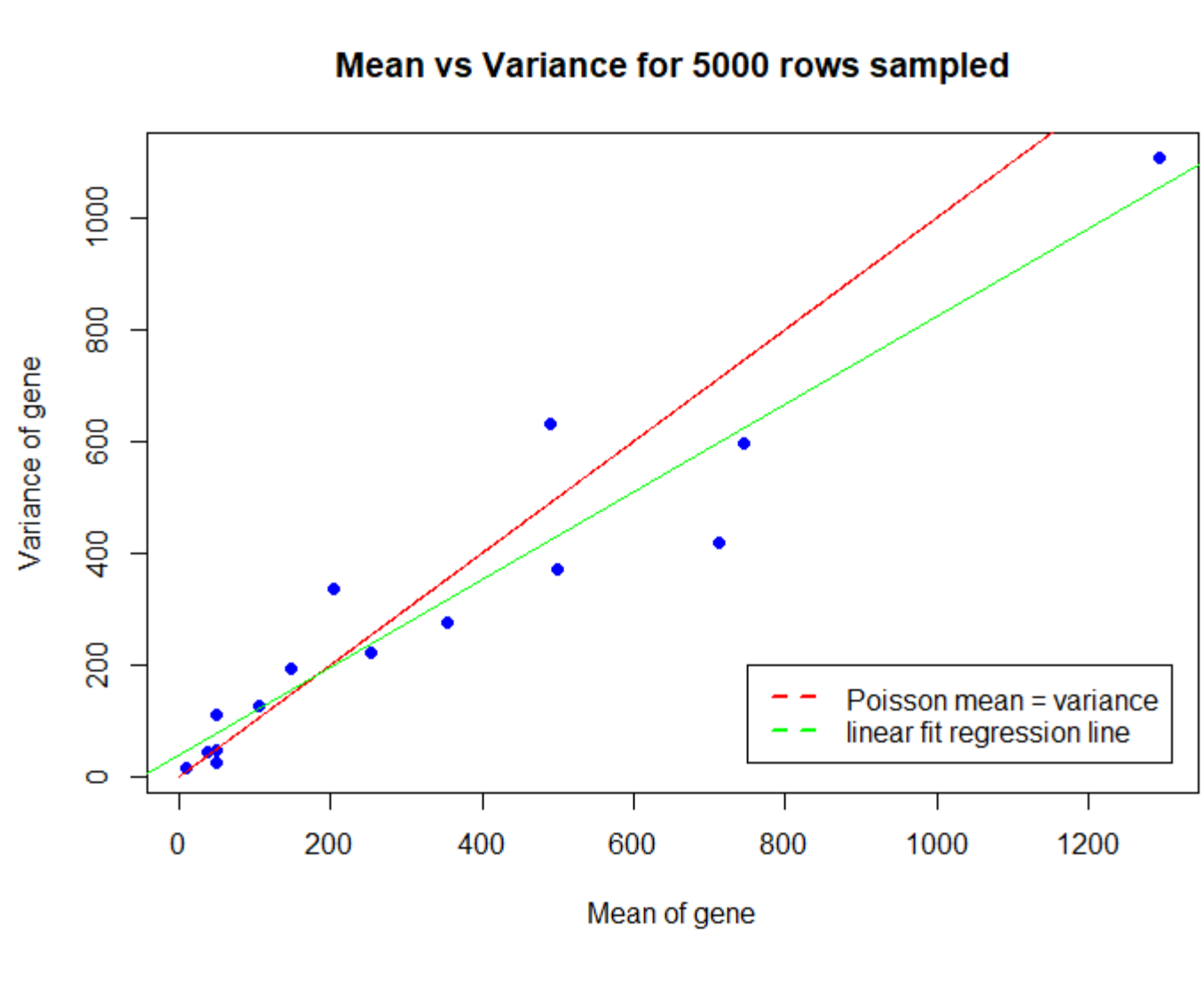


Question 2b. Do the same sampling experiment but sample 5000 rows each time. Again plot the mean vs. variance.

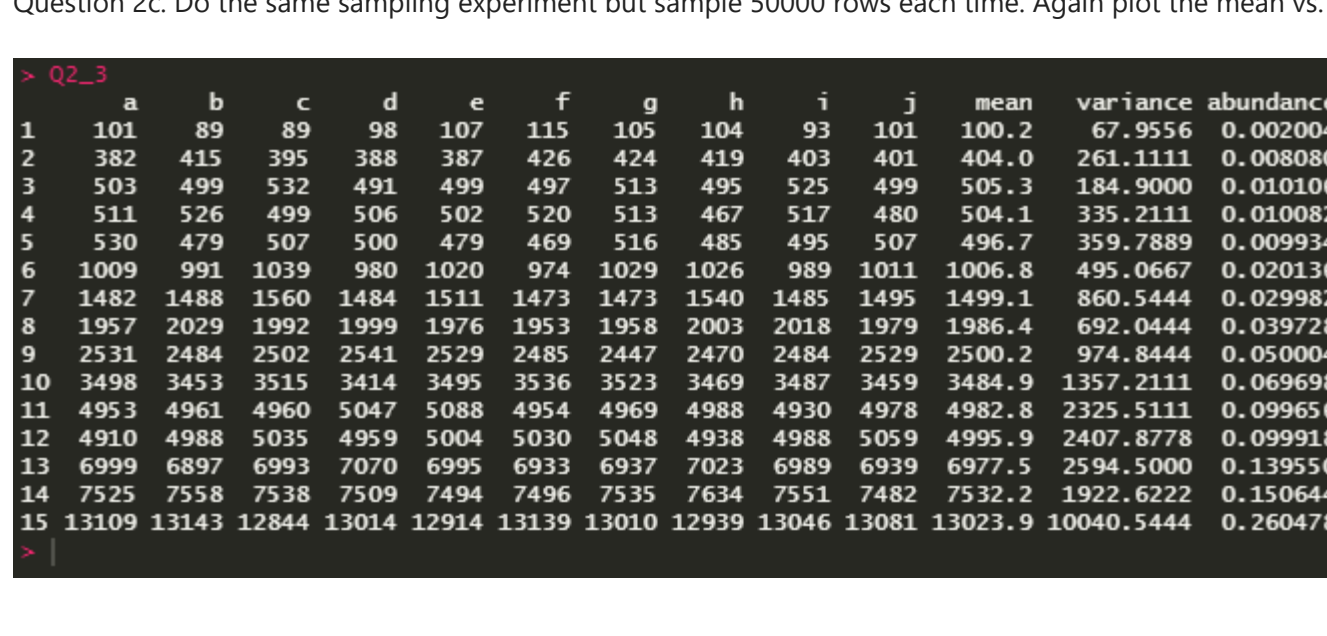


> Q2_2

	a	b	c	d	e	f	g	h	i	j	mean	variance	abundance
1	8	8	7	10	9	11	9	13	10	8	9.3	3.1222	0.00186
2	44	38	35	38	36	46	47	36	48	44	41.2	25.7333	0.00824
3	38	49	58	51	54	45	38	43	45	55	47.6	48.4889	0.00952
4	48	50	49	55	53	51	49	45	64	54	51.8	27.2889	0.01036
5	58	57	40	52	50	40	52	59	50	51	50.9	43.8778	0.01018
6	115	101	89	101	117	98	100	97	105	122	104.5	106.2778	0.02090
7	168	153	150	142	154	155	157	167	149	130	152.5	123.8333	0.03050
8	211	190	214	192	194	188	191	207	225	197	200.9	157.4333	0.04018
9	232	264	230	223	224	238	239	268	244	236	239.8	233.9556	0.04796
10	364	359	379	341	353	310	351	365	352	341	351.5	344.0556	0.07030
11	508	470	521	508	475	462	483	452	493	531	490.3	688.9000	0.09806
12	479	491	484	459	516	494	508	530	514	490	496.5	432.0556	0.09930
13	683	713	728	742	727	767	750	703	639	727	717.9	1330.9889	0.14358
14	755	743	724	747	758	760	762	785	751	755	754.0	239.7778	0.15080
15	1289	1314	1292	1339	1280	1335	1264	1230	1311	1259	1291.3	1196.4556	0.25826

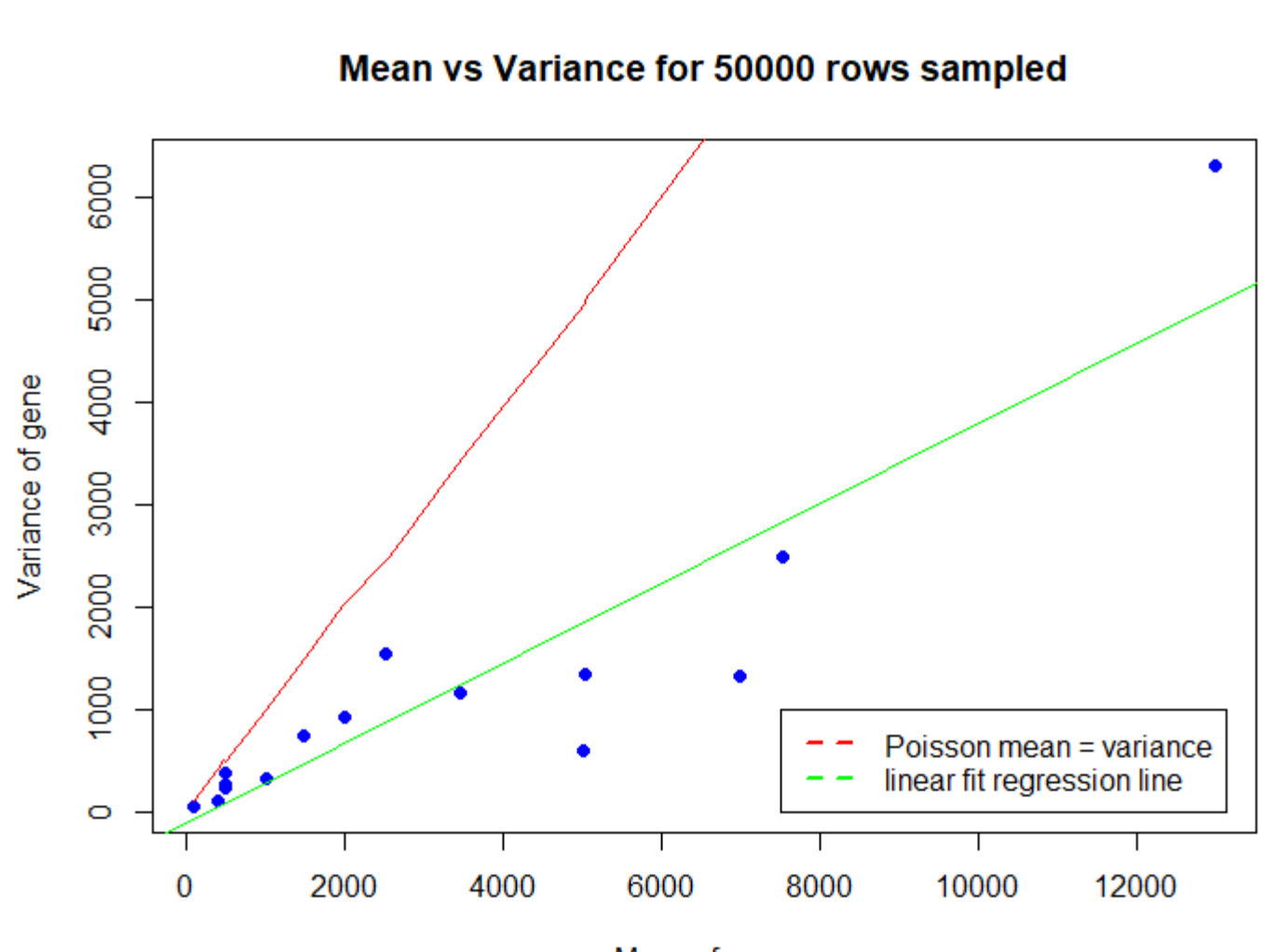


Question 2c. Do the same sampling experiment but sample 50000 rows each time. Again plot the mean vs. variance



> Q2_3

	a	b	c	d	e	f	g	h	i	j	mean	variance	abundance
1	101	89	89	98	107	115	105	104	93	101	100.2	67.9556	0.002004
2	382	415	395	388	387	426	424	419	403	401	404.0	261.1111	0.008080
3	503	499	532	491	499	497	513	495	525	499	505.3	184.9000	0.010106
4	511	526	499	506	502	520	513	467	517	480	504.1	335.2111	0.010082
5	530	479	507	500	479	469	516	485	495	507	496.7	359.7889	0.009934
6	1009	991	1039	980	1020	974	1029	1026	989	1011	1006.8	495.0667	0.020136
7	1482	1488	1560	1484	1511	1473	1473	1540	1485	1495	1499.1	860.5444	0.029982
8	1957	2029	1992	1999	1976	1953	1958	2003	2018	1979	1986.4	692.0444	0.039728
9	2531	2484	2502	2541	2529	2485	2447	2470	2484	2529	2500.2	974.8444	0.050004
10	3498	3453	3515	3414	3495	3536	3523	3469	3487	3459	3484.9	1357.2111	0.069698
11	4953	4961	4960	5047	5088	4954	4969	4988	4930	4978	4982.8	2325.5111	0.099656
12	4910	4988	5035	4959	5004	5030	5048	4938	4988	5059	4995.9	2407.8778	0.099918
13	6999	6897	6993	7070	6995	6933	6937	7023	6989	6939	6977.5	2594.5000	0.139550
14	7525	7558	7538	7509	7494	7496	7535	7634	7551	7482	7532.2	1922.6222	0.150644
15	13109	13143	12844	13014	12914	13139	13010	12939	13046	13081	13023.9	10040.5444	0.260478



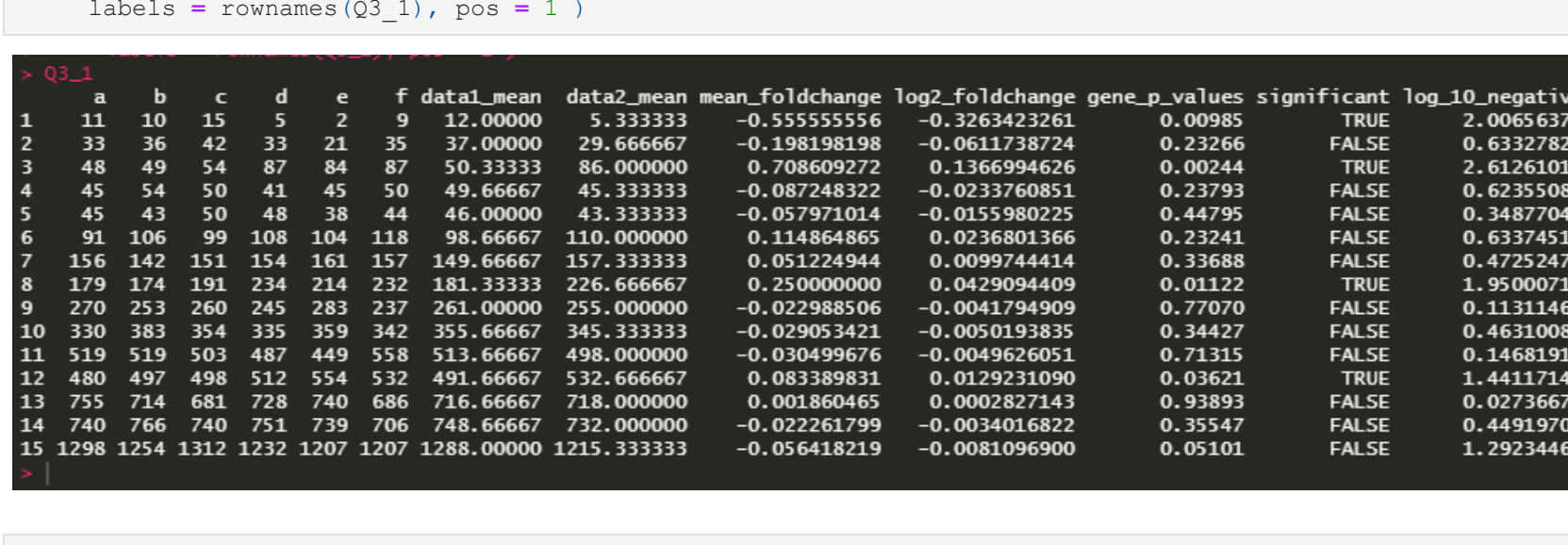
Question 2d. Is the variance greater in (a), (b) or (c)? What is the relationship between mean abundance and variance? Why?

```
Variance is greater in (c).
With a small sample size the mean vs ratio points are closet to poisson distribution,
But with an increase in sample size the points do not fit into the poisson distribution and largely vary.
In terms of scatter plot the correlation is positive but the data does't fit into the the distribution.
Here Relative Abundance of a gene increases with an increase variance.

OVer fitting an
```

Q3.Question 3. Differential Expression [20 pts]

Question 3a. Using the file from question 2 (data1.txt) along with data2.txt, randomly sample 5000 rows from each file. Sample 3 times for each file (this emulates making experimental replicates) and conduct a paired t-test for differential expression of each of the 15 genes. Which genes are significantly differentially expressed at the 0.05 level and what is their mean fold change? Ans: A Difference in the mean values of the two files show the following results.

```
library(reshape)
data0 = read.table("data1.txt", header = F)
data1 = as.vector(data0$V1)
data00 = read.table("data2.txt", header = F)
data2 = as.vector(data00$V1)


length(data1)
length(data2)

a = c()    # make a list of empty vectors
b = c()
c = c()
d = c()
e = c()
f = c()
# sample the two files each three times, calculate the count of each gene adn append to vector
a = as.vector(table(append(a , sample(data1, 5000))))
b = as.vector(table(append(b , sample(data1, 5000))))
c = as.vector(table(append(c , sample(data1, 5000))))
d = as.vector(table(append(d , sample(data2, 5000))))
e = as.vector(table(append(e , sample(data2, 5000))))
f = as.vector(table(append(f , sample(data2, 5000))))
total_sum = sum(a + b + c + d + e + f)    #check the data by calculating the total_sum i.e. denominator
total_sum

Q3_1 = data.frame(a,b,c,d,e,f)    #make dataframe of all the gene counts in each sampling
head(Q3_1)


Q3_1$data1_mean = rowMeans(Q3_1[,c(1:3)])    # calculate rowwise Mean
Q3_1$data2_mean = rowMeans(Q3_1[,c(4:6)])
Q3_1
#Q3_1$mean_diff = rowDiffs(as.matrix(Q3_1[,c(10,9)]))
Q3_1$log2_foldchange = (log2(Q3_1$data2_mean) - log2(Q3_1$data1_mean)) /
  log2(Q3_1$data1_mean)     # log2 mean foldchange (B - A / A)

t.test.all.genes <- function(x,s1,s2) {    #T.test function
  x1 <- x[s1]
  x2 <- x[s2]
  x1 <- as.numeric(x1)
  x2 <- as.numeric(x2)
  t.out <- t.test(x1,x2, alternative="two.sided", paired = TRUE,
                  conf.level = 0.95 )
  out <- as.numeric(t.out$p.value)
  return(out)
}

gene_p_values = round(apply(Q3_1[,1:6], 1, t.test.all.genes,
                            s1 = names(Q3_1)[1:3] ,
                            s2 = names(Q3_1)[4:6]),5)
Q3_1 = cbind(Q3_1, gene_p_values)
Q3_1
Q3_1$significant = Q3_1$gene_p_values < 0.05    #check if significant
Q3_1$log_10_negative = -log10(Q3_1$gene_p_values)    #calculate -log10
Q3_1

plot(y = Q3_1$log_10_negative, x = Q3_1$log2_foldchange, col = "red", pch = 16)
abline(0,0, v = -0.1)
abline(0,0, v = 0.1)
abline(0,0, h = -log10(0.05))
text(x = Q3_1$log2_foldchange, y = Q3_1$log_10_negative,
     labels = rownames(Q3_1), pos = 1 )
```

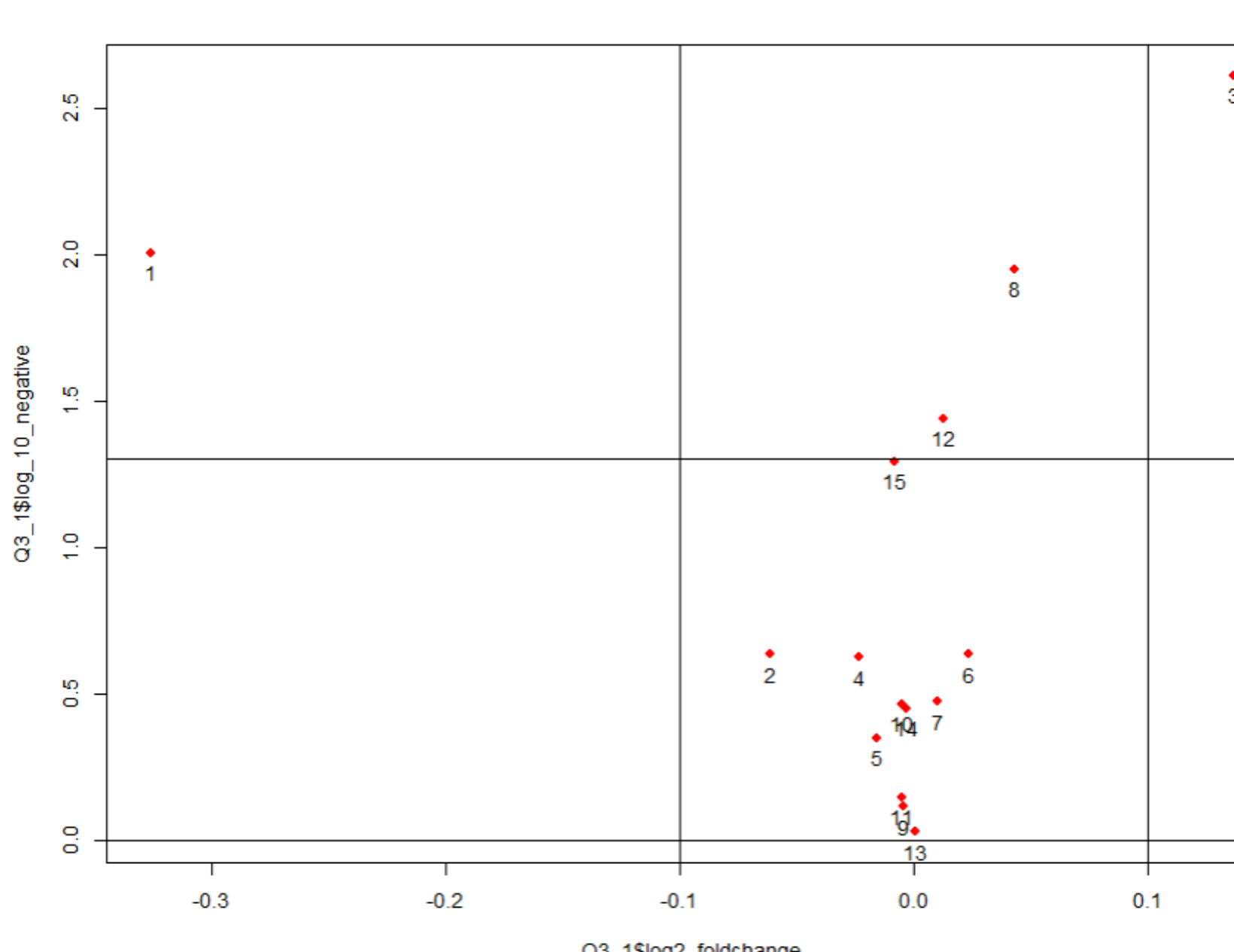


> Q3_1

	a	b	c	d	e	f	data1_mean	data2_mean	mean_foldchange	log2_foldchange	gene_p_values	significant	log_10_negative
1	11	10	15	5	2	9	12.00000	5.333333	-0.555555556	-0.3263423261	0.00985	TRUE	2.00656377
2	33	36	42	33	21	35	37.00000	29.666667	-0.198198198	-0.0611738724	0.23266	FALSE	0.63327828
3	48	49	54	87	84	87	50.33333	86.000000	0.708609272	0.1366994626	0.00244	TRUE	2.61261017
4	45	54	50	41	45	50	49.66667	45.333333	-0.087248322	-0.0233760851	0.23793	FALSE	0.62355080
5	45	43	50	48	38	44	46.00000	43.333333	-0.057971014	-0.0155980225	0.44795	FALSE	0.34877046
6	91	106	99	108	104	118	98.66667	110.000000	0.114864865	0.0236801366	0.23241	FALSE	0.63374519
7	156	142	151	154	161	157	149.66667	157.333333	0.051224944	0.0099744414	0.33688	FALSE	0.47252477
8	179	174	191	234	214	232	181.33333	226.666667	0.250000000	0.0429094409	0.01122	TRUE	1.95000714
9	270	253	260	245	283	237	261.00000	255.000000	-0.022988506	-0.0041794909	0.77070	FALSE	0.11311464
10	330	383	354	335	359	342	355.66667	345.333333	-0.029053421	-0.0050193835	0.34427	FALSE	0.46310082
11	519	519	503	487	449	558	513.66667	498.000000	-0.030499676	-0.0049626051	0.71315	FALSE	0.14681911
12	480	497	498	512	554	532	491.66667	532.666667	0.083389831	0.0129231090	0.03621	TRUE	1.44117148
13	755	714	681	728	740	686	716.66667	718.000000	0.001860465	0.0002827143	0.93893	FALSE	0.02736678
14	740	766	740	751	739	706	748.66667	732.000000	-0.022261799	-0.0034016822	0.35547	FALSE	0.44919705
15	1298	1254	1312	1232	1207	1207	1288.00000	1215.333333	-0.056418219	-0.0081096900	0.05101	FALSE	1.29234468

```
Ans: Q3a. Gene1, Gene3, Gene8, Gene12 have a P-value < 0.05. Hence They are significantly expressed.
    Their Mean fold change are -0.555, 0.7086, -0.25 and 0.083 respectively.
```

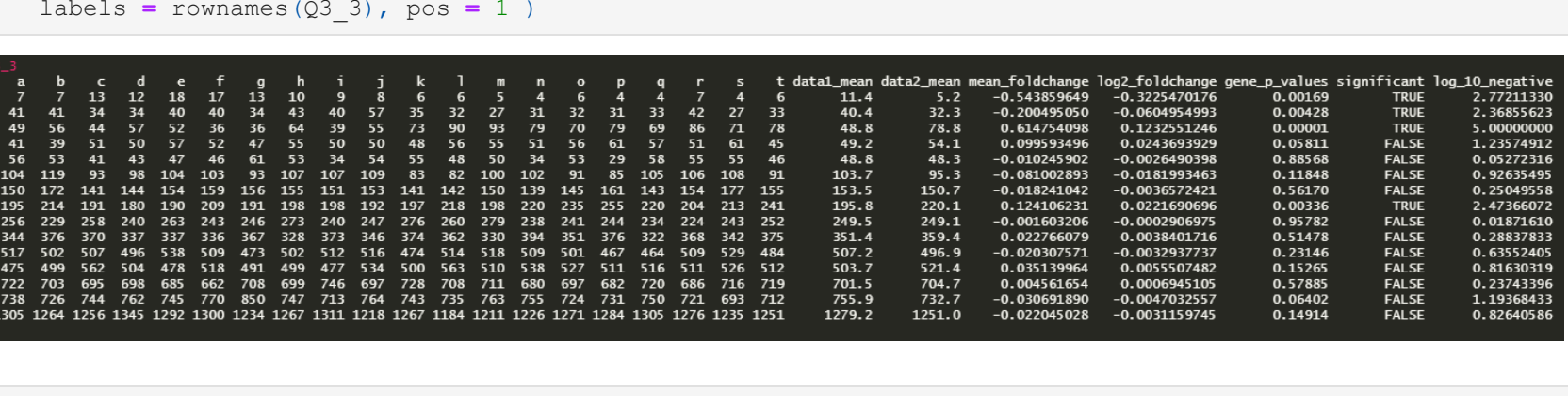
Question 3b. Make a volano plot of the data from part a: x-axis=log2(fold change of the mean expression of gene_i); y-axis=-log_10(p_value comparing the expression of gene_i). Label all of the genes that show a statistically siginificant change




```
Ans: 3b. In the volcano plot the gene1, gene3 and gene8 show a statistically significant change.
```

Question 3c. Now sample 5000 rows 10 times from each file, equivalent to making more replicates. Which genes are now significant at the 0.05 level and what is their mean fold change? Ans:

```
##Q3_c
a = c()
b = c()
c = c()
d = c()
e = c()
f = c()
g = c()
h = c()
i = c()
j = c()
k = c()
l = c()
m = c()
n = c()
o = c()
p = c()
q = c()
r = c()
s = c()
t = c()

a = as.vector(table(append(a , sample(data1, 5000))))
b = as.vector(table(append(b , sample(data1, 5000))))
c = as.vector(table(append(c , sample(data1, 5000))))
d = as.vector(table(append(d , sample(data1, 5000))))
e = as.vector(table(append(e , sample(data1, 5000))))
f = as.vector(table(append(f , sample(data1, 5000))))
g = as.vector(table(append(g , sample(data1, 5000))))
h = as.vector(table(append(h , sample(data1, 5000))))
i = as.vector(table(append(i , sample(data1, 5000))))
j = as.vector(table(append(j , sample(data1, 5000))))
k = as.vector(table(append(k , sample(data2, 5000))))
l = as.vector(table(append(l , sample(data2, 5000))))
m = as.vector(table(append(m , sample(data2, 5000))))
n = as.vector(table(append(n , sample(data2, 5000))))
o = as.vector(table(append(o , sample(data2, 5000))))
p = as.vector(table(append(p , sample(data2, 5000))))
q = as.vector(table(append(q , sample(data2, 5000))))
r = as.vector(table(append(r , sample(data2, 5000))))
s = as.vector(table(append(s , sample(data2, 5000))))
t = as.vector(table(append(t , sample(data2, 5000))))


total_sum = sum(a + b + c + d + e + f + g + h +
                  i + j + k + l + m + n + o + p + q + r + s + t)
total_sum

Q3_3 = data.frame(a,b,c,d,e,f, g, h, i, j, k, l, m, n, o, p, q, r, s, t)
Q3_3


Q3_3$data1_mean = rowMeans(Q3_3[,c(1:10)])
Q3_3$data2_mean = rowMeans(Q3_3[,c(11:20)])
#Q3_3$mean_diff = abs(rowDiffs(as.matrix(Q3_3[,c(22,21)])))
#Q3_3$foldchange = Q3_3$mean_diff / Q3_1$data1_mean
Q3_3$log2_foldchange = (log2(Q3_3$data2_mean) - log2(Q3_3$data1_mean)) /
  log2(Q3_3$data1_mean)     # foldchange (B - A / A)

t.test.all.genes <- function(x,s1,s2) {    #T.test function
  x1 <- x[s1]
  x2 <- x[s2]
  x1 <- as.numeric(x1)
  x2 <- as.numeric(x2)
  t.out <- t.test(x1,x2, alternative="two.sided", paired = TRUE, conf.level = 0.05 )
  out <- as.numeric(t.out$p.value)
  return(out)
}

gene_p_values = round(apply(Q3_3[,1:20], 1, t.test.all.genes,
                            s1 = names(Q3_3)[1:10] ,
                            s2 = names(Q3_3)[11:20]),5)
Q3_3 = cbind(Q3_3, gene_p_values)
Q3_3
Q3_3$significant = Q3_3$gene_p_values < 0.05

#Q3_3$log2_foldchange = log2(Q3_3$foldchange)
Q3_3$log_10_negative = -log10(Q3_3$gene_p_values)
Q3_3

plot(x = Q3_3$log2_foldchange, y = Q3_3$log_10_negative, col = "red", pch = 16)
abline(0,0, v = -0.1)
abline(0,0, v = 0.1)
abline(0,0, h = -log10(0.05))
text(x = Q3_3$log2_foldchange, y = Q3_3$log_10_negative,
     labels = rownames(Q3_3), pos = 1 )
```



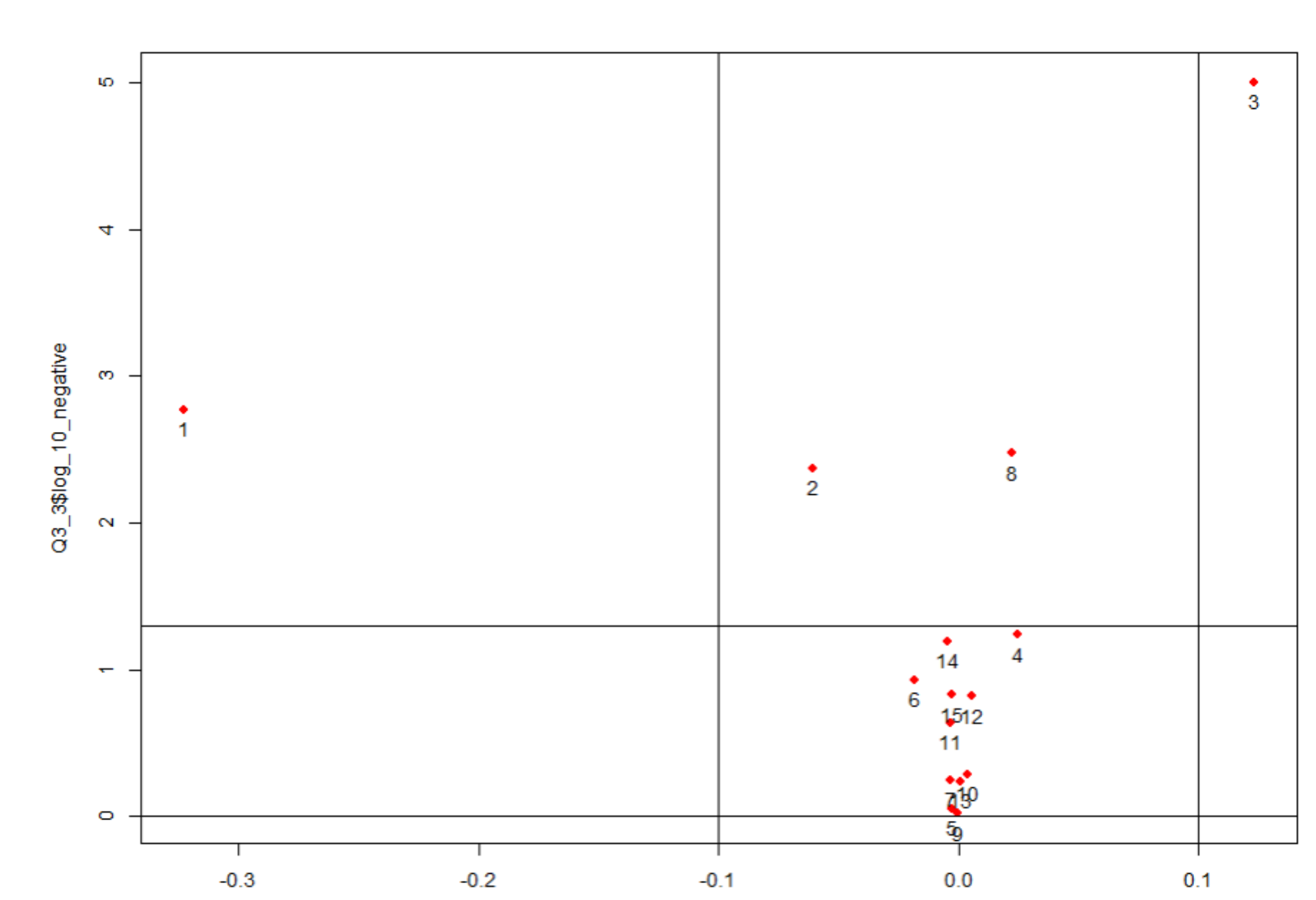
> Q3_3

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	data1_mean	data2_mean	mean_foldchange	log2_foldchange	gene_p_values	significant	log_10_negative
1	7	7	13	12	18	17	13	10	9	8	6	6	5	4	6	4	4	7	4	6	11.4	5.2	-0.543859649	-0.3225470176	0.00169	TRUE	2.77211330
2	41	41	34	34	40	40	34	43	40	57	35	32	27	31	32	31	33	42	27	33	40.4	32.3	-0.200495050	-0.0604954993	0.00428	TRUE	2.36855623
3	49	56	44	57	52	36	36	64	39	55	73	90	93	79	70	79	69	86	71	78	48.8	78.8	0.614754098	0.1232551246	0.00001	TRUE	5.00000000
4	41	39	51	50	57	52	47	55	50	50	48	56	55	51	56	61	57	51	61	45	49.2	54.1	0.099593496	0.0243693929	0.05811	FALSE	1.23574912
5	56	53	41	43	47	46	61	53	34	54	55	48	50	34	53	29	58	55	55	46	48.8	48.3	-0.010245902	-0.0026490398	0.88568	FALSE	0.05272316
6	104	119	93	98	104	103	93	107	107	109	83	82	100	102	91	85	105	106	108	91	103.7	95.3	-0.081002893	-0.0181993463	0.11848	FALSE	0.92635495
7	150	172	141	144	154	159	156	155	151	153	141	142	150	139	145	161	143	154	177	155	153.5	150.7	-0.018241042	-0.0036572421	0.56170	FALSE	0.25049558
8	195	214	191	180	190	209	191	198	198	192	197	218	198	220	235	255	220	204	213	241	195.8	220.1	0.124106231	0.0221690696	0.00336	TRUE	2.47366072
9	256	229	258	240	263	243	246	273	240	247	276	260	279	238	241	244	234	224	243	252	249.5	249.1	-0.001603206	-0.0002906975	0.95782	FALSE	0.01871610
10	344	376	370	337	337	336	367	328	373	346	374	362	330	394	351	376	322	368	342	375	351.4	359.4	0.022766079	0.0038401716	0.51478	FALSE	0.28837833
11	517	502	507	496	538	509	473	502	512	516	474	514	518	509	501	467	464	509	529	484	507.2	496.9	-0.020307571	-0.0032937737	0.23146	FALSE	0.63552405
12	475	499	562	504	478	518	491	499	477	534	500	563	510	538	527	511	516	511	526	512	503.7	521.4	0.035139964	0.0055507482	0.15265	FALSE	0.81630319
13	722	703	695	698	685	662	708	699	746	697	728	708	711	680	697	682	720	686	716	719	701.5	704.7	0.004561654	0.0006945105	0.57885	FALSE	0.23743396
14	738	726	744	762	745	770	850	747	713	764	743	735	763	755	724	731	750	721	693	712	755.9	732.7	-0.030691890	-0.0047032557	0.06402	FALSE	1.19368433
15	1305	1264	1256	1345	1292	1300	1234	1267	1311	1218	1267	1184	1211	1226	1271	1284	1305	1276	1235	1251	1279.2	1251.0	-0.022045028	-0.0031159745	0.14914	FALSE	0.82640586

```
Question 3c: Ans:
Here the significantly expressed genes are Gene1, Gene2, Gene3 and
Gene8 and their mean fold change are -0.543, -0.200, 0.614, 0.124 respectively
```

Question 3d. Make a volcano plot using the results from part c (label any statistically significant genes) Ans:

Answer 3d




```
Q3d In the volcano plot the statistically significant genes are gene1, gene2, gene8 and gene3
```

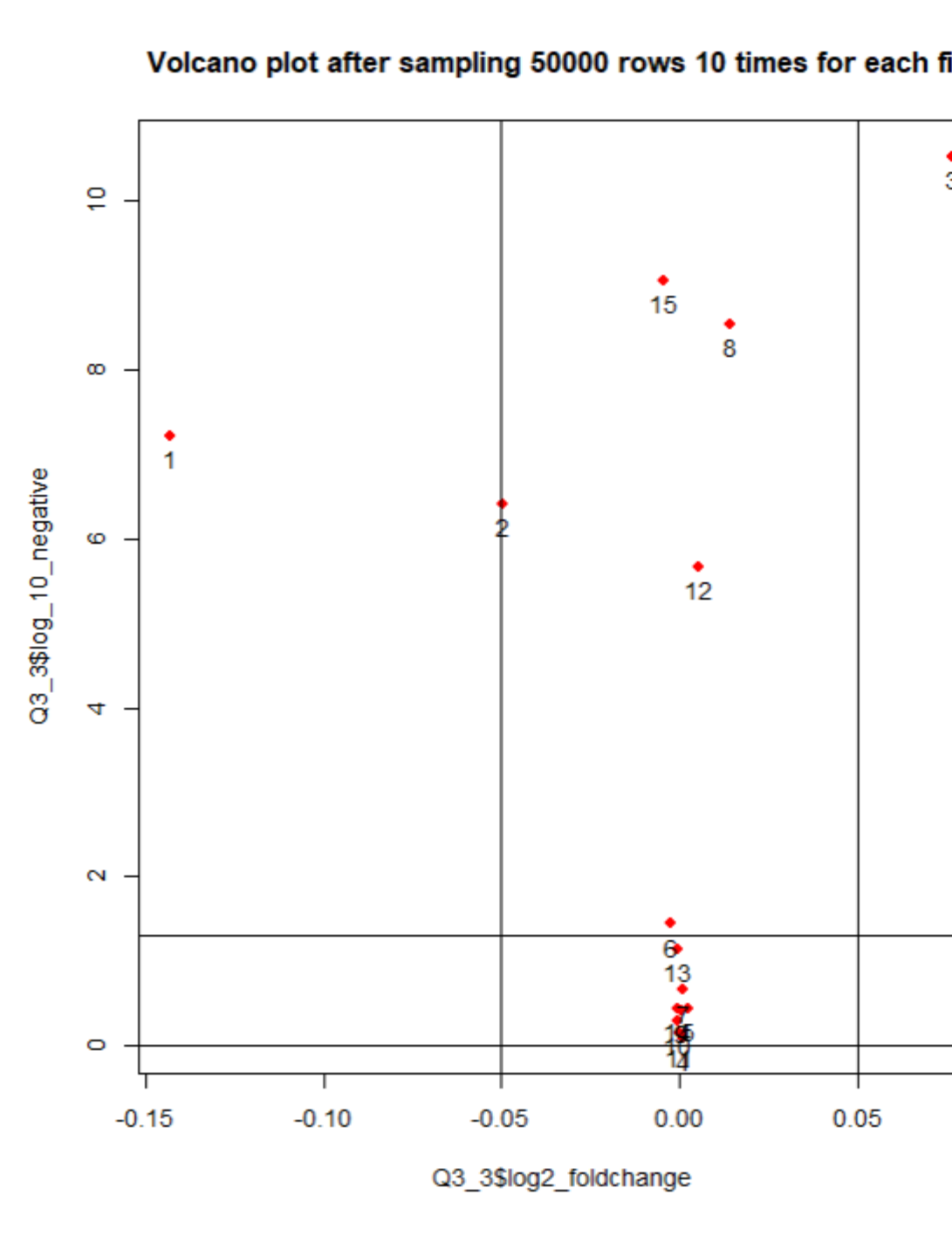
Question 3e. Perform the simulations from parts a/c but sample 50000 rows each time from each file. Which genes are significant and what is their mean fold change?

> Q3_3

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	data1_mean	data2_mean	mean_foldchange	log2_foldchange	gene_p_values	significant	log_10_negative
1	94	98	105	114	112	102	92	108	98	107	54	57	54	46	52	60	52	50	52	54	103.0	53.1	-0.484466019	-0.1429537869	5.920406e-08	TRUE	7.22764854
2	399	390	391	371	390	405	439	407	429	383	321	275	314	277	302	304	289	302	288	297	400.4	296.9	-0.258491508	-0.0499074610	3.864755e-07	TRUE	6.41287799
3	530	502	509	469	500	496	492	503	484	486	773	808	815	777	817	816	802	782	804	815	497.1	800.9	0.611144639	0.0768176712	2.910303e-11	TRUE	10.53606172
4	520	506	498	524	490	474	457	501	522	497	487	492	503	465	513	505	536	492	516	509	498.9	501.8	0.005812788	0.0009329653	8.125981e-01	FALSE	0.09012417
5	464	481	490	478	492	501	493	493	508	488	500	504	495	501	464	488	513	523	483	487	488.8	495.8	0.014320786	0.0022964016	3.613125e-01	FALSE	0.44211699
6	1009	975	972	992	1015	979	1015	1022	1043	1008	964	980	963	1014	1014	967	994	990	993	969	1003.0	984.8	-0.018145563	-0.0026498153	3.626778e-02	TRUE	1.44047900
7	1505	1505	1517	1492	1534	1485	1489	1489	1464	1481	1526	1523	1492	1507	1487	1481	1512	1526	1512	1518	1496.1	1508.4	0.008221376	0.0011199826	2.216722e-01	FALSE	0.65428869
8	1965	2030	1997	1990	2015	2015	2031	2026	2022	2026	2281	2235	2224	2231	2234	2212	2260	2268	2261	2251	2011.7	2245.7	0.116319531	0.0144657511	2.813733e-09	TRUE	8.55071718
9	2574	2503	2481	2428	2489	2498	2482	2457	2504	2463	2533	2513	2553	2430	2446	2509	2514	2508	2488	2495	2487.9	2498.9	0.004421400	0.0005642082	3.774184e-01	FALSE	0.42317694
10	3566	3510	3410	3560	3518	3525	3467	3534	3494	3502	3433	3518	3537	3445	3501	3543	3421	3517	3494	3522	3508.6	3493.1	-0.004417716	-0.0005423886	5.194849e-01	FALSE	0.28442708
11	4881	4951	4953	5014	4975	4942	5028	5031	5134	5062	4974	4994	4910	5015	5015	5051	4944	4931	5125	4919	4997.1	4987.8	-0.001861079	-0.0002187270	7.323825e-01	FALSE	0.13526203
12	5006	5058	5032	4973	4952	4930	5000	4941	4937	5072	5184	5251	5214	5259	5150	5181	5269	5290	5158	5185	4990.1	5214.1	0.044888880	0.0051567182	2.170996e-06	TRUE	5.66334105
13	7002	7010	6985	7002	6976	7022	7074	6977	6942	6876	6999	6972	6885	7009	6959	6940	6922	6871	6923	6953	6986.6	6943.3	-0.006197578	-0.0007023316	7.132492e-02	FALSE	1.14675870
14	7483	7502	7526	7558	7603	7574	7462	7492	7413	7562	7451	7432	7469	7477	7543	7480	7574	7579	7444	7509	7517.5	7495.8	-0.002886598	-0.0003238964	3.687928e-01	FALSE	0.43321751
15	13002	12979	13134	13035	12939	13052	12979	13019	13006	12987	12520	12446	12572	12547	12503	12463	12398	12371	12459	12517	13013.2	12479.6	-0.041004518	-0.0044194802	8.672099e-10	TRUE	9.06187576

```
Answer 3e: The significantly expressed genes are gene1, gene2, gene3, gene8, gene12, gene15.
    The mean fold change of these gene are -0.4844, -0.2584, 0.6111, -0.018, 0.044 and -0.041 respectively
```

Question 3f. Make a volcano plot from 5e (label any statistically significant genes) Ans:




```
Answer 3f, In the volcano plot the statistically significant genes are gene1 and gene3
```

Question 3g. Now examine the complete files: compare the fold change in the complete files vs the different subsamples making sure to address replicates and the size of the random sample.

Ans:

> Q3_g

	a	b	foldchange	log2_foldchange
1	200	100	-0.50000000	-0.130824021
2	800	600	-0.25000000	-0.043036467
3	1000	1600	0.60000000	0.068039994
4	1000	1000	0.00000000	0.000000000
5	1000	1000	0.00000000	0.000000000
6	2000	2000	0.00000000	0.000000000
7	3000	3000	0.00000000	0.000000000
8	4000	4500	0.12500000	0.014200908
9	5000	5000	0.00000000	0.000000000
10	7000	7000	0.00000000	0.000000000
11	10000	10000	0.00000000	0.000000000
12	10000	10500	0.05000000	0.005297325
13	14000	14000	0.00000000	0.000000000
14	15000	15000	0.00000000	0.000000000
15	26000	25000	-0.03846154	-0.003858084

```
Comparing Q3_a and Q3_g (5000 samples 3 times for each file v/s the complete file)
Here the fold change varies a lot between the complete file and the sampling data.
Almost All fold change values in the sampling data are greater than the complete data.
This small sampling of 5000 genes also gives one false positive significant genes

Comparing Q3_c and Q3_g (5000 samples 10 times for each file v/s the complete file)
Here the non-zero fold change genes in the complete file were similar to the fold change values
in the sampling data. (gene1, gene2, gene3, gene8, gene12 and15 had similar fold changes.
The zero fold change genes in the complete file though gave less but significant fold change in comparison
This size of sampling does not give any false positive,
but does not detect all the genes that are signinficant.

Comparing Q3_e and Q3_g (50000 samples 10 times for each file v/s the complete file)
Here the sampling fold change values were very much similar to the complete files.
genes with zero fold change in the complete files gave very little fold change
                    in the sampling data ( almost in power 10^ -3)
This size of sampling accurately gives all the genes with a significant fold change.


Thus with an increase in sample size the accuracy of fold change increases.
small size samples do not give correct fold changes.
```