DEG Viral Volcano Plots

Annika Jorgensen

2022-07-15

R Markdown

Title: “DEG_Analysis Viral Breakdown Tumor vs Tumor Adjacent”
Author: Annika Jorgensen
Date: 05/22/2022
Purpose: This document is for the author to parse out the viral etiology tumor vs. tumor adjacent code from the “DEG_changed_comparison” file as well as demonstrate understanding of the code and theory

Libraries

The first chunk of code is dedicated to installing the libraries. These libraries are to help execute the differential analysis and helps visualize the data. The code was not included for concision.

Environment parameters

This next section of code is dedicated to the environmental parameters. Environmental parameters are a series of variables and other code that will help make the rest of the script be easier to make and run later on.

Working Directory

A working directory is a code that iterates a file path on your computer th.t sets where the default location of any files that you read into R. Working directories work different in R files than R Markdowns. R Markdown files require directories to be defined at the end of each code chunk. Meaning from here on out you will see working directories being defined at the end of each code chunk.

```
setwd('~/R')
```

Defining Colors

This chunk defines color palette variables that are going to be used in plots later on the script. These variables are defined by conversiting BrewerCode palettes into palettes that can be used in R.

```
viralPalette <- brewer.pal(8, "Set1")
hbvColor <- viralPalette[1]
hcvColor <- viralPalette[2]
bothColor <- viralPalette[3]
neitherColor <- viralPalette[4]

sexTissuePalette <- brewer.pal(12, "Paired")
maleTumorColor <- sexTissuePalette[4]
maleAdjacentColor <- sexTissuePalette[3]
femaleTumorColor <- sexTissuePalette[6]
femaleAdjacentColor <- sexTissuePalette[5]
setwd('~/R')
```

Read in Data

Read in unfiltered gene lists for volcano plots and reading in filtered data for GO/KEGG Analysis.

```
M_HBV_unfiltered <-read.csv("M_HBV_unfiltered.csv")
M_HCV_unfiltered <-read.csv("M_HCV_unfiltered.csv")
M_Neither_unfiltered <-read.csv("M_Neither_unfiltered.csv")

DEGs_M_HBV <-read.csv("DEG_M_HBV.csv")
DEGs_M_HCV <-read.csv("DEG_M_HCV.csv")
DEGs_M_Neither <- read.csv("DEG_M_Neither.csv")

F_HBV_unfiltered <-read.csv("F_HBV_unfiltered.csv")
F_HCV_unfiltered <-read.csv("F_HCV_unfiltered.csv")
F_Neither_unfiltered <-read.csv("F_Neither_unfiltered.csv")

DEGs_F_HBV <-read.csv("DEG_F_HBV.csv")
head(DEGs_F_HBV)
```

```
##                    X   chr     start       end            TXNAME
## 1  ENSG00000237649.7  chr6  33391535  33409924 ENST00000428849.6
## 2 ENSG00000100526.19 chr14  54396848  54417951 ENST00000555837.5
## 3 ENSG00000186185.13 chr17  44924708  44947637 ENST00000587309.5
## 4 ENSG00000135451.12 chr12  49323235  49324296 ENST00000549275.5
## 5 ENSG00000112984.11  chr5 138178718 138182425 ENST00000513276.5
## 6 ENSG00000175063.16 chr20  45812575  45816953 ENST00000405520.5
##               GENEID gene_name length    logFC   AveExpr        t      P.Value
## 1  ENSG00000237649.7     KIFC1  18389 4.959939 1.4714948 12.59994 3.925759e-30
## 2 ENSG00000100526.19     CDKN3  21103 5.169373 0.9017477 12.37121 2.815240e-29
## 3 ENSG00000186185.13    KIF18B  22929 4.922340 0.7676218 12.29525 5.400059e-29
## 4 ENSG00000135451.12     TROAP   1061 5.151423 1.1320168 12.16973 1.579102e-28
## 5 ENSG00000112984.11    KIF20A   3707 5.758037 0.9974977 12.15945 1.734310e-28
## 6 ENSG00000175063.16     UBE2C   4378 5.532745 1.2093500 11.96584 8.977151e-28
##      adj.P.Val        B
## 1 5.254236e-26 57.50928
## 2 1.883959e-25 55.47993
## 3 2.409146e-25 54.85611
## 4 4.642402e-25 53.83688
## 5 4.642402e-25 53.74599
## 6 2.002503e-24 52.12861
```

```
DEGs_F_HBV_relax_p <-read.csv("DEG_F_HBV_relax_p.csv")
head(DEGs_F_HBV_relax_p)
```

```
##                    X   chr     start       end            TXNAME
## 1  ENSG00000237649.7  chr6  33391535  33409924 ENST00000428849.6
## 2 ENSG00000100526.19 chr14  54396848  54417951 ENST00000555837.5
## 3 ENSG00000186185.13 chr17  44924708  44947637 ENST00000587309.5
## 4 ENSG00000135451.12 chr12  49323235  49324296 ENST00000549275.5
## 5 ENSG00000112984.11  chr5 138178718 138182425 ENST00000513276.5
## 6 ENSG00000175063.16 chr20  45812575  45816953 ENST00000405520.5
##               GENEID gene_name length    logFC   AveExpr        t      P.Value
## 1  ENSG00000237649.7     KIFC1  18389 4.959939 1.4714948 12.59994 3.925759e-30
## 2 ENSG00000100526.19     CDKN3  21103 5.169373 0.9017477 12.37121 2.815240e-29
## 3 ENSG00000186185.13    KIF18B  22929 4.922340 0.7676218 12.29525 5.400059e-29
## 4 ENSG00000135451.12     TROAP   1061 5.151423 1.1320168 12.16973 1.579102e-28
## 5 ENSG00000112984.11    KIF20A   3707 5.758037 0.9974977 12.15945 1.734310e-28
## 6 ENSG00000175063.16     UBE2C   4378 5.532745 1.2093500 11.96584 8.977151e-28
##      adj.P.Val        B
## 1 5.254236e-26 57.50928
## 2 1.883959e-25 55.47993
## 3 2.409146e-25 54.85611
## 4 4.642402e-25 53.83688
## 5 4.642402e-25 53.74599
## 6 2.002503e-24 52.12861
```

```
DEGs_F_HCV <-read.csv("DEG_F_HCV.csv")
DEGs_F_HCV_relax_p <-read.csv("DEG_F_HCV_relax_p.csv")

DEGs_F_Neither <-read.csv("DEG_F_Neither.csv")
DEGs_F_Neither_relax_p <-read.csv("DEG_F_Neither_relaX_p.csv")
```

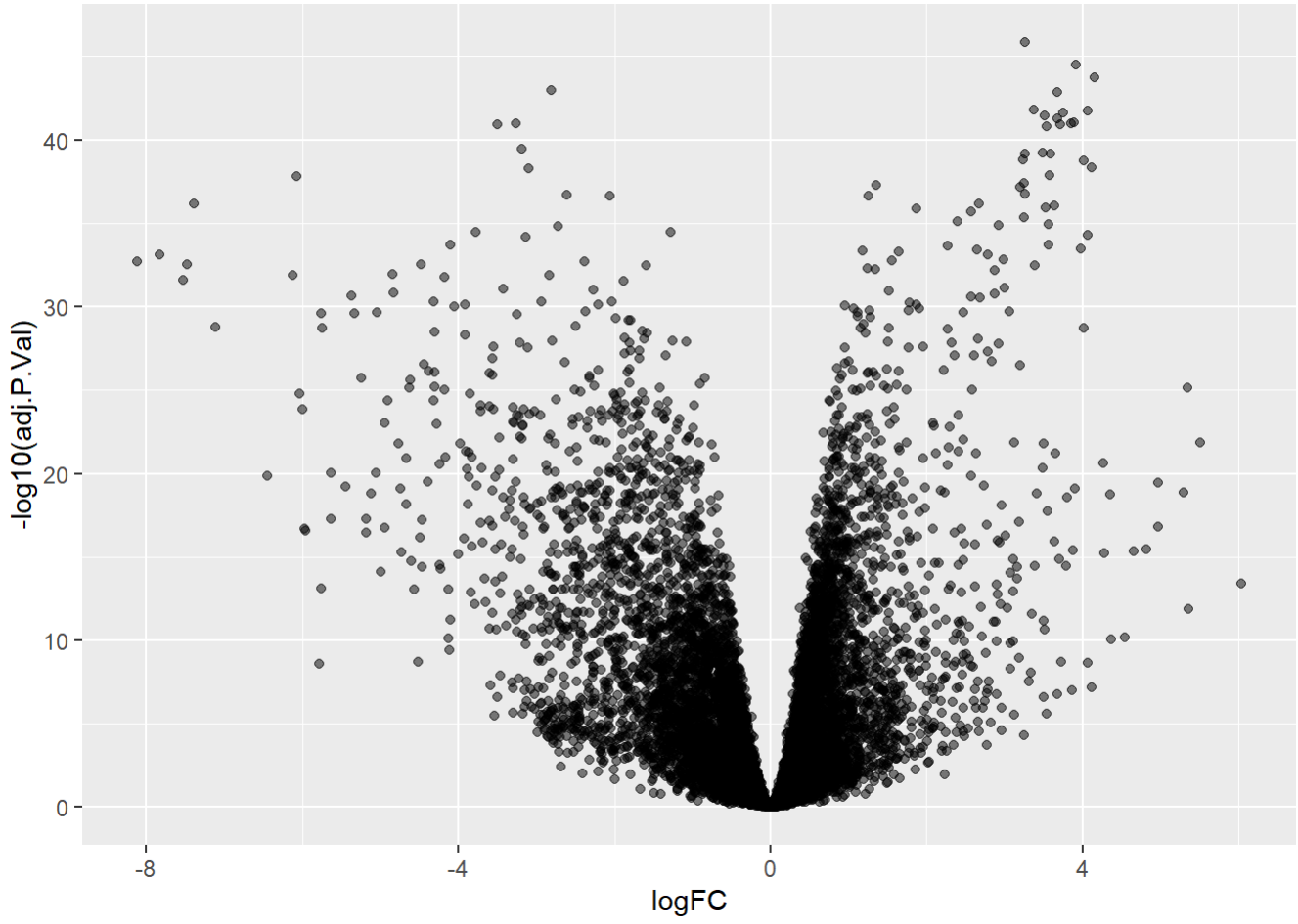
Data Visualization

This section creates Volcano plots for easy visualization of the gene lists.

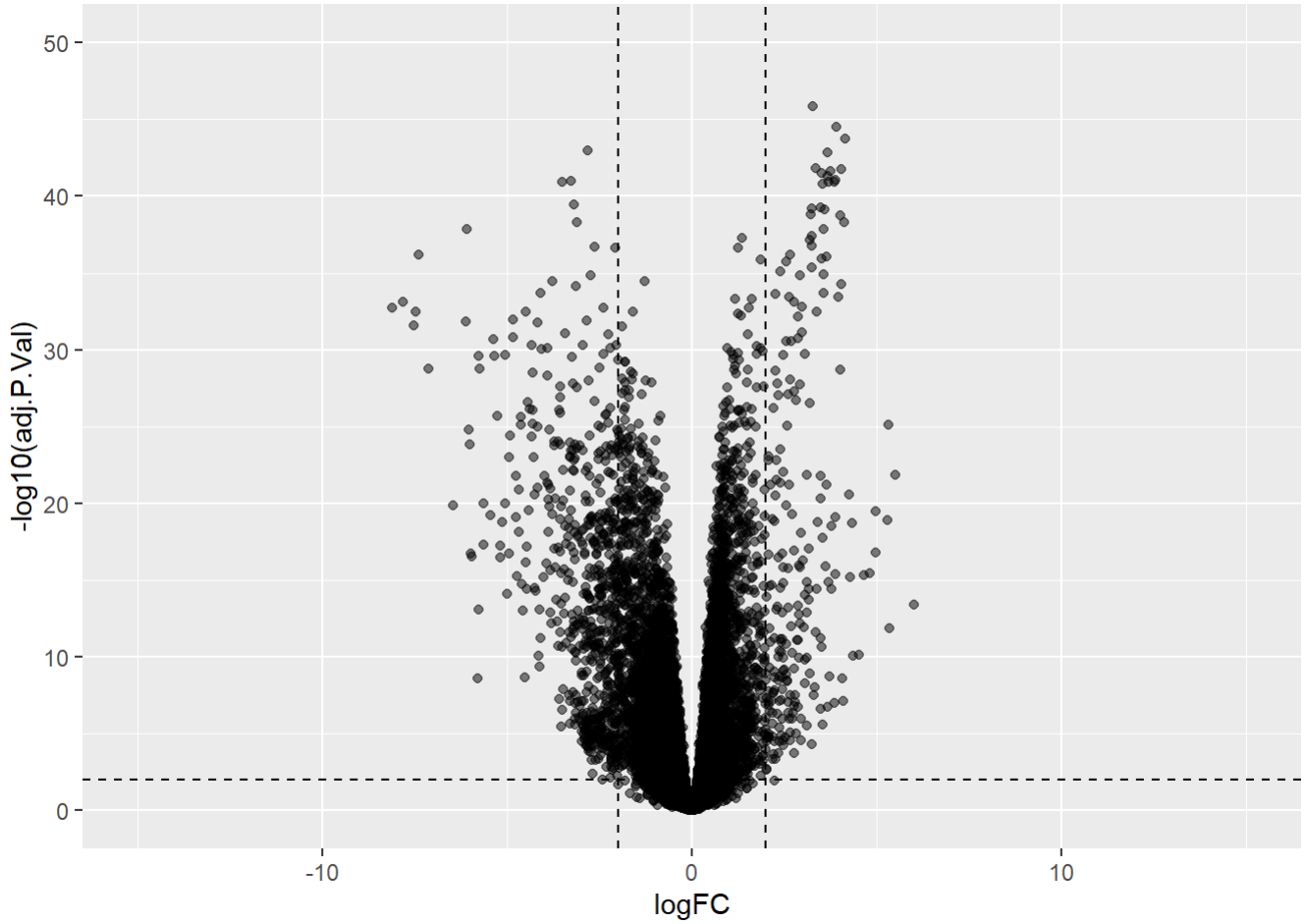
Volcano Plots Data used to create volcano plot of HBV tumor-tumor adjacent male sex.

This plot uses the unfiltered Male HBV list and is plotting genes with a P value greater than or equal to 0.05 and with an absolute logFC of 2. The x and y coordinate plane limits are x (-15, 15) and y (0, 50). The dashed lines are significance thresholds and are placed a x= -2, x=2, and y=2.

```
df <- data.frame(M_HBV_unfiltered$adj.P.Val, M_HBV_unfiltered$logFC, M_HBV_unfiltered$chr, M_HBV
_unfiltered$GENEID, M_HBV_unfiltered$gene_name)
colnames(df) <- c("adj.P.Val", "logFC", "chr", "id", "name")
#dfSig <- df[(abs(df$logFC) >= 2 & df$adj.P.Val <= 0.05),]$id
p <-ggplot(data=df, aes( x=logFC, y=-log10(adj.P.Val))) + geom_point(alpha=0.5)

p
```

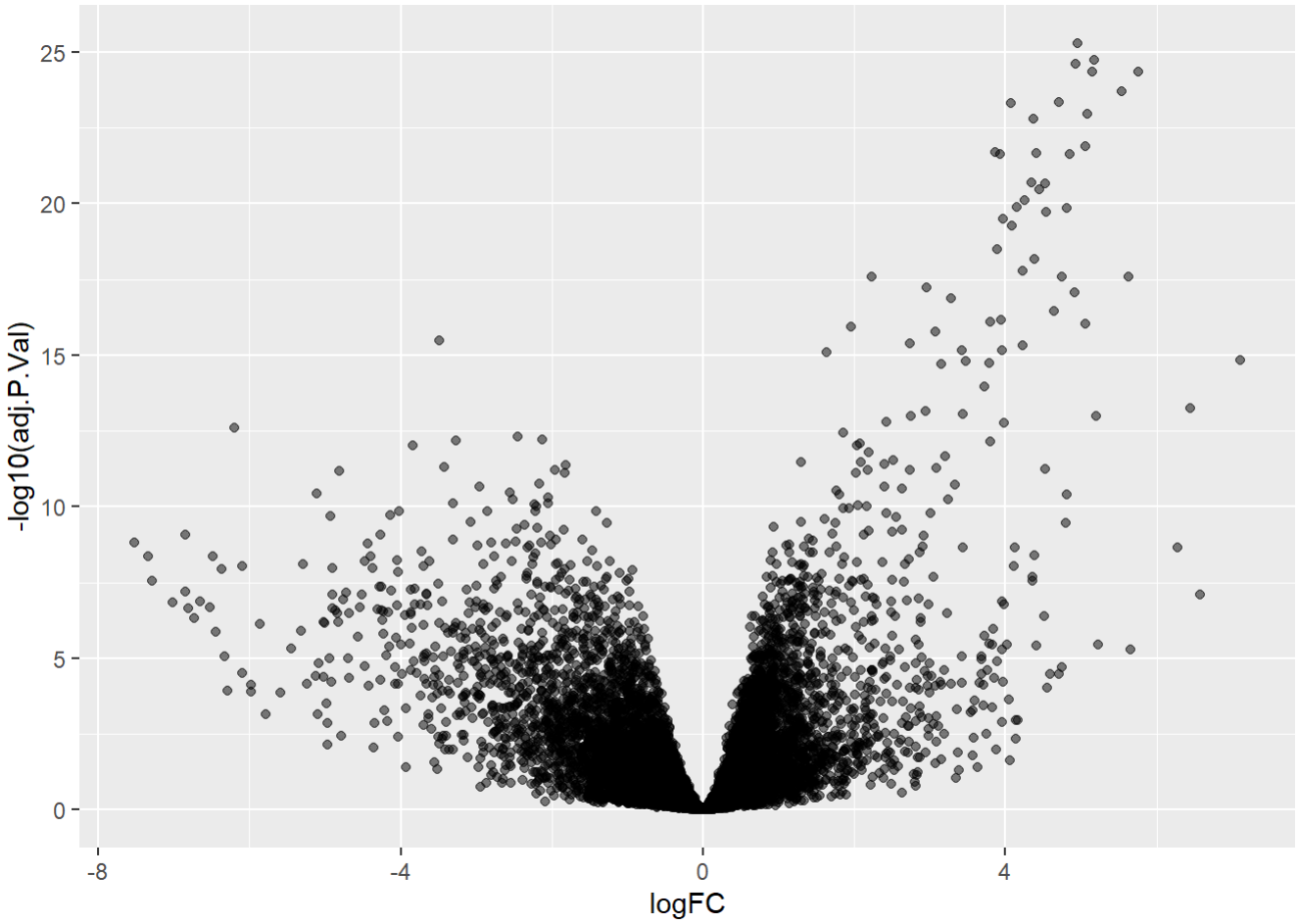


```
p2 <- p+ geom_vline(xintercept=c(-2, 2), linetype= "dashed") + geom_hline(yintercept= 2, linetyp
e= "dashed") +  xlim(c(-15, 15)) + ylim(c(0, 50))

p2
```

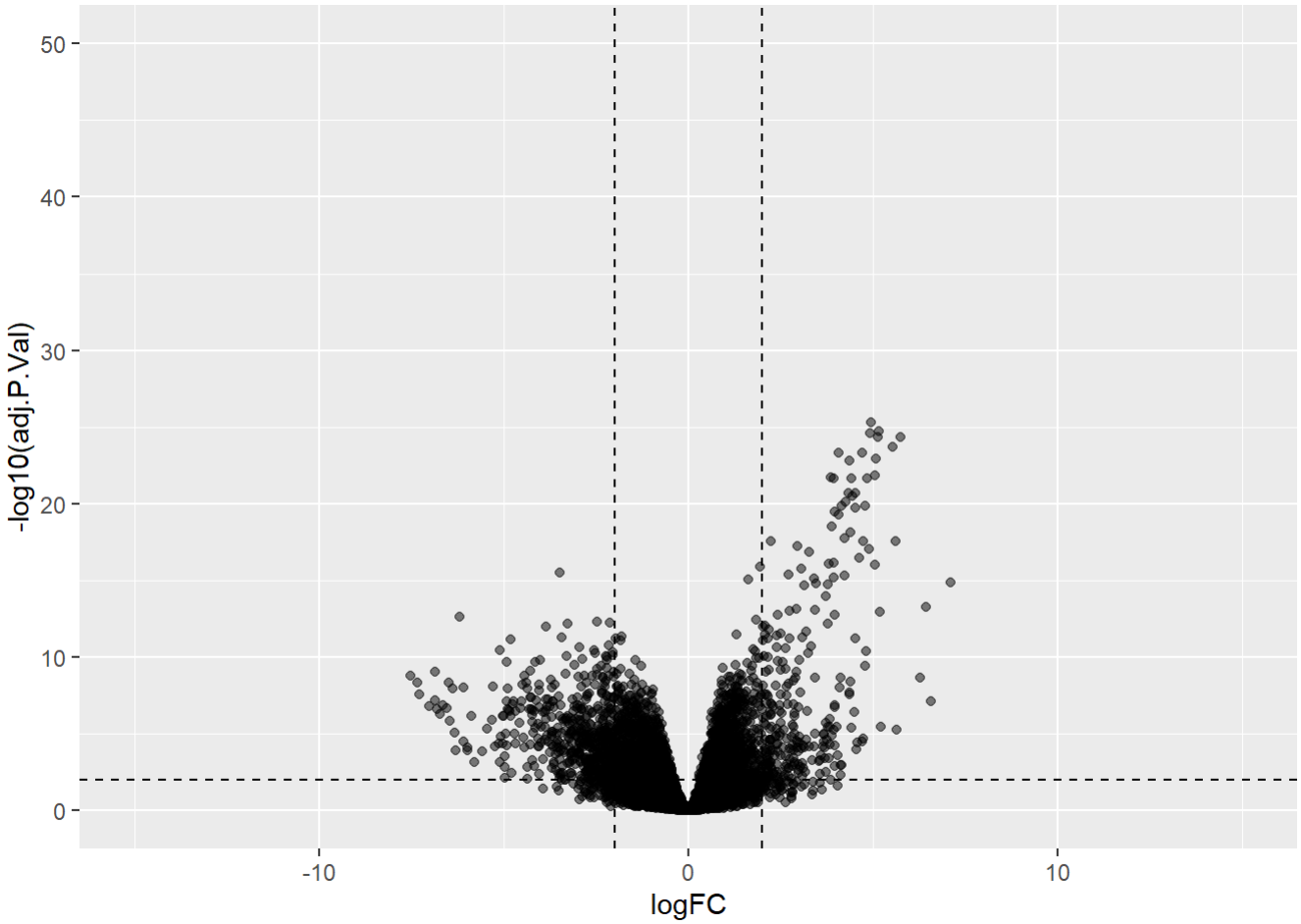


This plot uses the unfiltered Female HBV list and is plotting genes with a P value greater than or equal to 0.05 and with an absolute logFC of 2. The x and y coordinate plane limits are x (-15, 15) and y (0, 50). The dashed lines are significance thresholds and are placed a x= -2, x=2, and y=2.

```
df <- data.frame(F_HBV_unfiltered$adj.P.Val, F_HBV_unfiltered$logFC, F_HBV_unfiltered$chr, F_HBV
_unfiltered$GENEID, F_HBV_unfiltered$gene_name)
colnames(df) <- c("adj.P.Val", "logFC", "chr", "id", "name")
dfSig <- df[(abs(df$logFC) >= 2 & df$adj.P.Val <= 0.05),]$id
p <-ggplot(data=df, aes( x=logFC, y=-log10(adj.P.Val))) + geom_point(alpha=0.5)

p
```

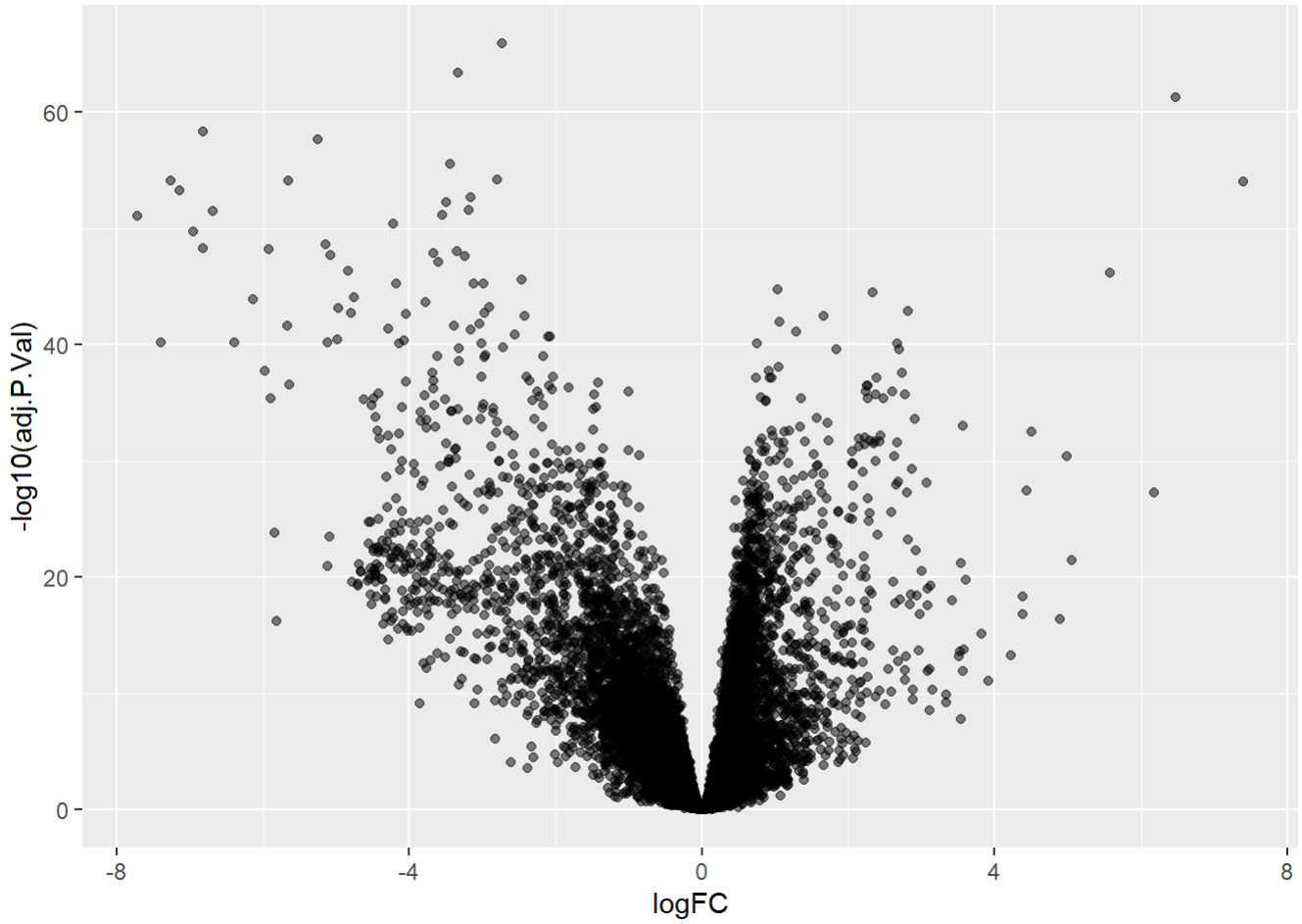



```
p2 <- p+ geom_vline(xintercept=c(-2, 2), linetype= "dashed") + geom_hline(yintercept= 2, linetyp
e= "dashed") +  xlim(c(-15, 15)) + ylim(c(0, 50))

p2
```



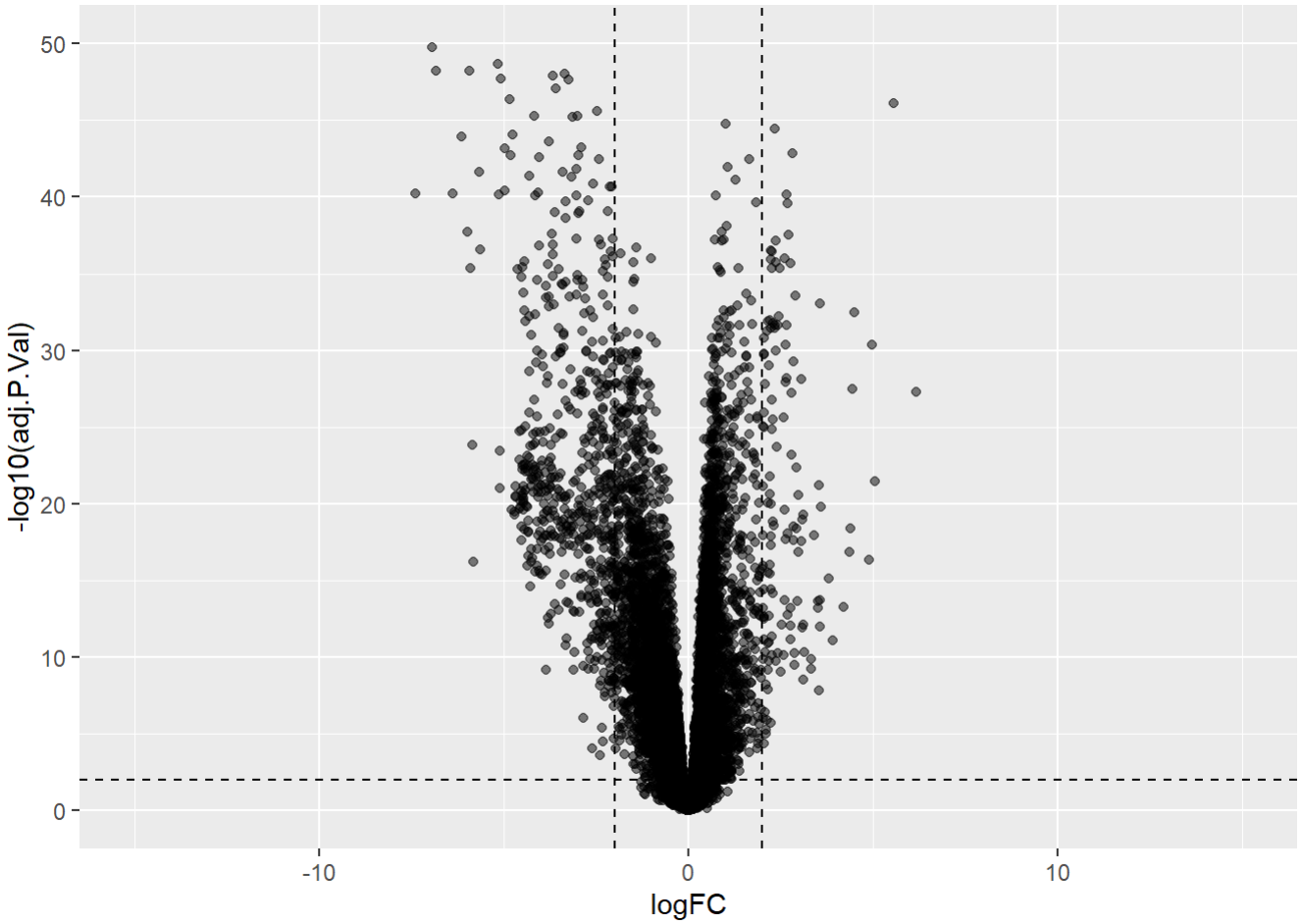
This plot uses the unfiltered Male HCV list and is plotting genes with a P value greater than or equal to 0.05 and with an absolute logFC of 2. The x and y coordinate plane limits are x (-15, 15) and y (0, 50). The dashed lines are significance thresholds and are placed a x= -2, x=2, and y=2.

```
df <- data.frame(M_HCV_unfiltered$adj.P.Val, M_HCV_unfiltered$logFC, M_HCV_unfiltered$chr, M_HCV
_unfiltered$GENEID, M_HCV_unfiltered$gene_name)
colnames(df) <- c("adj.P.Val", "logFC", "chr", "id", "name")
#dfSig <- df[(abs(df$logFC) >= 2 & df$adj.P.Val <= 0.05),]$id
p <-ggplot(data=df, aes( x=logFC, y=-log10(adj.P.Val))) + geom_point(alpha=0.5)

p
```



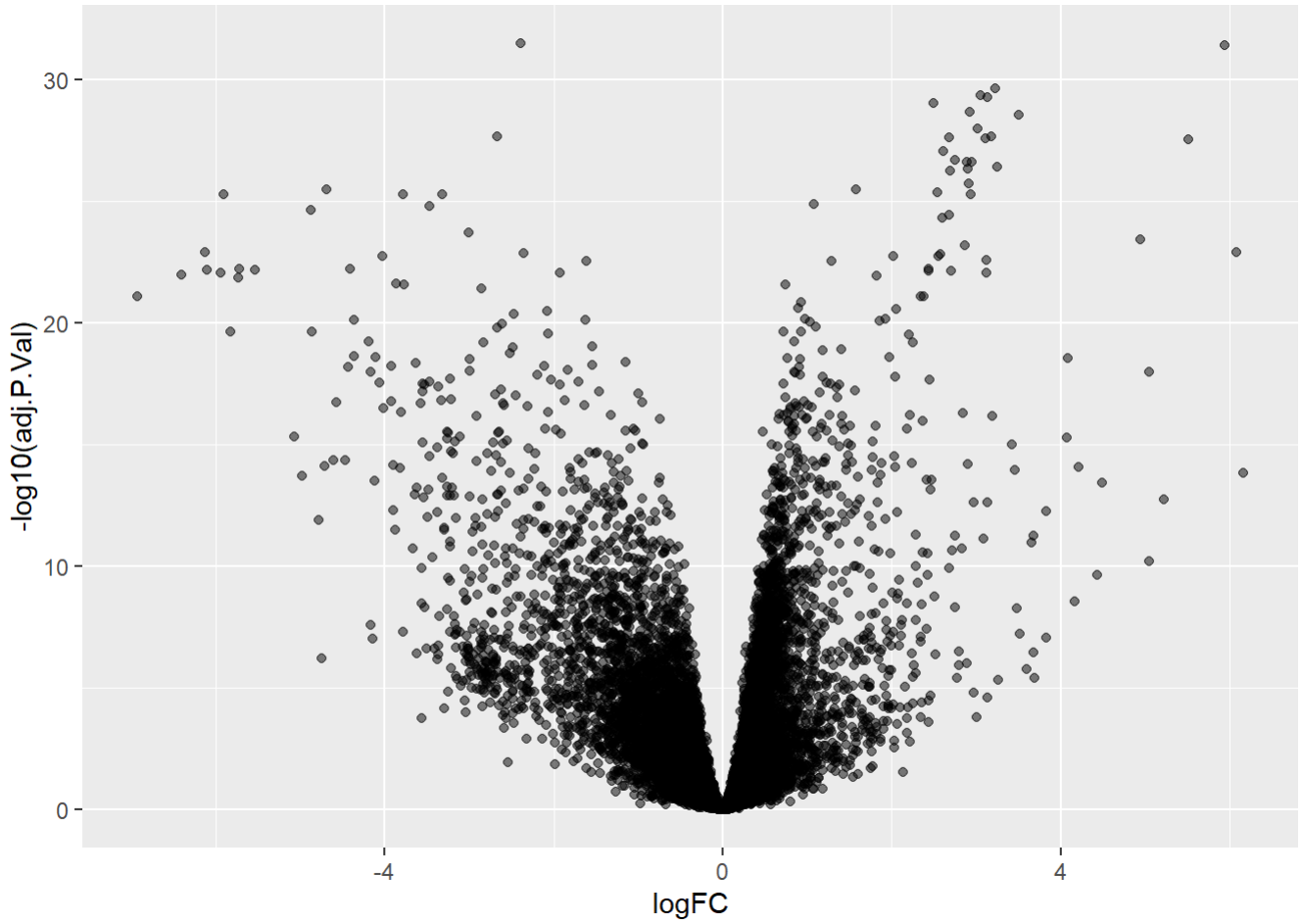

```
p2 <- p+ geom_vline(xintercept=c(-2, 2), linetype= "dashed") + geom_hline(yintercept= 2, linetyp
e= "dashed") +  xlim(c(-15, 15)) + ylim(c(0, 50))

p2
```

```
## Warning: Removed 18 rows containing missing values (geom_point).
```

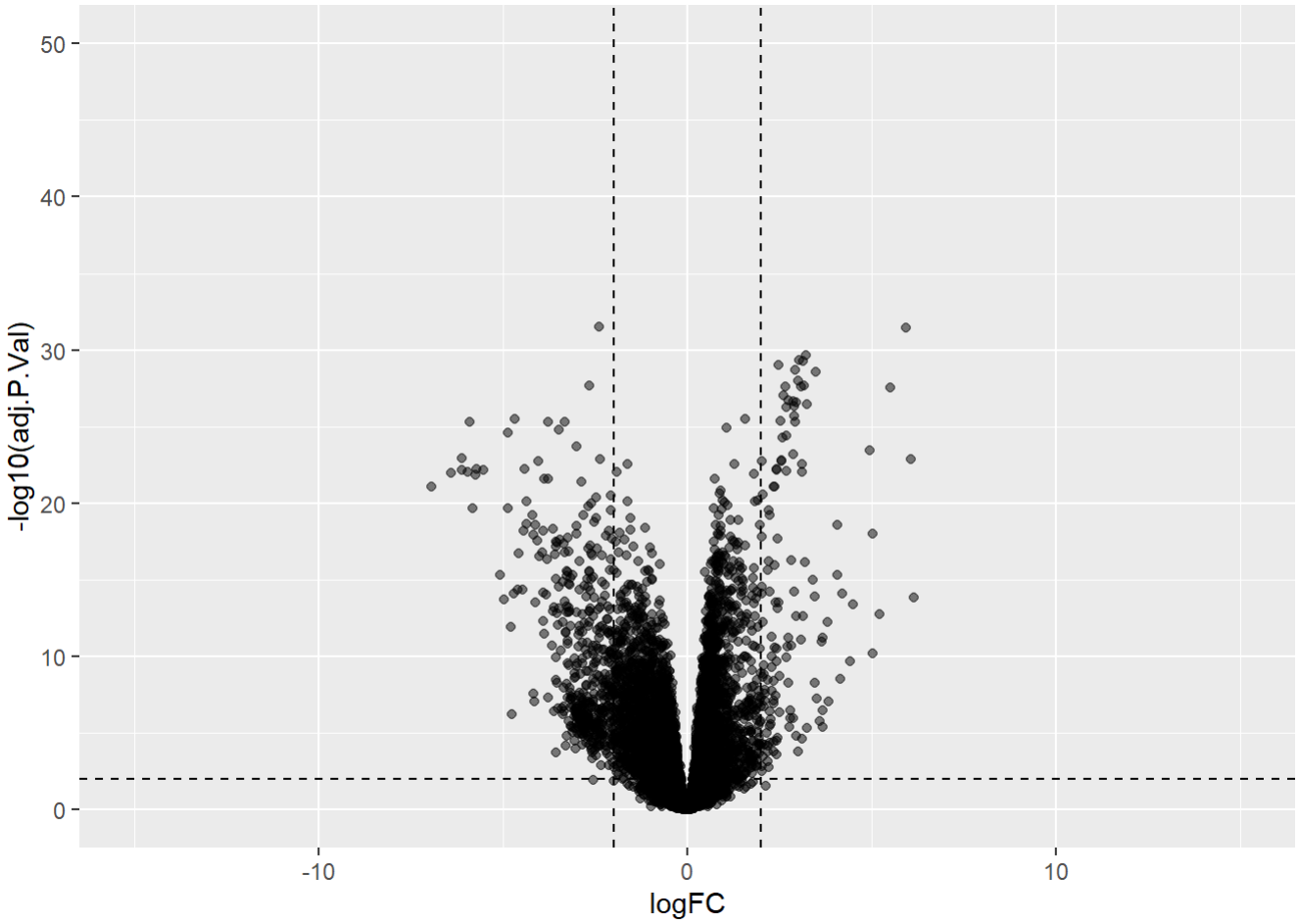


This plot uses the unfiltered Female HCV list and is plotting genes with a P value greater than or equal to 0.05 and with an absolute logFC of 2. The x and y coordinate plane limits are x (-15, 15) and y (0, 50). The dashed lines are significance thresholds and are placed a x= -2, x=2, and y=2.

```
df <- data.frame(F_HCV_unfiltered$adj.P.Val, F_HCV_unfiltered$logFC, F_HCV_unfiltered$chr, F_HCV
_unfiltered$GENEID, F_HCV_unfiltered$gene_name)
colnames(df) <- c("adj.P.Val", "logFC", "chr", "id", "name")
#dfSig <- df[(abs(df$logFC) >= 2 & df$adj.P.Val <= 0.05),]$id
p <-ggplot(data=df, aes( x=logFC, y=-log10(adj.P.Val))) + geom_point(alpha=0.5)

p
```

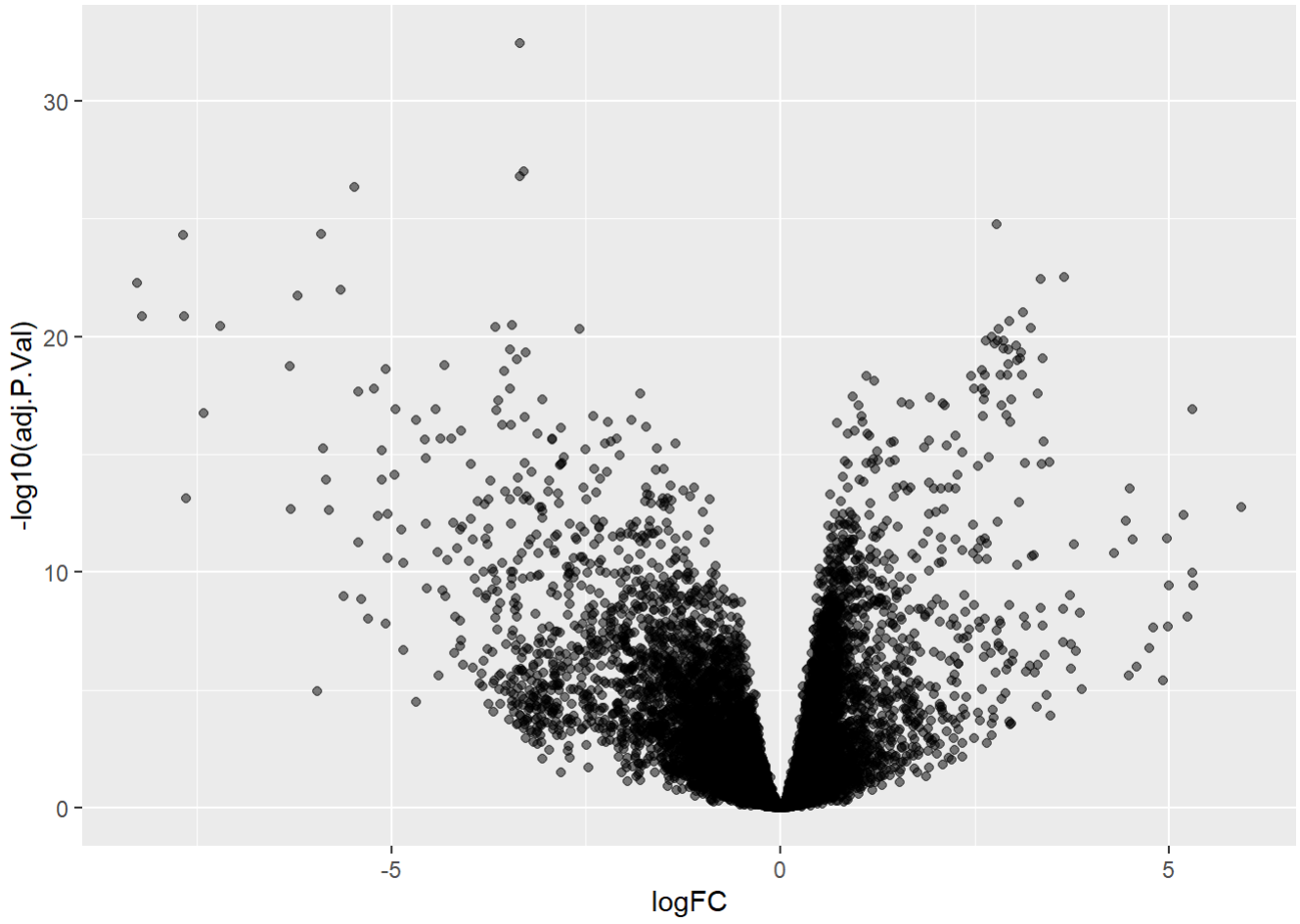



```
p2 <- p+ geom_vline(xintercept=c(-2, 2), linetype= "dashed") + geom_hline(yintercept= 2, linetyp
e= "dashed") +  xlim(c(-15, 15)) + ylim(c(0, 50))

p2
```

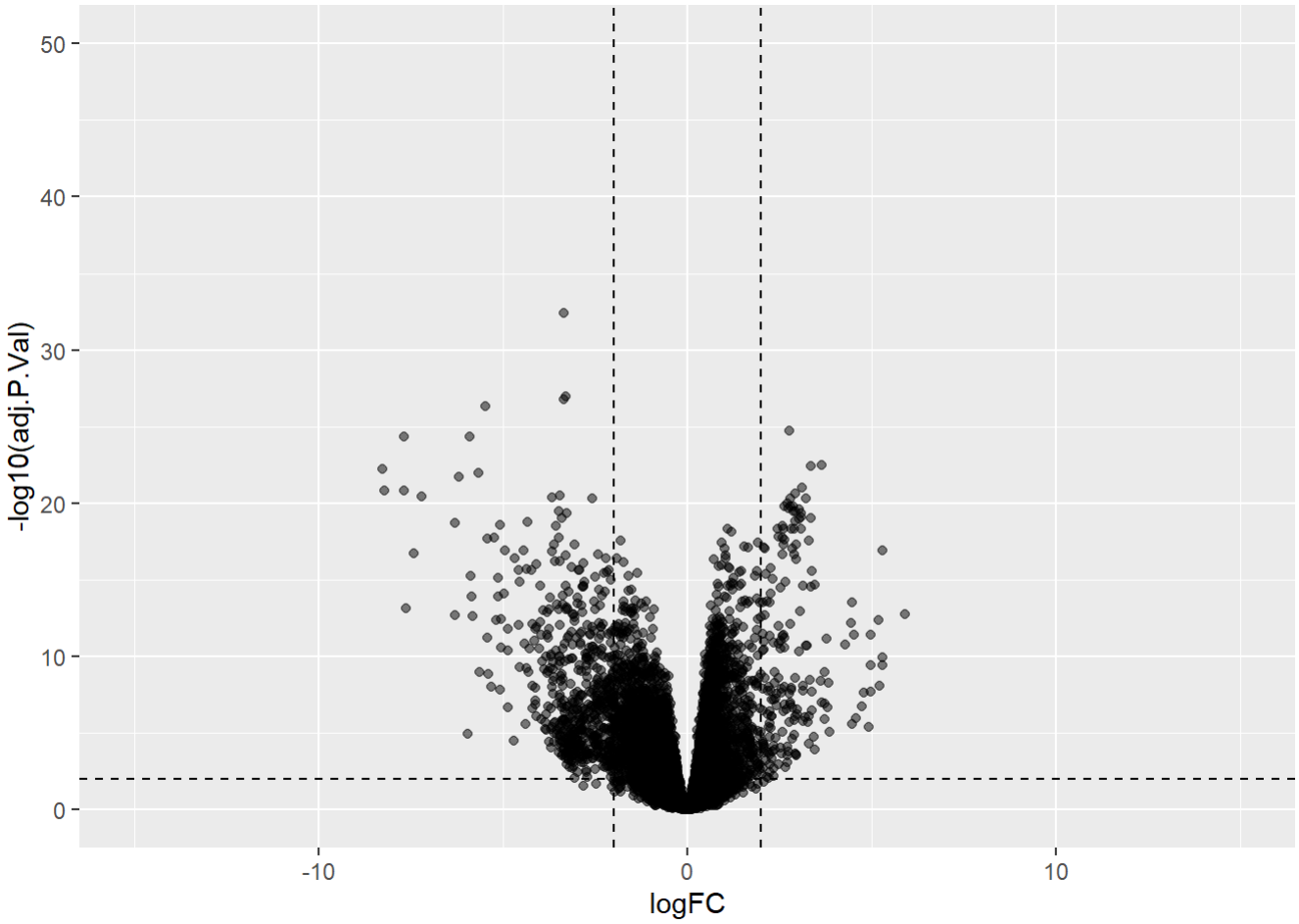


This plot uses the unfiltered Male Neither list and is plotting genes with a P value greater than or equal to 0.05 and with an absolute logFC of 2. The x and y coordinate plane limits are x (-15, 15) and y (0, 50). The dashed lines are significance thresholds and are placed a x= -2, x=2, and y=2.

```
df <- data.frame(M_Neither_unfiltered$adj.P.Val, M_Neither_unfiltered$logFC, M_Neither_unfiltere
d$chr, M_Neither_unfiltered$GENEID, M_Neither_unfiltered$gene_name)
colnames(df) <- c("adj.P.Val", "logFC", "chr", "id", "name")
#dfSig <- df[(abs(df$logFC) >= 2 & df$adj.P.Val <= 0.05),]$id
p <-ggplot(data=df, aes( x=logFC, y=-log10(adj.P.Val))) + geom_point(alpha=0.5)

p
```

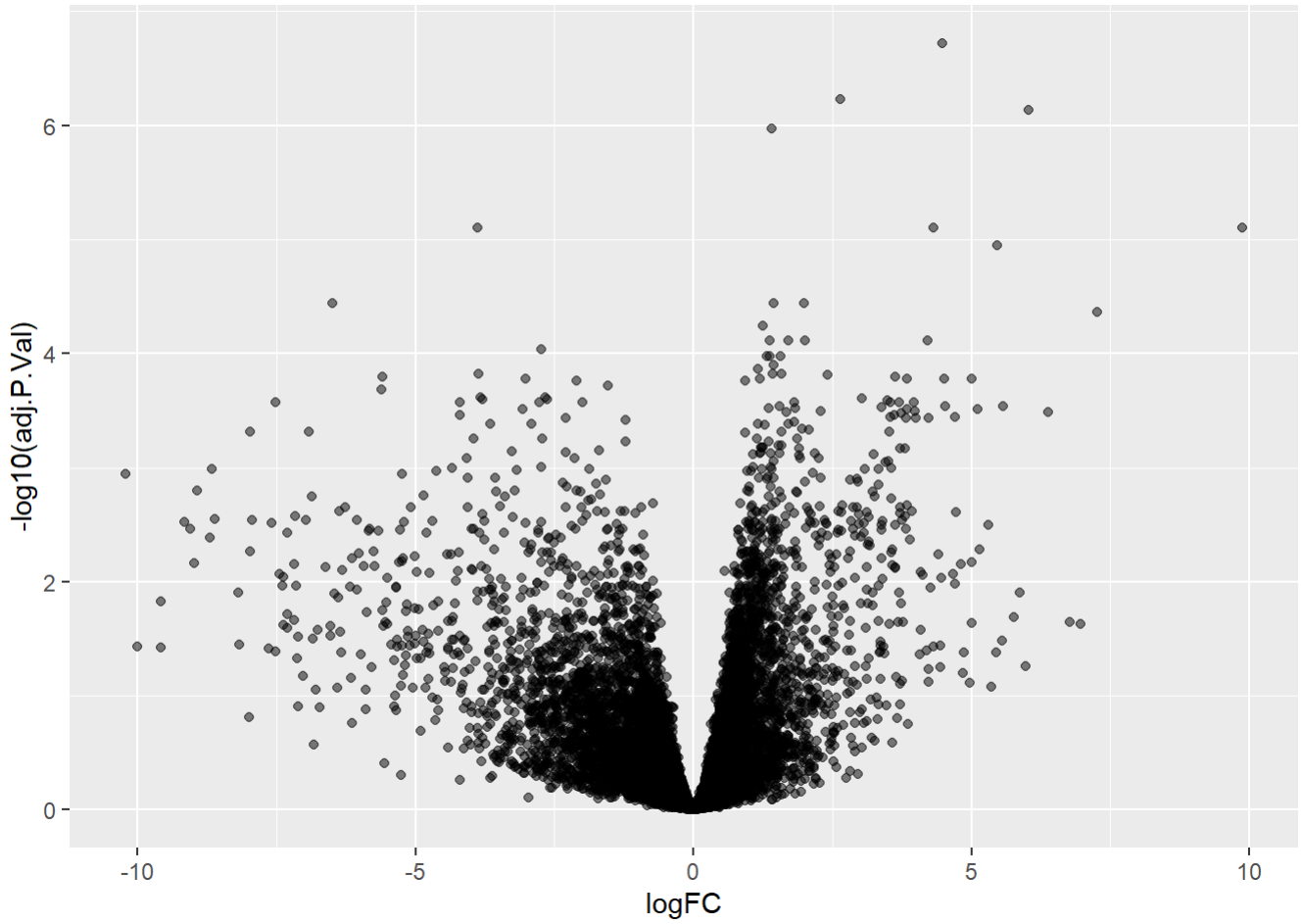



```
p2 <- p+ geom_vline(xintercept=c(-2, 2), linetype= "dashed") + geom_hline(yintercept= 2, linetyp
e= "dashed") +  xlim(c(-15, 15)) + ylim(c(0, 50))

p2
```

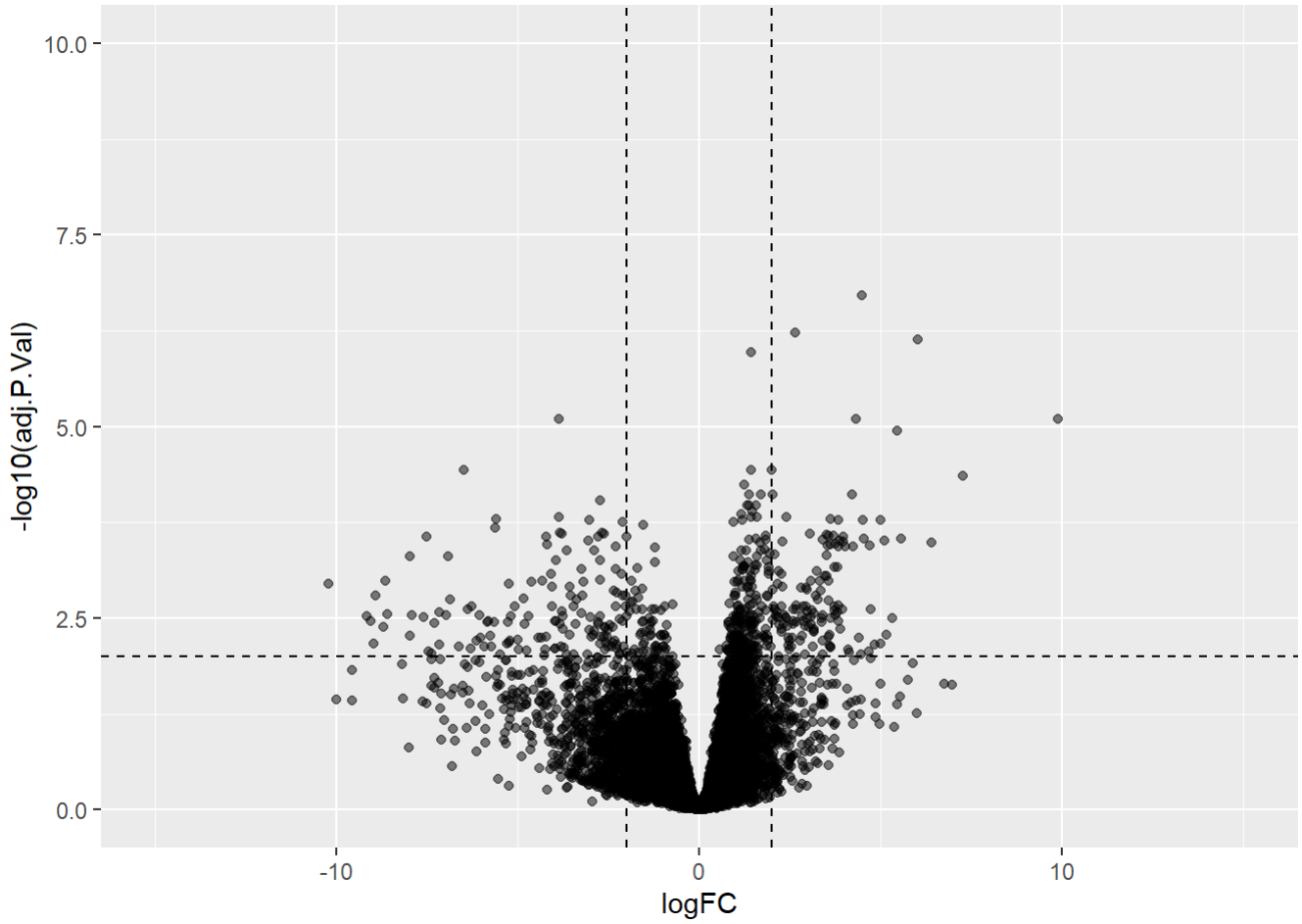


This plot uses the unfiltered Female Neither list and is plotting genes with a P value greater than or equal to 0.05 and with an absolute logFC of 2. The x and y coordinate plane limits are x (-15, 15) and y (0, 50). The dashed lines are significance thresholds and are placed a x= -2, x=2, and y=2.

```
df <- data.frame(F_Neither_unfiltered$adj.P.Val, F_Neither_unfiltered$logFC, F_Neither_unfiltere
d$chr, F_Neither_unfiltered$GENEID, F_Neither_unfiltered$gene_name)
colnames(df) <- c("adj.P.Val", "logFC", "chr", "id", "name")
#dfSig <- df[(abs(df$logFC) >= 2 & df$adj.P.Val <= 0.05),]$id
p <-ggplot(data=df, aes( x=logFC, y=-log10(adj.P.Val))) + geom_point(alpha=0.5)

p
```

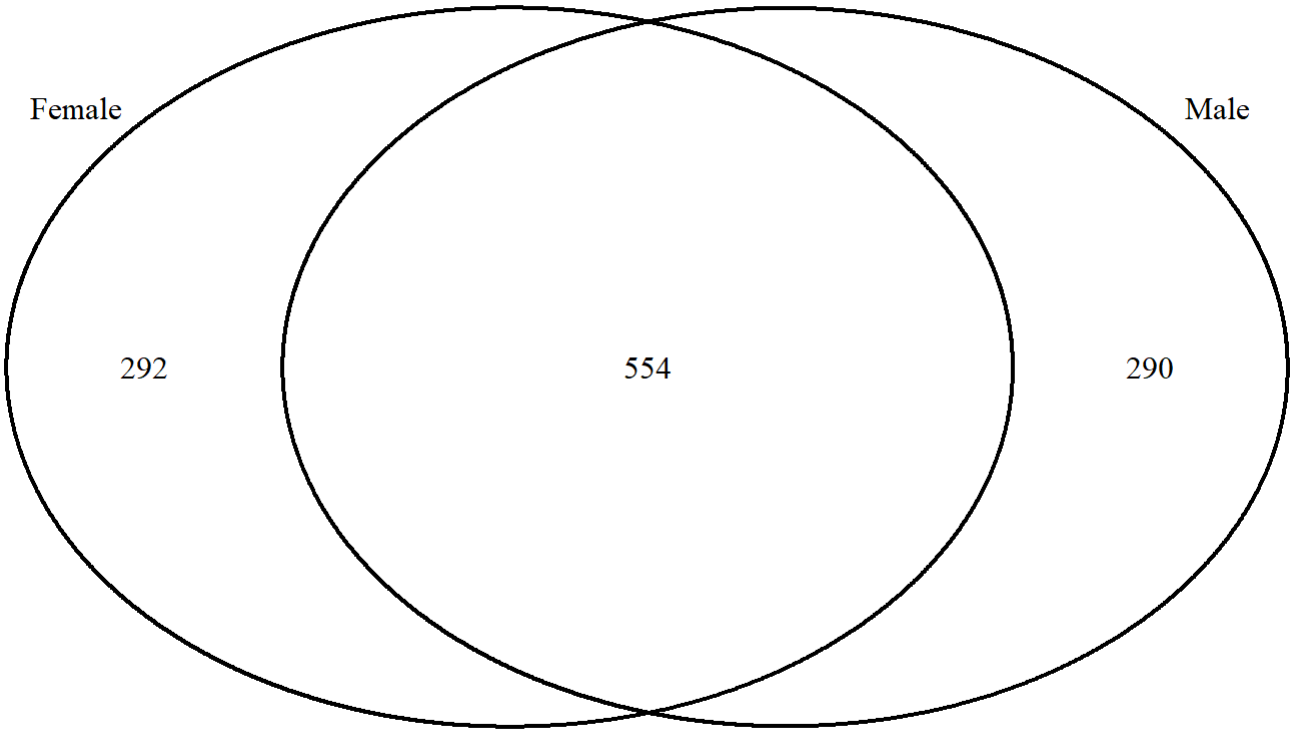



```
p2 <- p+ geom_vline(xintercept=c(-2, 2), linetype= "dashed") + geom_hline(yintercept= 2, linetyp
e= "dashed") +  xlim(c(-15, 15)) + ylim(c(0, 10))

p2
```

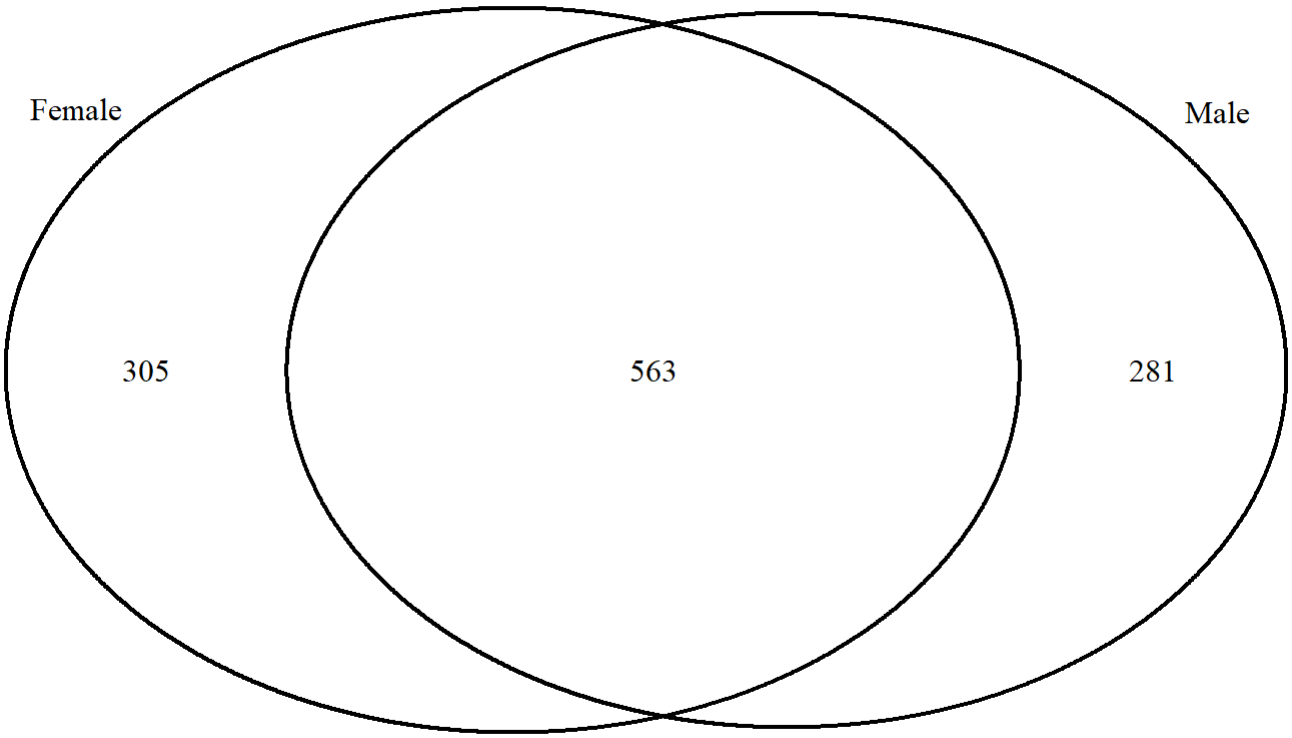


Using the filtered gene list p value 0.05 of males and females separated by etiology I am comparing the amount of genes that coincide using a VennDiagram. I am also using a filtered female gene list with the p value relaxed to 0.1 and comparing to the filtered male gene list. These two Venn Diagrams show if any genes one the female list are excluded do to sample size.

```
library(VennDiagram)
library(grDevices)

venn3<- venn.diagram(List("Female"=DEGs_F_HBV$gene_name, "Male"=DEGs_M_HBV$gene_name),filename = NULL)
grid.newpage()

grid.draw(venn3)
```

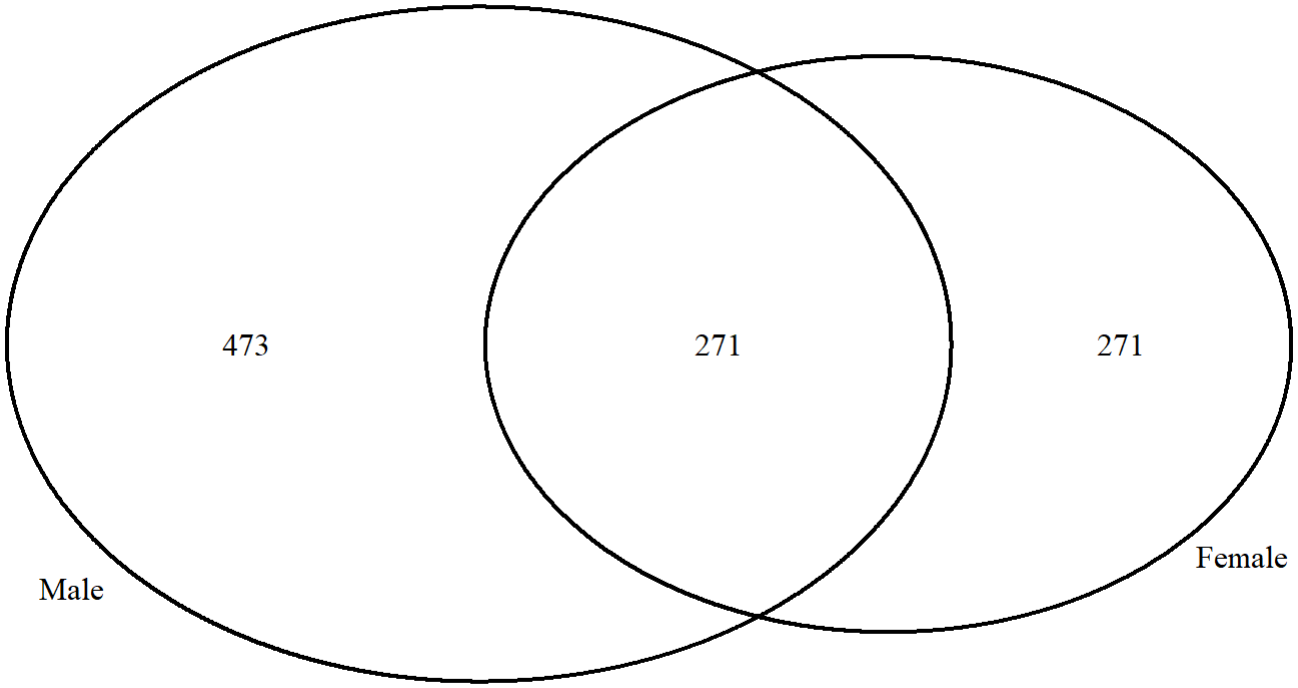



```
#pdf(file="venn1.pdf")

venn4<-venn.diagram(List("Female"=DEGs_F_HBV_relax_p$gene_name, "Male"=DEGs_M_HBV$gene_name),  f
ilename = NULL)
grid.newpage()
grid.draw(venn4)
```

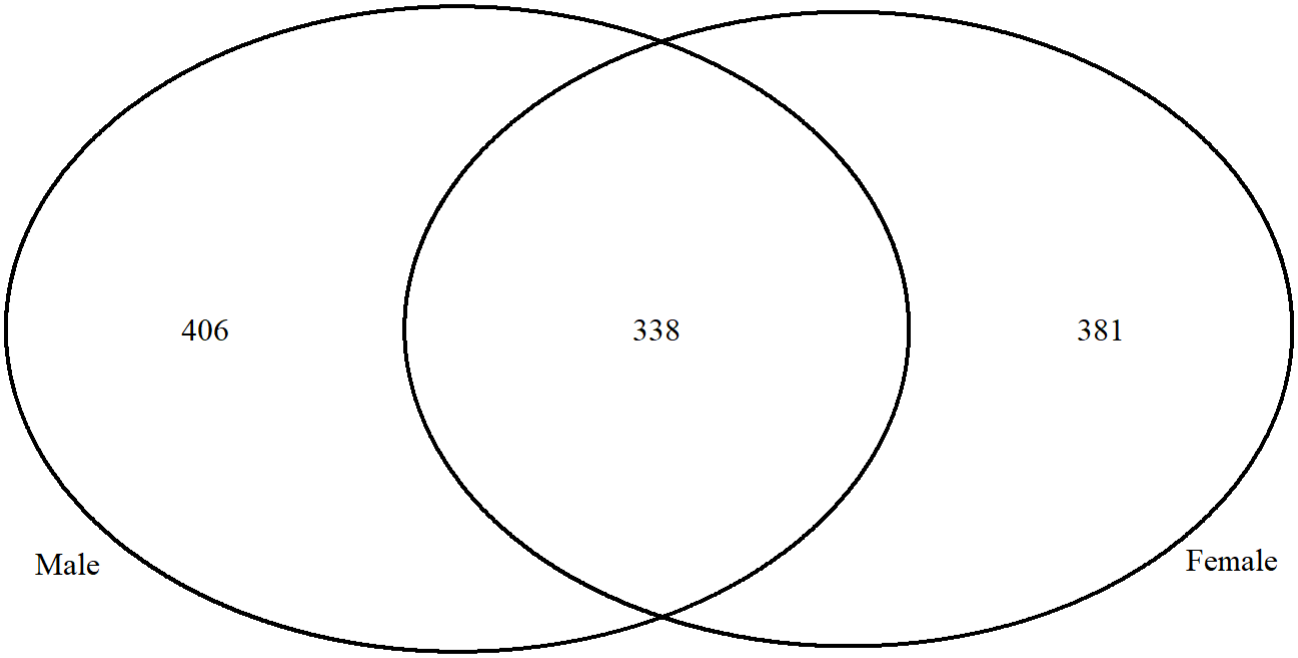



```
#pdf(file="venn2.pdf")
```

```
library(VennDiagram)
library(grDevices)

venn5<- venn.diagram(List("Female"=DEGs_F_Neither$gene_name, "Male"=DEGs_M_Neither$gene_name),fi
lename = NULL)
grid.newpage()

grid.draw(venn5)
```

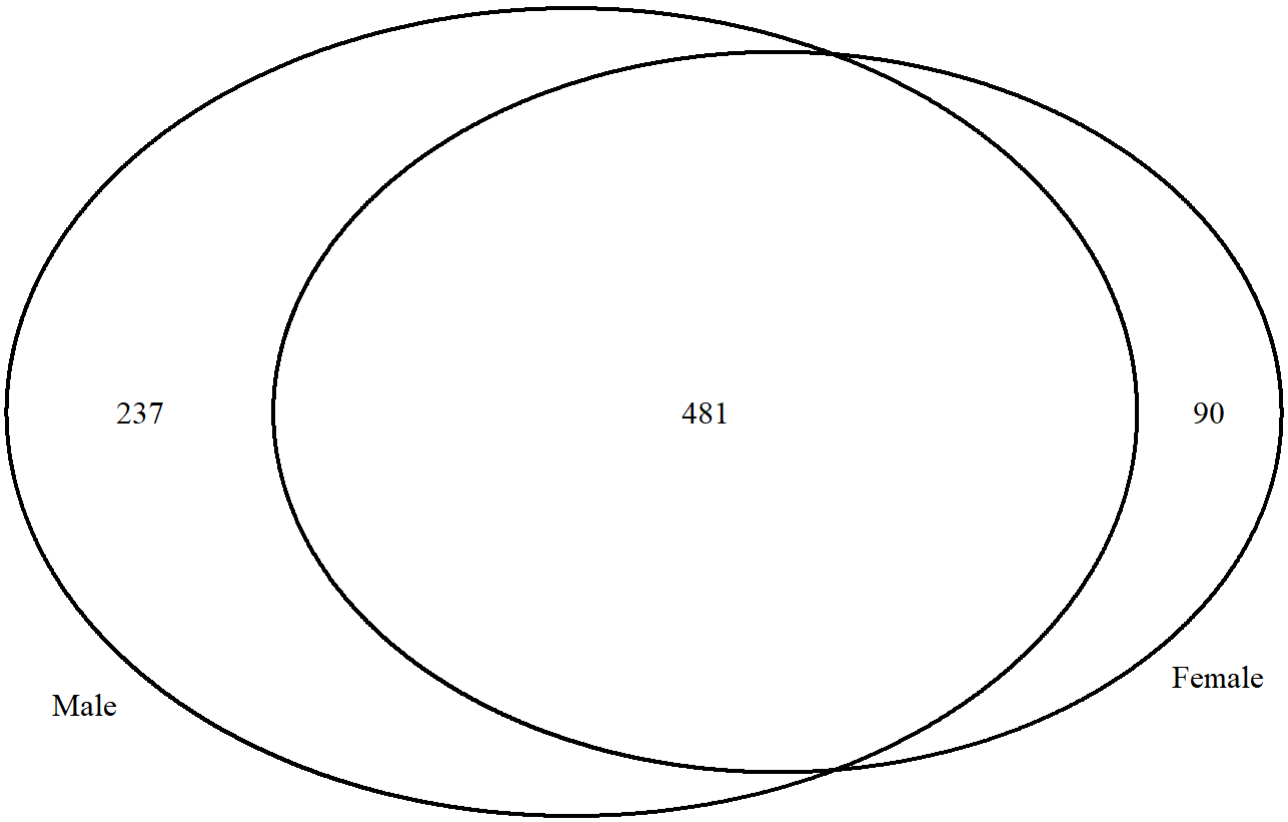



```
#pdf(file="venn1.pdf")

venn6<-venn.diagram(List("Female"=DEGs_F_Neither_relax_p$gene_name, "Male"=DEGs_M_Neither$gene_n
ame),  filename = NULL)
grid.newpage()
grid.draw(venn6)
```

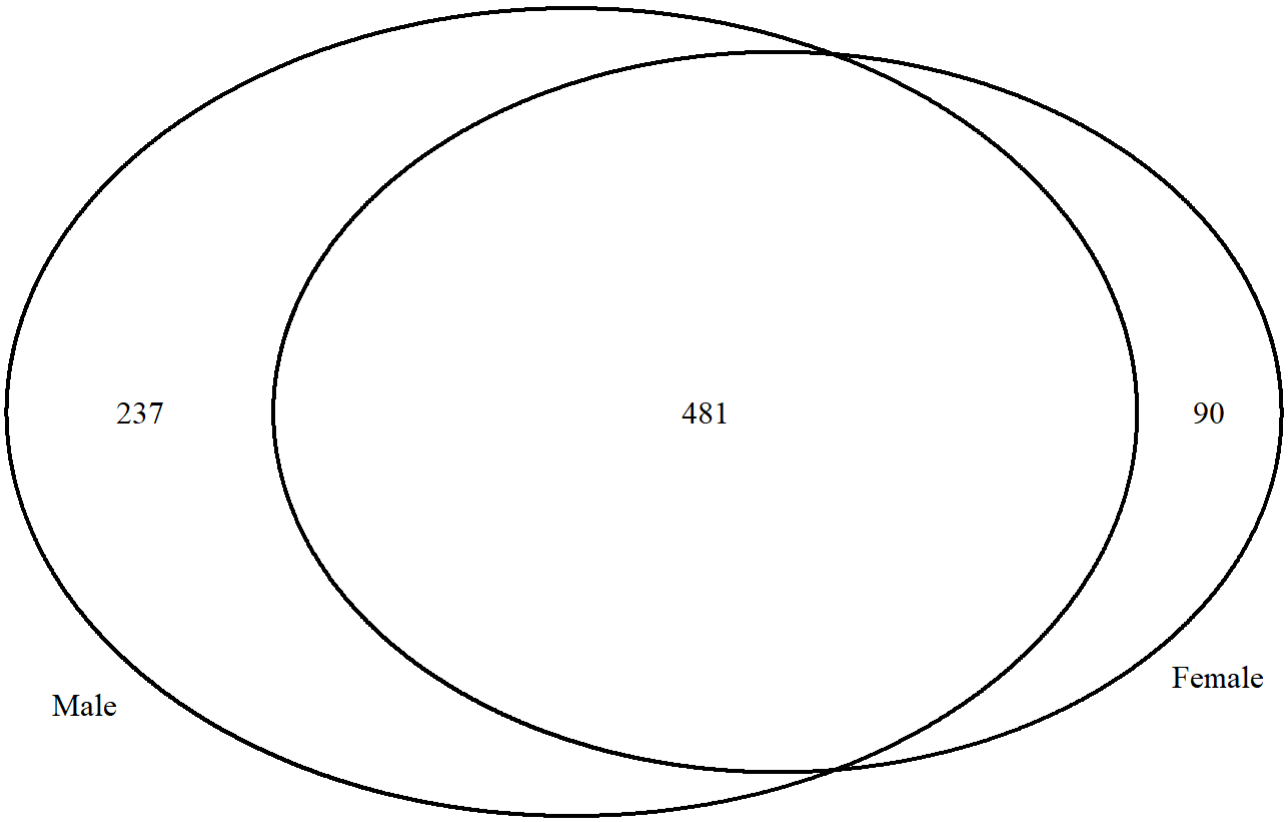



```
#pdf(file="venn2.pdf")
```

```
library(VennDiagram)
library(grDevices)

venn7<- venn.diagram(List("Female"=DEGs_F_HCV$gene_name, "Male"=DEGs_M_HCV$gene_name),filename = NULL)
grid.newpage()

grid.draw(venn7)
```




```
#pdf(file="venn1.pdf")

venn8<-venn.diagram(List("Female"=DEGs_F_HCV_relax_p$gene_name, "Male"=DEGs_M_HCV$gene_name),  f
ilename = NULL)
grid.newpage()
grid.draw(venn8)
```

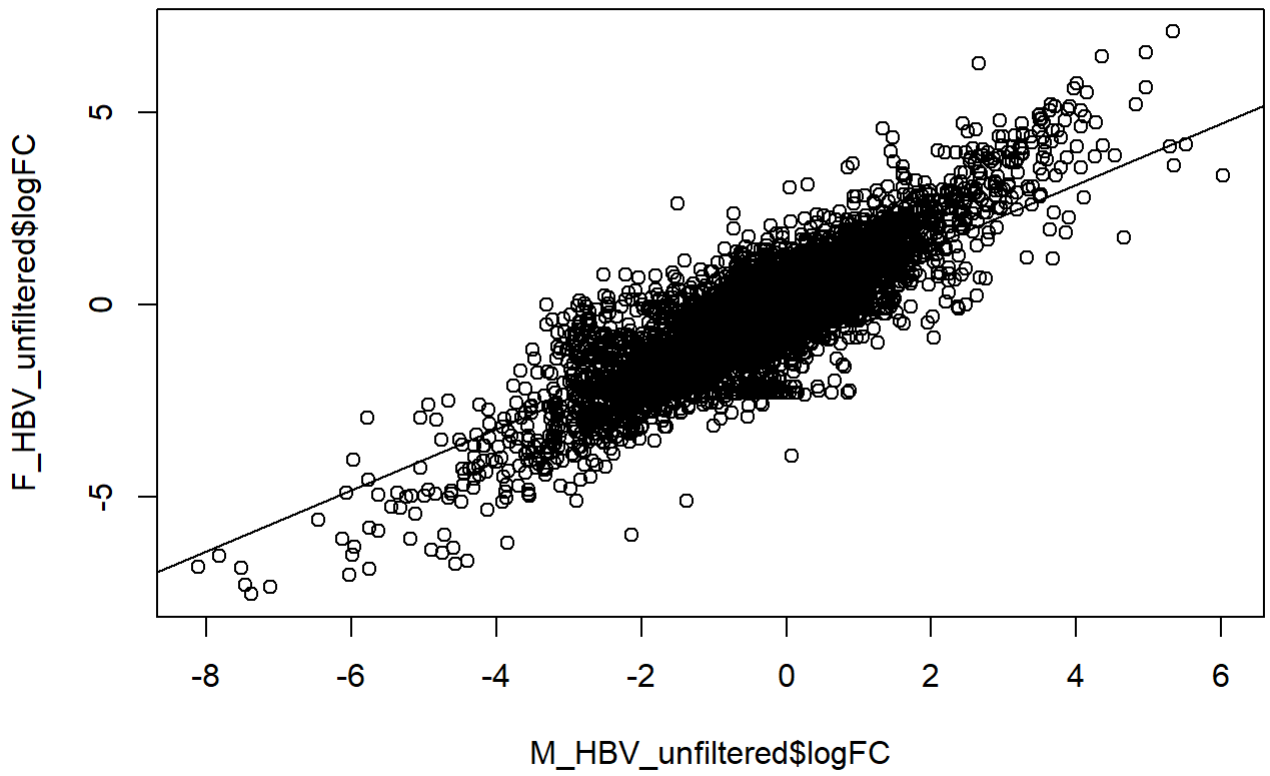



```
#pdf(file="venn2.pdf")
```

```
#ggplot(mu_diff, aes(x, y)) +        # ggplot2 plot with confidence intervals
 # geom_point() +
  #geom_errorbar(aes(ymin = lower, ymax = upper))

#matching the male gene list to the female gene list and making sure they are the same size and
 plotting the log FC
#should see a linear trend
F_HBV_unfiltered <- F_HBV_unfiltered[match( M_HBV_unfiltered$gene_name, F_HBV_unfiltered$gene_na
me), ]
identical(M_HBV_unfiltered$gene_name, F_HBV_unfiltered$gene_name)
```

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

```
plot(M_HBV_unfiltered$logFC, F_HBV_unfiltered$logFC)

# Calculate and display the regression line
regression <- lm(M_HBV_unfiltered$logFC~F_HBV_unfiltered$logFC)
abline(regression)
```




```
#Show regression formula
print(regression)
```

```
##
## Call:
## lm(formula = M_HBV_unfiltered$logFC ~ F_HBV_unfiltered$logFC)
##
## Coefficients:
##            (Intercept)  F_HBV_unfiltered$logFC
##               -0.04601                 0.79507
```

```
#Create data frame for sample data
#Sample data is data frame with columns of male logFC, male gene names, male p values, female lo
g FC, female gene names, female p values respectively
sample_data <- data.frame(M_HBV_unfiltered$logFC, M_HBV_unfiltered$gene_name, M_HBV_unfiltered$a
dj.P.Val, F_HBV_unfiltered$logFC, F_HBV_unfiltered$gene_name, F_HBV_unfiltered$adj.P.Val)

#Calculate residuals
sample_data$residuals <- residuals(regression)

#Threshold of 0.5
outlier_threshold <- 0.5

#Print only name of outliers
outlier <- sample_data[which(abs(sample_data$residuals) > 0.5), ]
is.numeric(sample_data$residuals)
```

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

```
#geneids <- data.frame(outlier$M_HBV_unfiltered.GENEID, outlier$F_HBV_unfiltered.GENEID)

#write.csv(geneids, "geneids.csv")
```

```
#creating data frame for data in ggplot and renaming columns for ggplot
df <- sample_data
colnames(df) <- c("M_logFC", "M_name", "M_adj.P.Val", "F_logFC", "F_name", "F_adj.P.Val","residu
als")

#limiting the genes in the dataset to be significant with a p value of 0.05
df<-df[which(df$M_adj.P.Val<0.05 | df$F_adj.P.Val <0.05), ]
ggp<- ggplot(df, mapping=aes(x=M_logFC, y= F_logFC)) + geom_point()

#getting coefficients from regression function to create line for ggplot
regression <- lm(df$F_logFC~df$M_logFC)
regression
```

```
##
## Call:
## lm(formula = df$F_logFC ~ df$M_logFC)
##
## Coefficients:
## (Intercept)   df$M_logFC
##     0.02019      0.93518
```

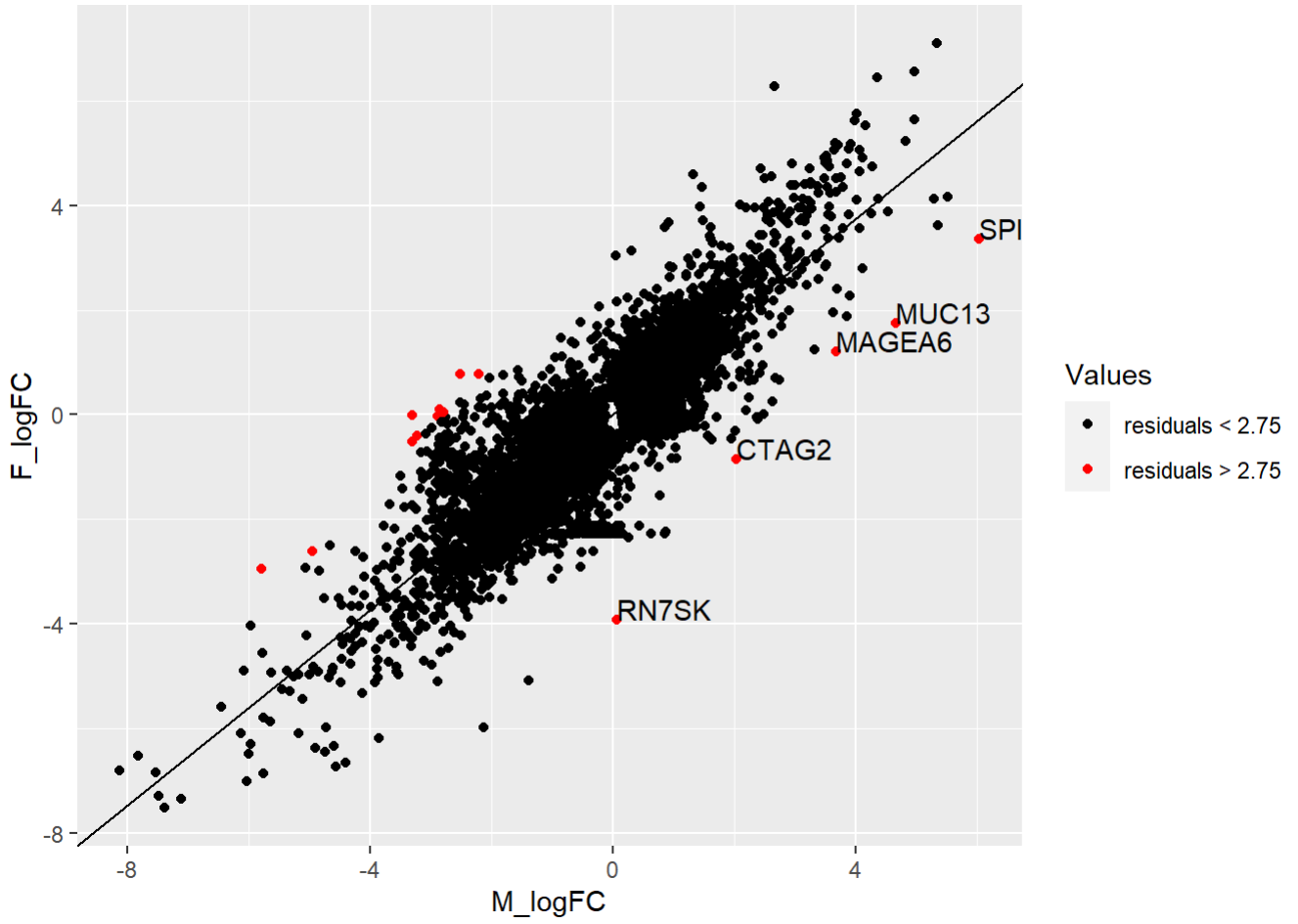
```
coeff<-coefficients(regression)
coeff
```

```
## (Intercept)  df$M_logFC
##  0.02018953  0.93517772
```

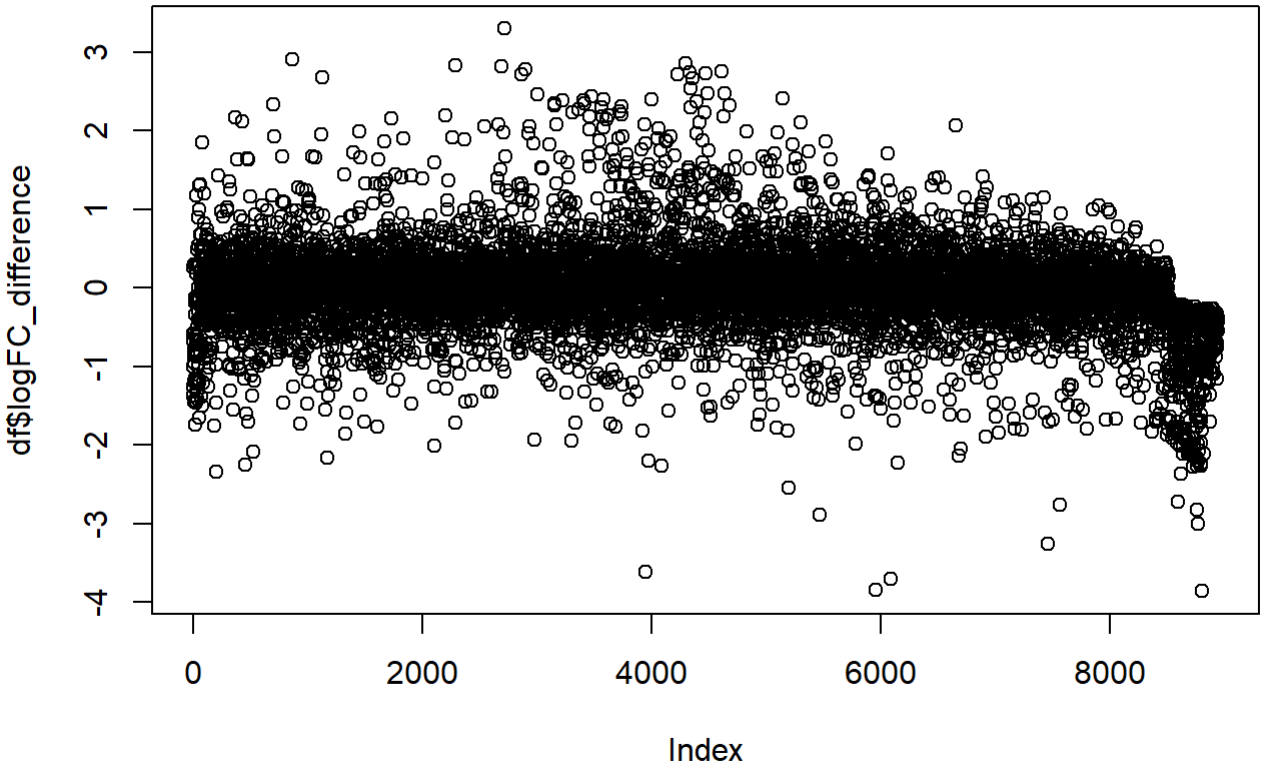
```
intercept<-coeff[1]
slope<-coeff[2]

#plotting with regression line

ggp2 <- ggp+geom_abline(intercept=intercept, slope=slope)


#plotting with color scheme. If residuals is >2.75, then the point will be colored red.

ggp3<- ggp2 +geom_point(aes(color=ifelse( abs(residuals) >2.75, "red", "black"))) +scale_color_m
anual(labels=c("residuals < 2.75", "residuals > 2.75"), values= c("black", "red")) +labs(color=
"Values")

#plotting with gene label if the residual is >2.75, then the point will be labeled with the gene
name.
ggp4 <- ggp3 + geom_text(aes(label=ifelse(residuals>2.75, as.character(M_name), '')), hjust=0, v
just=0)
pdf("~/R/effect_size_ggplot_HBV.pdf", width=12, height=12)
ggp4
dev.off()
```

```
## png
##   2
```

```
ggp4
```




```
#plot with the log fold change difference of male log FC subtracted from female log FC
df$logFC_difference <- abs(df$M_logFC)-abs(df$F_logFC)
plot(df$logFC_difference)
```




```
#logFC_difference <- data.frame(df$logFC_difference)
#colnames(logFC_difference)<-c("logFCdifference")
#logFC_difference
#ggp1<- ggplot(logFC_difference, )
```

```
#printing gene with logFC greater than 2
gene_names_M_HBV <- sample_data[which(abs(sample_data$M_HBV_unfiltered.logFC) > 2), ]
gene_names_F_HBV <- sample_data[which(abs(sample_data$F_HBV_unfiltered.logFC) > 2), ]

#printing out number of genes in list first output is the number of genes

dim(gene_names_M_HBV)
```

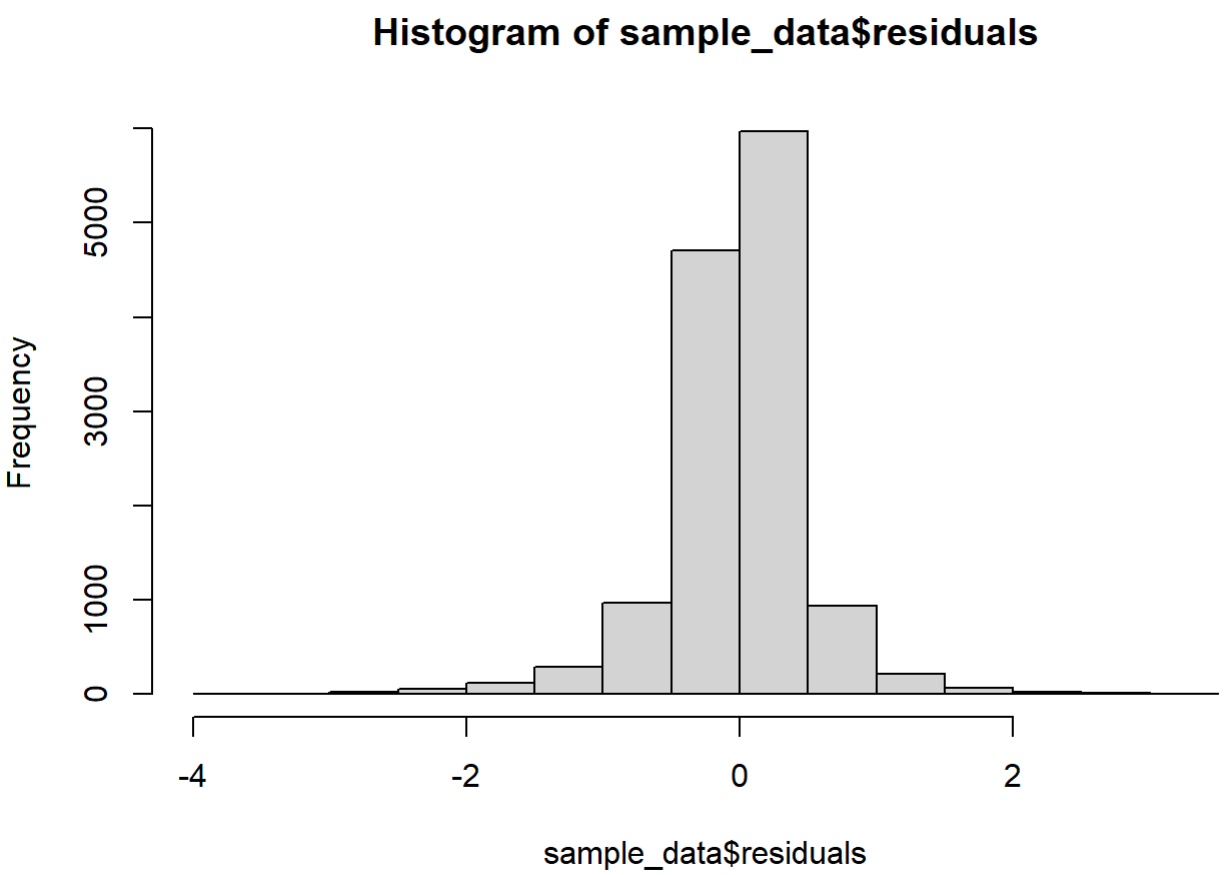
```
## [1] 844   7
```

```
dim(gene_names_F_HBV)
```

```
## [1] 968   7
```

```
write.csv(gene_names_M_HBV, "Regression_gene_lists_M_HBV.csv")
write.csv(gene_names_F_HBV, "Regression_gene_lists_F_HBV.csv")
```

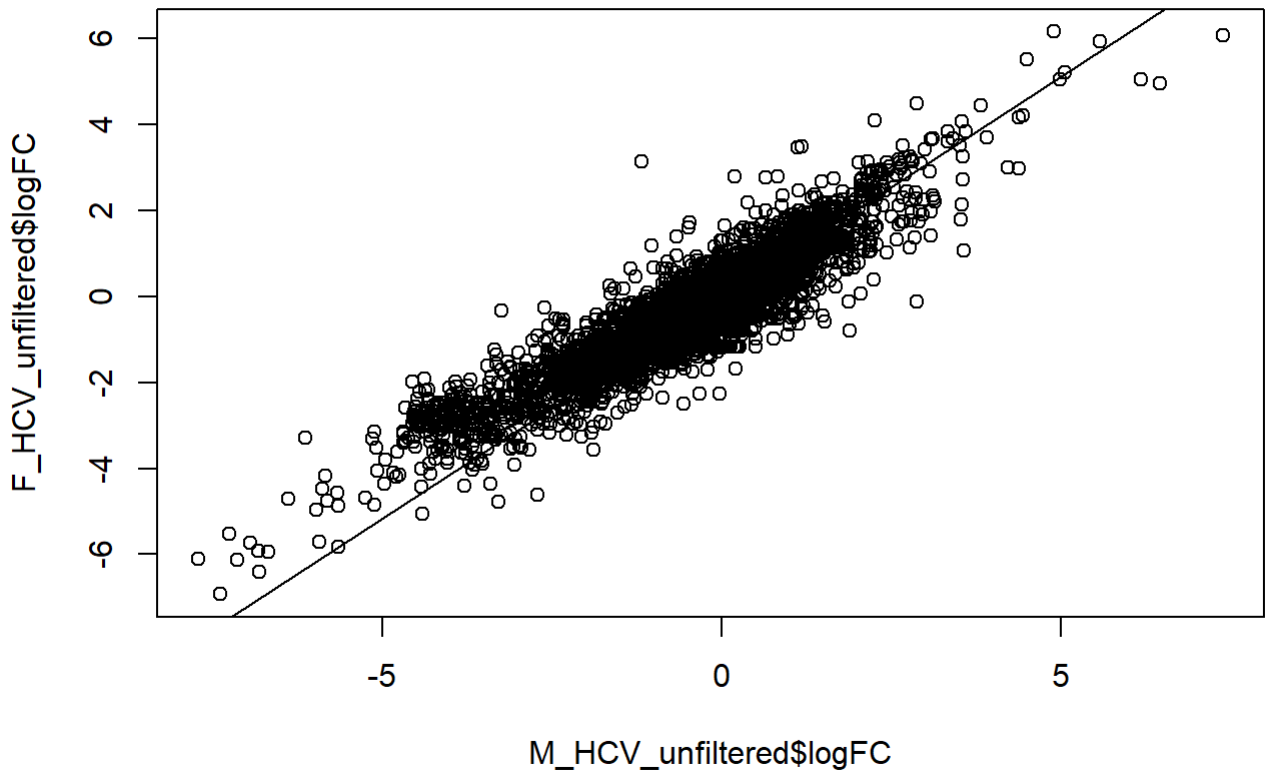
```
#histograms of residuals
#should expect most of genes to be around 0
hist(sample_data$residuals)
```




```
#ggplot(mu_diff, aes(x, y)) +       # ggplot2 plot with confidence intervals
 # geom_point() +
  #geom_errorbar(aes(ymin = lower, ymax = upper))

#matching the male gene list to the female gene list and making sure they are the same size and
 plotting the log FC
#should see a linear trend
F_HCV_unfiltered <- F_HCV_unfiltered[match( M_HCV_unfiltered$gene_name, F_HCV_unfiltered$gene_na
me), ]
identical(M_HCV_unfiltered$gene_name, F_HCV_unfiltered$gene_name)
```

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

```
plot(M_HCV_unfiltered$logFC, F_HCV_unfiltered$logFC)

# Calculate and display the regression line
regression <- lm(M_HCV_unfiltered$logFC~F_HCV_unfiltered$logFC)
abline(regression)
```




```
#Show regression formula
print(regression)
```

```
##
## Call:
## lm(formula = M_HCV_unfiltered$logFC ~ F_HCV_unfiltered$logFC)
##
## Coefficients:
##            (Intercept)  F_HCV_unfiltered$logFC
##                -0.02488                 1.02806
```

```
#Create data frame for sample data
#Sample data is data frame with columns of male logFC, male gene names, male p values, female lo
g FC, female gene names, female p values respectively
sample_data <- data.frame(M_HCV_unfiltered$logFC, M_HCV_unfiltered$gene_name, M_HCV_unfiltered$a
dj.P.Val, F_HCV_unfiltered$logFC, F_HCV_unfiltered$gene_name, F_HCV_unfiltered$adj.P.Val)

#Calculate residuals
sample_data$residuals <- residuals(regression)

#Threshold of 0.5
outlier_threshold <- 0.5

#Print only name of outliers
outlier <- sample_data[which(abs(sample_data$residuals) > 0.5), ]
is.numeric(sample_data$residuals)
```

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

```
#geneids <- data.frame(outlier$M_HBV_unfiltered.GENEID, outlier$F_HBV_unfiltered.GENEID)



#write.csv(geneids, "geneids.csv")
```

```
#creating data frame for data in ggplot and renaming columns for ggplot
df <- sample_data
colnames(df) <- c("M_logFC", "M_name", "M_adj.P.Val", "F_logFC", "F_name", "F_adj.P.Val","residu
als")
df<-df[which(df$M_adj.P.Val<0.05 | df$F_adj.P.Val <0.05), ]
ggp<- ggplot(df, mapping=aes(x=M_logFC, y= F_logFC)) + geom_point()

#getting coefficients from regression function to create line for ggplot
regression <- lm(df$F_logFC~df$M_logFC)
regression
```

```
##
## Call:
## lm(formula = df$F_logFC ~ df$M_logFC)
##
## Coefficients:
## (Intercept)   df$M_logFC
##    0.004992     0.817150
```

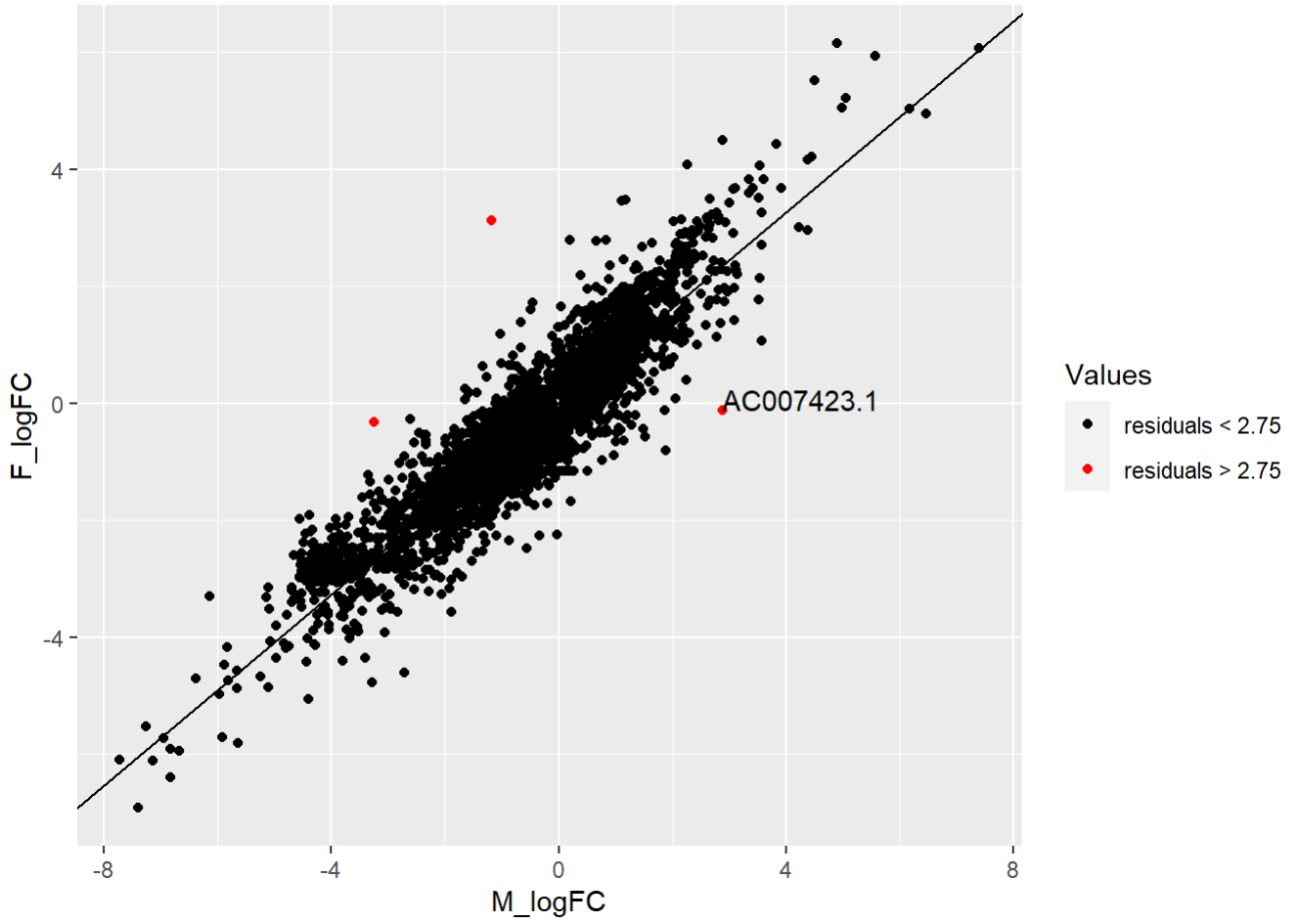
```
coeff<-coefficients(regression)
coeff
```

```
## (Intercept)  df$M_logFC
## 0.004991509 0.817150314
```

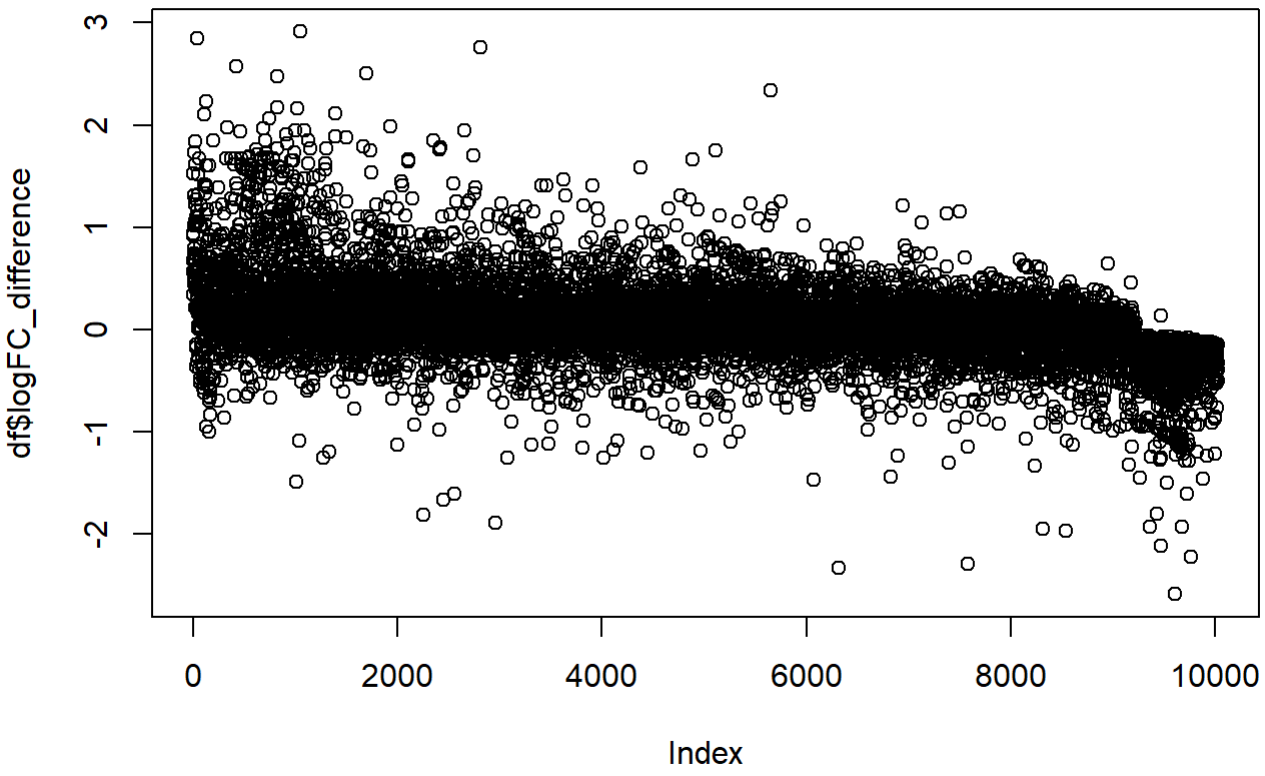
```
intercept<-coeff[1]
slope<-coeff[2]

#plotting with regression line

ggp2 <- ggp+geom_abline(intercept=intercept, slope=slope)


#plotting with color scheme. If residuals is >2.75, then the point will be colored red.

ggp3<- ggp2 +geom_point(aes(color=ifelse( abs(residuals) >2.75, "red", "black"))) +scale_color_m
anual(labels=c("residuals < 2.75", "residuals > 2.75"), values= c("black", "red")) +labs(color=
"Values")

#plotting with gene label if the residual is >2.75, then the point will be labeled with the gene
name.
ggp4 <- ggp3 + geom_text(aes(label=ifelse(residuals>2.75, as.character(M_name), '')), hjust=0, v
just=0)
pdf("~/R/effect_size_ggplot_HCV.pdf", width=12, height=12)
ggp4
dev.off()
```

```
## png
##   2
```

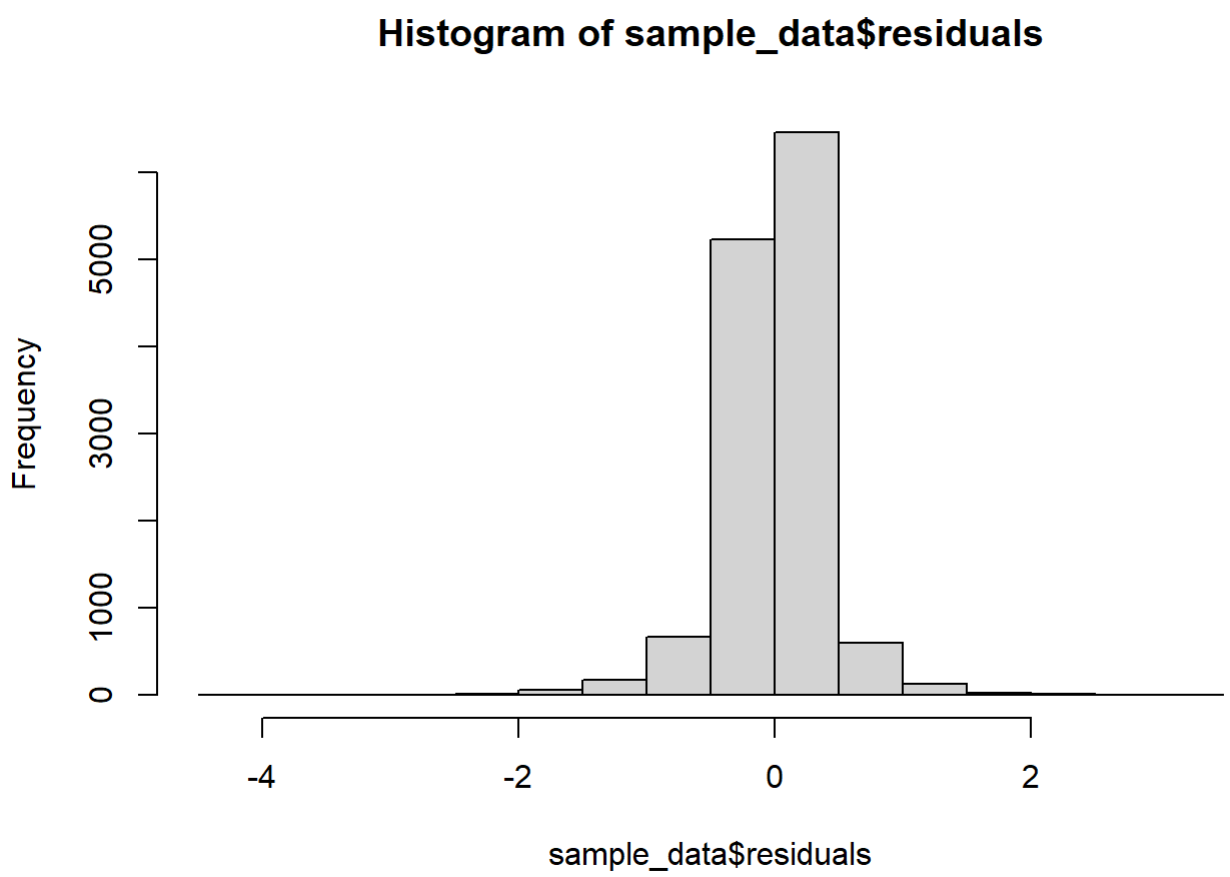
```
ggp4
```




```
#plot with the log fold change difference of male log FC subtracted from female log FC
df$logFC_difference <- abs(df$M_logFC)-abs(df$F_logFC)
plot(df$logFC_difference)
```




```
#histogram of residuals should expect for most samples to be around 0
hist(sample_data$residuals)
```




```
#printing gene with logFC greater than 2
gene_names_M_HCV <- sample_data[which(abs(sample_data$M_HCV_unfiltered.logFC) > 2), ]
gene_names_F_HCV <- sample_data[which(abs(sample_data$F_HCV_unfiltered.logFC) > 2), ]

#printing out number of genes in list first output is the number of genes

dim(gene_names_M_HCV)
```

```
## [1] 718   7
```

```
dim(gene_names_F_HCV)
```

```
## [1] 571   7
```

```
write.csv(gene_names_M_HCV, "Regression_gene_lists_M_HCV.csv")
write.csv(gene_names_F_HCV, "Regression_gene_lists_F_HCV.csv")
```