Class12 Transcriptomics and RNA-Seq data

AUTHOR
Ryan Chung A15848050

```
#library(BiocManager)
#library(DESeq2)
```

Here we will use the DeSeq2 package for RNASeq analysis the data comes from a studty (Himes et al. 2014) on airway smooth muscle cells treated with steroids.

Importing countData and colData

We need two things for this analysis: - **countData** (counts for every transcript/gene) - **ColData** (metadata that describes experimental setup)

```
countData <- read.csv("airway_scaledcounts.csv", row.names=1)
head(countData)
```

```
                SRR1039508 SRR1039509 SRR1039512 SRR1039513 SRR1039516
ENSG00000000003        723        486        904        445       1170
ENSG00000000005          0          0          0          0          0
ENSG00000000419        467        523        616        371        582
ENSG00000000457        347        258        364        237        318
ENSG00000000460         96         81         73         66        118
ENSG00000000938          0          0          1          0          2
                SRR1039517 SRR1039520 SRR1039521
ENSG00000000003       1097        806        604
ENSG00000000005          0          0          0
ENSG00000000419        781        417        509
ENSG00000000457        447        330        324
ENSG00000000460         94        102         74
ENSG00000000938          0          0          0
```

```
metadata <-  read.csv("airway_metadata.csv")
metadata
```

```
         id     dex celltype     geo_id
1 SRR1039508 control   N61311 GSM1275862
2 SRR1039509 treated   N61311 GSM1275863
3 SRR1039512 control  N052611 GSM1275866
4 SRR1039513 treated  N052611 GSM1275867
5 SRR1039516 control  N080611 GSM1275870
6 SRR1039517 treated  N080611 GSM1275871
7 SRR1039520 control  N061011 GSM1275874
8 SRR1039521 treated  N061011 GSM1275875
```

> Q1. How many genes are in this dataset?

```
nrow(countData)
```

```
[1] 38694
```

38694 genes

> Q2. How many 'control' cell lines do we have?

```
table(metadata$dex)
```

```
control treated
      4       4
```

```
#another method
metadata$dex == 'control' #gives u T/F values
```

```
[1]  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE
```

```
sum(metadata$dex == 'control') #actually gives the number
```

```
[1] 4
```

There are 4 control cell lines

Toy differential gene expression

- Step 1/ Calculate the mean of the control samples (i.e. columns in countData) Calculate the mean of the treated samples

a. We need to find which columns in countData in "control" samples

- look in the metadata - our colData(metadata) dex column

Calculating the control treatment means

LAB SHEET WAY - double pound = code, single = comments

```
#[r,c]
#index metadata for all rows where dex = control store as control

##control <- metadata[metadata[ ,"dex" ]  == 'control',]
```

```
#now index the control count data by using the control ID's from the metadata

##control.counts <- countData[ ,control$id]
##head(countData[ ,control$id])

#take the mean of each treatment
##control.mean <- rowSums( control.counts )/4

##head(control.mean)
```

IN CLASS WAY

```
control.inds <- metadata$dex == 'control'
```

b. Extract all the control columns from `countData` nad call it `control.counts`

```
control.counts <- countData [ , control.inds]
```

c. Calculate the mean value across the rows of `control.counts` i.e. calculate the mean count values for each gene in the control samples

```
control.means <- rowMeans(control.counts)
head(control.means)
```

```
ENSG00000000003 ENSG00000000005 ENSG00000000419 ENSG00000000457 ENSG00000000460
         900.75             0.00           520.50           339.75            97.25
ENSG00000000938
           0.75
```

> Q3. How would you make the above code in either approach more robust?

I would condense the calculating mean code into a function in a way where I can calculate means for both treatment and controls.

> Q4. Follow the same procedure and calculate the treatment means

Calculating treatment means

a. Index for treatment

```
treat.inds <-  metadata$dex == 'treated'
```

b. Extract treatment columns

```
treat.counts <- countData[ ,treat.inds]
```

c. take the mean values across the rows

```
treat.means <- rowMeans(treat.counts)
head(treat.means)
```

```
ENSG00000000003 ENSG00000000005 ENSG00000000419 ENSG00000000457 ENSG00000000460
         658.00            0.00          546.00          316.50           78.75
ENSG00000000938
           0.00
```

Store the means for book keeping.

```
meancounts <- data.frame(control.means,treat.means )
```

> Q5. Create a scatter plot of the means using base R and ggplot

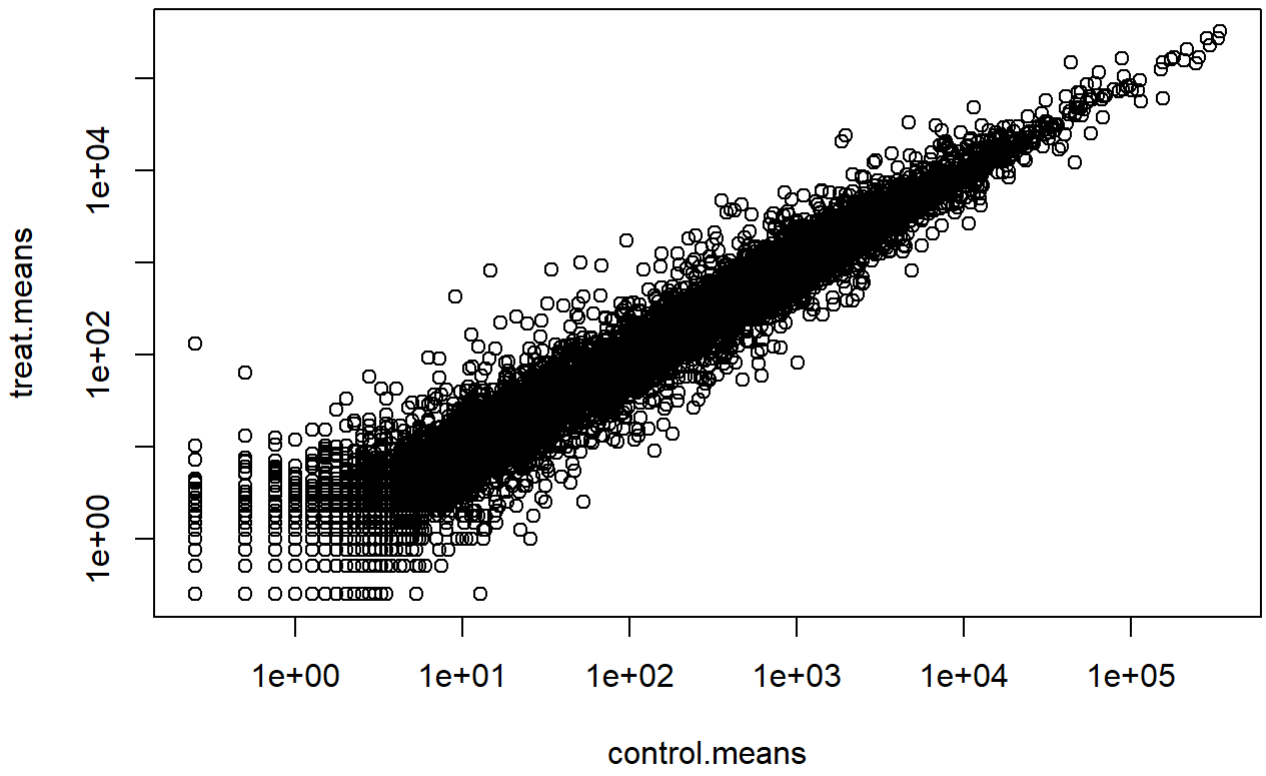
See below

> Q6 Plot both axes on a log scale

See below

```
library(ggplot2)
plot(meancounts, log = 'xy')
```

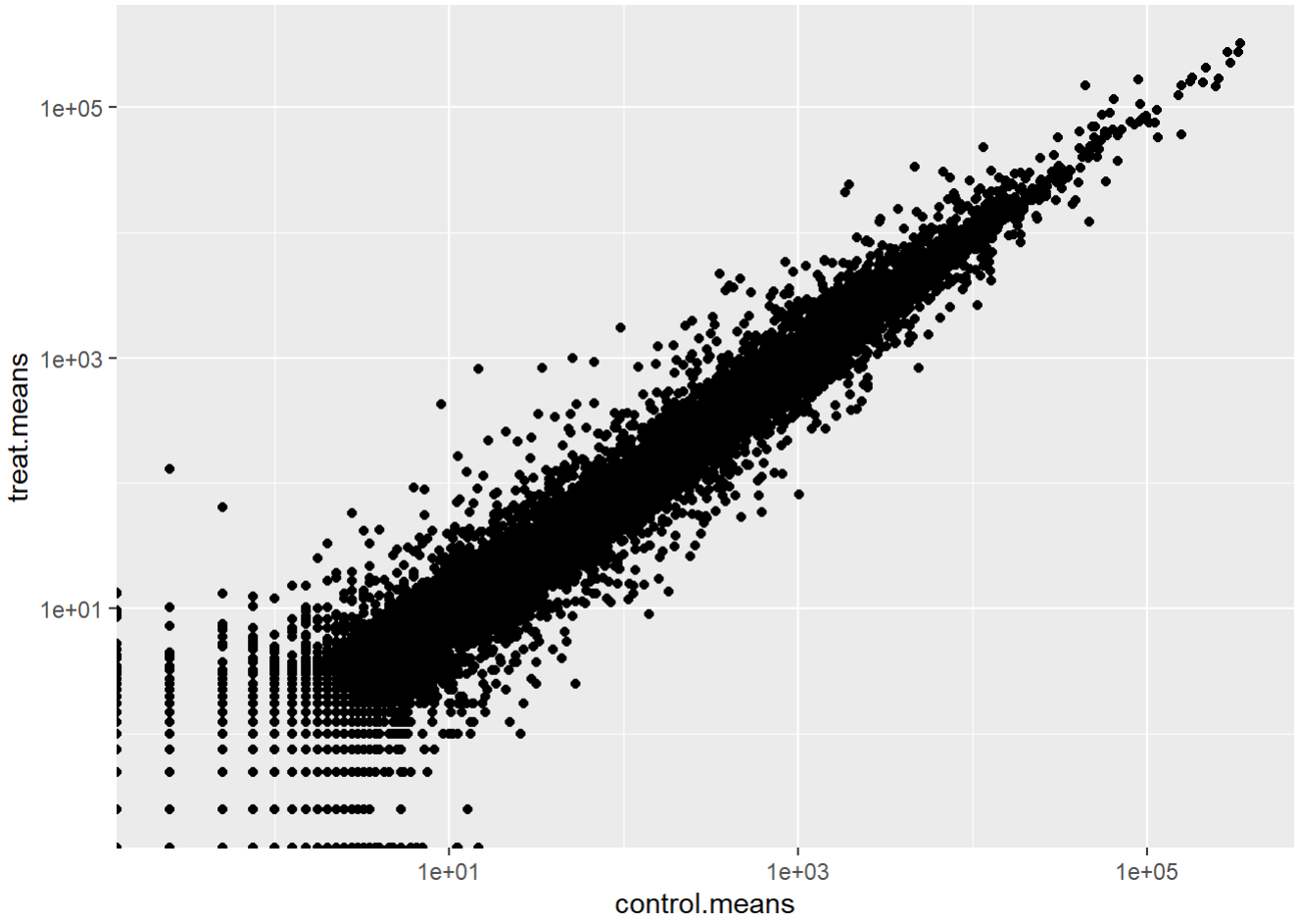
```
Warning in xy.coords(x, y, xlabel, ylabel, log): 15032 x values <= 0 omitted
from logarithmic plot

Warning in xy.coords(x, y, xlabel, ylabel, log): 15281 y values <= 0 omitted
from logarithmic plot
```




```
ggplot(meancounts)+
  aes(control.means, treat.means) +
  geom_point() +
  scale_x_log10() +
  scale_y_log10()
```

```
Warning: Transformation introduced infinite values in continuous x-axis
```

```
Warning: Transformation introduced infinite values in continuous y-axis
```



We can use log transforms for skewed data and because we care more about relative changes in magnitude

We most often use log2 as our transform as themath is easier to interpret htan log10

If we have no change - i.ie. same values in control and treated we will have a log2 value of 0

```
(20/20)
```

```
[1] 1
```

```
log2(20/20) #if same values = 0
```

```
[1] 0
```

```
log2(20/10) # if decrease after treatment = +1 pos value log2 fold-change of +1 if double the amo
```

```
[1] 1
```

```
log2(10/20) # if increase after treatment = -1 neg value
```

```
[1] -1
```

```
log2(40/10) # two fold change
```

```
[1] 2
```

```
meancounts$log2fc <- log2(meancounts$treat.means/meancounts$control.means)
head(meancounts)
```

```
                control.means treat.means      log2fc
ENSG00000000003        900.75      658.00 -0.45303916
ENSG00000000005          0.00        0.00         NaN
ENSG00000000419        520.50      546.00  0.06900279
ENSG00000000457        339.75      316.50 -0.10226805
ENSG00000000460         97.25       78.75 -0.30441833
ENSG00000000938          0.75        0.00        -Inf
```

> Q: How many genes are up regulated at the common threshold of +2 log2FC values

Use the tables through excluding na/inf values

```
table(meancounts$log2fc >= 2)
```

```
FALSE  TRUE
23348  1910
```

```
sum(meancounts$log2fc >= 2, na.rm = TRUE)
```

```
[1] 1910
```

Hold on what about stats! Yes these are big changes but are these changes significant?

To do this properly we will turn into the DESeq2 package.

DESeq2 analysis

```
library(DESeq2)
```

To use DESeq we need our input contData and colData in a specific format that DESeq wants:

```
dds <- DESeqDataSetFromMatrix(countData = countData,
                       colData = metadata,
                       design = ~dex)
```

```
converting counts to integer mode
```

```
Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
design formula are characters, converting to factors
```

To run the analysis I can now use the main DESeq2 function called `DESeq()` with `dds` as an input

```
dds <- DESeq(dds)
```

```
estimating size factors

estimating dispersions

gene-wise dispersion estimates

mean-dispersion relationship

final dispersion estimates

fitting model and testing
```

To get the results out of this `dds` object we can use the `results()` function from the package

```
res <- results(dds)
head(res)
```

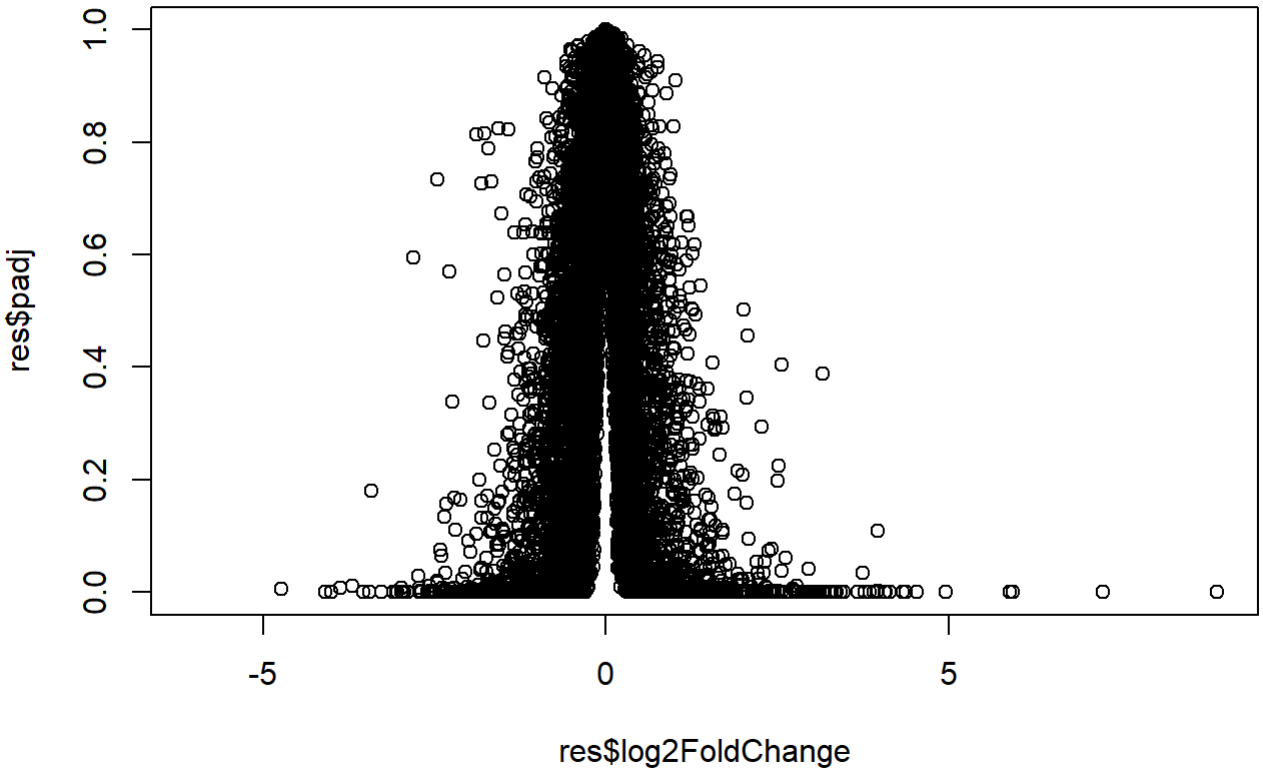
```
log2 fold change (MLE): dex treated vs control
Wald test p-value: dex treated vs control
DataFrame with 6 rows and 6 columns
                  baseMean log2FoldChange     lfcSE      stat    pvalue
                 <numeric>      <numeric> <numeric> <numeric> <numeric>
ENSG00000000003 747.194195     -0.3507030  0.168246 -2.084470 0.0371175
ENSG00000000005   0.000000             NA        NA        NA        NA
ENSG00000000419 520.134160      0.2061078  0.101059  2.039475 0.0414026
ENSG00000000457 322.664844      0.0245269  0.145145  0.168982 0.8658106
ENSG00000000460  87.682625     -0.1471420  0.257007 -0.572521 0.5669691
ENSG00000000938   0.319167     -1.7322890  3.493601 -0.495846 0.6200029
                     padj
                <numeric>
ENSG00000000003  0.163035
ENSG00000000005        NA
ENSG00000000419  0.176032
ENSG00000000457  0.961694
ENSG00000000460  0.815849
ENSG00000000938        NA
```

```
#p adj is there bc 0.05 (5%) in our dataset of 38k observations is a decently big number
#deseq uses benhamini and Hochberg method: 1) rank the genes by p-value 2) multiply each p value
```

Volcano plot - log2FC vs PADJ

Let's make a final (for today) plot of log2 fold change vs adjusted P-value

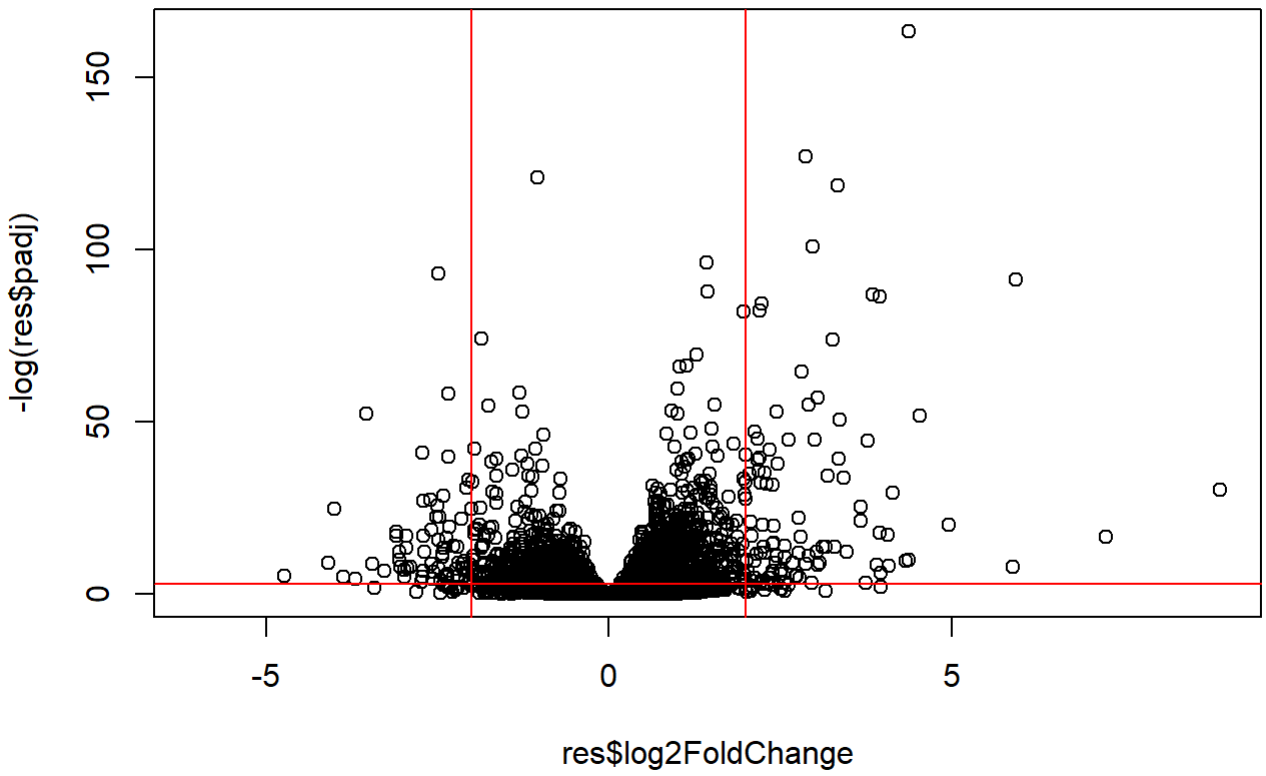
```
plot(res$log2FoldChange, res$padj)
```




```
#plot shows some skew
```

It is the low P=values that we care about and these are lost in the skewed plot above. Let's take the log of the $padj values for out plot

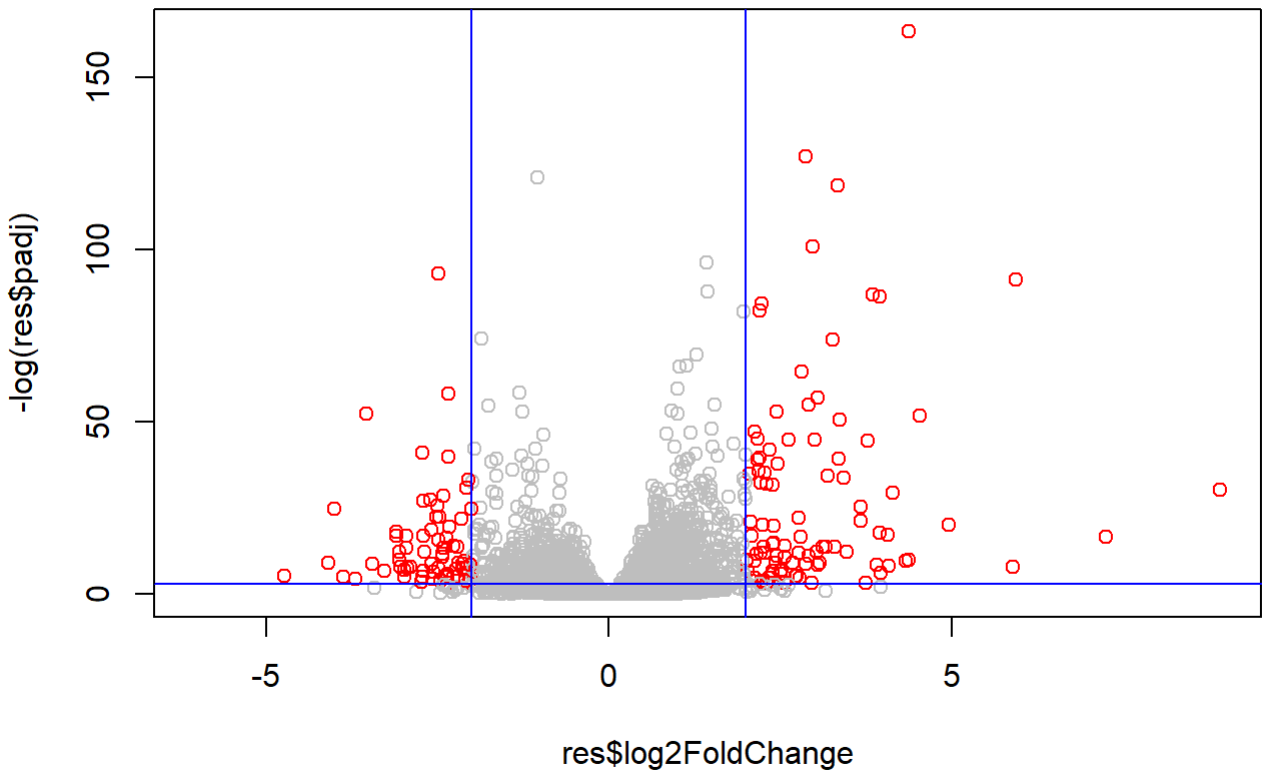
```
plot(res$log2FoldChange, -log(res$padj))
abline(v = c(+2,-2), col = 'red')
abline(h = -log(0.05), col = 'red')
```




```
#all points away from 0, x axis shows amount of change and y axis is as it goes higher it becomes
```

Finally we can make a color vector to use in the polt to better highlight the genes we care about.

```
mycols <-  rep("gray", nrow(res))
mycols[abs(res$log2FoldChange) >= 2] <- 'red'
mycols[res$padj > 0.05] <- 'gray'

plot(res$log2FoldChange, -log(res$padj), col = mycols)
abline(v = c(+2,-2), col = 'blue')
abline(h = -log(0.05), col = 'blue')
```


-log(res$padj)
0
50
100
150
-5
0
5
res$log2FoldChange