Stat 30280 - Assignment 4

Daniel Williams 21203054

18/04/2022

Exercise 1

Loading the data using the code provided in the assignment.

```
prostmat=as.matrix(read.csv("http://hastie.su.domains/CASI_files/DATA/prostmat.csv"))
prostmat=prostmat[1:200,]
cases=c(rep(FALSE,50),rep(TRUE,52))
```

Part a

Assume gene activity is normally distributed with mean mu, and variance sigma squared.
Hypothesis.
H0: mu_control = mu_case.
HA: mu_control != mu_case. Separating the control results, and case results by selecting correct columns.
Testing them for an equal mean assuming equal variance, with a 2 sample t test.

```
g1_control <- prostmat[1,1:50]
g1_case   <- prostmat[1,51:102]
t_test <- t.test(g1_control, g1_case, var.equal = TRUE)
t_test
```

```
##
##  Two Sample t-test
##
## data:  g1_control and g1_case
## t = -1.4812, df = 100, p-value = 0.1417
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.9222718  0.1338032
## sample estimates:
##  mean of x  mean of y
## -0.1860100  0.2082243
```

P value: 0.1417 > 0.05 (alpha) significance level. Confidence interval includes zero. Fail to reject H0. No evidence to suggest that the difference is not equal to zero.

Part b

Firstly setting my empty vector which will hold my results.
Looping over the 200 rows performing the same operation as in 1a.
Inside the loop, I select each gene in turn, separate case & control, then perform a t.test as per 1a.
The resulting p value is stored in the all_genes vector created earlier.
Finally returning the row numbers which have a p value of < 0.05, thus ones where we would reject H0.

```
all_genes <- rep(NA, 200)
rows <- 1:200

for (b in rows) {
  b_control <- prostmat[b,1:50]
  b_case <- prostmat[b,51:102]
  b_test <- t.test(b_control, b_case, var.equal = TRUE)
  all_genes[b] <- b_test$p.value
  }

genes_different <- which(all_genes < 0.05)
genes_different
```

```
##  [1]   2  11  26  35  37  38  44  55  61  68  72  73  74  78  79  81  85  90  98
## [20] 126 130 134 144 152 190
```

Part c

Controlling for the false discovery rate, to allow us to account for the fact that completing 200 tests in this scenario would originally result in an expectation that 10 tests would be type 1 errors.
Firstly calculating the FDR threshold.
Using the Benjamini-Hochberg Algorithm.
Ordering the p-values from lowest to highest.
Rejecting all H0 where sorted p values are less than FDR threshold.
Retaining all with sorted p values higher than FDR threshold.

```
FDR=function(N,q=0.1) q*(1:N)/N
FDR_Threshold <- FDR(200)

BHA <- sort(all_genes)
TF_BHA_FDR <- BHA <= FDR_Threshold
sum(TF_BHA_FDR)
```

```
## [1] 2
```

```
which(all_genes <= BHA[2])
```

```
## [1]  2 11
```

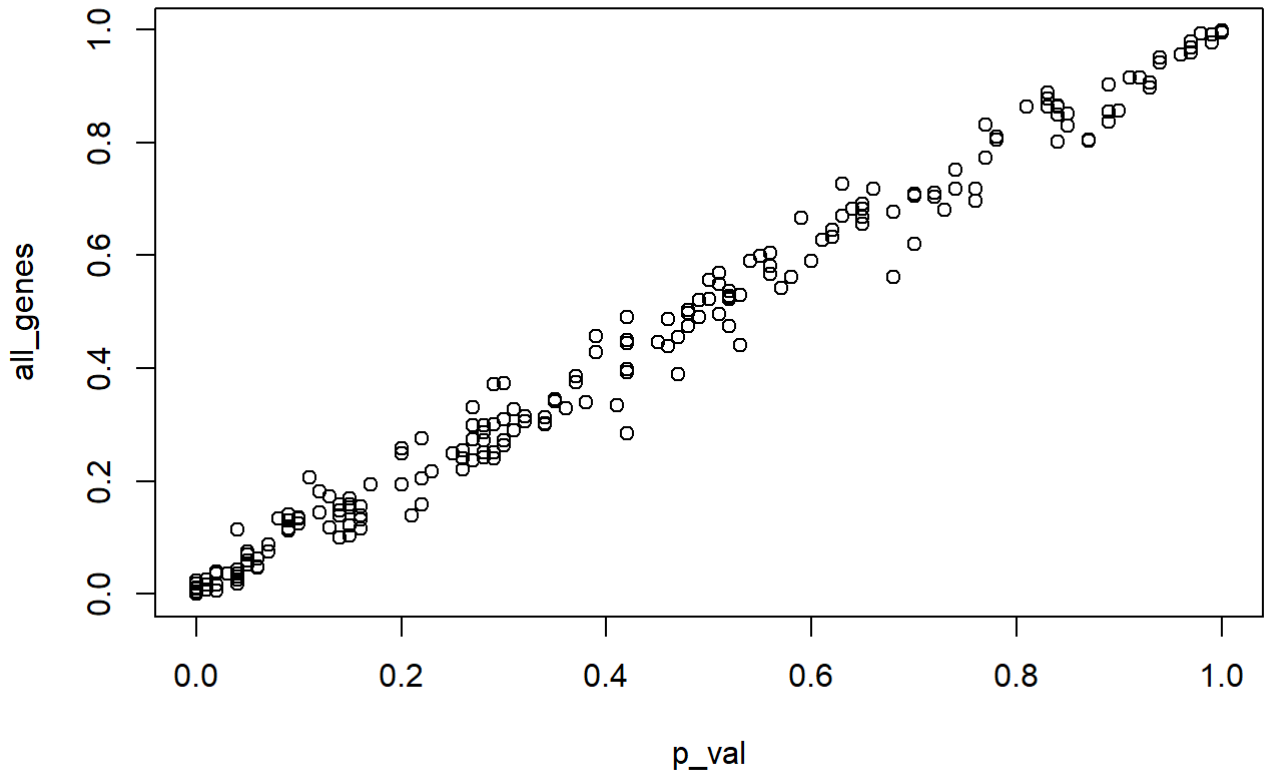
Part d

Commented the code due to the complex nature.

```
set.seed(436) # ensuring I get replicable results
B <- 100 # 100 permutations
mu_r_delta <- rep(NaN, 100) # setting empty vector for results
all_control <- prostmat[,1:50] # separating control & case data
all_case <- prostmat[,51:102]
mu_delta_obs <- rowMeans(all_control) - rowMeans(all_case) # observed average mu delta by gen
e
p_val <- rep(NaN, 200) # empty p value vector for results

for (r in rows) {
  gene_r <- (prostmat[r,])  # selecting the gene (looping through all 200)

  for (w in 1:B) { # for each gene, looping through 100 permutations
    gene_r_samp <- sample(gene_r)  # taking a random sample within this gene
    gene_r_cont <- gene_r_samp[cases == FALSE] # separating control & case data 50 & 52
    gene_r_case <- gene_r_samp[cases == TRUE] # using case vector from the top
    mu_r_delta[w] <- mean(gene_r_cont) - mean(gene_r_case)
    # calculating if observed or sample mu delta is higher
    p_val[r] <- sum(abs(mu_r_delta) > abs(mu_delta_obs[r])) / B
    # summing out of the 100 iterations how many times the mu delta within the permutation wa
s above the observed result, and normalising this by dividing by 100 iterations.
  }
}

comp <- cbind(p_val, all_genes)
# observing numerically the similarity of the 200 p values, one for each gene, and one for ea
ch testing methodology.
head(comp)
```

```
##      p_val    all_genes
## [1,]  0.09 0.1416871081
## [2,]  0.00 0.0003819891
## [3,]  0.97 0.9779276803
## [4,]  0.26 0.2546974208
## [5,]  0.83 0.8881183896
## [6,]  0.38 0.3397385064
```

```
# observing graphically the similarity of result from the two methods
# we see a reasonable y=x trend, showing the two methods produce similar results
plot(p_val, all_genes)
```




```
# checking the correlation using the inbuilt r function cor
# high correlation value again backs up the methods similarity.
cor(p_val, all_genes)
```

```
## [1] 0.9926025
```

Part e

Adapting my code to calculate median differences rather than mean.
Code chunks very similar, but I will comment the differences to part d.

```
set.seed(436)
library(matrixStats)  # package needed to utilise rowMeadians function
```

```
## Warning: package 'matrixStats' was built under R version 4.1.1
```

```
med_delta_obs <- rowMedians(all_control) - rowMedians(all_case) # median delta observed
p_val_med <- rep(NaN, 200) # empty median p value vector
med_r_delta <- rep(NaN, 100) # empty vector for use in the for loop

for (r in rows) {
  gene_r <- (prostmat[r,])  # as above

  for (w in 1:B) {
    gene_r_samp <- sample(gene_r)
    gene_r_cont <- gene_r_samp[cases == FALSE]
    gene_r_case <- gene_r_samp[cases == TRUE]
    med_r_delta[w] <- median(gene_r_cont) - median(gene_r_case)  # median delta calc
    p_val_med[r] <- sum(abs(med_r_delta) > abs(med_delta_obs[r])) / B
  }
}

genes_different_med <- which(p_val_med < 0.05) # finding genes which have a p value of less t
han 0.05 (alpha)
genes_different_med # the genes for which we would reject H0, that there is no difference in
 medians between case & control patients
```

```
##  [1]   2  11  17  24  34  35  37  38  44  50  52  55  69  73  74  77  81  82  83
## [20] 144 152 162 167 181 189 190 196
```

Part f

```
set.seed(436) # repeatable results
BHA_med <- sort(p_val_med) # Using the Benjamini-Hochberg Algorithm
TF_BHA_med_FDR <- BHA_med <= FDR_Threshold # using FDR threshold from above, to find genes th
at pass the threshold
sum(TF_BHA_med_FDR) # summing the number of genes which meet this requirement
```

```
## [1] 11
```

```
which(all_genes <= BHA[11])
```

```
##  [1]  2 11 26 35 37 44 68 73 78 90 98
```

```
# using the result of 11 from the sum, to get the gene numbers of the genes which we would re
ject the null hypothesis
```

Part b - t tests on means, 25 genes rejecting H0

Part c - t tests correcting for FDR, 2 genes rejecting H0.
Part e - medians permutation test, 27 genes rejecting H0.
Part f - medians permutation, correcting for FDR, 11 genes rejecting H0.

We see that for the medians, we reject a far greater amount of H0's, compared to the mean once we have corrected for FDR.
We would need to interrogate the data more to see which of these measures would be most appropriate in this scenario. We have not done any anyalsis on the spread of data, nor the skewness to inform this decision.

Thank you