Assignment 4 - Data Modeling & DGE

Abrar Faruque

October 2025

Assignment 4 - Data Modeling & Differential Gene Expression

The dataset can be accessed here: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29

The phenotype attributions are as follows:

1) ID number
2) Diagnosis (M = malignant, B = benign) 3-32)

Ten real-valued features are computed for each cell nucleus:

a) radius (mean of distances from center to points on the perimeter)
b) texture (standard deviation of gray-scale values)
c) perimeter
d) area
e) smoothness (local variation in radius lengths)
f) compactness (perimeter^2 / area - 1.0)
g) concavity (severity of concave portions of the contour)
h) concave points (number of concave portions of the contour)
i) symmetry
j) fractal dimension ("coastline approximation" - 1)

Here is some code to read in the dataset

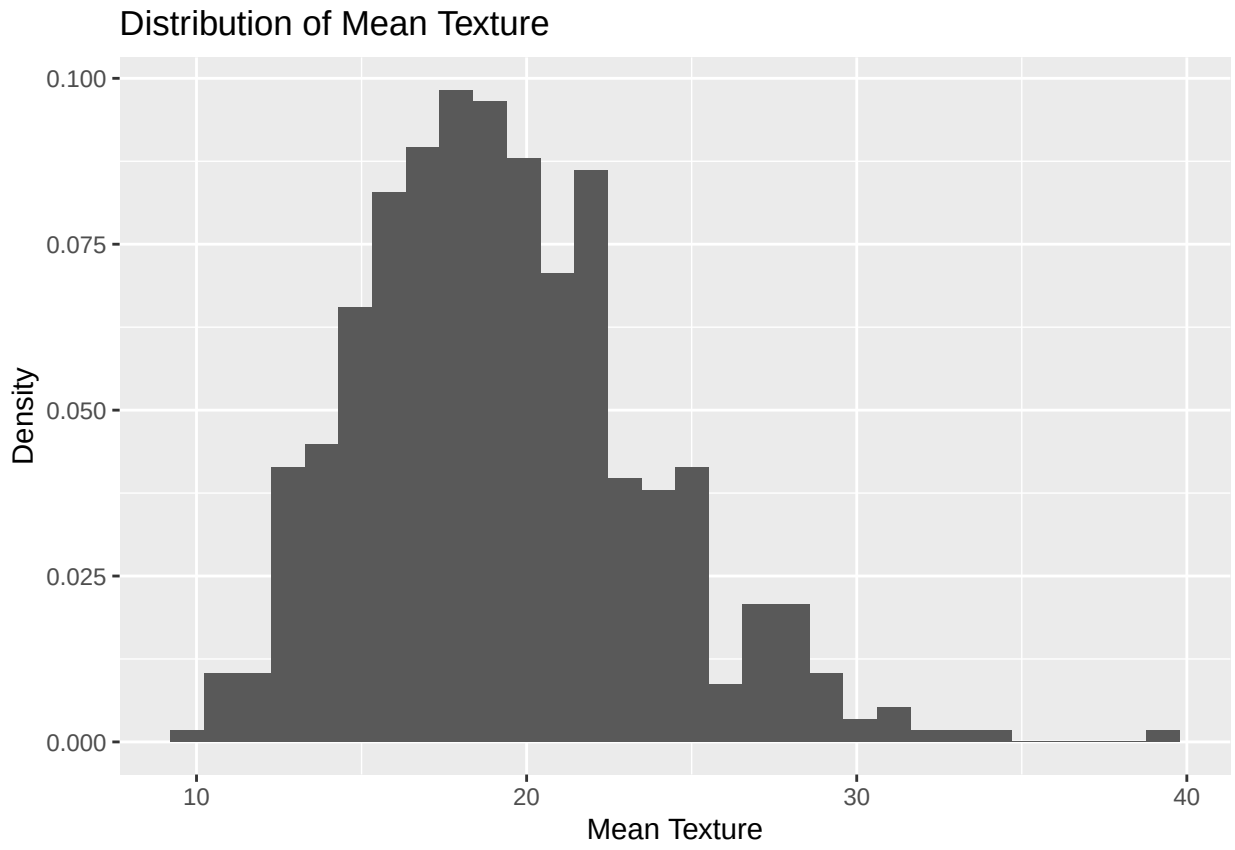
bCancer <- read.csv("breastCancer.csv", header=T)

1. Draw a histogram and a qqplot of the continuous variable `texture_mean`. [0.25]

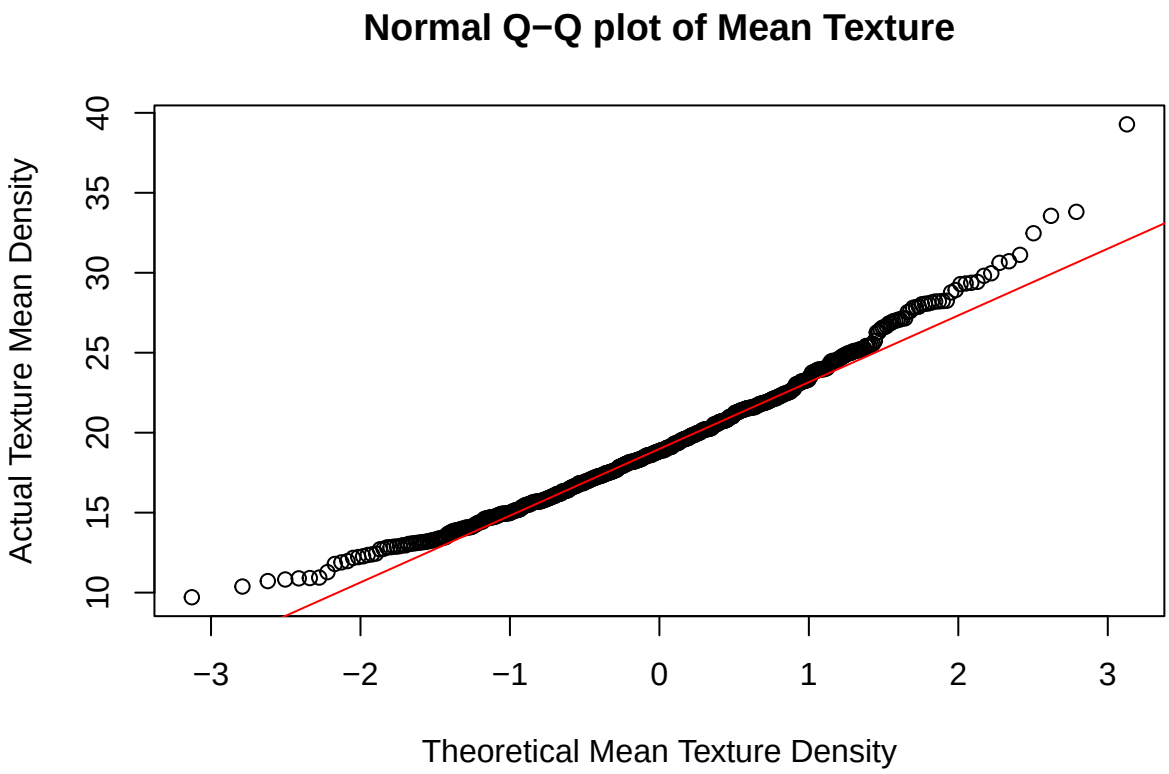
```
library(ggplot2)

bCancer <- read.csv("breastCancer.csv", header=T)
ggplot(bCancer,aes(x=texture_mean,y=..density..))+
  geom_histogram()+
  labs(title="Distribution of Mean Texture",x="Mean Texture", y="Density")
```

```
## Warning: The dot-dot notation (‘..density..‘) was deprecated in ggplot2 3.4.0.
## i Please use ‘after_stat(density)‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
## `stat_bin()` using `bins = 30`. Pick better value `binwidth`.
```




```
qqnorm(bCancer$texture_mean, main = "Normal Q-Q plot of Mean Texture",
       xlab = "Theoretical Mean Texture Density", ylab="Actual Texture Mean Density")
qqline(bCancer$texture_mean, col = "red")
```



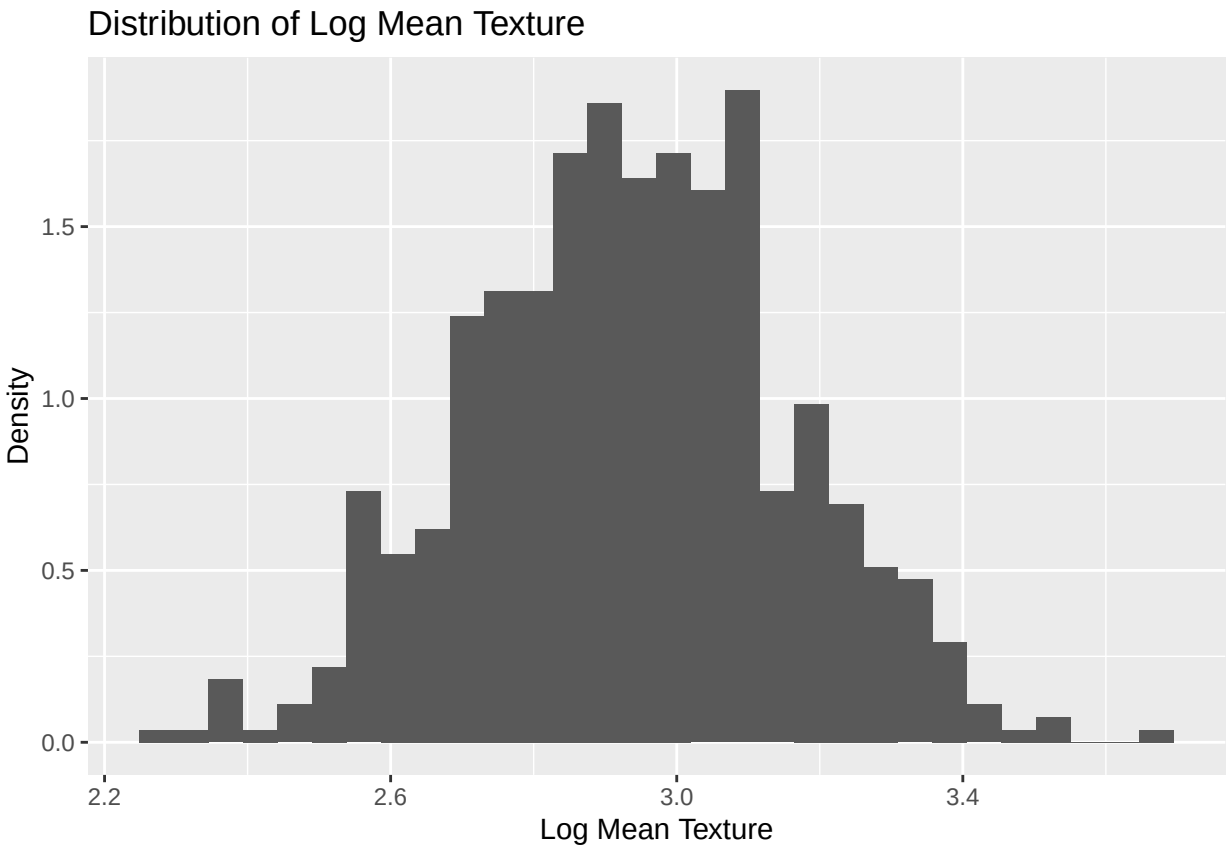
a) What is your interpretation of the qqplot? [0.25]

```
#Data apears to be right skewed visually based on the histogram, the data mostly falls along the diagon
```

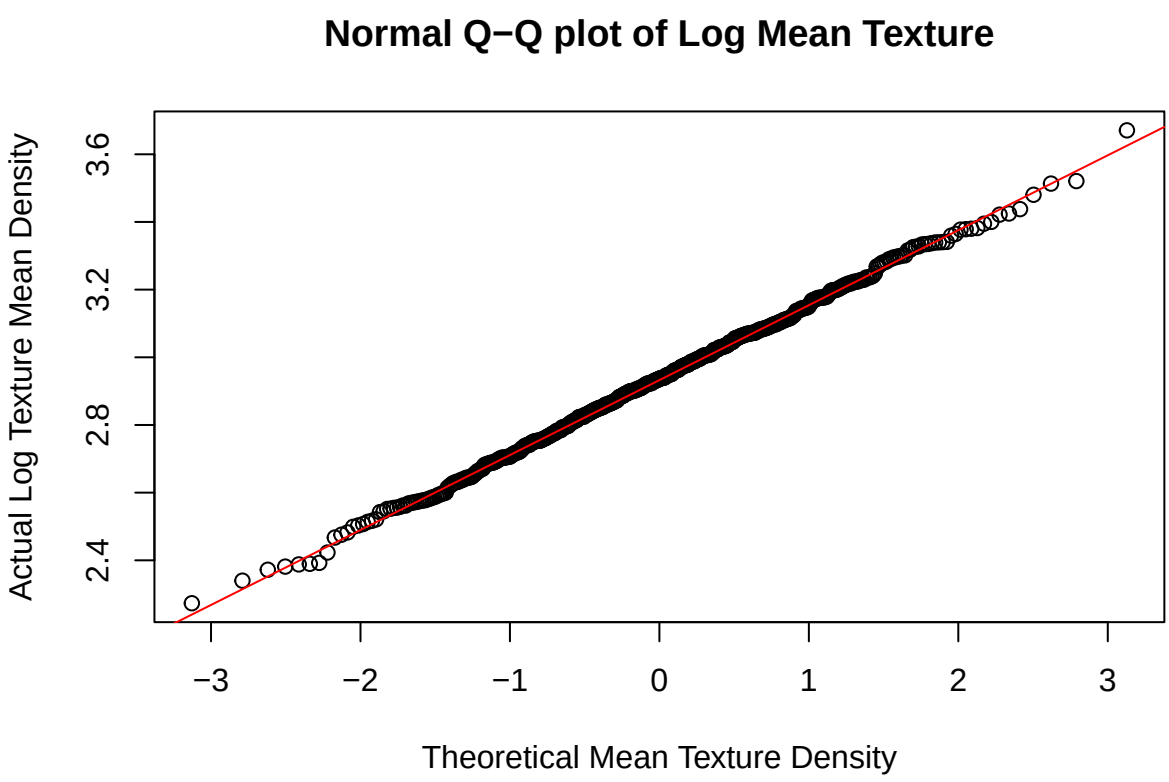
b) Try applying a logarithmic transformation to the variable and justify whether you will be proceeding with the transformed or untransformed data when running your linear model. [0.5]

```
bCancer <- read.csv("breastCancer.csv", header=T)
logTextureMean <- data.frame (log_texture_mean <- log(bCancer$texture_mean))
ggplot(logTextureMean, aes(x=log_texture_mean,y=..density..))+
  geom_histogram()+
  labs(title="Distribution of Log Mean Texture",x="Log Mean Texture", y="Density")
```

```
## `stat_bin()` using `bins = 30`. Pick better value `binwidth`.
```




```
qqnorm(logTextureMean$log_texture_mean, main = "Normal Q-Q plot of Log Mean Texture",
       xlab = "Theoretical Mean Texture Density", ylab="Actual Log Texture Mean Density")
qqline(logTextureMean$log_texture_mean, col = "red")
```




```
#I would use the log data for the linear model as the texture mean data is no longer skewed right, the
```

2. Run a linear model with `texture_mean` as a function of `area_mean`. Are the variables significantly associated with each other? [0.75]

```
bCancer <- read.csv("breastCancer.csv", header=T)
logTextureMean <- log_texture_mean <- log(bCancer$texture_mean)
linear_model <- lm(logTextureMean ~ bCancer$area_mean)
model_summary <- summary(linear_model)
print(model_summary)
```

```
##
## Call:
## lm(formula = logTextureMean ~ bCancer$area_mean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.66633 -0.14054  0.00268  0.13629  0.68098
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       2.8010375  0.0185101 151.325  < 2e-16 ***
## bCancer$area_mean 0.0002050  0.0000249   8.231 1.28e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.2089 on 567 degrees of freedom
## Multiple R-squared:  0.1067, Adjusted R-squared:  0.1052
## F-statistic: 67.75 on 1 and 567 DF,  p-value: 1.283e-15
```

```
coefficients <- coef(linear_model)
intercept <- coefficients[1]
slope <- coefficients[2]
equation <- paste0("y = ", round(intercept, 2), " + ", slope , "x")
print(equation)
```

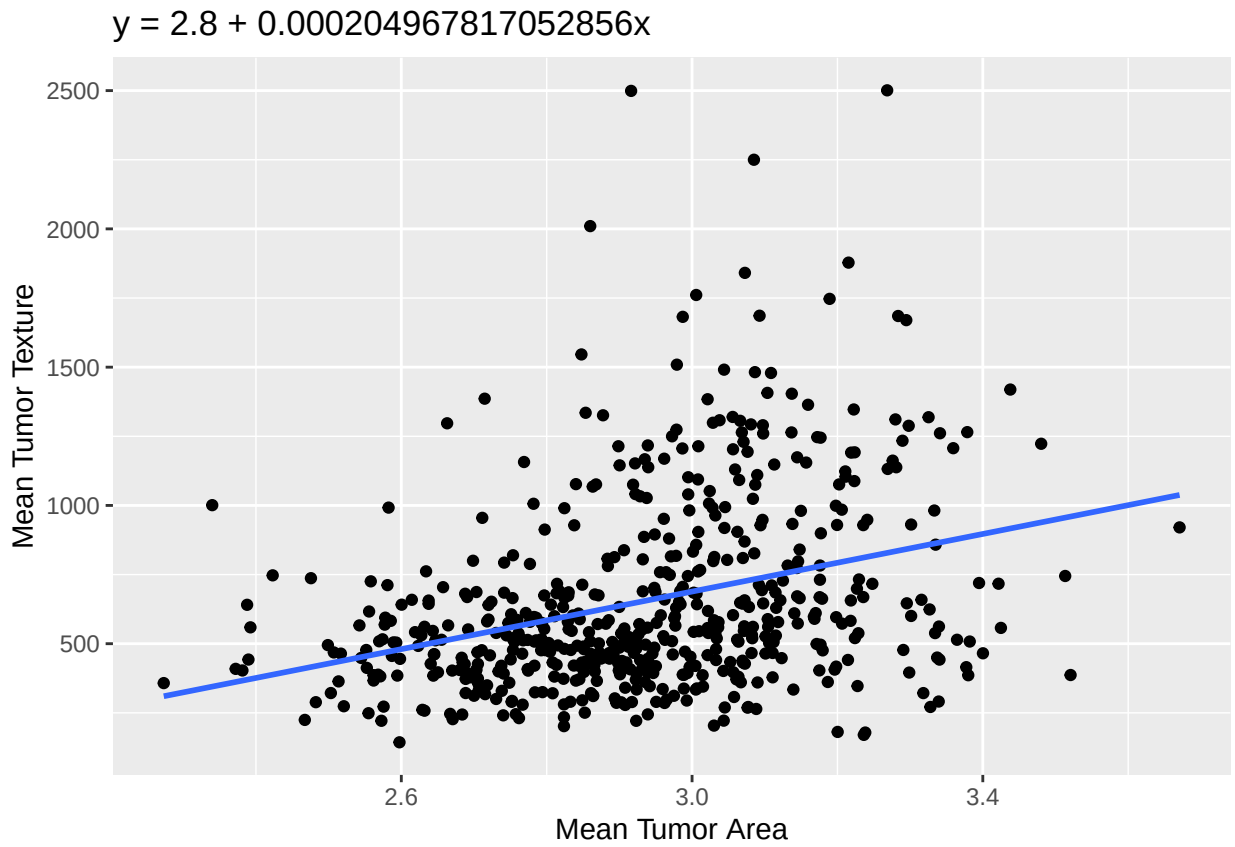
```
## [1] "y = 2.8 + 0.000204967817052856x"
```

```
#the R^2 value is small (close to 0) but positive indicating that there is a very weak positive associa
```

3. Visualize the linear model. Include the data points from the original data and the line for the linear model. Include the formula for the equation in the title of the plot. [1]

```
linear_model <- lm(logTextureMean ~ bCancer$area_mean)
coefficients <- coef(linear_model)
intercept <- coefficients[1]
slope <- coefficients[2]
equation <- paste0("y = ", round(intercept, 2), " + ", slope , "x")

bCancer <- read.csv("breastCancer.csv", header=T)
bCancer$log_texture_mean <- log(bCancer$texture_mean)
bCancer_texture <- c(bCancer$log_texture_mean)
ggplot(bCancer,aes(x=log_texture_mean,y=area_mean))+
  geom_point()+
  geom_smooth(method = "lm", se = FALSE)+
  labs(title=equation, x = "Mean Tumor Area", y = "Mean Tumor Texture")
```

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



The next exercise continues from the Week 8's practice problems. Here is the same code from class to get you started:

Use the following code to download the `parathyroidSE` package. This package provides RangedSummarizedExperiment objects of read counts in genes and exonic parts for paired-end RNA-Seq data from experiments on primary cultures of parathyroid tumors.

The sequencing was performed on tumor cultures from 4 patients at 2 time points over 3 conditions (DPN, OHT and control).

The data were presented in the article "Evidence of a Functional Estrogen Receptor in Parathyroid Adenomas" by Haglund F, Ma R, Huss M, Sulaiman L, Lu M, Nilsson IL, Hoog A, Juhlin CC, Hartman J, Larsson C, J Clin Endocrinol Metab. jc.2012-2484, Epub 2012 Sep 28, PMID: 23024189. The raw sequencing data provided by NCBI Gene Expression Omnibus is under accession number GSE37211.

```
# install.packages("BiocManager")
# BiocManager::install("parathyroidSE")
# BiocManager::install("DESeq2")
library("parathyroidSE")
```

```
## Loading required package: SummarizedExperiment

## Loading required package: MatrixGenerics

## Loading required package: matrixStats

##
## Attaching package: 'MatrixGenerics'
```

```
## The following objects are masked from 'package:matrixStats':
##
##     colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,
##     colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
##     colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
##     colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
##     colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
##     colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
##     colWeightedMeans, colWeightedMedians, colWeightedSds,
##     colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,
##     rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
##     rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
##     rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
##     rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
##     rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
##     rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
##     rowWeightedSds, rowWeightedVars

## Loading required package: GenomicRanges

## Loading required package: stats4

## Loading required package: BiocGenerics

## Loading required package: generics

##
## Attaching package: 'generics'

## The following objects are masked from 'package:base':
##
##     as.difftime, as.factor, as.ordered, intersect, is.element, setdiff,
##     setequal, union

##
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, aperm, append, as.data.frame, basename, cbind,
##     colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
##     get, grep, grepl, is.unsorted, lapply, Map, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rownames, sapply, saveRDS, table, tapply, unique,
##     unsplit, which.max, which.min

## Loading required package: S4Vectors
```

```
##
## Attaching package: 'S4Vectors'

## The following object is masked from 'package:utils':
##
##     findMatches

## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname

## Loading required package: IRanges

##
## Attaching package: 'IRanges'

## The following object is masked from 'package:grDevices':
##
##     windows

## Loading required package: GenomeInfoDb

## Loading required package: Biobase

## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.

##
## Attaching package: 'Biobase'

## The following object is masked from 'package:MatrixGenerics':
##
##     rowMedians

## The following objects are masked from 'package:matrixStats':
##
##     anyMissing, rowMedians

## Warning in fun(libname, pkgname): Package 'parathyroidSE' is deprecated and will be removed from
##   Bioconductor version 3.22
```

```
data("parathyroidGenesSE")
```

First, build the object using count and sample data.

```
# save count data into object
para <- assay(parathyroidGenesSE)

# save sample data into object
sample <- colData(parathyroidGenesSE)

# get names of samples using `run` column
colData(parathyroidGenesSE)$run
```

```
##  [1] "SRR479052" "SRR479053" "SRR479054" "SRR479055" "SRR479056" "SRR479057"
##  [7] "SRR479058" "SRR479059" "SRR479060" "SRR479061" "SRR479062" "SRR479063"
## [13] "SRR479064" "SRR479065" "SRR479066" "SRR479067" "SRR479068" "SRR479069"
## [19] "SRR479070" "SRR479071" "SRR479072" "SRR479073" "SRR479074" "SRR479075"
## [25] "SRR479076" "SRR479077" "SRR479078"
```

```
# save these sample names into columns
colnames(para) <- colData(parathyroidGenesSE)$run

# check all our objects: count matrix and sample data
# para
# sample
```

Next, we can construct our DESeq object. We will specify a function that tests for the effect of treatment while controlling for the patient.

```
library(DESeq2)
para_object <- DESeqDataSetFromMatrix(
  countData = para,
  colData = sample,
  design = ~ patient + treatment
)
```

Finally, let's subset from the full dataset only data from after 48 hours of treatment.

```
para_object_subset <- para_object[ , para_object$time == "48h" ]
```

4. Filter out the genes that contain 2 or less columns of count data. How many feature genes and samples are in this filtered subset? [0.75]

```
para_genes_subset <- para_object_subset[rowSums(para)>2,]

print(nrow(para_genes_subset))
```

```
## [1] 29142
```

```
print(ncol(para_genes_subset))
```

```
## [1] 14
```

```
#'There are 29142 feature genes and 14 samples after filtering genes that contain
#'atleast 3 or more counts
```

5. Run the differential expression function to fit this object using generalized linear models. [0.25]

```
DESeq2_para_genes_subset <- DESeq(para_genes_subset)
```

```
## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing
```

6. See the code below to get results for differentially expressed genes between the control and OHT treatment. Do the same for the DPN group. What are the top 5 differentially-expressed genes in the DPN group? How many genes are up- and down-regulated? [0.75]

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:Biobase':
##
##     combine

## The following objects are masked from 'package:GenomicRanges':
##
##     intersect, setdiff, union

## The following object is masked from 'package:GenomeInfoDb':
##
##     intersect

## The following objects are masked from 'package:IRanges':
##
##     collapse, desc, intersect, setdiff, slice, union

## The following objects are masked from 'package:S4Vectors':
##
##     first, intersect, rename, setdiff, setequal, union
```

```
## The following objects are masked from 'package:BiocGenerics':
##
##     combine, intersect, setdiff, setequal, union

## The following object is masked from 'package:generics':
##
##     explain

## The following object is masked from 'package:matrixStats':
##
##     count

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# differentially expressed genes between control and OHT treatment
results_OHT <- results(DESeq2_para_genes_subset, contrast = c("treatment", "Control", "OHT"))
# differentially expressed genes between control and DPN treatment
results_DPN <- results(DESeq2_para_genes_subset, contrast = c("treatment", "Control", "DPN"))

DPN_results_DF <- data.frame(genes = results_DPN@rownames,
                                  results_DPN@listData)
#Top 5 differentially expressed genes in DNP group based on p-value (significance)
top_genes <- DPN_results_DF %>%
  slice_min(pvalue,n=5)
#'The top genes are ENSG00000168542, ENSG00000044574, ENSG00000092621,
#'ENSG00000091137, and ENSG00000166598
print(top_genes)
```

```
##             genes  baseMean log2FoldChange      lfcSE      stat       pvalue
## 1 ENSG00000168542 1913.8578      0.4857226 0.05484462  8.856339 8.268482e-19
## 2 ENSG00000044574 4032.3633     -0.4933679 0.05685304 -8.677951 4.029640e-18
## 3 ENSG00000092621  459.9058     -0.8969675 0.11975996 -7.489711 6.902552e-14
## 4 ENSG00000091137  971.6163      0.6783086 0.09725872  6.974270 3.074622e-12
## 5 ENSG00000166598 4082.6300     -0.3210482 0.05308229 -6.048123 1.465432e-09
##           padj
## 1 9.494698e-15
## 2 2.313618e-14
## 3 2.642067e-10
## 4 8.826470e-09
## 5 3.348982e-06
```

```
#Number of genes up/down regulated
up_regulated <- DPN_results_DF %>%
  filter(DPN_results_DF$log2FoldChange>0)
down_regulated <- DPN_results_DF %>%
  filter(DPN_results_DF$log2FoldChange<0)
```

```r
#There are 13778 genes that are upregulated and 14380 genes that are downregulated
nrow(up_regulated)
```

```
## [1] 13778
```

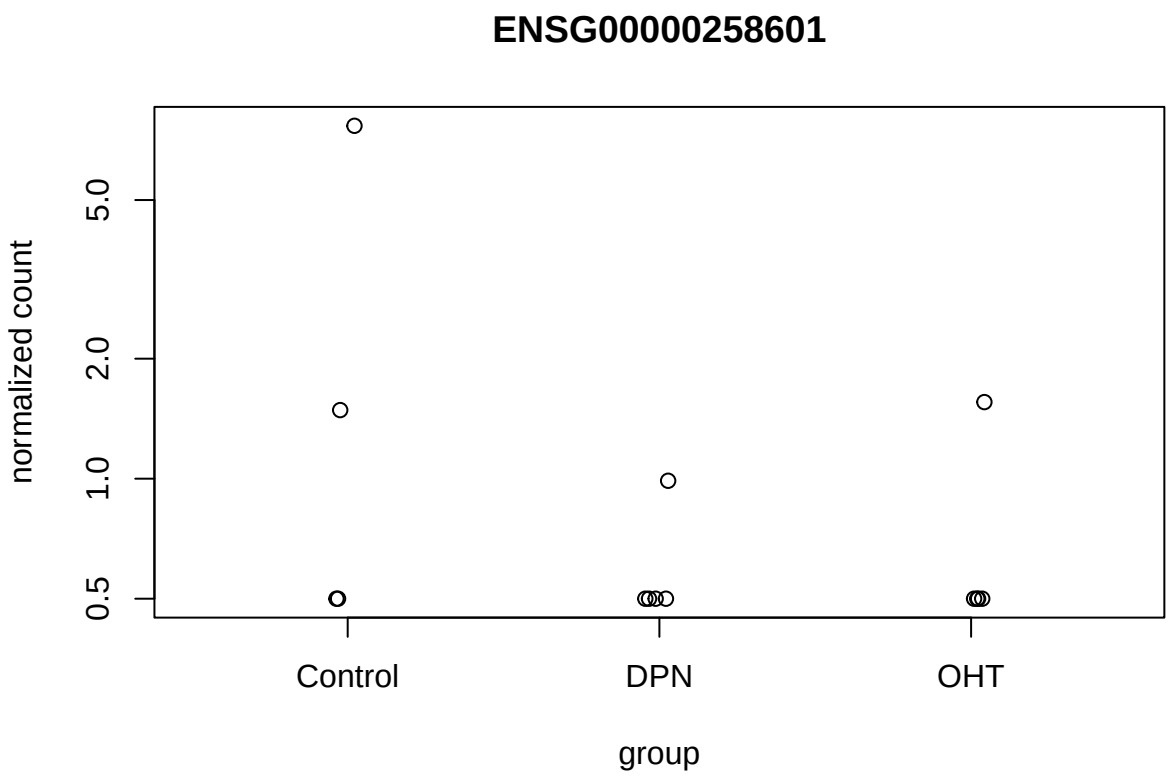
```r
nrow(down_regulated)
```

```
## [1] 14380
```

7. Compare the gene expression across treatments of the gene with the highest log-fold change. [0.5]

Hint: use `which.max()` to get the highest value for log-fold change.

```r
plotCounts(DESeq2_para_genes_subset,
           DPN_results_DF$genes[which.max(DPN_results_DF$log2FoldChange)], intgroup = "treatment")
```



8. Create a volcano plot of the differentially-expressed genes only in the DPN treatment using a p-value cutoff of 0.05. [0.75]

```r
library(ggplot2)

DPN_results_DF$diffexpressed <- "NO"
```

```
DPN_results_DF$diffexpressed[DPN_results_DF$log2FoldChange > 0.6 & DPN_results_DF$pvalue < 0.05] <- "UP
DPN_results_DF$diffexpressed[DPN_results_DF$log2FoldChange < -0.6 & DPN_results_DF$pvalue < 0.05] <- "D

DPN_results_DF <- DPN_results_DF %>%
  filter(!is.na(pvalue))
ggplot(DPN_results_DF, aes(x = log2FoldChange,
                           y = -log(pvalue),
                           color = diffexpressed))+
  geom_vline(xintercept = c(-0.6, 0.6), col = "gray", linetype = 'dashed') +
  geom_hline(yintercept = -log((0.05)), col = "gray", linetype = 'dashed') +
  geom_point(size = 2) +
  scale_color_manual(values = c("#00AFBB", "grey", "#bb0c00"),
                     labels = c("Downregulated", "Not significant", "Upregulated"))+
  labs(color = 'Expression', #legend_title,
       x = expression("log"[2]*"FC"), y = expression("-log"[10]*"p-value"),
  title = 'Parathyroid Gene expression after 24h treatment with DPN') # Plot title
```