Analysis of Differential Expression of Fruit Flies Using Voom: Code Execution

Data Preprocessing

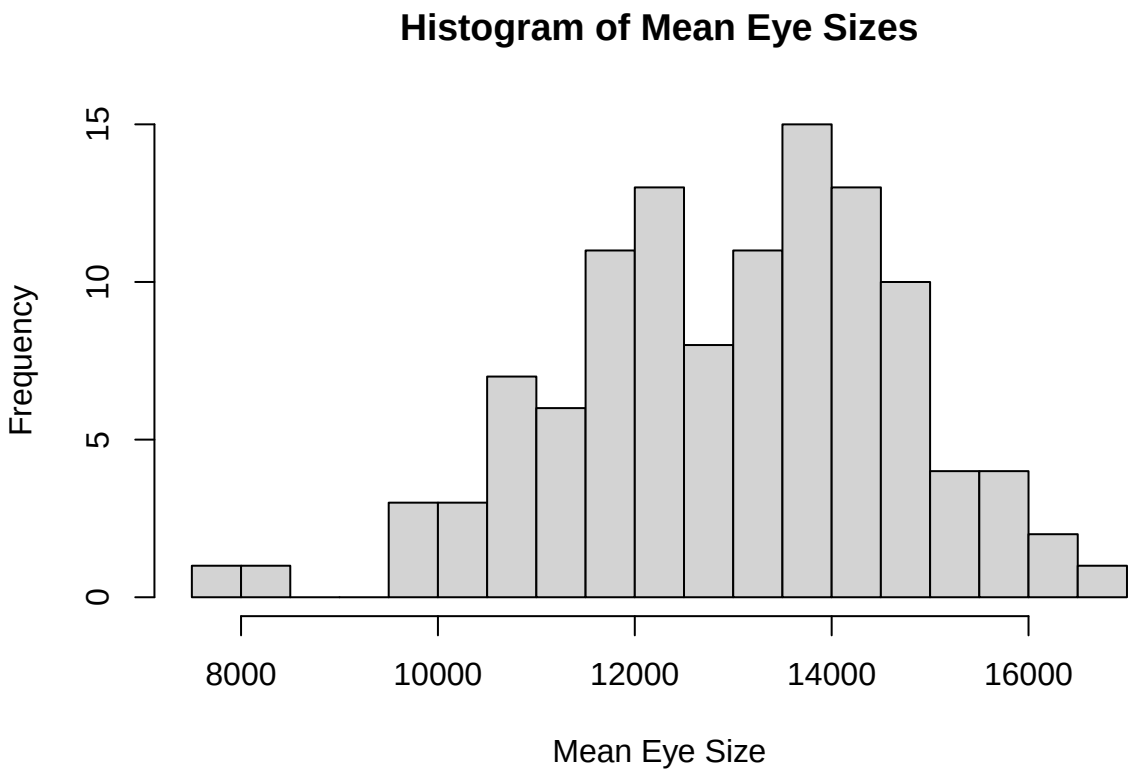
This step goes through both the expression set and the eye size set to remove any strains that aren't found in both.

```
#load in data
expresData <- read.delim("dgrp_expression_Female.txt", row.names = 1)
eyeSizeData <- read.delim("rpr.txt")
#remove unnecessary characters and split groups
colnames(expresData) <- gsub("line_", "", colnames(expresData))
group1names <- seq(1, length(expresData), 2)
splitData <- expresData[, group1names]
colnames(splitData) <- gsub(".1$", "", colnames(splitData))
#check what strains are in the data and create a finalized data frame for both
finalNames <- NULL
for (i in 1:nrow(eyeSizeData)) {
  target <- as.character(eyeSizeData[i,1])
  if(target %in% colnames(splitData)){
    finalNames <- append(finalNames, target)
  }
}
finalExpressData <- splitData[, finalNames]
droppedValues <- NULL
for (i in 1:nrow(eyeSizeData)) {
  target <- as.character(eyeSizeData[i, 1])
  if((target %in% finalNames)==FALSE){
    droppedValues <- append(droppedValues, i)
  }
}
finalSizeData <- eyeSizeData[-droppedValues,]
```

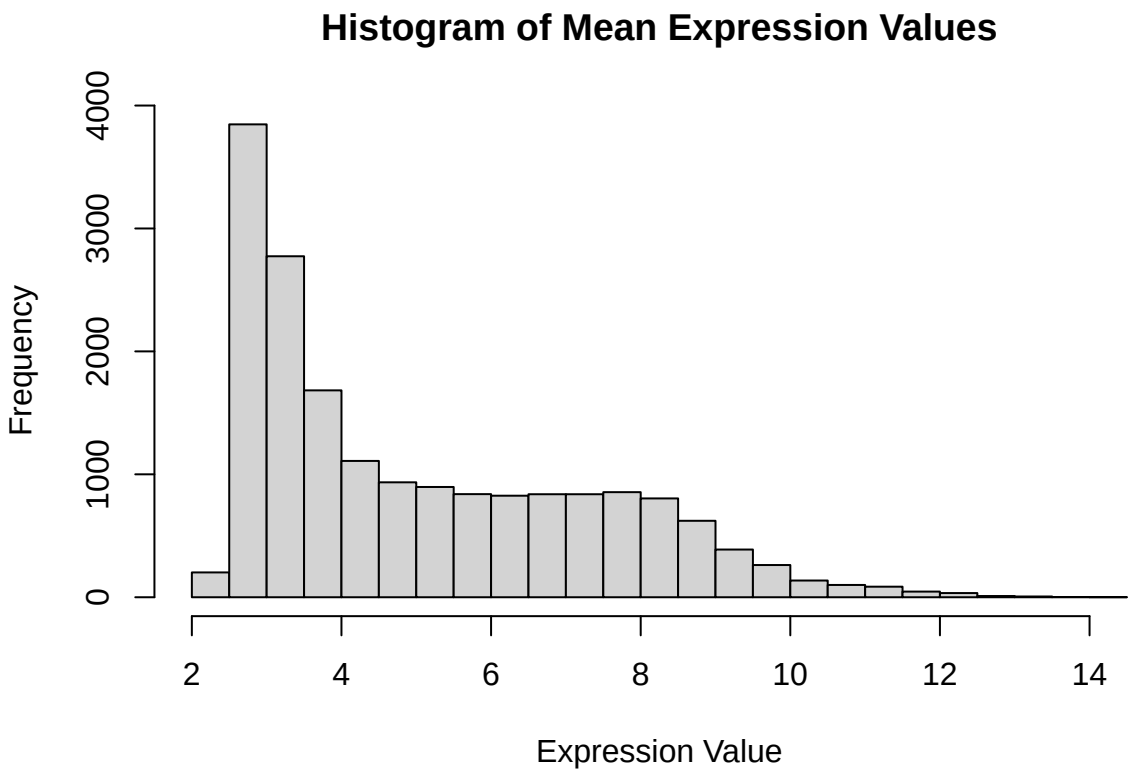
Visualizations

Early visualizations to get a better idea of the data.

```
#visualize
hist(finalSizeData$Mean.Eye.Size, breaks = 20, xlab = "Mean Eye Size", main = "Histogram of Mean Eye Si
```



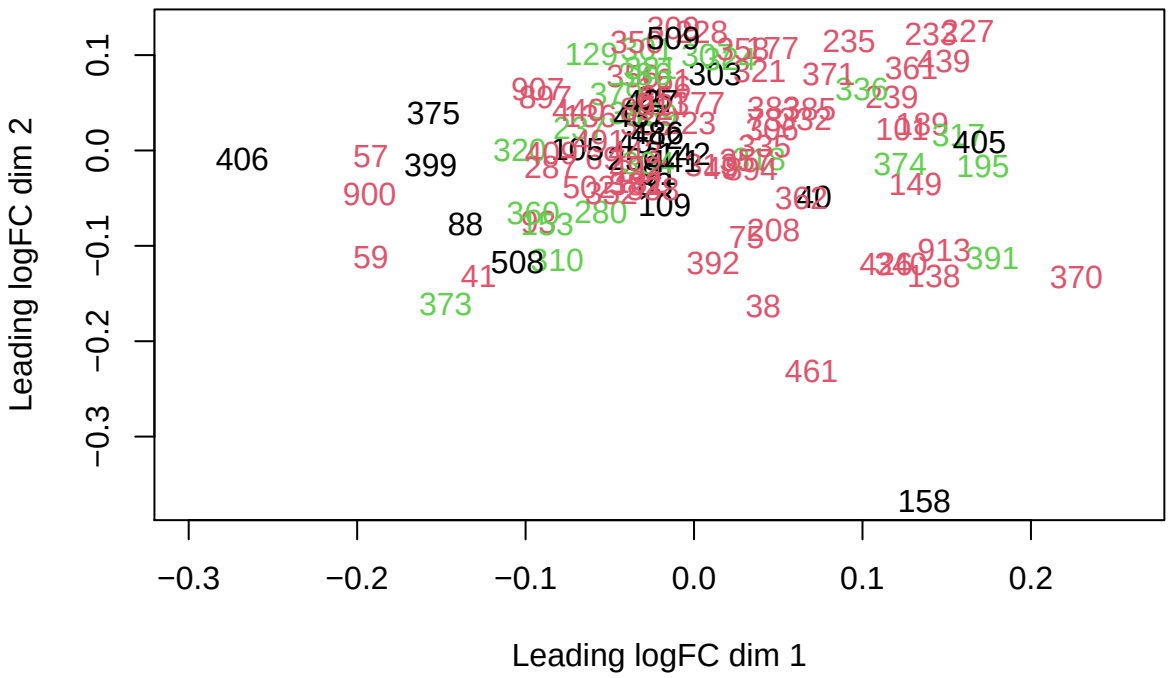

```
expressMeans <- rowMeans(finalExpressData)
hist(expressMeans, breaks = 20, xlab = "Expression Value", main = "Histogram of Mean Expression Values")
```



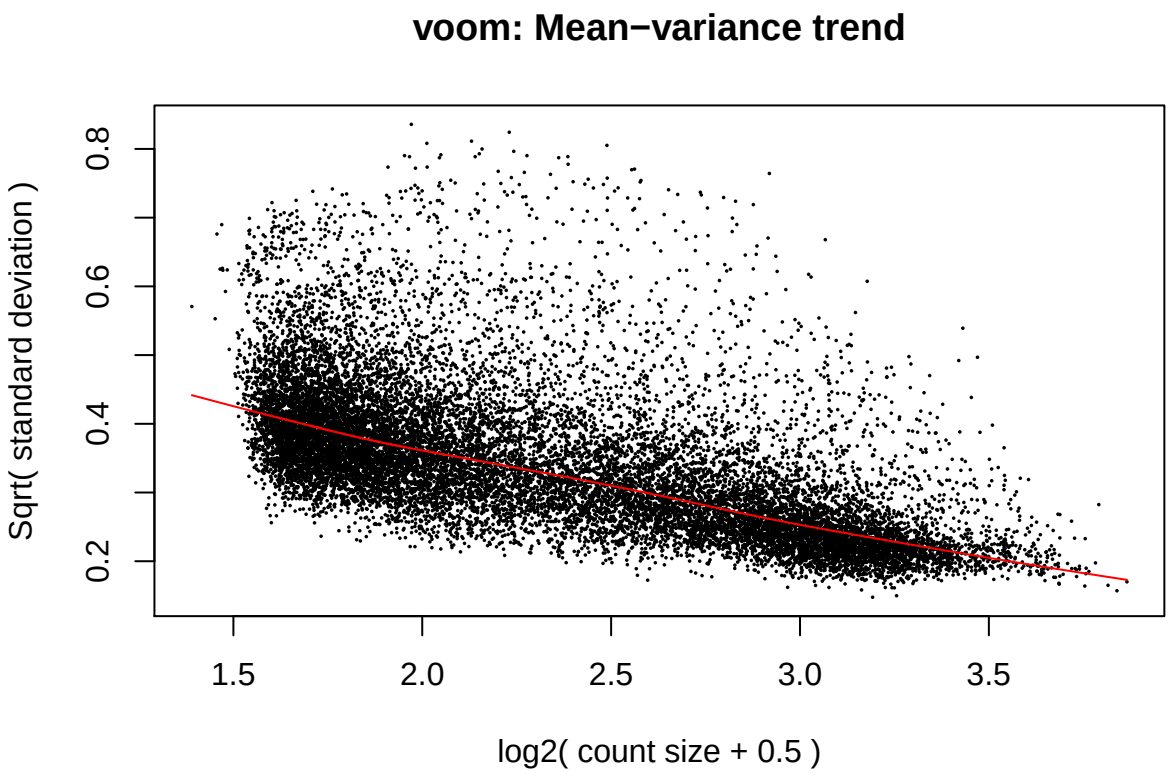
Methods

Applying the actual methods necessary to get the results.

```
#calculate quantiles
quantiles <- quantile(finalSizeData$Mean.Eye.Size, probs = c(0, 0.2, 0.8, 1))
groups <- c("small", "medium", "large")
eyeSizes <- cut(finalSizeData$Mean.Eye.Size, breaks = quantiles, include.lowest = TRUE, labels = groups)
#create a DGElist and recalculate normalization factors
expresList <- DGEList(as.matrix(finalExpressData))
expresList <- calcNormFactors(expresList)
#Filter out low expression genes
droppedGenes <- which(apply(cpm(expresList), 1, max) < 30)
#no dropped genes
#create a multidimensional scaling plot using the col names as labels
plotMDS(expresList, col = as.numeric(eyeSizes))
```




```
#create a model matrix
modelMat <- model.matrix(~0 + eyeSizes)
#use voom and apply it to a table to be read
vObj <- voom(expresList, modelMat, plot = T)
```




```
vFit <- lmFit(vObj, design = modelMat)
vFitBayes <- eBayes(vFit)
fitTable <- topTable(vFitBayes)
#use contrasts to get better results
contrasts <- makeContrasts(eyeSizeslarge - eyeSizesmedium, eyeSizesmedium - eyeSizessmall, eyeSizeslarg
contrFit <- contrasts.fit(vFitBayes, contrasts)
contrFitBayes <- eBayes(contrFit)
contrFitTable <- topTable(contrFitBayes, resort.by = "p")
```

Results

The results of applying Voom to our data.

```
#print out the table data
fitTable
```

```
##             eyeSizessmall eyeSizesmedium eyeSizeslarge  AveExpr       F
## FBgn0038148      7.287831       7.290465      7.296877 7.291657 3264815
## FBgn0030034      6.638433       6.645713      6.647636 6.644810 3039458
## FBgn0036373      6.712366       6.705609      6.710535 6.708219 2984343
## FBgn0261599      7.205400       7.205409      7.208299 7.206481 2682629
## FBgn0000556      7.266014       7.267794      7.267514 7.267991 2679452
## FBgn0025286      7.141547       7.136611      7.140674 7.138796 2457481
## FBgn0004045      7.314151       7.318543      7.319410 7.318111 2430338
```

```
## FBgn0034968      7.143617       7.134498      7.139784 7.137901 2377172
## FBgn0038108      6.609892       6.617027      6.614519 6.615485 2363221
## FBgn0001324      6.698803       6.704103      6.701064 6.702318 2313534
##                   P.Value     adj.P.Val
## FBgn0038148 2.268610e-280 4.115259e-276
## FBgn0030034 1.323431e-278 1.200352e-274
## FBgn0036373 3.745579e-278 2.264827e-274
## FBgn0261599 1.603485e-275 6.222749e-272
## FBgn0000556 1.715201e-275 6.222749e-272
## FBgn0025286 2.341081e-273 7.077869e-270
## FBgn0004045 4.401849e-273 1.140708e-269
## FBgn0034968 1.547913e-272 3.509893e-269
## FBgn0038108 2.163083e-272 4.359814e-269
## FBgn0001324 7.239712e-272 1.313284e-268
```

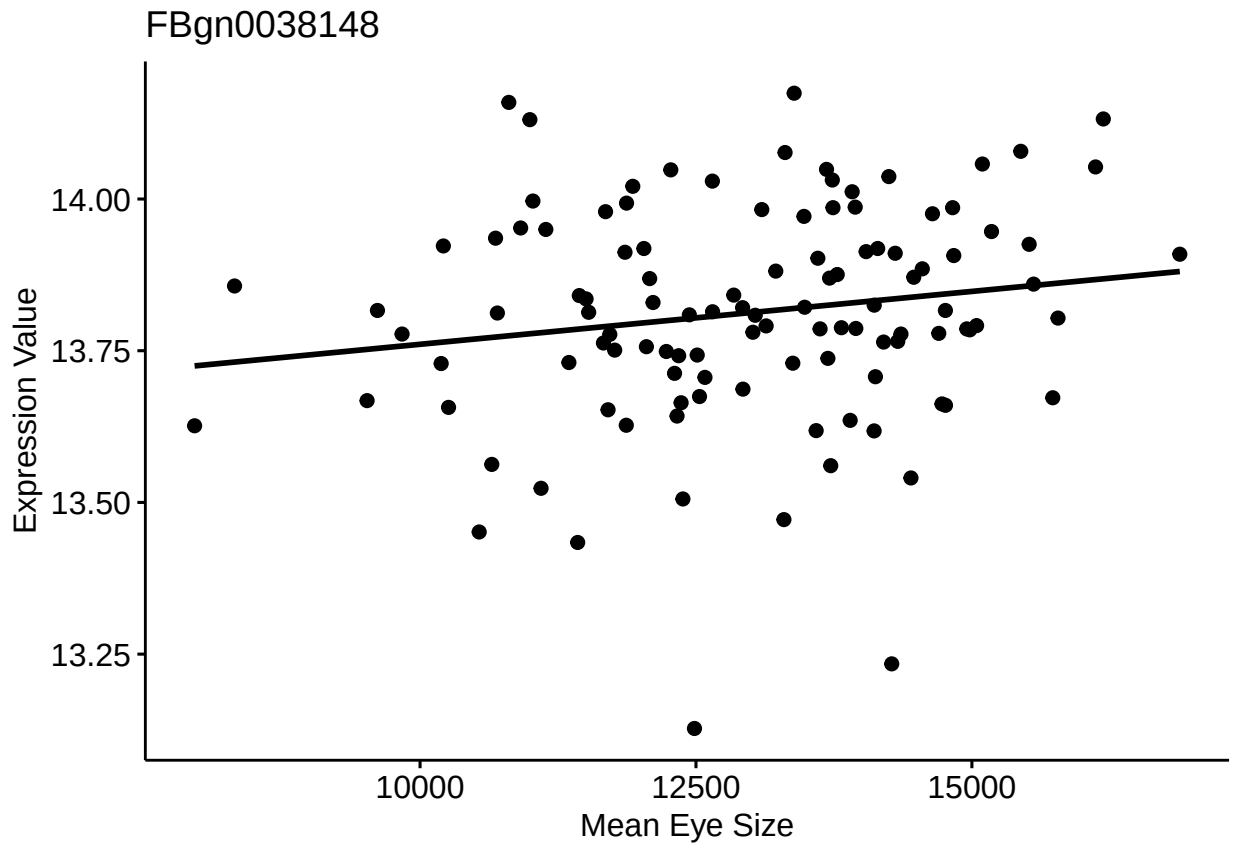
```
contrFitTable
```

```
##             eyeSizeslarge...eyeSizesmedium eyeSizesmedium...eyeSizessmall
## FBgn0035620                     0.017449266                     -0.17936296
## FBgn0027584                    -0.004958454                     -0.36814858
## FBgn0030594                    -0.036535755                     -0.14746926
## FBgn0036993                     0.032478218                     -0.13520362
## FBgn0039342                     0.022616888                     -0.12652274
## FBgn0039085                    -0.064496052                     -0.12711583
## FBgn0035667                     0.043672269                     -0.09374099
## FBgn0054043                    -0.061284733                     -0.12348902
## FBgn0036518                     0.001984224                      0.04503949
## FBgn0036659                    -0.046976902                     -0.07803003
##             eyeSizeslarge...eyeSizessmall  AveExpr         F      P.Value
## FBgn0035620                    -0.16191370 5.858138 12.357517 1.392186e-05
## FBgn0027584                    -0.37310703 5.834361 10.597994 6.019004e-05
## FBgn0030594                    -0.18400501 6.157179 10.344829 7.453699e-05
## FBgn0036993                    -0.10272540 5.959036  9.523969 1.499141e-04
## FBgn0039342                    -0.10390585 6.786504  9.513390 1.512788e-04
## FBgn0039085                    -0.19161188 5.460847  9.033089 2.286222e-04
## FBgn0035667                    -0.05006872 6.690378  9.024299 2.303631e-04
## FBgn0054043                    -0.18477375 5.874565  8.651390 3.181096e-04
## FBgn0036518                     0.04702372 6.579366  8.508924 3.600275e-04
## FBgn0036659                    -0.12500693 5.670124  8.380254 4.027073e-04
##             adj.P.Val
## FBgn0035620 0.2525425
## FBgn0027584 0.4507004
## FBgn0030594 0.4507004
## FBgn0036993 0.5488396
## FBgn0039342 0.5488396
## FBgn0039085 0.5969696
## FBgn0035667 0.5969696
## FBgn0054043 0.6563564
## FBgn0036518 0.6563564
## FBgn0036659 0.6563564
```

```
#transpose the data so ggscatter stops complaining
transData <- rbind(finalExpressData, as.vector(finalSizeData[, 2]))
```

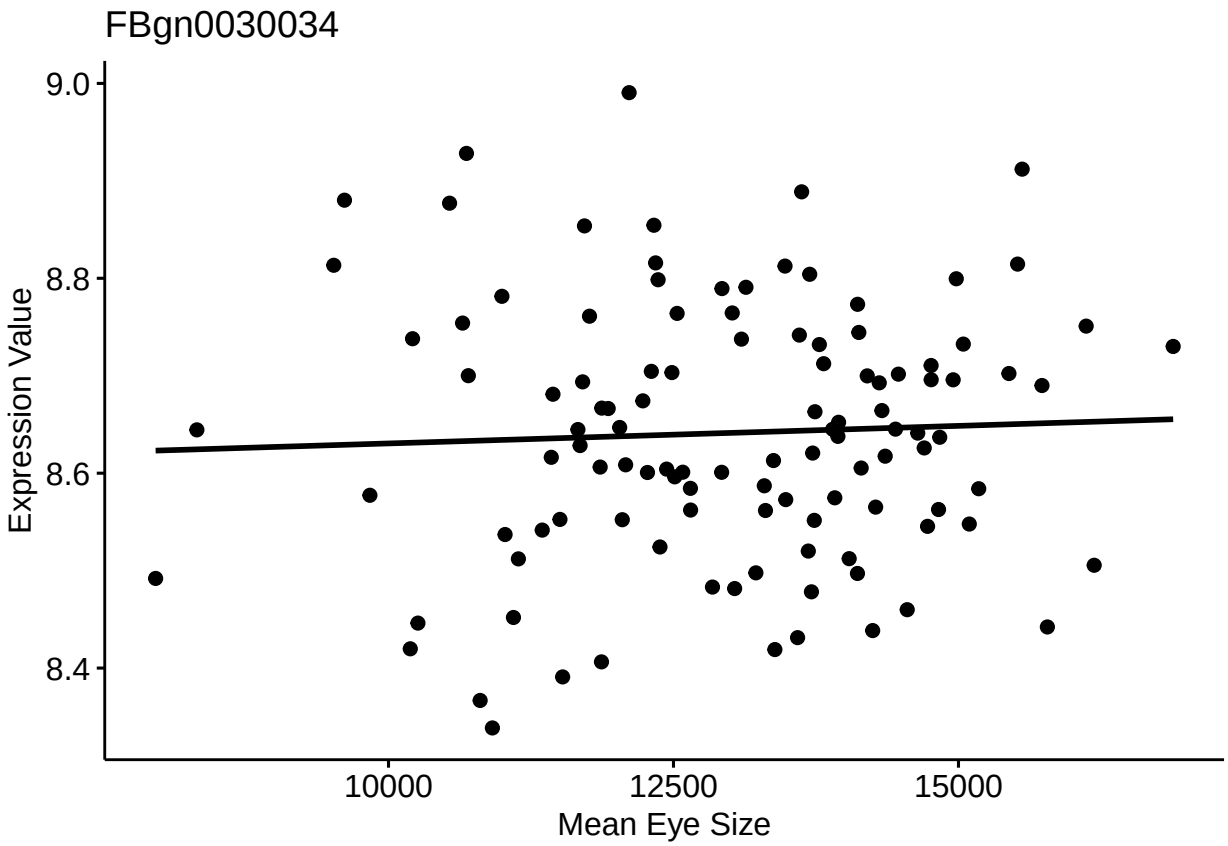
```
transCol <- colnames(transData)
transRow <- rownames(transData)
transData <- transpose(transData)
rownames(transData) <- transCol
colnames(transData) <- transRow
colnames(transData)[colnames(transData) == "18141"] <- "eyeSize"
#do the plots for the top 5 genes found in the normal fitted table
ggscatter(data = transData, x = "eyeSize", y = rownames(fitTable[1,]), add = "reg.line", xlab = "Mean Ey
```

```
## `geom_smooth()` using formula 'y ~ x'
```



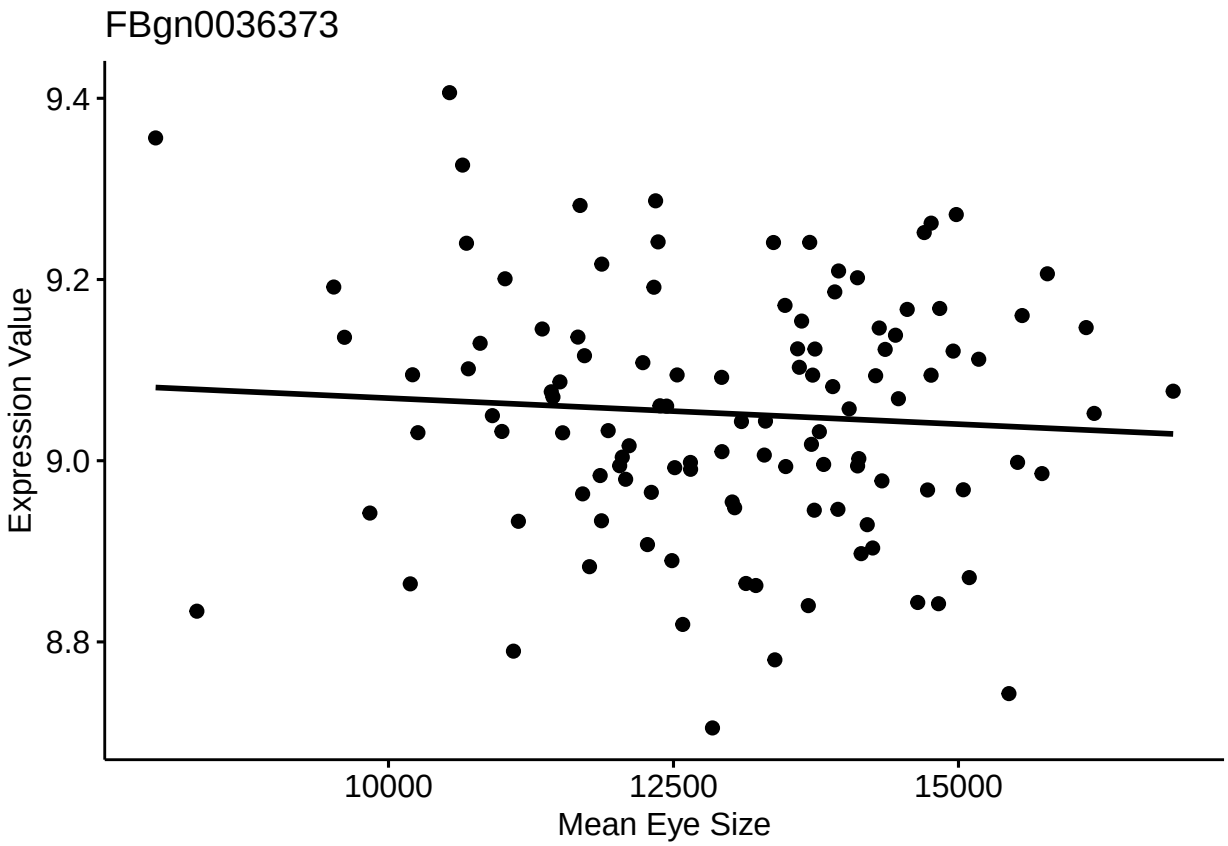

```
ggscatter(data = transData, x = "eyeSize", y = rownames(fitTable[2,]), add = "reg.line", xlab = "Mean Ey
```

```
## `geom_smooth()` using formula 'y ~ x'
```



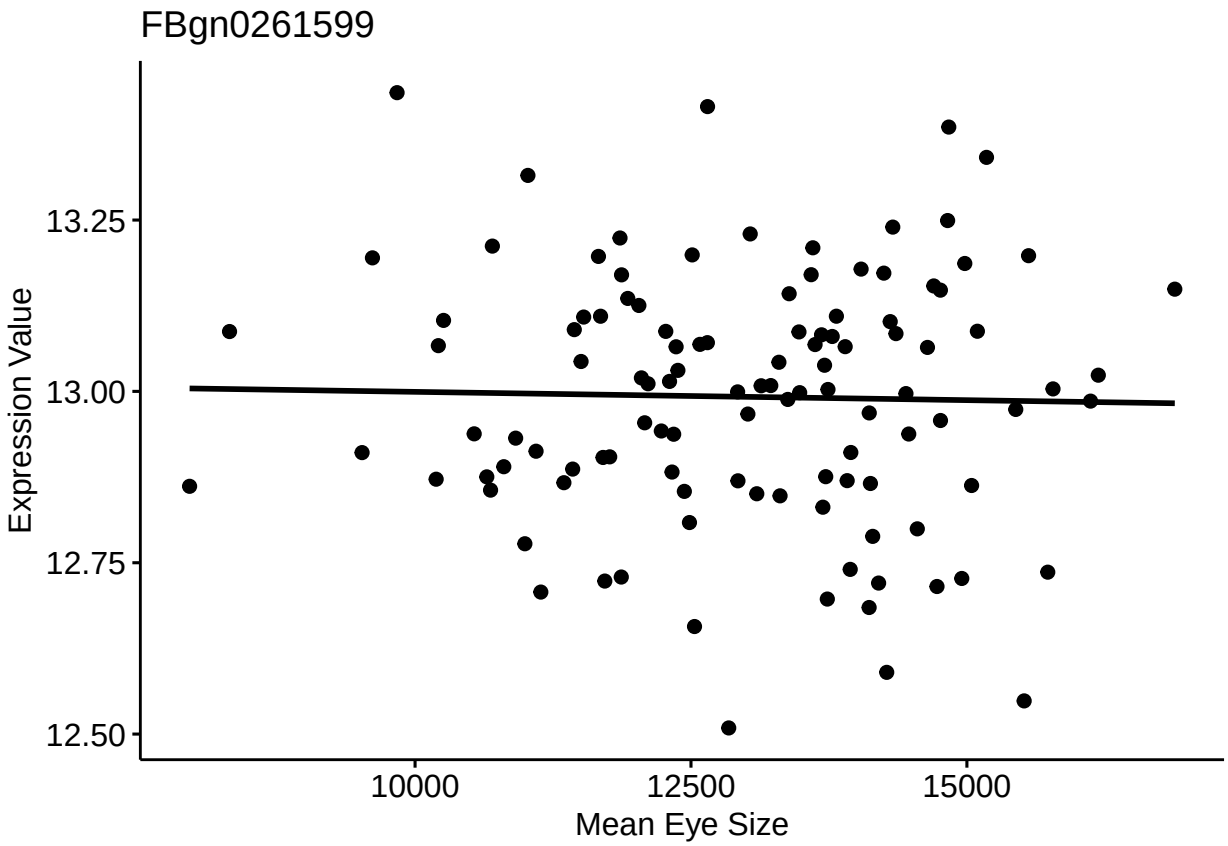

```
ggscatter(data = transData, x = "eyeSize", y = rownames(fitTable[3,]), add = "reg.line", xlab = "Mean Ey
```

```
## `geom_smooth()` using formula 'y ~ x'
```



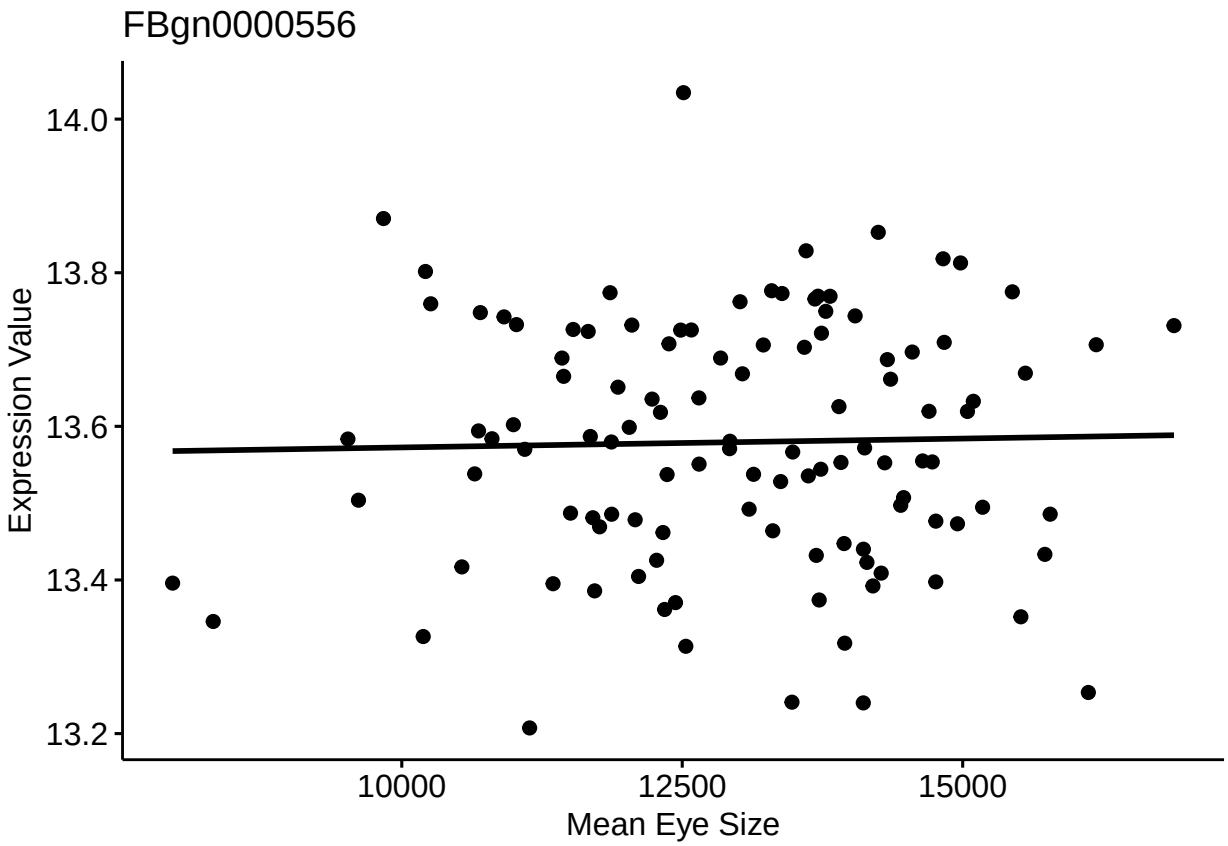

```
ggscatter(data = transData, x = "eyeSize", y = rownames(fitTable[4,]), add = "reg.line", xlab = "Mean Ey
```

```
## `geom_smooth()` using formula 'y ~ x'
```



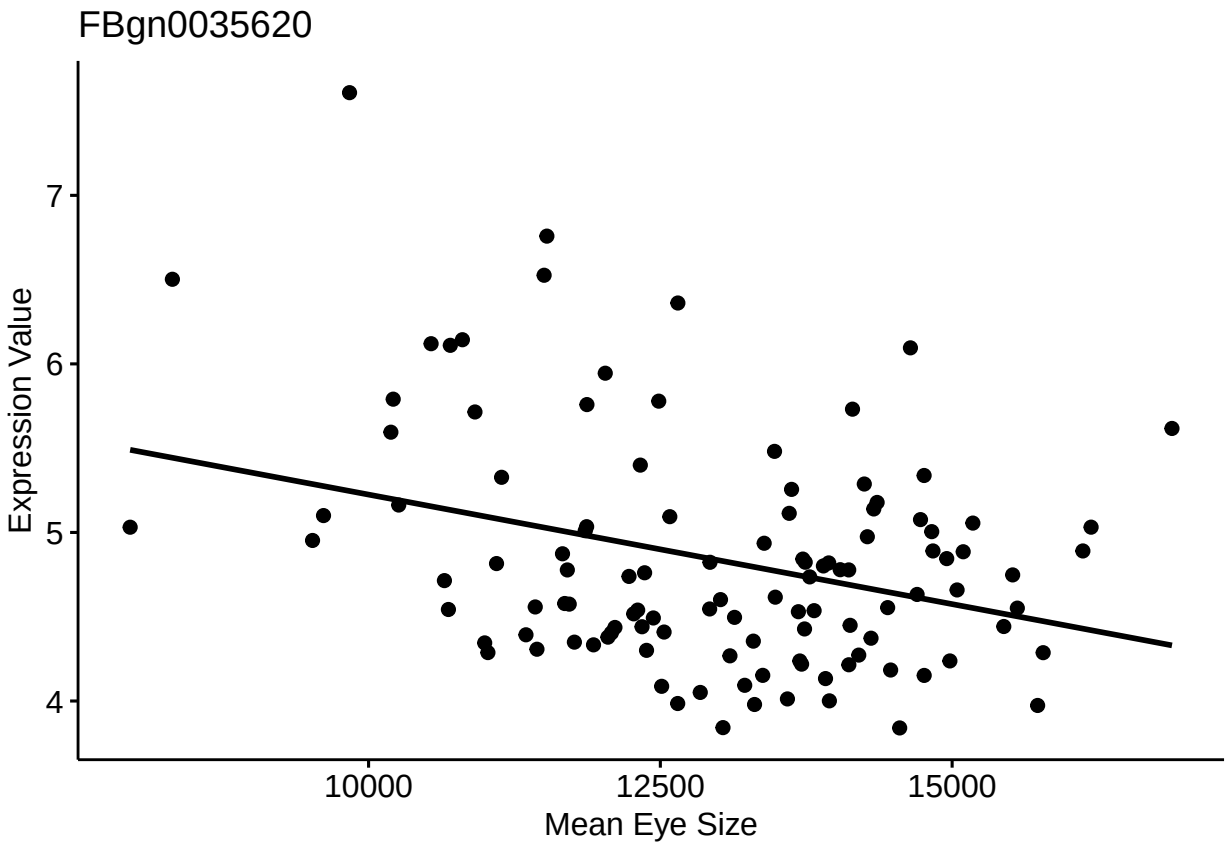

```
ggscatter(data = transData, x = "eyeSize", y = rownames(fitTable[5,]), add = "reg.line", xlab = "Mean Ey
```

```
## `geom_smooth()` using formula 'y ~ x'
```



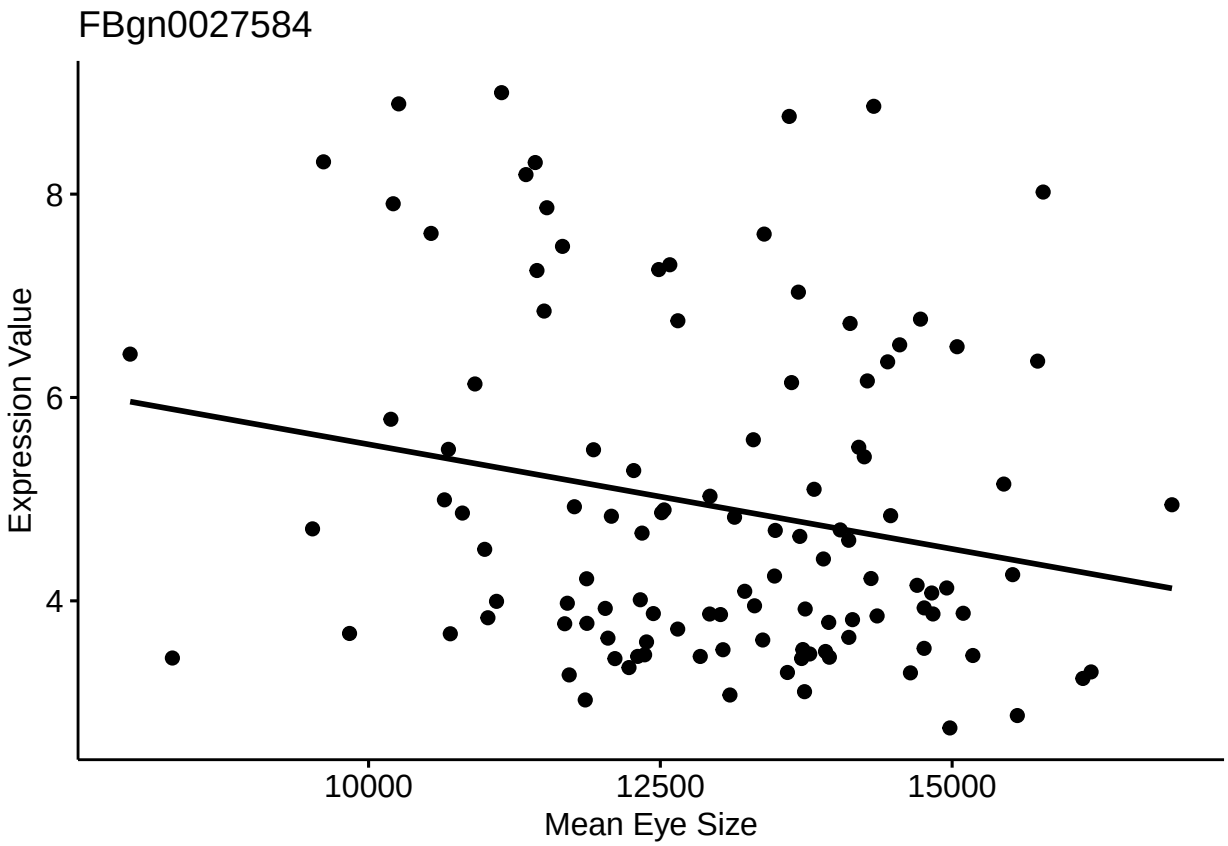

```
#do the plots for the top 5 genes in the contrast fitted table
ggscatter(data = transData, x = "eyeSize", y = rownames(contrFitTable[1,]), add = "reg.line", xlab = "M
```

```
## `geom_smooth()` using formula 'y ~ x'
```



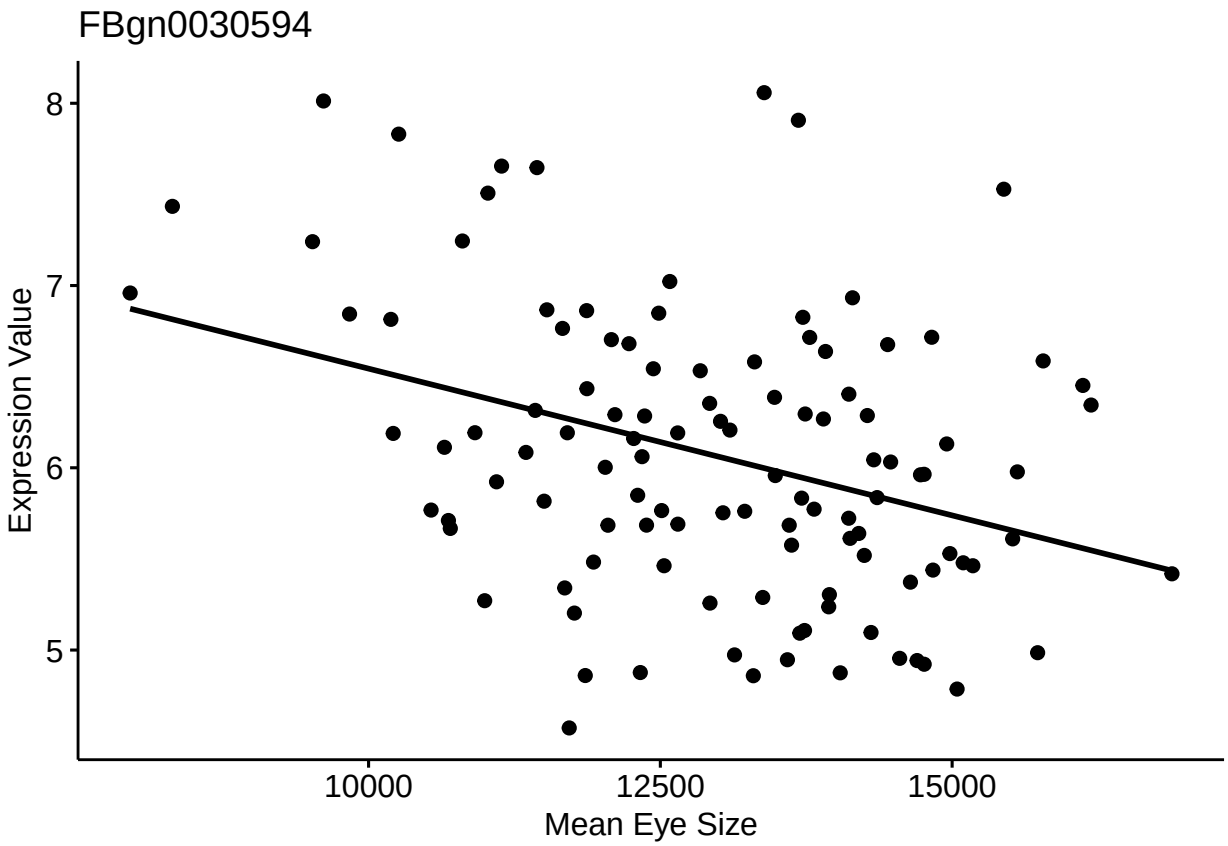

```
ggscatter(data = transData, x = "eyeSize", y = rownames(contrFitTable[2,]), add = "reg.line", xlab = "M
```

```
## `geom_smooth()` using formula 'y ~ x'
```



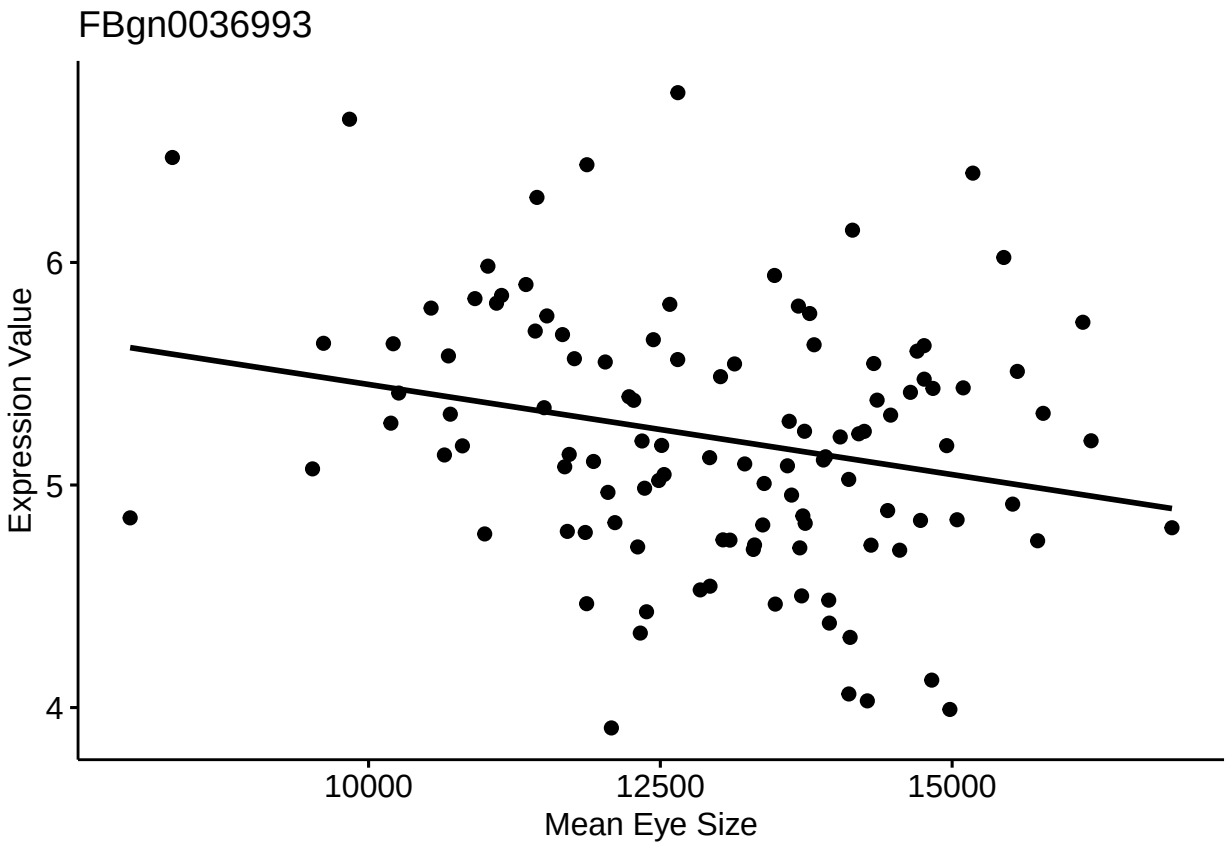

```
ggscatter(data = transData, x = "eyeSize", y = rownames(contrFitTable[3,]), add = "reg.line", xlab = "M
```

```
## `geom_smooth()` using formula 'y ~ x'
```



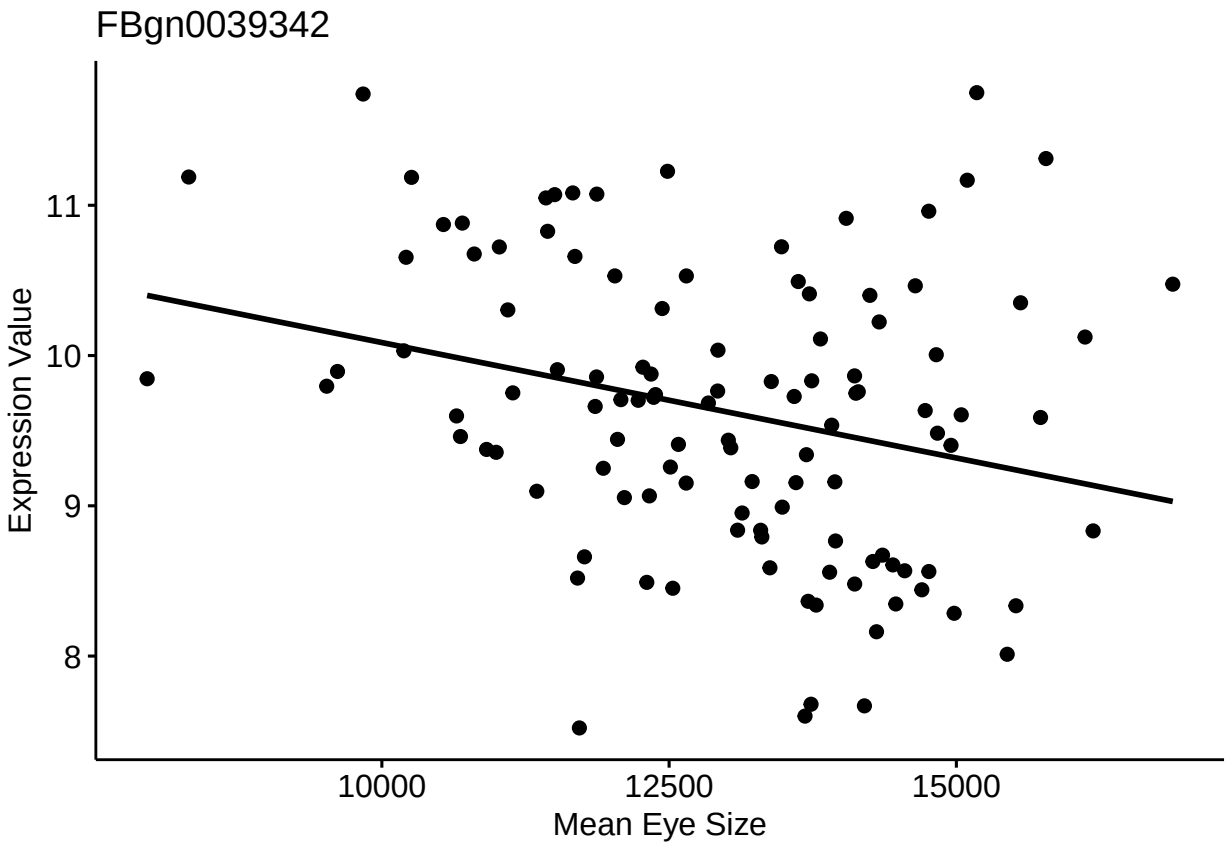

```
ggscatter(data = transData, x = "eyeSize", y = rownames(contrFitTable[4,]), add = "reg.line", xlab = "M
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
ggscatter(data = transData, x = "eyeSize", y = rownames(contrFitTable[5,]), add = "reg.line", xlab = "M
```

```
## `geom_smooth()` using formula 'y ~ x'
```


FBgn0039342
Expression Value
11
10
9
8
10000
12500
15000
Mean Eye Size