HW1_final

2024-03-24

```
# ***** 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.
```

```r
# We will go over similar code in  the class in a few weeks.

# Create a sparse matrix of word presence

library(gamlr)
```

```
## Loading required package: Matrix
```

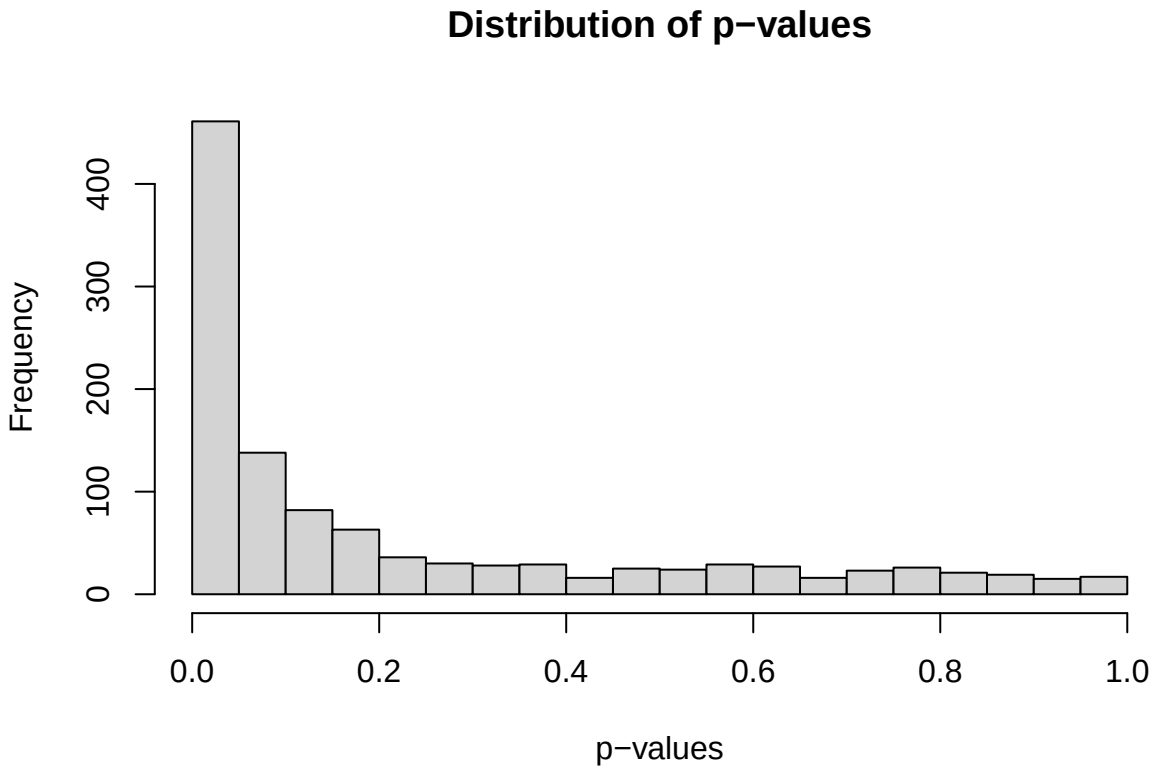
```r
spm<-sparseMatrix(i=doc_word[,1],
                  j=doc_word[,2],
                  x=doc_word[,3],
                  dimnames=list(id=1:nrow(data),words=words))
dim(spm)
```

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

```r
# 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
names(mrgpvals) <- colnames(P)
# The p-values are stored in mrgpvals
```

1.

```r
hist(mrgpvals, breaks = 30, main = "Distribution of p-values", xlab = "p-values")
```



The frequency of p-values are skewed right, which implies that the values are generally positive.

2.

```
alpha_005 <- sum(mrgpvals < 0.05)
alpha_001 <- sum(mrgpvals < 0.01)
alpha_005
```

```
## [1] 461
```

```
alpha_001
```

```
## [1] 348
```

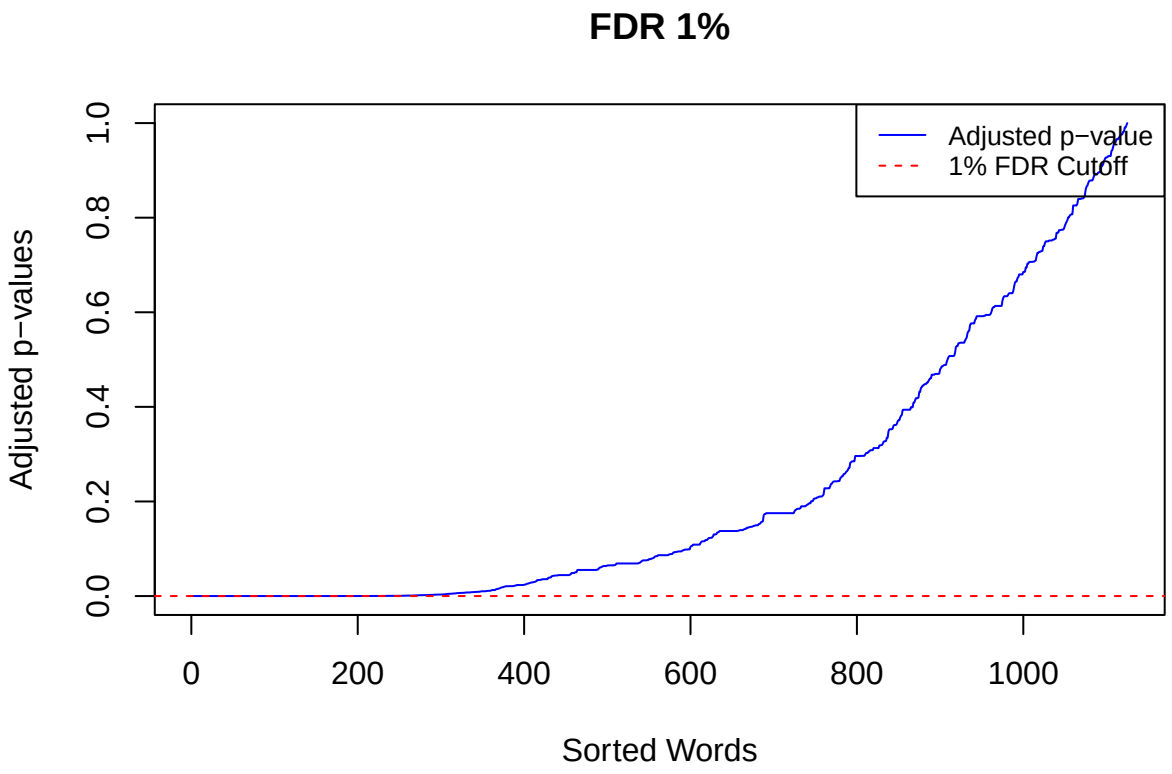
461 significant tests at the alpha level of 0.05 (alpha_005). 348 significant tests at the alpha level of 0.01 (alpha_001).

3.

```
sorted_pvalues <- sort(mrgpvals)
m <- length(sorted_pvalues)
alpha <- 0.01
critical_value <- min((which(sorted_pvalues <= (1:m) / m * alpha)))
pvalue_cutoff <- sorted_pvalues[critical_value]
cat("P-value cutoff for 1% FDR:", pvalue_cutoff, "\n")
```

```
## P-value cutoff for 1% FDR: 1.568776e-165
```

```r
plot(1:m, sorted_pvalues, type = "l", col = "blue", xlab = "Sorted Words", ylab = "Adjusted p-values", m
abline(h = pvalue_cutoff, col = "red", lty = 2)
legend("topright", legend = c("Adjusted p-value", "1% FDR Cutoff"), col = c("blue", "red"), lty = c(1, 1
```



4.

```r
sorted_pvals <- sort(mrgpvals)
m <- length(sorted_pvals)
fdr_level <- 0.01
bh_critical_values <- (1:m / m) * fdr_level
cutoff_index <- max(which(sorted_pvals < bh_critical_values))
num_discoveries <- cutoff_index
expected_false_discoveries <- num_discoveries * (1 - fdr_level)
cat("Number of discoveries at q = 0.01:", num_discoveries, "\n")
```

```
## Number of discoveries at q = 0.01: 290
```

The output appears to be reasonable and consistent with the principles of False Discovery Rate (FDR) control. The number of discoveries at the specified FDR threshold of q = 0.01 is 290, indicating the total count of significant findings identified in the dataset. Additionally, the expected number of false discoveries at this FDR threshold is approximately 287.1. This suggests that while a substantial number of discoveries are deemed significant, the proportion of false discoveries remains relatively low, in line with the intended control of Type I errors under the FDR framework. Therefore, the output indicates that the approach used to calculate the number of discoveries and expected false discoveries effectively manages the balance between identifying true positives and controlling the rate of false positives within the dataset. Based on this information, it indicates that a large proportion of the discoveries are expected to be false positives relative to the total number of discoveries identified. In other words, the majority of the significant findings are likely to be incorrect or spurious, which may imply that the FDR is relatively high in this scenario.

5.

```
top_10_indices <- order(mrgpvals)[1:10]
top_10_words <- words[top_10_indices]
top_10_words
```

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

The 10 most significant words are “not”, “horrible”, “great”, “bad”, “nasty”, “disappointed”, “new”, “but”, “same”, and “poor”. These words reflect a spectrum of opinions commonly found in consumer feedback, ranging from positive attributes like “great” to negative descriptors such as “horrible” and “disappointed”, but more tilted towards negative descriptors. The presence of words like “not” and “but” underscores the importance of negation and contrast in shaping opinions. The results of the False Discovery Rate (FDR) analysis appear reasonable, capturing both positive and negative sentiments expressed in the reviews.