Data Setup (copied from HW1.Rmd with extraneous stuff removed)

```
# READ REVIEWS
data = read.table("~/Desktop/Code/BUSN Assignments/HW1/Data/Review_subset.csv",header=TRUE)
dim(data)
```

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

```
colnames(data)
```

```
## [1] "ProductId"     "UserId"        "Score"         "Time"
## [5] "Summary"       "Nrev"          "Length"        "Prod_Category"
## [9] "Prod_Group"
```

```
# READ WORDS
words = read.table("~/Desktop/Code/BUSN Assignments/HW1/Data/words.csv")
words = words[,1]
length(words)
```

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

```
# READ text-word pairings file
doc_word = read.table("~/Desktop/Code/BUSN Assignments/HW1/Data/word_freq.csv")
colnames(doc_word) = c("Review ID","Word ID","Times Word")
```

```
# Create a matrix of word presence

spm = matrix(0L,nrow = nrow(data),ncol=length(words))
for (index in 1:nrow(doc_word)) {
  i = doc_word[index,1]
  j = doc_word[index,2]
  spm[i,j] = doc_word[index,3]
}
colnames(spm) = words
dim(spm)
```

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

```
P = as.data.frame(as.matrix(spm>0))

stars = data$Score
```

```
margreg = function(p){
    fit = lm(stars~P[[p]])
    sf = summary(fit)
    return(sf$coef[2,4])
}
margreg(10) #the pval for the 10th word
```

```
## [1] 0.004430345
```

```
library(parallel) # BASE R package. Should come pre-installed
ncores = detectCores()-1 #good form on your own computer to lose 1
ncores
# Make a cluster
cl = makeCluster(ncores)
# Export data to the cluster
```

```
clusterExport(cl,c("stars","P"))
# Run the regressions in parallel
# The same syntax as sapply, but first we tell it the cluster name.
pvals = parSapply(cl,1:length(words),margreg)
# About 2 seconds on my computer w/ 7 cores.
# Turn off cluster
stopCluster(cl)
```

```
pvals = sapply(1:length(words),margreg)
```

```
names(pvals) = words
```

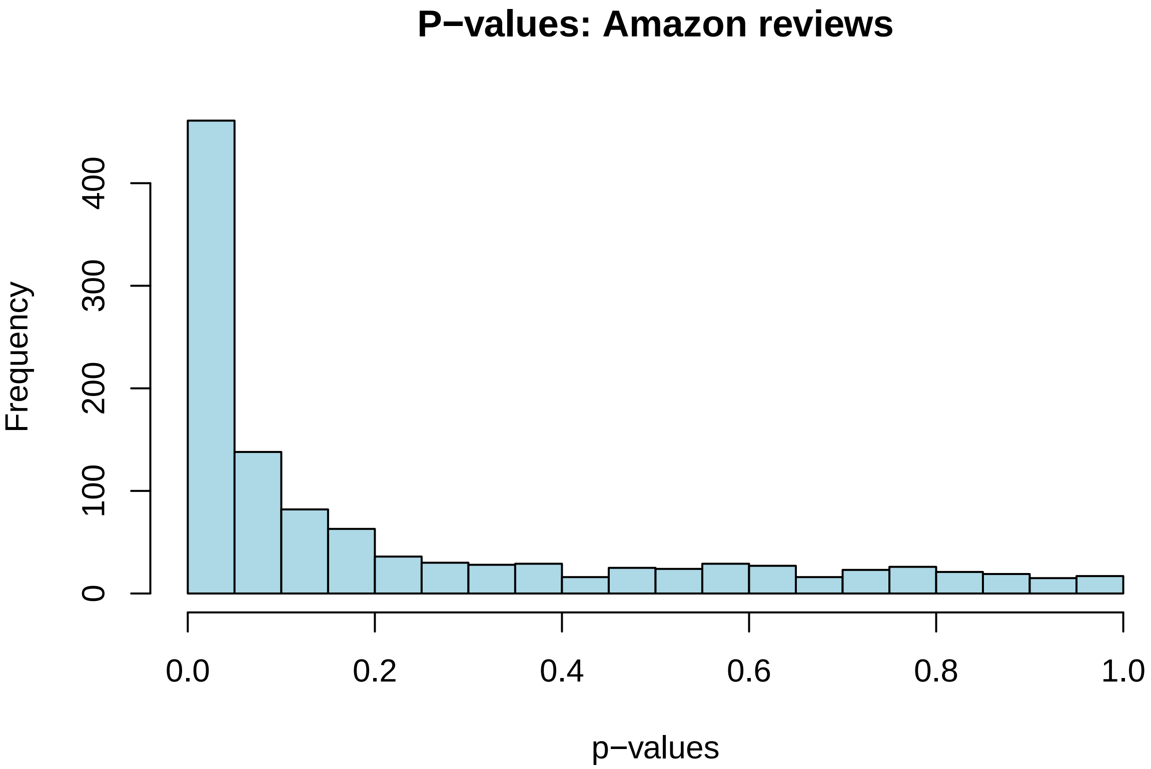
Now we have a vector of pvals, each of which corresponds to a word.

Homework Questions:

Q1

Plot the p-values and comment on their distribution (2 sentence max).

```
hist(pvals, main='', xlab='p-values',breaks=30,col="lightblue")
title("P-values: Amazon reviews")
```



Q2

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

```
significant_5 <- pvals[pvals<=0.05]
length(significant_5)
```

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

```
#461
significant_1 <- pvals[pvals<=0.01]
length(significant_1)
```

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

```
#348
```

Q3

What is the p-value cutoff for 1% FDR? Plot the rejection region.

```
fdr_cut = function(pvals, q){
  pvals = pvals[!is.na(pvals)]
  K = length(pvals)
  k = rank(pvals, ties.method="min")
  alpha = max(pvals[ pvals<= (q*k/K) ])
  alpha
}
wordscut <- fdr_cut(pvals, 0.01)
wordscut #this should be the p-value cutoff
```

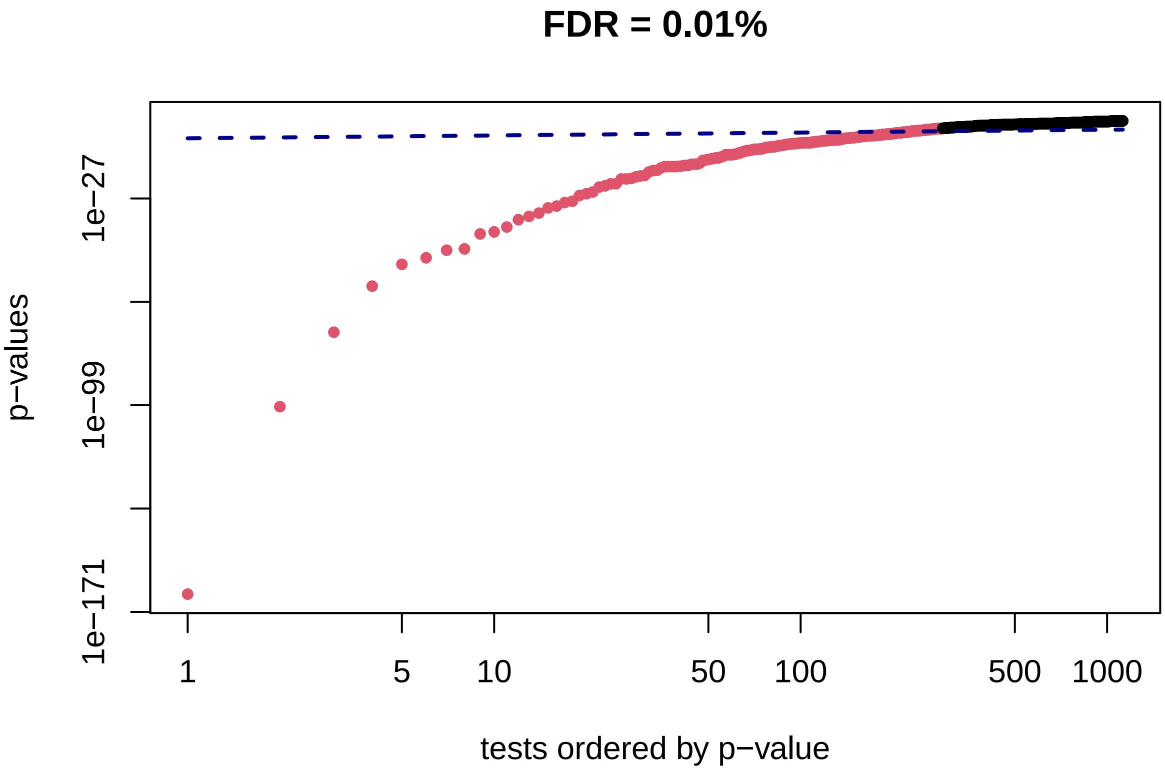
```
## [1] 0.002413249
```

Plotting this:

```
sig = factor(pvals<=wordscut)
levels(sig) = c("lightgrey","maroon")

o = order(pvals)
K = length(pvals)

plot(pvals[o], col=sig[o], pch=20,
     ylab="p-values", xlab="tests ordered by p-value",
     main = 'FDR = 0.01%',log="xy") + lines(1:K,  0.001*(1:K)/K,col="darkblue",lty=2,lwd=2)
```



```
## integer(0)
```

Note: This is what I did before noticing you already wrote out the code on the homework examples. This didn't work; not sure why honestly.

```r
pvals_corrected <- pvals[order(pvals)]
library(FSA)
```

```
## ## FSA v0.8.32. See citation('FSA') if used in publication.
## ## Run fishR() for related website and fishR('IFAR') for related book.
```

```r
data <- as.data.frame(pvals_corrected)
data$BH = p.adjust(pvals_corrected,
                method = "BH")
data_cutoff <- filter(data, data$BH < 0.01)
max(data_cutoff)
```

```
## [1] NA
```

Q4

How many discoveries do you find at q=0.01 and how many do you expect to be false?

```r
passed <- pvals[pvals<wordscut]
```

There are 289 values that passed this cutoff (interestingly, I got 290 on my `FSA` method above, so something worked I guess?), which means that I expect 289*.01 = 2.89 = 3 to be false.

Q5

What are the 10 most significant words? Was 'addictive' significant'? Do these results make sense to you? (2 sentence max)

```r
names(pvals) = words
names(pvals)[order(pvals)[1:10]]
```

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

While a few of these certainly make sense (horrible, great, bad, etc.) make sense, but several of the words don't, notably "not", "but", and "same"; I'm assuming that this has something to do with the underlying data collection process and how we structure our assumptions about the reviews and the ratings. What I mean is that because not all people review a product they buy, and might only review a product if they have particularly positive or negative views. Also, `addictive` was significant.

Q6

What are the advantages and disadvantages of our FDR analysis? (4 sentence max)

Advantages: FDR increases with number of hypothesis tested, which is good for large amounts of data. Disadvantages: have to rely on assumption that all tests are independent, and selection of `q` is arbitrary. This assumption isn't necessarily great here; very positive or negative words might be more likely to appear alongside each other.