Lab3

Power and sample size

Jessica Temporal 7547611

August 29, 2016

Contents

1. Get the Eisen DLBCL data set.

```
file <- "eisen.txt"
```

2. Load into R, using read.table and arguments: `header=T`, `na.strings="NA"`, `blank.lines.skip=F`. There are missing values in this data frame because we're working with cDNA data. Make sure that you names the row names as the first column values and then remove this first column.

```
eisen_data <- read.table(file, header = T, na.strings = "NA", blank.lines.skip = F)
rownames(eisen_data) <- as.character(eisen_data$UID)
eisen_data$UID <- NULL
```

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

```
file2 <- "eisenClasses.txt"
eisen_classes <- read.table(file2, header = T)
```

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.

```
# eisen_classes$class
class_1 <- subset(eisen_classes, eisen_classes$class == 1)
eisen_c1 <- subset(eisen_data, select = class_1$sample)
class_2 <- subset(eisen_classes, eisen_classes$class == 2)
eisen_c2 <- subset(eisen_data, select = class_2$sample)
dimnames(eisen_data)[[2]]
```

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

```
dimnames(eisen_c1)[[2]]
```

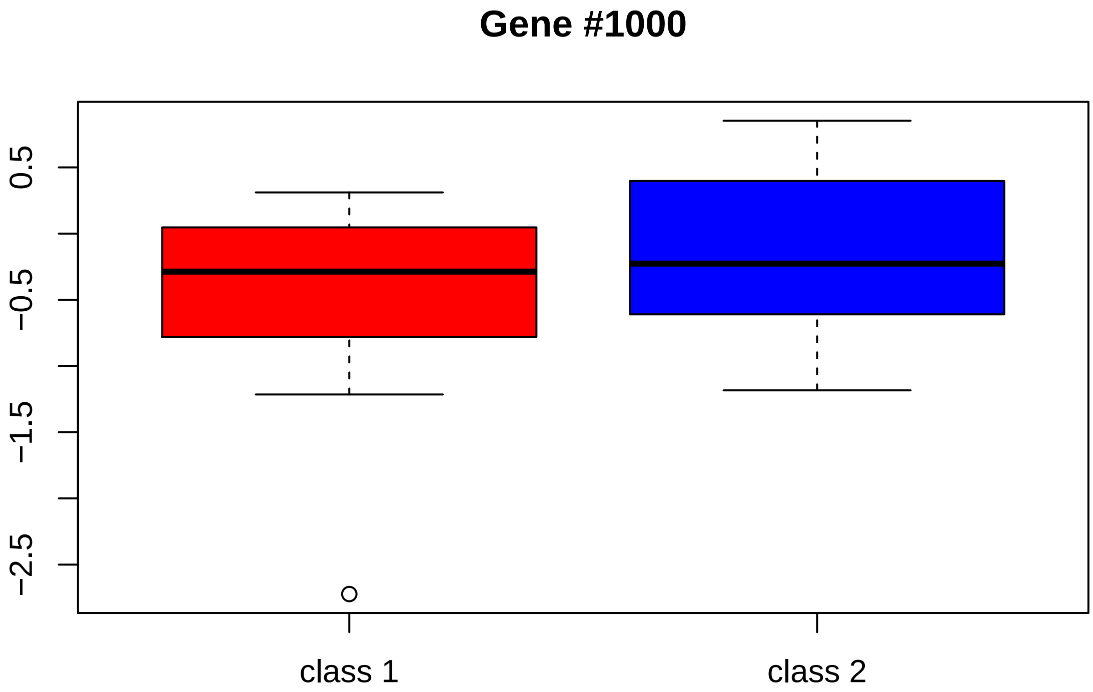
```
##  [1] "DLCL.0012" "DLCL.0024" "DLCL.0003" "DLCL.0026" "DLCL.0023"
##  [6] "DLCL.0015" "DLCL.0010" "DLCL.0030" "DLCL.0034" "DLCL.0018"
## [11] "DLCL.0032" "DLCL.0036" "DLCL.0001" "DLCL.0008" "DLCL.0004"
## [16] "DLCL.0029" "DLCL.0009" "DLCL.0020" "DLCL.0033"
```

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

```r
# gene 1000
gene <- "1000"
c1 <- as.numeric(eisen_c1[gene,])
c1 <- c1[!is.na(c1)]
c2 <- as.numeric(eisen_c2[gene,])
c2 <- c2[!is.na(c2)]
```

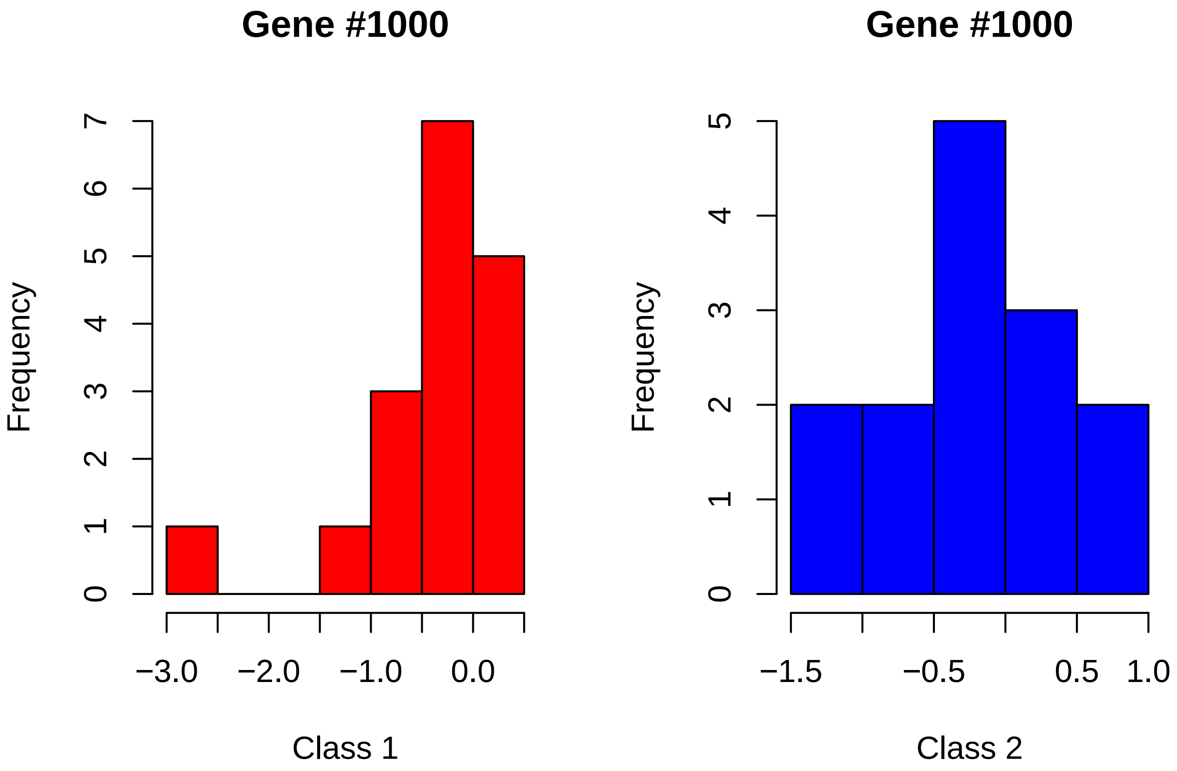
a) boxplot (use the argument `col=c("red", "blue")` to color separate boxes)

```r
boxplot(list(c1,c2), col = c("red", "blue"),
        main = "Gene #1000", names = c("class 1", "class 2"))
```



b) 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.

```r
par(mfrow=c(1,2))
hist(c1, col = "red", main = "Gene #1000", xlab = "Class 1")
hist(c2, col = "blue", main = "Gene #1000", xlab = "Class 2")
```



6. Calculate the standard deviation (sd) for both classes for the gene you chose, use the larger of the two, and calculate the minimum sample size necessary to detect a 1.5 fold difference (at 80% power and 99% confidence).

```r
c1_sd <- sd(c1)
c2_sd <- sd(c2)
power.t.test(delta = log(1.5),
             sd = max(c1_sd, c2_sd),
             power = 0.8,
             sig.level = 0.01)
```

```
##
##      Two-sample t test power calculation
##
##               n = 77.45077
##           delta = 0.4054651
##              sd = 0.7303008
##       sig.level = 0.01
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

7. Now calculate the power obtained when using the maximum number of replicates between the 2 classes for your gene (assuming 99% confidence). Set 'n' to the larger of the two classes. Also, start with the assumption that you want to detect a 2 fold difference between the two classes. Hint: `fold <- log(2)` (fold is now the value used for the 'delta' argument).

```r
power.t.test(n = max(length(c1), length(c2)),
             delta = log(2),
             sig.level = 0.01,
             sd = max(c1_sd, c2_sd))
```

```
##
##      Two-sample t test power calculation
##
##               n = 17
##           delta = 0.6931472
##              sd = 0.7303008
##       sig.level = 0.01
##           power = 0.5191489
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```