PH240C Assignment 1

Xiangyu Hu

September 24, 2014

1. Packages

2. Loading data

```
  data(mainz)
# gene expression: 22,283 by 200 subjects
expression = exprs(mainz)
```

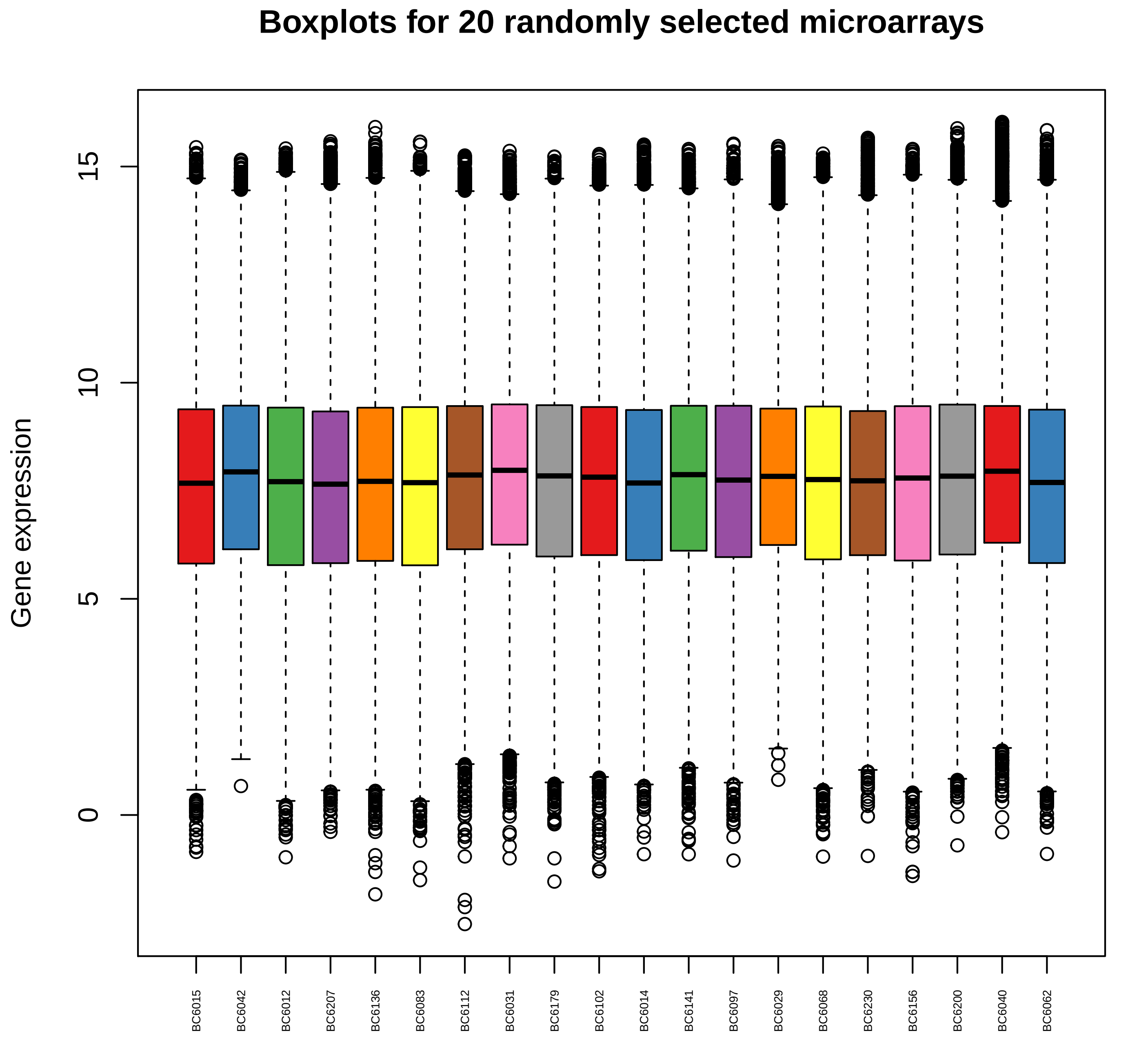
3. Problems

Problem 2(a).

```
# random select 20 of 200 microarrays
subjects = 1:200
subjects_sample1 = sample(subjects, 20)

exprs_sample1 = expression[,subjects_sample1]
```

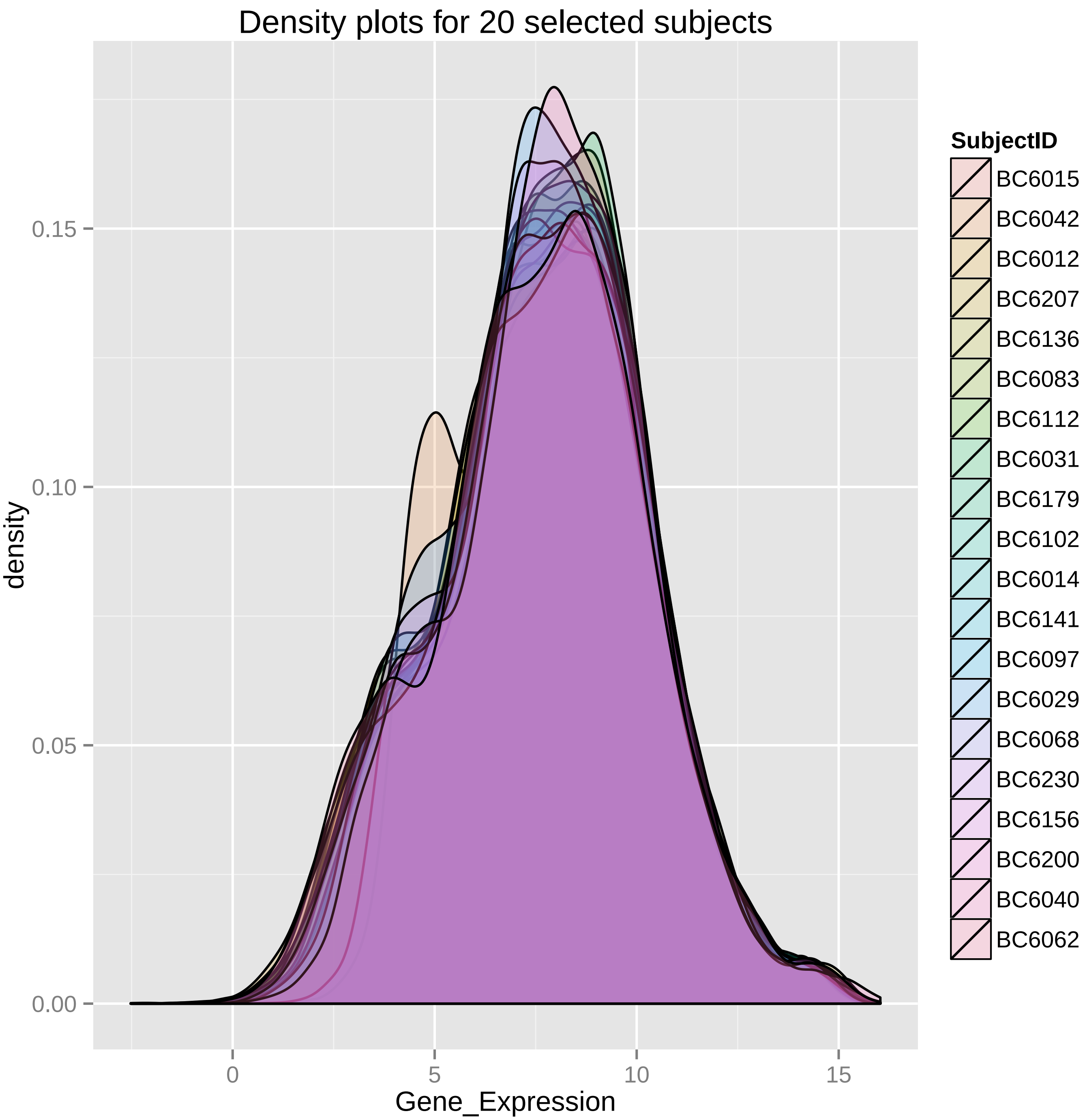
2.a.1. Boxplot: to check the symmetry of the distribution, and mainly compares the location and dispersion differences bewteen the 20 microarrays.The boxplot shows that the dispersions and location are similar for all selected 20 microarrays. Their medians are all around 7.

```
colors = brewer.pal(9, "Set1")
names = sub("MAINZ_","",colnames(exprs_sample1))
colnames(exprs_sample1) = names
boxplot(exprs_sample1, col = colors, xaxt='n',
        main="Boxplots for 20 randomly selected microarrays",
        ylab="Gene expression")
axis(1,at = 1:20, labels = names, cex.axis=0.4, las = 2)
```



2.a.2. Density plot: the density plot enables us to have an overview of the entire distribution of the data. From these density plots, we can see that the distribution for all 20 subjects are really similar

```
exprs_sample1_long = melt(exprs_sample1)
colnames(exprs_sample1_long) = c("Gene", "SubjectID", "Gene_Expression")
layer = geom_density(alpha = 0.2)
dp1 = ggplot(exprs_sample1_long, mapping = aes(fill = SubjectID, x = Gene_Expression)) + layer

dp1 + ggtitle("Density plots for 20 selected subjects")
```



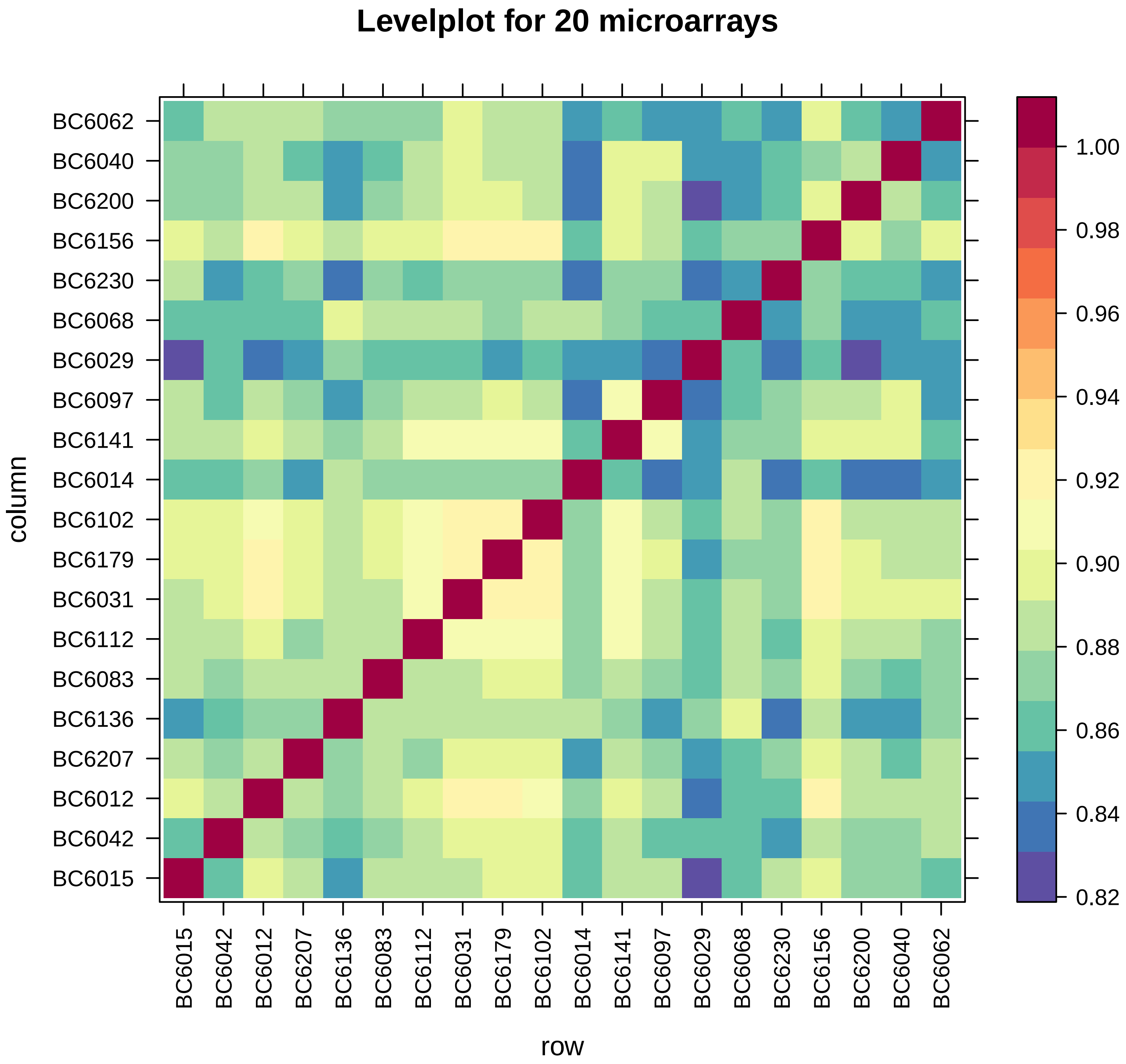
2.a.3 Numerical summaries

```
summary(exprs_sample1)

##      BC6015           BC6042          BC6012           BC6207
##  Min.   :-0.854   Min.   : 0.67   Min.   :-0.978   Min.   :-0.389
##  1st Qu.: 5.816   1st Qu.: 6.15   1st Qu.: 5.779   1st Qu.: 5.825
##  Median : 7.676   Median : 7.94   Median : 7.711   Median : 7.653
##  Mean   : 7.564   Mean   : 7.89   Mean   : 7.551   Mean   : 7.561
##  3rd Qu.: 9.384   3rd Qu.: 9.47   3rd Qu.: 9.424   3rd Qu.: 9.335
##  Max.   :15.447   Max.   :15.16   Max.   :15.418   Max.   :15.583
##      BC6136          BC6083          BC6112          BC6031
##  Min.   :-1.83   Min.   :-1.51   Min.   :-2.52   Min.   :-1.00
##  1st Qu.: 5.88   1st Qu.: 5.78   1st Qu.: 6.15   1st Qu.: 6.25
##  Median : 7.72   Median : 7.69   Median : 7.87   Median : 7.97
##  Mean   : 7.60   Mean   : 7.56   Mean   : 7.75   Mean   : 7.81  
##  3rd Qu.: 9.42   3rd Qu.: 9.44   3rd Qu.: 9.46   3rd Qu.: 9.50  
##  Max.   :15.91   Max.   :15.57   Max.   :15.26   Max.   :15.36  
##      BC6179          BC6102          BC6014           BC6141     
##  Min.   :-1.54   Min.   :-1.30   Min.   :-0.906   Min.   :-0.909  
##  1st Qu.: 5.98   1st Qu.: 6.01   1st Qu.: 5.896   1st Qu.: 6.113  
##  Median : 7.84   Median : 7.82   Median : 7.679   Median : 7.873  
##  Mean   : 7.66   Mean   : 7.67   Mean   : 7.607   Mean   : 7.740  
##  3rd Qu.: 9.48   3rd Qu.: 9.44   3rd Qu.: 9.369   3rd Qu.: 9.466  
##  Max.   :15.23   Max.   :15.29   Max.   :15.508   Max.   :15.405  
##      BC6097          BC6029           BC6068           BC6230     
##  Min.   :-1.05   Min.   : 0.814   Min.   :-0.963   Min.   :-0.947  
##  1st Qu.: 5.97   1st Qu.: 6.245   1st Qu.: 5.912   1st Qu.: 6.010  
##  Median : 7.75   Median : 7.833   Median : 7.760   Median : 7.730  
##  Mean   : 7.65   Mean   : 7.842   Mean   : 7.630   Mean   : 7.666  
##  3rd Qu.: 9.47   3rd Qu.: 9.401   3rd Qu.: 9.449   3rd Qu.: 9.344  
##  Max.   :15.53   Max.   :15.473   Max.   :15.299   Max.   :15.663  
##      BC6156          BC6200           BC6040           BC6062     
##  Min.   :-1.41   Min.   :-0.701   Min.   :-0.398   Min.   :-0.901  
##  1st Qu.: 5.89   1st Qu.: 6.026   1st Qu.: 6.297   1st Qu.: 5.828  
##  Median : 7.79   Median : 7.838   Median : 7.953   Median : 7.692  
##  Mean   : 7.61   Mean   : 7.700   Mean   : 7.841   Mean   : 7.544  
##  3rd Qu.: 9.46   3rd Qu.: 9.494   3rd Qu.: 9.462   3rd Qu.: 9.377  
##  Max.   :15.41   Max.   :15.885   Max.   :16.030   Max.   :15.840  
```

Problem 2(b).

2.b.1 Correlation heatmap: the heatmap is helpful for visualizing the correlation. The lowest correlation is around 0.82, with the majority around 0.9, which is reasonable since we are calculating correlation bewteen the subjects. These people were all breast cancer patients, so it is not suprising that their gene expressions are highly correlated.

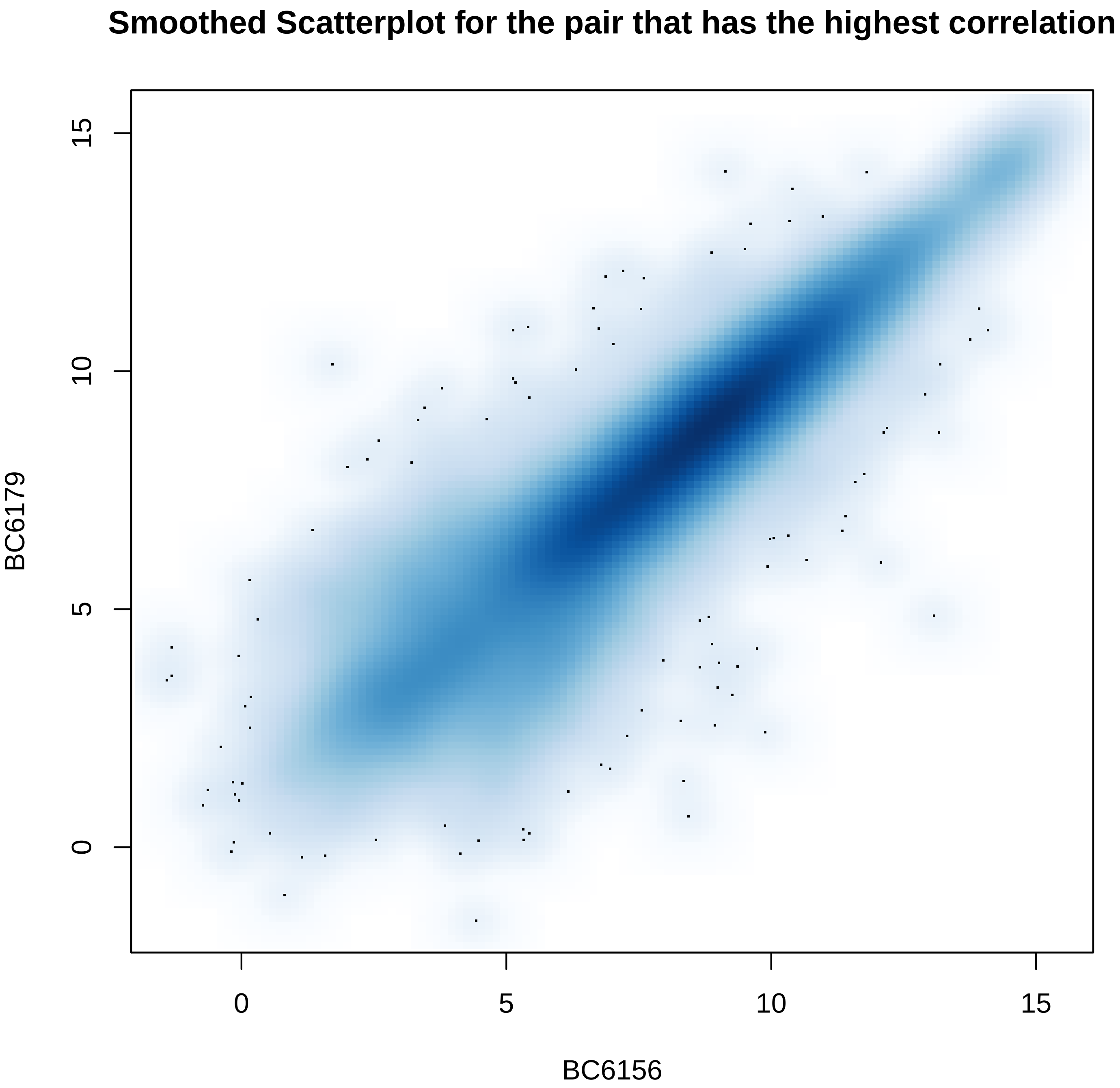
```
myPalette = colorRampPalette(rev(brewer.pal(11, "Spectral")))

correlation = cor(exprs_sample1)
levelplot(correlation,col.regions=myPalette,
          main=" Levelplot for 20 microarrays", scales=list(x=list(rot=90)))
```



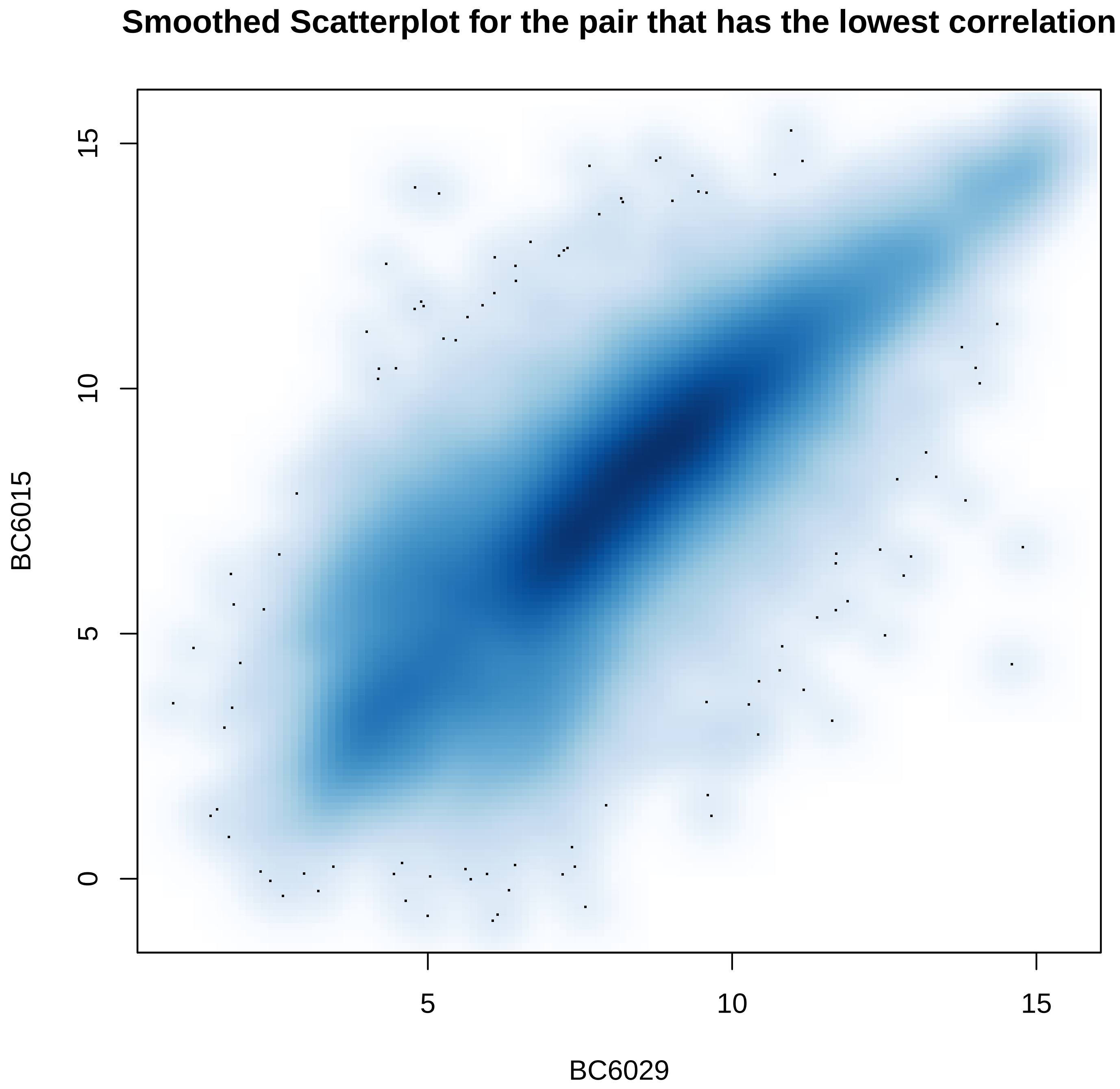
2.b.2 Smoothed scatterplot:the plot tells us what gene expressions two subjects have most in common, or density of the points. It can also tell us the correlation and functional form at the same time. Hypothetically, we can draw such plot for each pair of two subjects(i.e. 190 pairs), but there are too many. So I decide to just check the pairs that have either the biggest correlation(excluding the pait of themselves) or the lowest correlation

```
max.cor = max(correlation[correlation!=1])
min.cor = min(correlation)

cor.df=as.data.frame(correlation)
max.index=which(correlation==max.cor, arr.ind=T)
min.index=which(correlation==min.cor, arr.ind=T)

max.pairname = rownames(max.index)
min.pairname = rownames(min.index)
```

```
smoothScatter(exprs_sample1[,max.pairname[1]],exprs_sample1[,max.pairname[2]],
              main="Smoothed Scatterplot for the pair that has the highest correlation",
              xlab=max.pairname[1],
              ylab=max.pairname[2])
```



```
smoothScatter(exprs_sample1[,min.pairname[1]],exprs_sample1[,min.pairname[2]],
              main="Smoothed Scatterplot for the pair that has the lowest correlation",
              xlab=min.pairname[1],
              ylab=min.pairname[2])
```



2.b.3 Numerical summaries

```
correlation
```

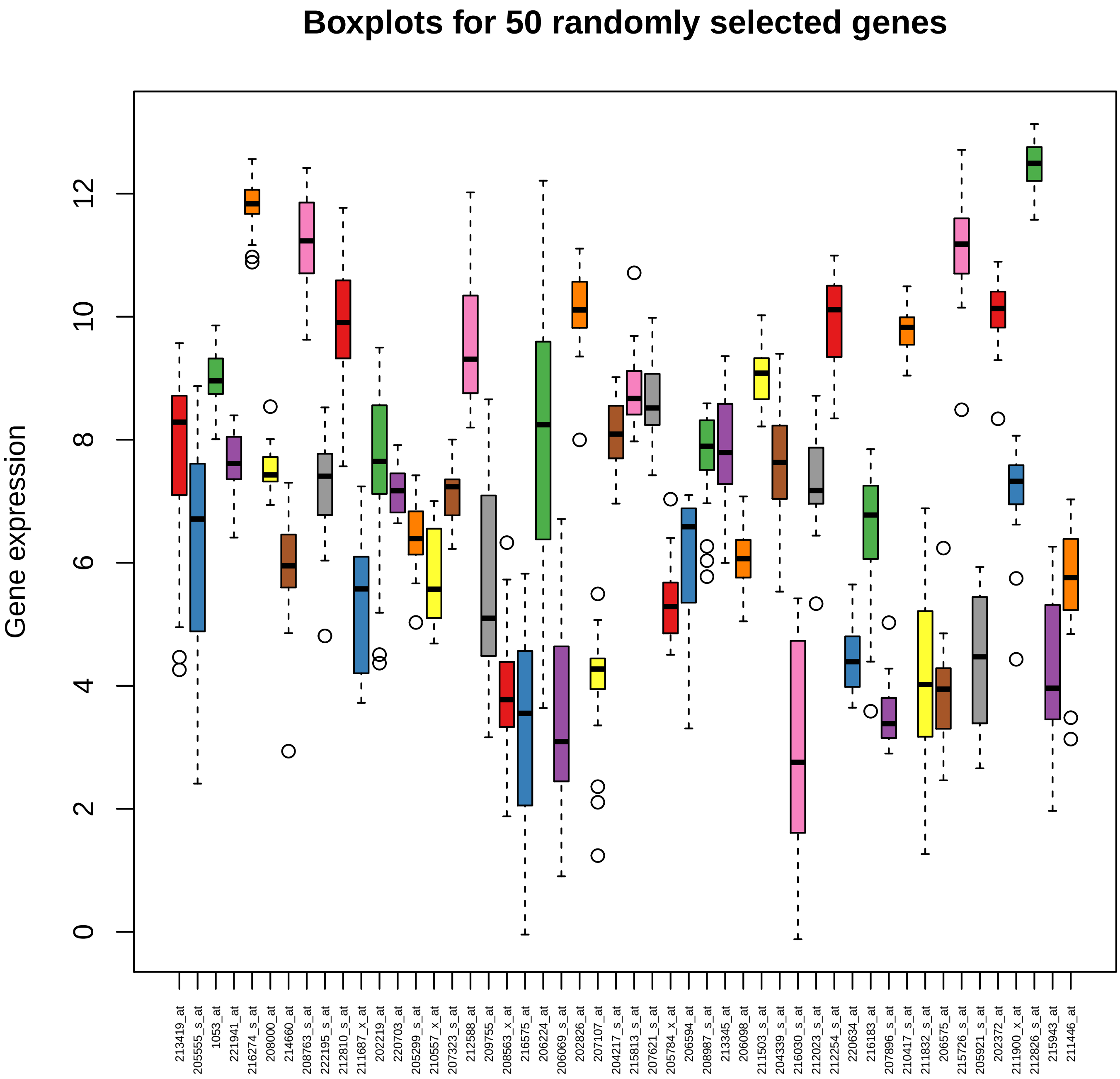
```
##        BC6015 BC6042 BC6012 BC6207 BC6136 BC6083 BC6112 BC6031 BC6179
## BC6015 1.0000 0.8621 0.8954 0.8895 0.8544 0.8883 0.8815 0.8880 0.9000
## BC6042 0.8621 1.0000 0.8895 0.8732 0.8654 0.8746 0.8836 0.9028 0.8967
## BC6012 0.8954 0.8895 1.0000 0.8883 0.8726 0.8857 0.9031 0.9178 0.9237
## BC6207 0.8895 0.8732 0.8883 1.0000 0.8697 0.8881 0.8772 0.8928 0.9013
## BC6136 0.8544 0.8654 0.8726 0.8697 1.0000 0.8823 0.8846 0.8846 0.8816
## BC6083 0.8883 0.8746 0.8857 0.8881 0.8823 1.0000 0.8905 0.8892 0.9027
## BC6112 0.8815 0.8836 0.9031 0.8772 0.8846 0.8905 1.0000 0.9107 0.9088
## BC6031 0.8880 0.9028 0.9178 0.8928 0.8846 0.8892 0.9107 1.0000 0.9204
## BC6179 0.9000 0.8967 0.9237 0.9013 0.8816 0.9027 0.9088 0.9204 1.0000
## BC6102 0.8987 0.8948 0.9113 0.8981 0.8857 0.9028 0.9043 0.9198 0.9213
## BC6014 0.8564 0.8574 0.8733 0.8503 0.8847 0.8743 0.8769 0.8741 0.8698
## BC6141 0.8897 0.8795 0.9012 0.8842 0.8709 0.8889 0.9130 0.9140 0.9070
## BC6097 0.8848 0.8605 0.8837 0.8758 0.8509 0.8790 0.8884 0.8885 0.8954
## BC6029 0.8306 0.8602 0.8391 0.8468 0.8693 0.8575 0.8620 0.8577 0.8474
## BC6068 0.8570 0.8597 0.8644 0.8645 0.9003 0.8812 0.8901 0.8816 0.8777
## BC6230 0.8802 0.8512 0.8651 0.8690 0.8382 0.8744 0.8653 0.8721 0.8733
## BC6156 0.8913 0.8905 0.9186 0.8966 0.8886 0.9018 0.9031 0.9162 0.9239
## BC6200 0.8778 0.8700 0.8906 0.8869 0.8546 0.8729 0.8831 0.8943 0.8986
## BC6040 0.8684 0.8702 0.8824 0.8655 0.8470 0.8619 0.8830 0.8944 0.8843
## BC6062 0.8650 0.8792 0.8797 0.8851 0.8705 0.8780 0.8719 0.8914 0.8847
##        BC6102 BC6014 BC6141 BC6097 BC6029 BC6068 BC6230 BC6156 BC6200
## BC6015 0.8987 0.8564 0.8897 0.8848 0.8306 0.8570 0.8802 0.8913 0.8778
## BC6042 0.8948 0.8574 0.8795 0.8605 0.8602 0.8597 0.8512 0.8905 0.8700
## BC6012 0.9113 0.8733 0.9012 0.8837 0.8391 0.8644 0.8651 0.9186 0.8906
## BC6207 0.8981 0.8503 0.8842 0.8758 0.8468 0.8645 0.8690 0.8966 0.8869
## BC6136 0.8857 0.8847 0.8709 0.8509 0.8693 0.9003 0.8382 0.8886 0.8546
## BC6083 0.9028 0.8743 0.8889 0.8790 0.8575 0.8812 0.8744 0.9018 0.8729
## BC6112 0.9043 0.8769 0.9130 0.8884 0.8620 0.8901 0.8653 0.9031 0.8831
## BC6031 0.9198 0.8741 0.9140 0.8885 0.8577 0.8816 0.8721 0.9162 0.8943
## BC6179 0.9213 0.8698 0.9070 0.8954 0.8474 0.8777 0.8733 0.9239 0.8986
## BC6102 1.0000 0.8755 0.9051 0.8863 0.8558 0.8831 0.8784 0.9214 0.8906
## BC6014 0.8755 1.0000 0.8640 0.8366 0.8531 0.8804 0.8349 0.8663 0.8363
## BC6141 0.9051 0.8640 1.0000 0.9036 0.8468 0.8729 0.8693 0.9023 0.8952
## BC6097 0.8863 0.8366 0.9036 1.0000 0.8335 0.8552 0.8726 0.8827 0.8857
## BC6029 0.8558 0.8531 0.8468 0.8335 1.0000 0.8609 0.8314 0.8611 0.8307
## BC6068 0.8831 0.8804 0.8729 0.8552 0.8609 1.0000 0.8433 0.8750 0.8539
## BC6230 0.8784 0.8349 0.8693 0.8726 0.8314 0.8433 1.0000 0.8705 0.8554
## BC6156 0.9214 0.8663 0.9023 0.8827 0.8611 0.8750 0.8705 1.0000 0.8946
## BC6200 0.8906 0.8363 0.8952 0.8857 0.8307 0.8539 0.8554 0.8946 1.0000
## BC6040 0.8806 0.8364 0.8981 0.8918 0.8443 0.8472 0.8595 0.8781 0.8878
## BC6062 0.8905 0.8506 0.8659 0.8451 0.8473 0.8590 0.8527 0.8974 0.8587
##        BC6040 BC6062
## BC6015 0.8684 0.8650
## BC6042 0.8702 0.8792
## BC6012 0.8824 0.8797
## BC6207 0.8655 0.8851
## BC6136 0.8470 0.8705
## BC6083 0.8619 0.8780
## BC6112 0.8830 0.8719
## BC6031 0.8944 0.8914
## BC6179 0.8843 0.8847
## BC6102 0.8806 0.8905
## BC6014 0.8364 0.8506
## BC6141 0.8981 0.8659
## BC6097 0.8918 0.8451
## BC6029 0.8443 0.8473
## BC6068 0.8472 0.8590
## BC6230 0.8595 0.8527
## BC6156 0.8781 0.8974
## BC6200 0.8878 0.8587
## BC6040 1.0000 0.8451
## BC6062 0.8451 1.0000
```

Problem 3.

3.1 Boxplot: the boxplot shows that the dispersions and location are quite different for each of 50 genes.

```
# sample a subset of 50 genes
genes = 1:nrow(exprs_sample1)
genes_sample1 = sample(genes,50)
```

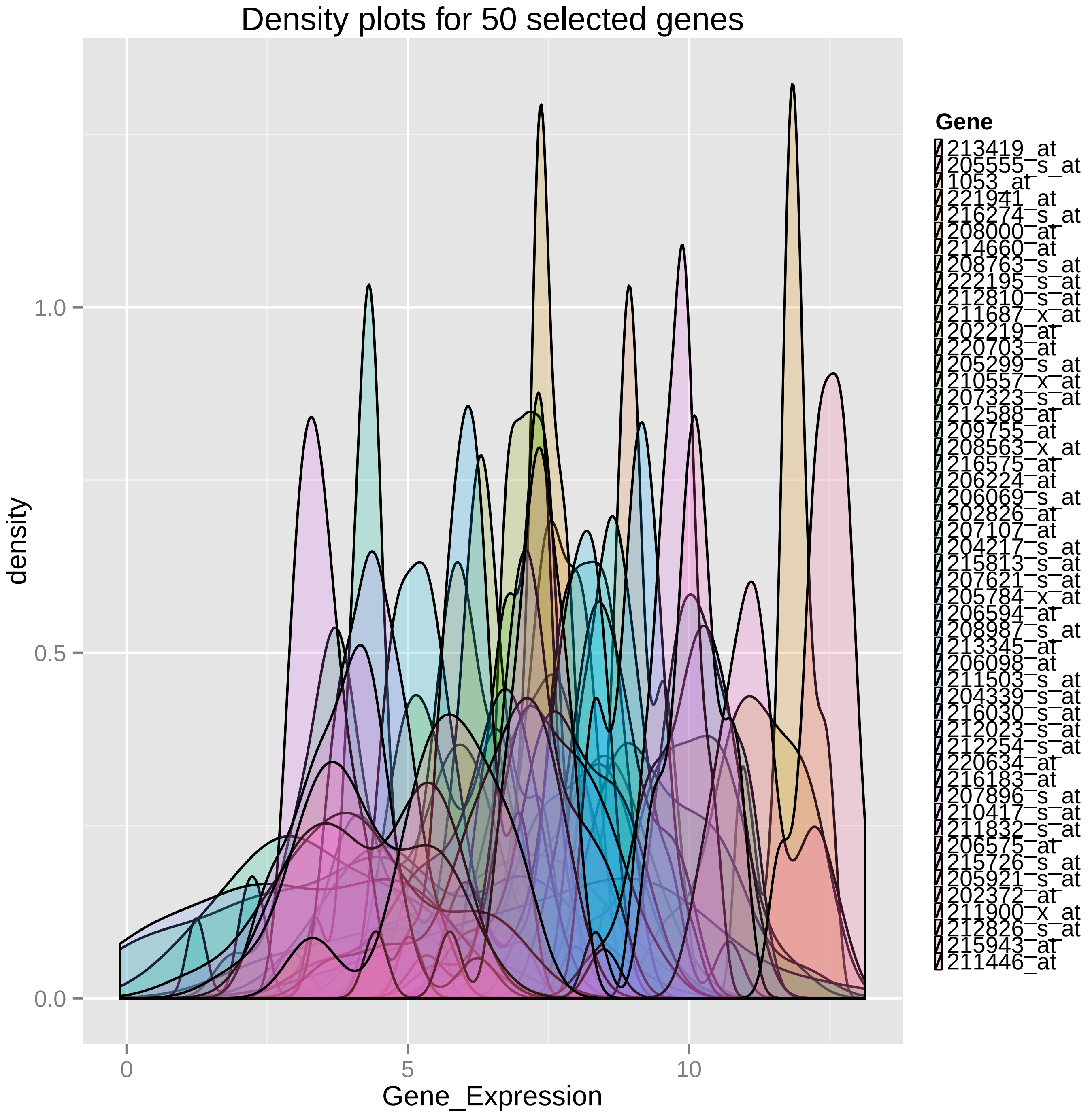
```
exprs_sample2 = t(exprs_sample1[genes_sample1,])

# boxplot
boxplot(exprs_sample2, col = colors, xaxt='n',
        main="Boxplots for 50 randomly selected genes",
        ylab="Gene expression")
axis(1,at = 1:50, labels = colnames(exprs_sample2), cex.axis=0.4, las = 2)
```



3.2 Density plot: the density plot enables us to have an overview of the entire distribution of the data. The conclusion made from density plots is similar to that from boxplots, that is, the distributions for different genes varies a lot.

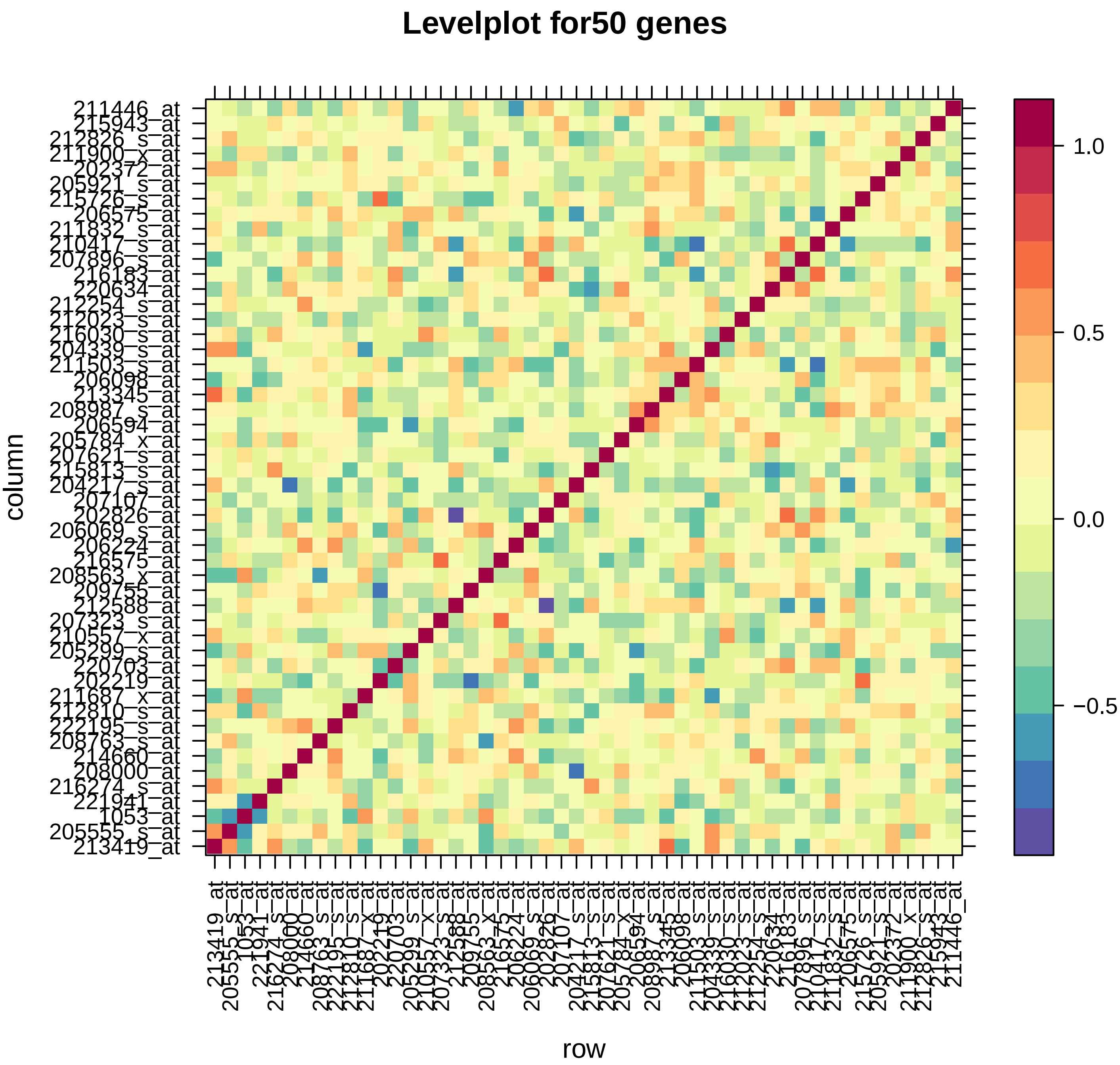
```
exprs_sample2_long = melt(exprs_sample2)
colnames(exprs_sample2_long) = c("SubjectID","Gene", "Gene_Expression")
```

```
layer2 = geom_density (alpha = 0.2)
require(grid)

## Loading required package:  grid

legend.size = theme(legend.key.size = unit(0.1, "cm"))
gg = ggplot(exprs_sample2_long, mapping = aes(fill = Gene, x = Gene_Expression))
dp2 = gg + layer2 + legend.size
dp2 + ggtitle("Density plots for 50 selected genes")
```



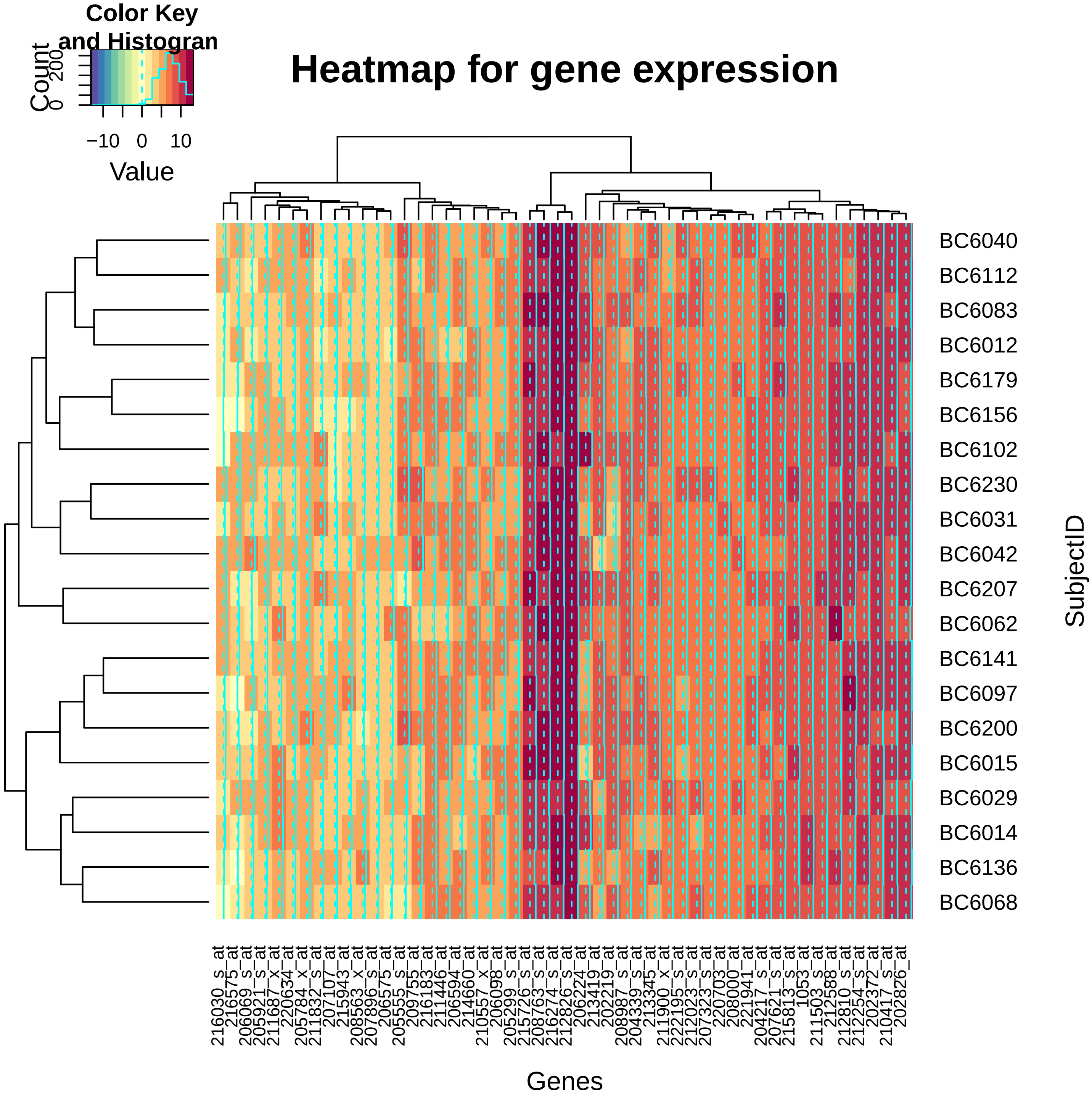
3.3 Correlation heatmap: give the number of pairs, which is quite big in this case, a visual heatmap like this is much helpful than a correlation matrix.

```
correlation2 = cor(exprs_sample2)
levelplot(correlation2,col.regions=myPalette,
          main=" Levelplot for50 genes", scales=list(x=list(rot=90)))
```



3.4 Heatmap: visualize higher and lower expressed genes across genes and across subjects. It is very useful for us to identify clusters and gene expression patterns

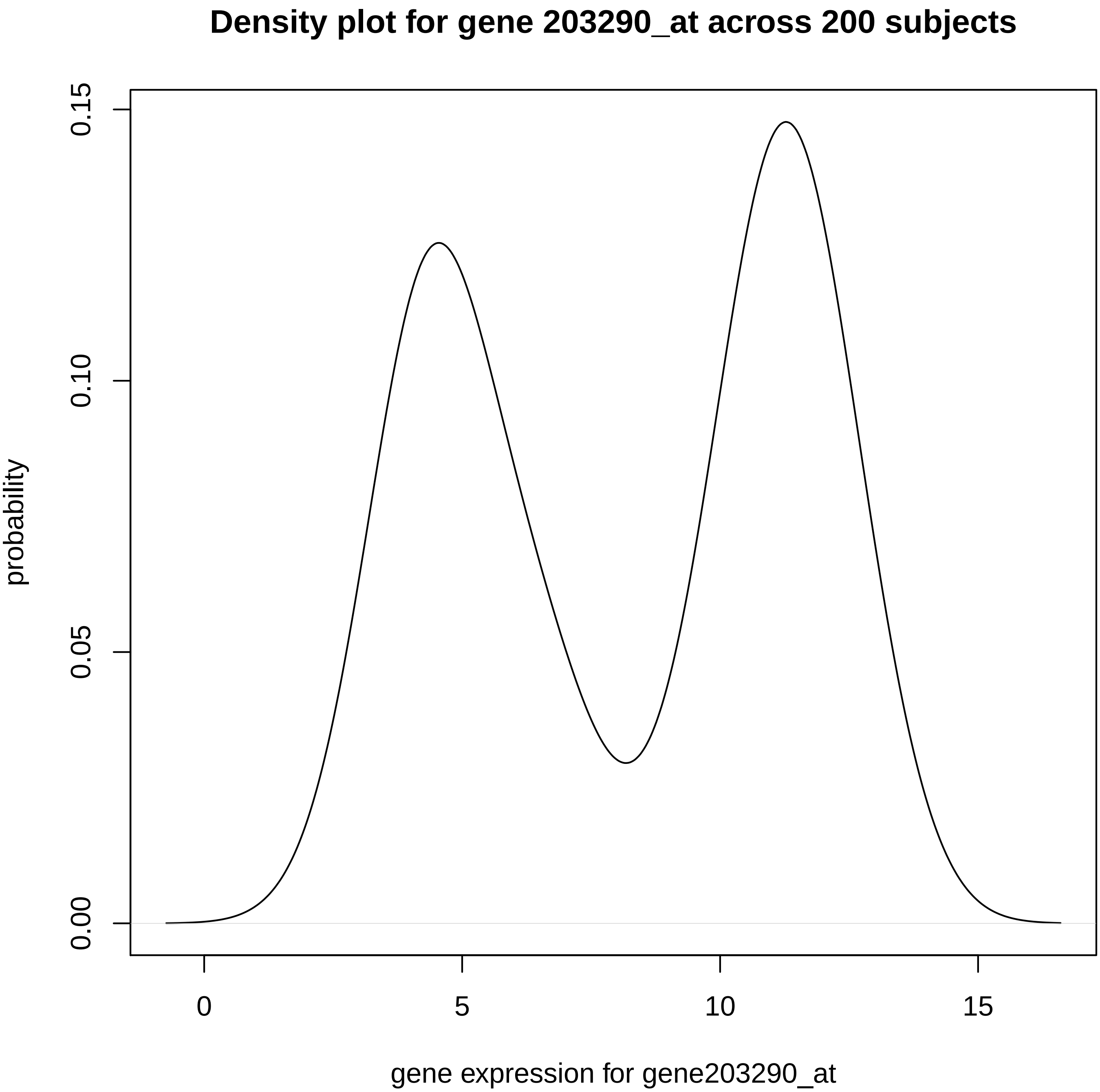
```
heatmap.2(exprs_sample2, main ="Heatmap for gene expression",
          col = myPalette, xlab = "Genes", ylab="SubjectID", keysize=1, cex.main = 0.5,
          margins = c(7,7))
```



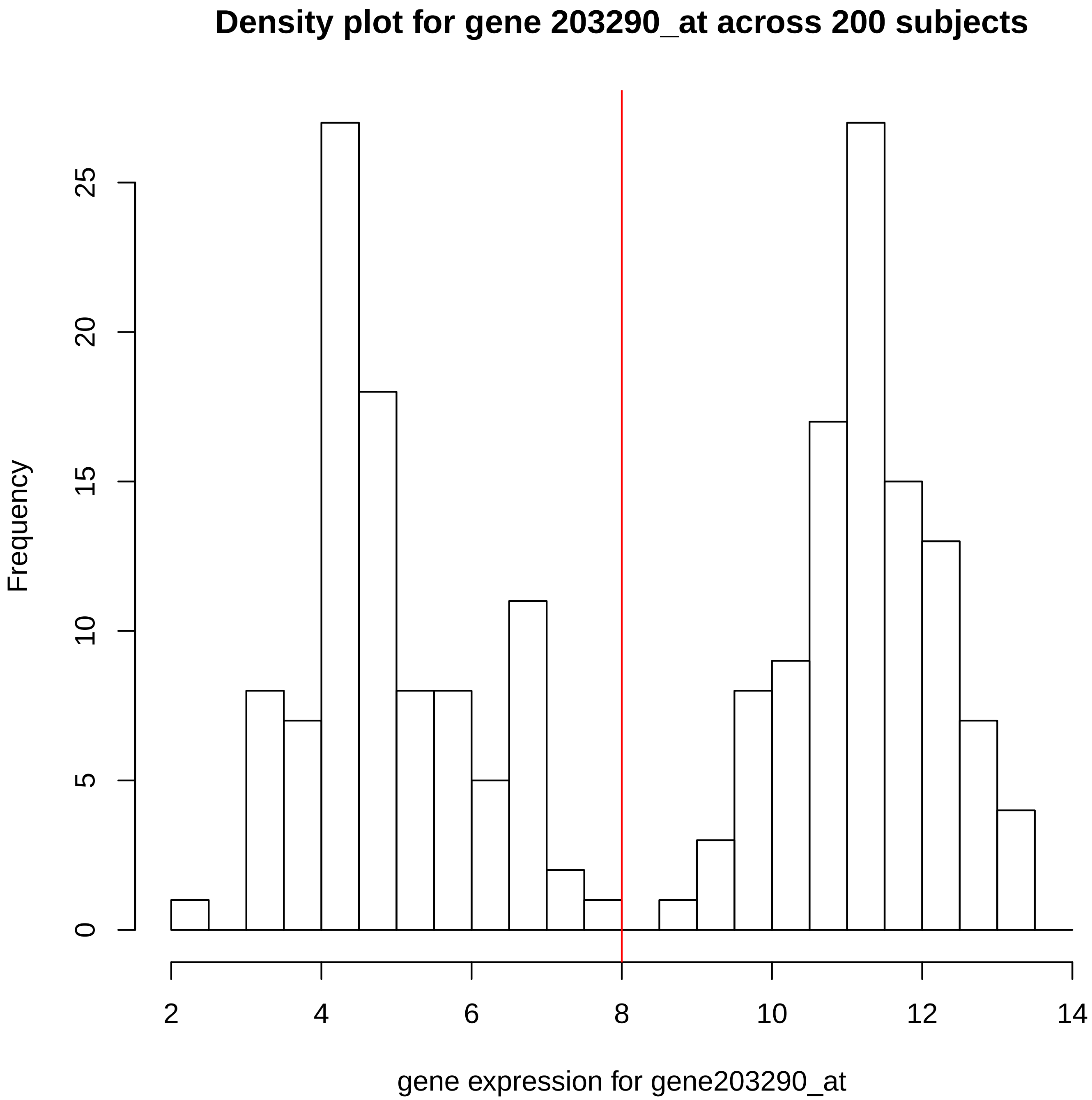
Problem 4.

The density plot shows us that there are two groups.
The histogram gives us a more detailed look on the gene expression level to find the threshold for the separation.
From the histogram, we find that gene expression at 8 is a very clear threshold.

```
gene203290_at = expression["203290_at",]

plot(density(gene203290_at), xlab="gene expression for gene203290_at",
     ylab="probability", main="Density plot for gene 203290_at across 200 subjects")
```



```r
hist(gene203290_at,breaks=seq(2,14,by=0.5),
     xlab="gene expression for gene203290_at",
     main="Density plot for gene 203290_at across 200 subjects")
abline(v=8,col="red")
```



Problem 5.

```
library("breastCancerMAINZ")
# from the package
feature = fData(mainz)
colnames(feature)

##  [1] "probe"                "Gene.title"
##  [3] "Gene.symbol"          "Gene.ID"
##  [5] "EntrezGene.ID"        "UniGene.title"
##  [7] "UniGene.symbol"       "UniGene.ID"
##  [9] "Nucleotide.Title"     "GI"
## [11] "GenBank.Accession"    "Platform_CLONEID"
## [13] "Platform_ORF"         "Platform_SPOTID"
## [15] "Chromosome.location"   "Chromosome.annotation"
## [17] "GO.Function"           "GO.Process"
## [19] "GO.Component"          "GO.Function.1"
## [21] "GO.Process.1"          "GO.Component.1"
```

```
gene.feature = feature[feature$probe == "203290_at",c("Gene.symbol","Nucleotide.Title",
                                                      "Gene.title" , "Chromosome.location",
                                                      "Chromosome.annotation")]


# Now matching the notaiton using hgu133a.db
# 1. Map between Manufacturer Identifiers and Gene Symbols
x <- hgu133aSYMBOL
mapped_probes <- mappedkeys(x)
# Convert to a list
MI.to.GeneSym <- as.list(x[mapped_probes])
# Acquire the Gene symbol for gene 203290_at
MI.to.GeneSym[["203290_at"]]
```

```
## [1] "HLA-DQA1"
```

```
# Check if it is the same as the one in mainz
MI.to.GeneSym[["203290_at"]] == gene.feature$Gene.symbol
```

```
## [1] TRUE
```

```
# 2. Map Manufacturer IDs to Chromosomal Location
x2 <- hgu133aCHRLOC
# Get the probe identifiers that are mapped to chromosome locations
mapped_probes2 <- mappedkeys(x2)
# Convert to a list
MI.to.ChrLoc <- as.list(x2[mapped_probes2])
# Acquire the Chromosome Location for gene 203290_at
MI.to.ChrLoc[["203290_at"]]
```

```
##        6
## 32605183
```

```
# Check if it is the same as the one in mainz
MI.to.ChrLoc[["203290_at"]] == gene.feature$Chromosome.location
```

```
##     6
## FALSE
```

```
# The Chromosome location recorded in mainz
gene.feature$Chromosome.location
```

```
## [1] "6p21.3"
```

```
# The Chromosome annotation recorded in mazin
gene.feature$Chromosome.annotation
```

```
## [1] "Chromosome 6, NC_000006.11 (32605183..32611429)"
```

```
x3 <- hgu133aCHRLOCEND
# Get the probe identifiers that are mapped to chromosome locations
mapped_probes3 <- mappedkeys(x3)
# Convert to a list
MI.to.ChrLocEnd <- as.list(x3[mapped_probes3])
# Acquire the Chromosome Location for gene 203290_at
MI.to.ChrLocEnd[["203290_at"]]
```

```
##        6
## 32611429

# 3. description:
x3 <- hgu133aGENENAME
# Get the probe identifiers that are mapped to a gene name
mapped_probes3 <- mappedkeys(x3)
# Convert to a list
MI.to.name <- as.list(x3[mapped_probes3])
# Acquire gene name for gene 203290_at
MI.to.name[["203290_at"]]

## [1] "major histocompatibility complex, class II, DQ alpha 1"

# Compare it with the one in mainz
gene.feature$Nucleotide.Title

## [1] "Homo sapiens major histocompatibility complex, class II, DQ alpha 1 (HLA-DQA1), mRNA"

MI.to.name[["203290_at"]] == gene.feature$Gene.title

## [1] TRUE

# They are recorded in different format, but they are the same
```

If we look at the chormosome annotation, we know that the gene is on chromosome 6,from 32605183 to 32611429. The start number matches with the number acquired using hgu133aCHRLOC, the end number mathes with the number acquired using hgu133aCHRLOCEND. Both of these two results also has 6 on the first row of the output, indicating that it is from chromosome 6. The way that chromosome location is recorded in data mainz is in a different format. However, we can conclude that the record matches.