Lab 3: Power and Sample Size

Ryan Yancey

17 June 2021

In this lab, we are working with a data set from Alizadeh et al. at Stanford. In this study, the investigators were evaluating diffuse large B-cell lymphoma (DLBCL). Using expression profiling and hierarchical clustering (a topic that we will visit later in this class), they were able to identify 2 distinct forms of DLBCL that indicate different stages of B-cell differentiation. "One type expressed genes characteristic of germinal centre B cells ('germinal centre B-like DLBCL'); the second type expressed genes normally induced during in vitro activation of peripheral blood B cells ('activated B-like DLBCL')." They also found that the germinal centre B-like DLBCL patients had a better survival rate.

We will use this data set to evaluate the power and sample size in this experiment. We will also look for the necessary number of samples to appropriately power the study. First we will calculate the power and n required using a single gene calculation for illustration of the formula, then we will conduct a more multivariate summary that gives an idea of the power or n required for a specific percentage of genes/probes in the experiment. Remember that when we calculate these statistics for a microarray, we are dealing with more than a single variable, so general power formulas do not apply when attempting to summarize all genes/probes on an array.

The paper entitled *Distinct type of diffuse large B-cell lymphoma identified by gene expression profiling* can be found here.

ABSTRACT Diffuse large B-cell lymphoma (DLBCL), the most common subtype of non-Hodgkin's lymphoma, is clinically heterogeneous: 40% of patients respond well to current therapy and have prolonged survival, whereas the remainder succumb to the disease. We proposed that this variability in natural history reflects unrecognized molecular heterogeneity in the tumours. Using DNA microarrays, we have conducted a systematic characterization of gene expression in B-cell malignancies. Here we show that there is diversity in gene expression among the tumours of DLBCL patients, apparently reflecting the variation in tumour proliferation rate, host response and differentiation state of the tumour. We identified two molecularly distinct forms of DLBCL which had gene expression patterns indicative of different stages of B-cell differentiation. One type expressed genes characteristic of germinal centre B cells ('germinal centre B-like DLBCL'); the second type expressed genes normally induced during in vitro activation of peripheral blood B cells ('activated B-like DLBCL'). Patients with germinal centre B-like DLBCL had a significantly better overall survival than those with activated B-like DLBCL. The molecular classification of tumours on the basis of gene expression can thus identify previously undetected and clinically significant subtypes of cancer.

1.) Download the Eisen DLBCL data set and save as a text file (go to class web site or see syllabus for paper URL).

```
(pre_unzip <- dir())
```

```
## [1] "eisen.zip"                "eisenClasses.zip"
## [3] "gedav-lab3.Rproj"         "lab-3.pdf"
## [5] "ryancey3-gedav-lab3.Rmd"
```

```
system(command = "unzip -o eisen.zip")
data1 <- setdiff(dir(), pre_unzip)
```

2.) Load into R, using read.table and arguments:

```
header=T
na.strings="NA"
blank.lines.skip=F
row.names=1
```

There are missing values in this data frame because we're working with cDNA data.

```
eisen <-
    read.table(
        file = data1,
        header = TRUE,
        na.strings = "NA",
        blank.lines.skip = FALSE,
        row.names = 1
    )
```

3.) Get the class label file "eisenClasses.txt" from the class web site and read it into R. Use the header=T argument.

```
system(command = "unzip -o eisenClasses.zip")
data2 <- setdiff(dir(), append(pre_unzip, data1))
meta <- read.table(file = data2, header = TRUE)
```

4.) Subset the data frame with the class labels and look at the positions so you know where one class ends and the other begins. Remember that 'subset' means to re-index (i.e. reorder) the column headers. If you look at the original column name order with dimnames(dat)[[2]] both before and after you reorder them, you will see what this has done.

```
# reorder the columns based on the order they appear in metadata
colnames(eisen)
```

```
##  [1] "DLCL.0001" "DLCL.0002" "DLCL.0003" "DLCL.0004" "DLCL.0005" "DLCL.0006"
##  [7] "DLCL.0007" "DLCL.0008" "DLCL.0009" "DLCL.0010" "DLCL.0011" "DLCL.0012"
## [13] "DLCL.0013" "DLCL.0014" "DLCL.0015" "DLCL.0016" "DLCL.0017" "DLCL.0018"
## [19] "DLCL.0020" "DLCL.0021" "DLCL.0023" "DLCL.0024" "DLCL.0025" "DLCL.0026"
## [25] "DLCL.0027" "DLCL.0028" "DLCL.0029" "DLCL.0030" "DLCL.0031" "DLCL.0032"
## [31] "DLCL.0033" "DLCL.0034" "DLCL.0036" "DLCL.0037" "DLCL.0039" "DLCL.0040"
## [37] "DLCL.0041" "DLCL.0042" "DLCL.0048" "DLCL.0049"
```

```
eisen <- eisen[, meta[,2]]
colnames(eisen)
```

```
##  [1] "DLCL.0012" "DLCL.0024" "DLCL.0003" "DLCL.0026" "DLCL.0023" "DLCL.0015"
##  [7] "DLCL.0010" "DLCL.0030" "DLCL.0034" "DLCL.0018" "DLCL.0032" "DLCL.0037"
## [13] "DLCL.0001" "DLCL.0008" "DLCL.0004" "DLCL.0029" "DLCL.0009" "DLCL.0020"
## [19] "DLCL.0033" "DLCL.0005" "DLCL.0011" "DLCL.0048" "DLCL.0027" "DLCL.0013"
## [25] "DLCL.0007" "DLCL.0028" "DLCL.0025" "DLCL.0021" "DLCL.0016" "DLCL.0002"
## [31] "DLCL.0017" "DLCL.0040" "DLCL.0014" "DLCL.0031" "DLCL.0039" "DLCL.0042"
## [37] "DLCL.0041" "DLCL.0049" "DLCL.0006"
```

```
# categorize based on their class
germinal_center <- meta[c(1:19), 2]
activated <- meta[c(20:39), 2]
```

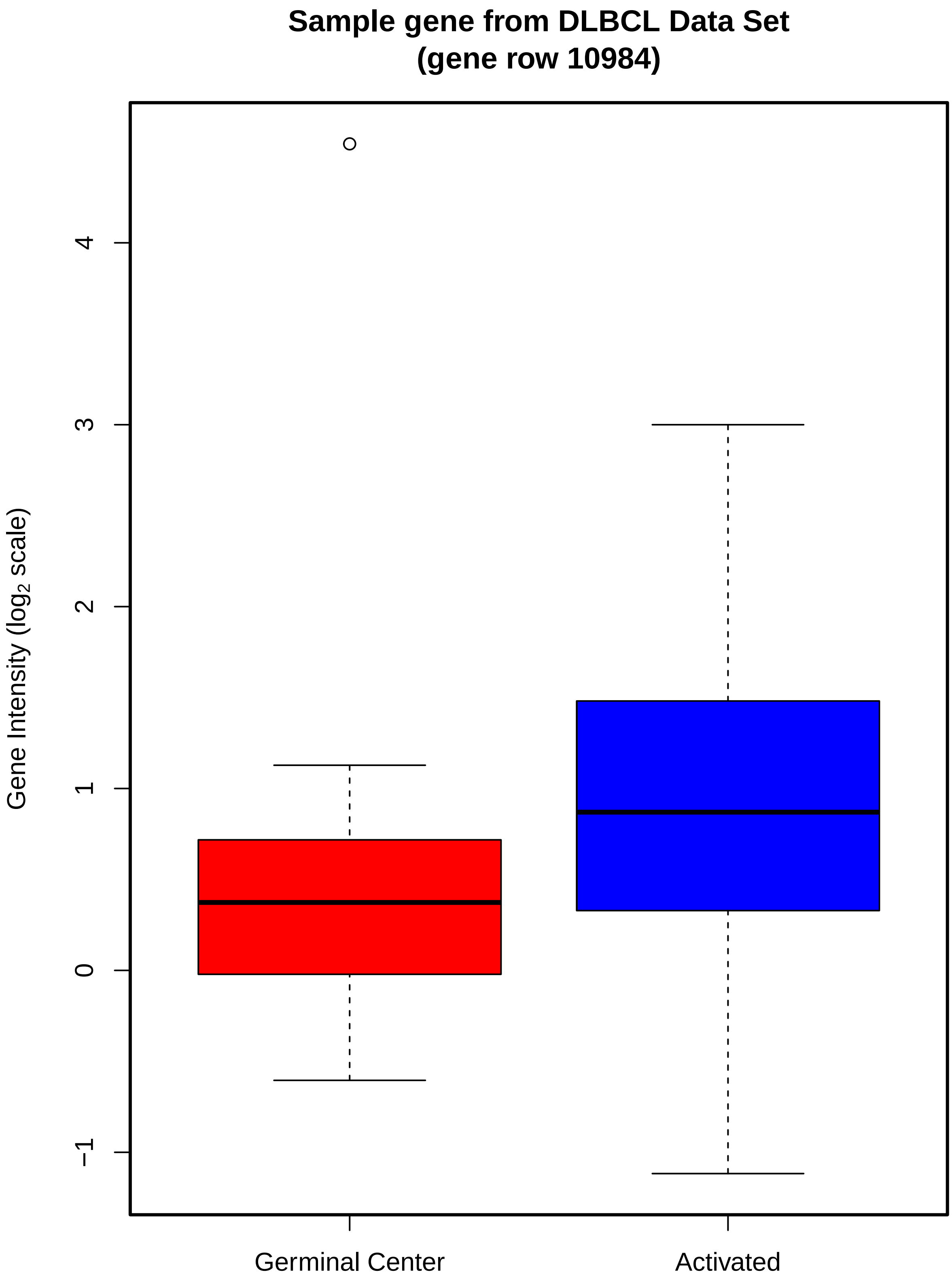
5.) Pick a gene, remove cells that have “NAs”, and plot the values for both classes with a:

- boxplot (use the argument col=c(“red”,”blue”) to color separate boxes)
- histogram (this should have 2 separate histogram plots on 1 page; use the par(mfrow=c(2,1)) function prior to plotting the first).

Color each class something different in the boxplot and histogram.

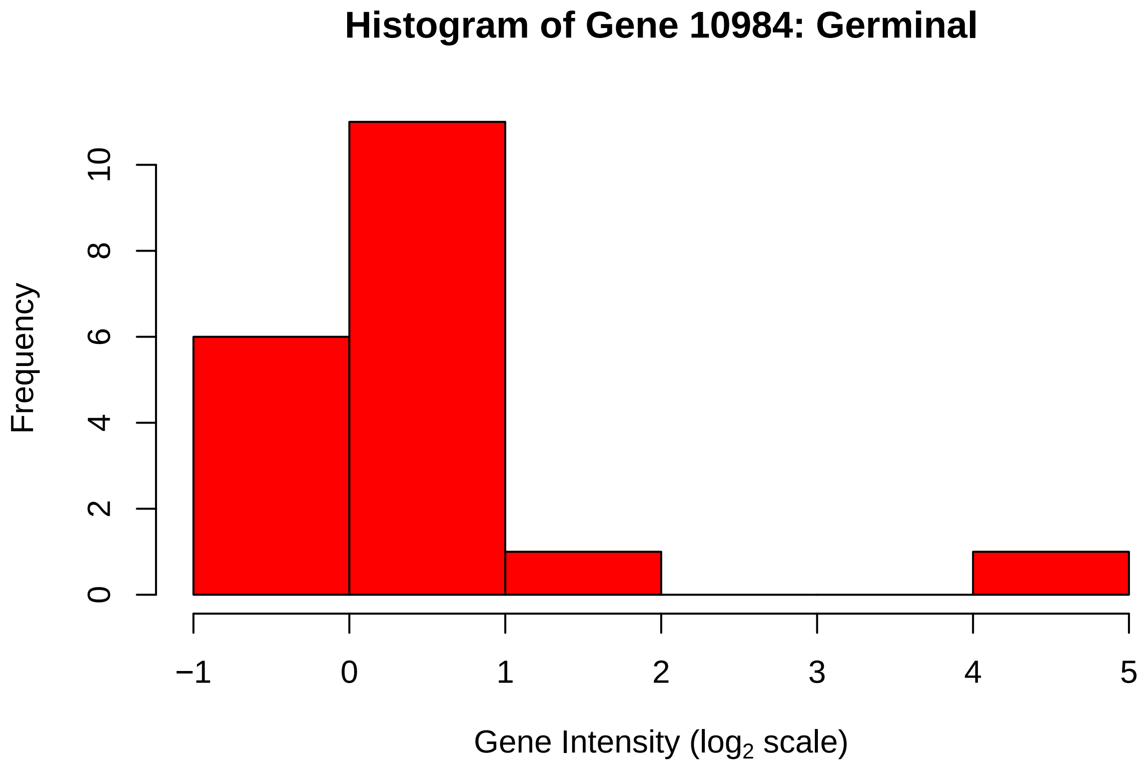
First, we can choose a random gene. Then, we’ll subset using that random number and the class categorization from the `eisen` data frame and remove `NA` values.

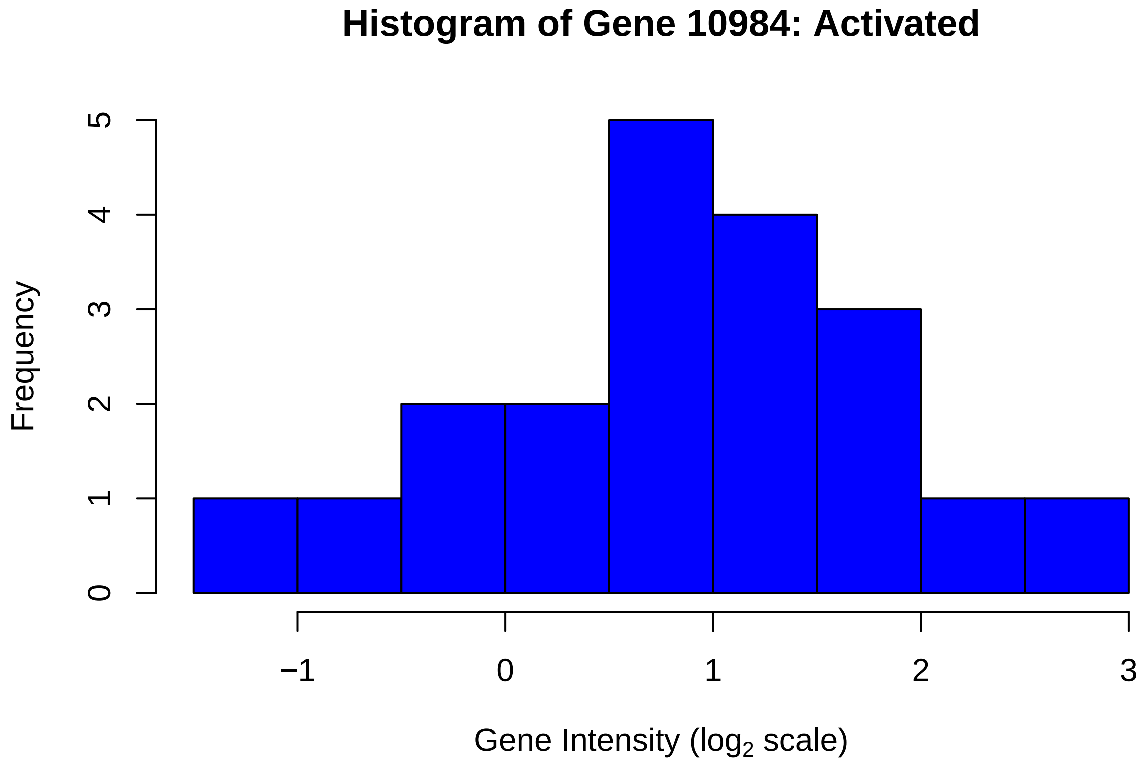
```
# choose a random gene
gene <- 10984

# subset the gene from eisen
x <- eisen[gene, germinal_center]
x <- x[!is.na(x)]
y <- eisen[gene, activated]
y <- y[!is.na(y)]

# make a list
xylist <- list(Germinal.Center = x[!is.na(x)], Activated = y[!is.na(y)])

# boxplot of random gene
boxplot(
  xylist,
  axes = FALSE,
  col = c("red", "blue"),
  ylab = expression(paste("Gene Intensity (", log[2], " scale)"))
)
box(lwd = 2)
axis(2)
axis(1,
     at = c(1, 2),
     labels = sub(".", " ", names(xylist), fixed = TRUE))
title(paste0("Sample gene from DLBCL Data Set\n(gene row ", gene, ")"))
```


Sample gene from DLBCL Data Set
(gene row 10984)
Gene Intensity (log2 scale)
−1
0
1
2
3
4
Germinal Center
Activated

Now we can plot the histogram, too.

```r
# histogram of random gene
par(mfrow = c(2, 1))
hist(
  xylist[["Germinal.Center"]],
  col = "red",
  xlab = expression(paste("Gene Intensity (", log[2], " scale)")),
  main = paste0("Histogram of Gene ", gene, ": Germinal")
)
hist(
  xylist[["Activated"]],
  col = "blue",
  xlab = expression(paste("Gene Intensity (", log[2], " scale)")),
  main = paste0("Histogram of Gene ", gene, ": Activated")
)
```


Histogram of Gene 10984: Germinal
Frequency
0
2
4
6
8
10
−1
0
1
2
3
4
5
Gene Intensity ($\log_2$ scale)


Histogram of Gene 10984: Activated
Frequency
0
1
2
3
4
5
−1
0
1
2
3
Gene Intensity ($\log_2$ scale)

6.) Calculate the pooled variance as coded in the lecture notes, and calculate the minimum sample size necessary to detect a 1.5 fold difference (at 80% power and 99% confidence).

```
# install.packages("pwr")
library("pwr")
nx <- length(x)
ny <- length(y)

# pooled variance
(pv <- (((nx - 1) * var(x)) + ((ny - 1) * var(y))) / (nx + ny - 2))
```

```
## [1] 1.130074
```

```
dif.1.5.fold <- log2(1.5)/sqrt(pv)

pl.ss.1.5 <-
  pwr.t.test(
    d = dif.1.5.fold,
    sig.level = 0.01,
    power = 0.8,
    type = "two.sample"
  )

# view the minimum sample size
round(pl.ss.1.5[["n"]])
```

```
## [1] 79
```

Note: the minimum sample size is *per sample*, so in reality, a total of 158, with 79 in each group, is minimally required.

7.) Now calculate the sample size required for the same gene selected in #5 using the empirically determined delta between the two groups, assuming 99% confidence and 80% power.

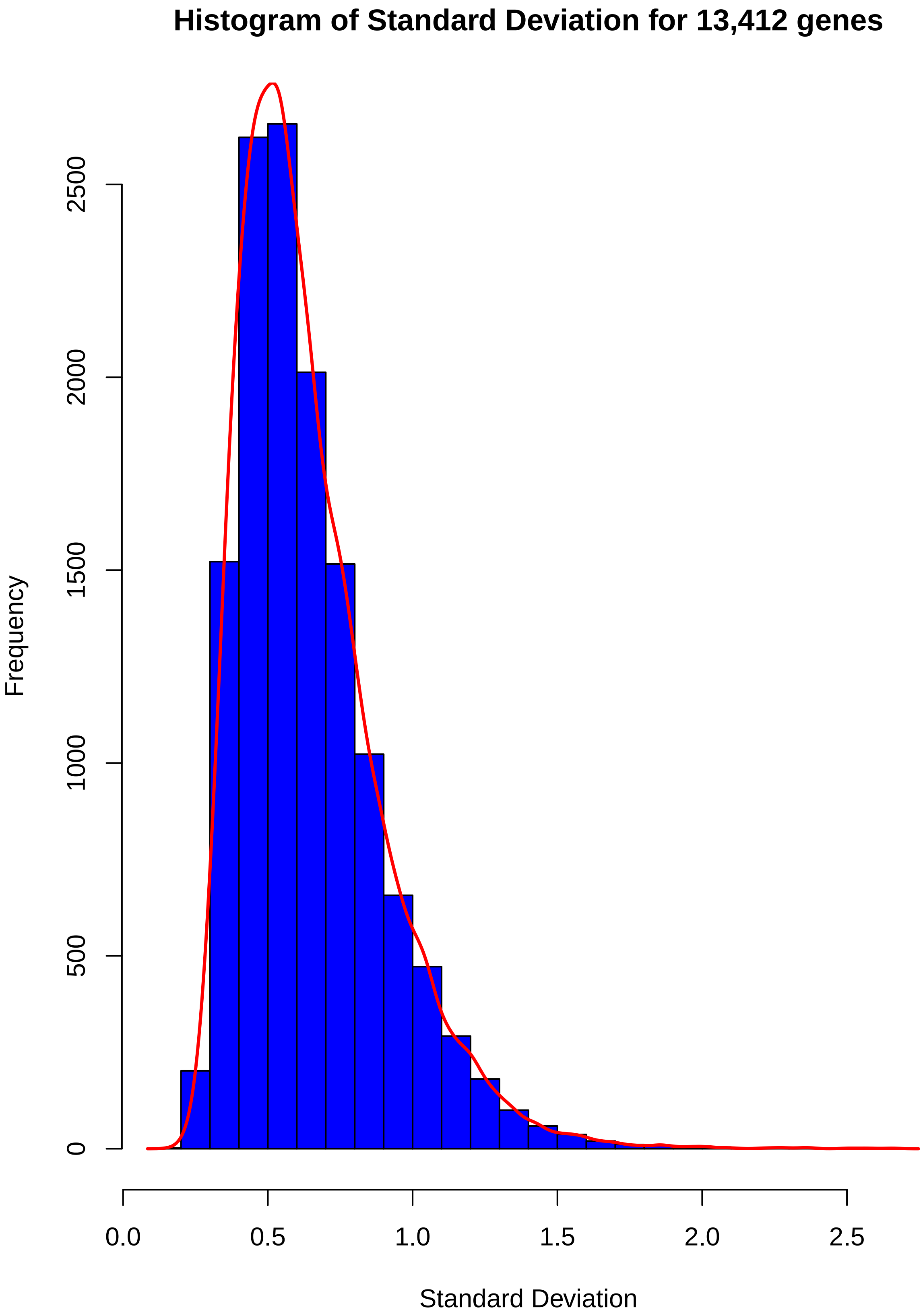
```r
dif.delta <- abs(mean(x) - mean(y)) / sqrt(pv)
pl.ss.delta <-
  pwr.t.test(
    d = dif.delta,
    sig.level = 0.01,
    power = 0.8,
    type = "two.sample"
  )
round(pl.ss.delta[["n"]])
```

```
## [1] 204
```

For the gene selected, the minimum sample size of 204 (for each group) would be required. Thus, a total of 408 is required.

8.) Now load the ssize and gdata libraries, calculate the standard deviation for each gene in the matrix (Hint: use the na.rm=T argument), and plot a histogram of the standard deviations. Label the plot accordingly.

```
suppressPackageStartupMessages(library("ssize"))
suppressPackageStartupMessages(library("gdata"))
# stdev for each gene (MAR=1 for rows)
mat.sd <- as.numeric(apply(
  X = eisen,
  MAR = 1,
  FUN = function(x) sd(x, na.rm = TRUE)
  ))
# histogram
hist(
  mat.sd,
  n = 30,
  col = "blue",
  # add title with comma-separated sample size
  main = paste(
    "Histogram of Standard Deviation for",
    format(length(mat.sd), big.mark = ",", scientific = FALSE),
    "genes",
    sep = " "
  ),
  xlab = "Standard Deviation"
)
# density line
d <- density(mat.sd)
lines(
  x = d$x,
  y = (d$y * par("usr")[4] / max(d$y)),
  col = "red",
  lwd = 2
)
```


Histogram of Standard Deviation for 13,412 genes
Frequency
0
500
1000
1500
2000
2500
0.0
0.5
1.0
1.5
2.0
2.5
Standard Deviation

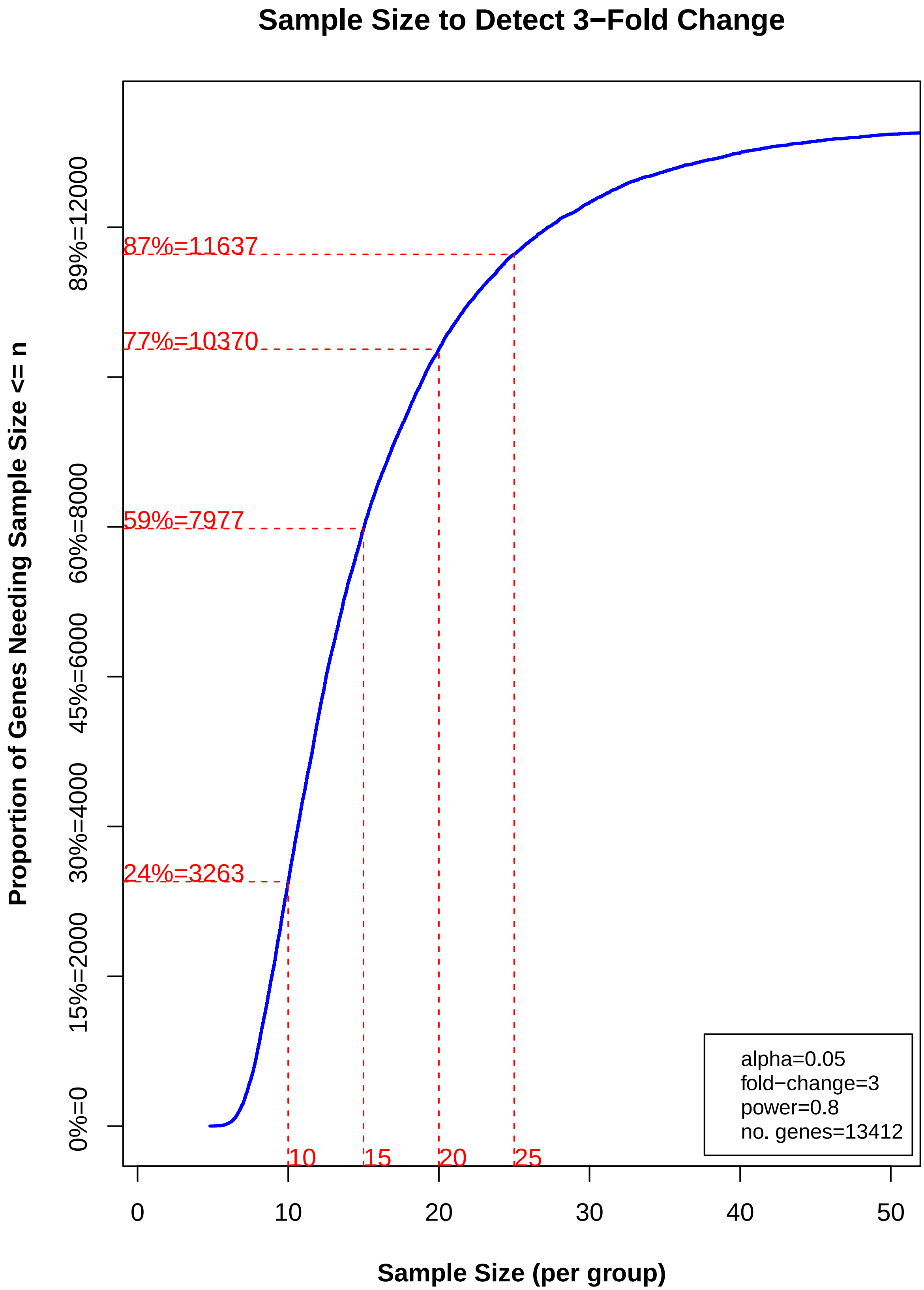
9.) Calculate and plot a proportion of genes vs. sample size graph to get an idea of the number of genes that have an adequate sample size for confidence=95%, effect size=3 (log2 transform for the function), and power=80%.

Calculate the required sample sizes for each sample using `ssize::ssize()`.

```
# stats interested in
p <- 0.05 # p-value
fc <- 3.0 # effect size
pow <- 0.80 # power

all.size <-
  ssize(
    sd = mat.sd,
    delta = log2(fc),
    sig.level = p,
    power = pow
  )
```

Now, we'll plot the power-vs-sample size data to gain an understanding of how the power of the experiments changes with respect to sample size.

```
ssize.plot(
  all.size,
  lwd = 2,
  col = "blue",
  xlim = c(1, 50),
  marks = c(10, 15, 20, 25),
  font.lab = 2
)
title(main = "Sample Size to Detect 3-Fold Change")
legend(
  "bottomright",
  legend = strsplit(
    paste0(
      "alpha=", p ,",",
      "fold-change=", fc ,",",
      "power=", pow, ",",
      "no. genes=", length(mat.sd)
    ), split = ",")[[1]],
  cex = 0.8,
  inset = 0.01
)
```


Sample Size to Detect 3−Fold Change
Proportion of Genes Needing Sample Size <= n
0%=0
15%=2000
30%=4000
45%=6000
60%=8000
89%=12000
87%=11637
77%=10370
59%=7977
24%=3263
10
15
20
25
0
10
20
30
40
50
Sample Size (per group)
alpha=0.05
fold−change=3
power=0.8
no. genes=13412