BUSN 41201 HW1

Group 11

2024-03-22

```
# ***** AMAZON REVIEWS

# READ REVIEWS

data<-read.table("Review_subset.csv",header=TRUE)
dim(data)
```

```
## [1] 13319     9
```

```
# 13319 reviews
# ProductID: Amazon ASIN product code
# UserID:  id of the reviewer
# Score: numeric from 1 to 5
# Time: date of the review
# Summary: text review
# nrev: number of reviews by this user
# Length: length of the review (number of words)

# READ WORDS

words<-read.table("words.csv")
words<-words[,1]
length(words)
```

```
## [1] 1125
```

```
#1125 unique words

# READ text-word pairings file

doc_word<-read.table("word_freq.csv")
names(doc_word)<-c("Review ID","Word ID","Times Word" )
# Review ID: row of the file  Review_subset
# Word ID: index of the word
# Times Word: number of times this word occurred in the text


# We'll do 1125 univariate regressions of
# star rating on word presence, one for each word.
# Each regression will return a p-value, and we can
# use this as an initial screen for useful words.

# Don't worry if you do not understand the code now.
# We will go over similar code in  the class in a few weeks.
```

```
# Create a sparse matrix of word presence


library(gamlr)
```

```
## Warning:   'gamlr' R 4.3.3
```

```
##      Matrix
```

```
spm<-sparseMatrix(i=doc_word[,1],
                  j=doc_word[,2],
                  x=doc_word[,3],
                  dimnames=list(id=1:nrow(data),words=words))

dim(spm)
```

```
## [1] 13319  1125
```

```
# 13319 reviews using 1125 words

# Create a dense matrix of word presence

P <- as.data.frame(as.matrix(spm>0))

library(parallel)

margreg <- function(p){
    fit <- lm(stars~p)
    sf <- summary(fit)
    return(sf$coef[2,4])
}

# The code below is an example of parallel computing
# No need to understand details now, we will discuss more later

cl <- makeCluster(detectCores())

# Pull out stars and export to cores

stars <- data$Score

clusterExport(cl,"stars")

# Run the regressions in parallel

mrgpvals <- unlist(parLapply(cl,P,margreg))

# If parallel stuff is not working,
# you can also just do (in serial):
# mrgpvals <- c()
# for(j in 1:1125){
#   print(j)
#   mrgpvals <- c(mrgpvals,margreg(P[,j]))
# }
# make sure we have names
```

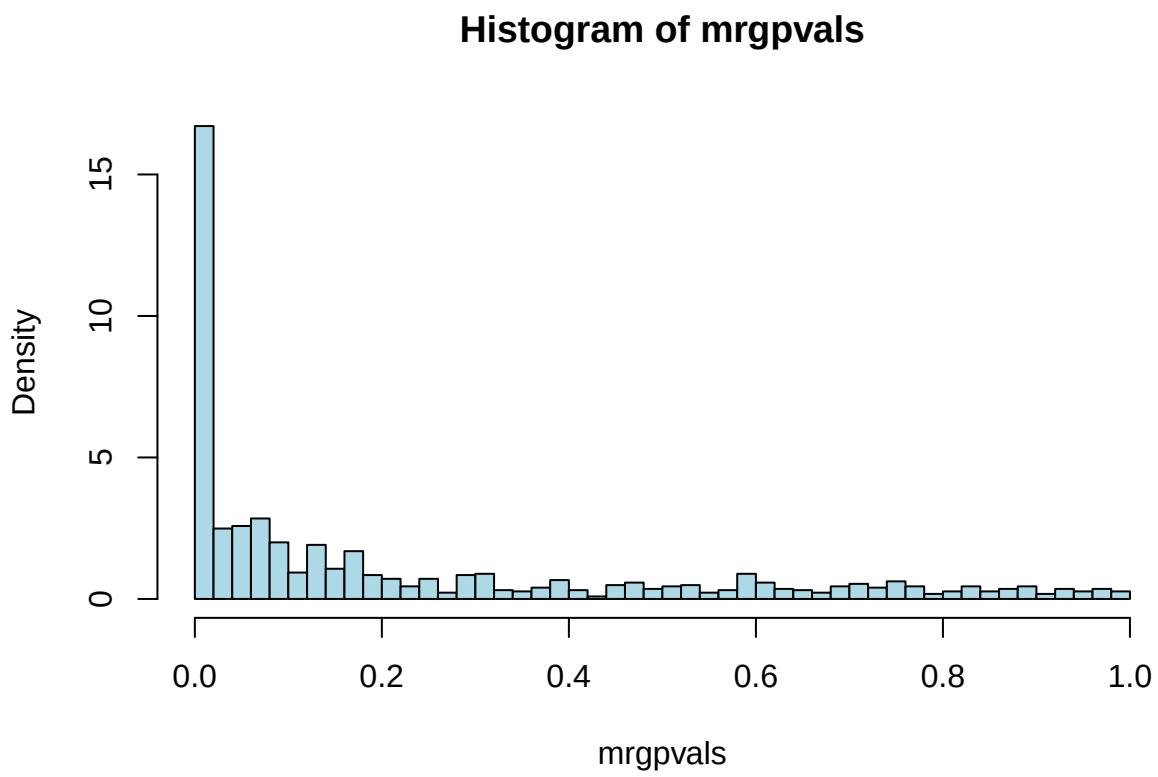
```r
names(mrgpvals) <- colnames(P)

# The p-values are stored in mrgpvals
```

Questions

(1) Plot the p-values and comment on their distribution. (1 point)

```r
# Plot density histogram of p-values
hist(mrgpvals, freq = FALSE, col = 'lightblue', breaks=50)
```



From the graph, a high peak around 0.0 is observed. This shows that we have found many words significant.

(2) Let's do standard statistical testing. How many tests are significant at the alpha level 0.05 and 0.01? (1 point)

```r
sig_0.05 <- sum(mrgpvals < 0.05)
sig_0.01 <- sum(mrgpvals < 0.01)
print(paste('The number of significant tests at 0.05:', sig_0.05))
```

```
## [1] "The number of significant tests at 0.05: 461"
```

```r
print(paste('The number of significant tests at 0.01:', sig_0.01))
```

```
## [1] "The number of significant tests at 0.01: 348"
```

(3) What is the p-value cutoff for 1% FDR? Plot and describe the rejection region. (1 point)

```
fdr_cut <- function(pvals, q){
  pvals <- pvals[!is.na(pvals)]
  N <- length(pvals)

  k <- rank(pvals, ties.method="min")
  alpha <- max(pvals[ pvals<= (q*k/N) ])

  return(alpha)
}

# @ 1% FDR

cutoff1 <- fdr_cut(mrgpvals,q=.01)

print(cutoff1)
```

```
## [1] 0.002413249
```

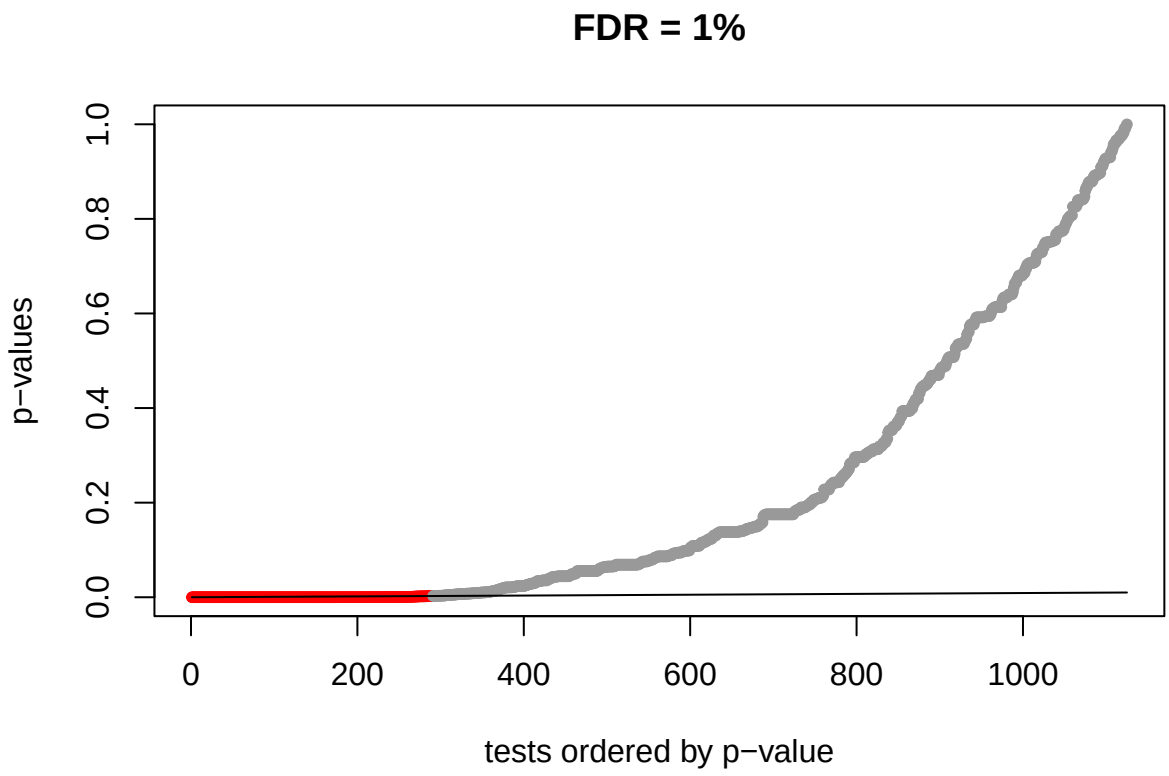
```
print(sum(mrgpvals<=cutoff1))
```

```
## [1] 290
```

```
## visualize the B+H FDR algorithm

sig <- factor(mrgpvals<=cutoff1)

o <- order(mrgpvals)
N <- length(mrgpvals)
plot(mrgpvals[o], col=c("grey60","red")[sig[o]], pch=20,
     ylab="p-values", xlab="tests ordered by p-value", main = 'FDR = 1%')
lines(1:N,  0.01*(1:N)/N)
```



(4) How many discoveries do you find at q=0.01 and how many do you expect to be false? (1 point)

```
# @ 1% FDR
cutoff1 <- fdr_cut(mrgpvals,q=.01)
print(cutoff1)
```

```
## [1] 0.002413249
```

```
print(sum(mrgpvals<=cutoff1))
```

```
## [1] 290
```

290 significant p-values are found at q=0.01. Out of these significant tests, around 3 are expected to be false discovery.

(5) What are the 10 most significant words? Do these results make sense to you? What are the advantages and disadvantages of our FDR anaysis? (1 point)

```
# top 10 words to investigate
names(mrgpvals)[order(mrgpvals)[1:10]]
```

```
##  [1] "not"          "horrible"     "great"        "bad"          "nasty"
##  [6] "disappointed" "new"          "but"          "same"         "poor"
```

The top ten words are “not, horrible, great, bad, nasty, disappointed, new, but, same, poor”.

These words make a lot of sense. “not, horrible, bad, nasty, disappointed, poor” are more related to negative review, whereas “great, new” are more related to positive reviews.

Advantages of our FDR analysis: We conducted multiple tests correction by choosing a relatively small FDR value to ensure that only a very small portion of our result is due to false discovery. In our case, around 3 out of 290 tests.

Disadvantages of our FDR analysis: The number of observations could be expanded more so that our result would be more reliable.