Exercise_2

Exercise 2

Problem 1

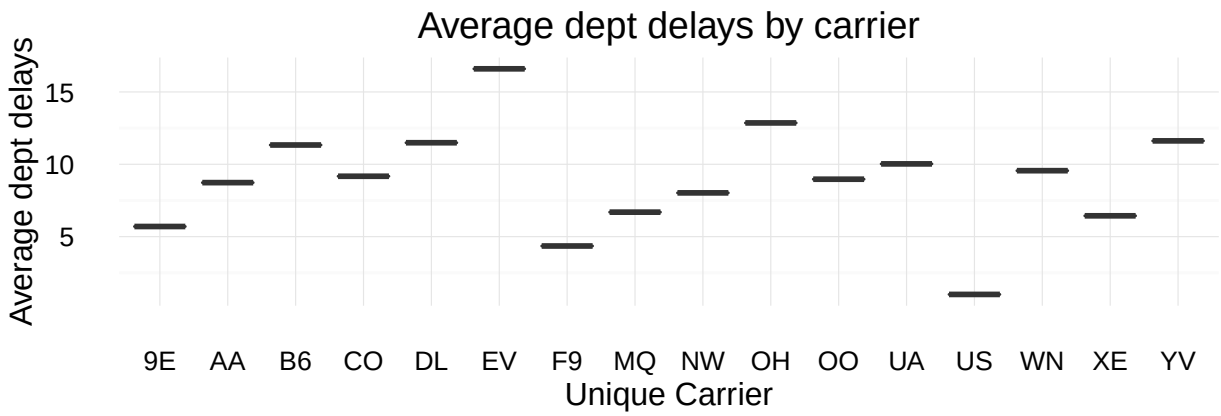
```
suppressMessages(library(ggplot2))
suppressMessages(library(dplyr))
suppressMessages(library(gridExtra))
suppressMessages(library(maps))
suppressMessages(library(mapdata))
airlines <- read.csv("ABIA.csv")
airlines$Year = NULL
airlines$Month= as.factor(airlines$Month)
airlines$Cancelled = as.factor(airlines$Cancelled)
```

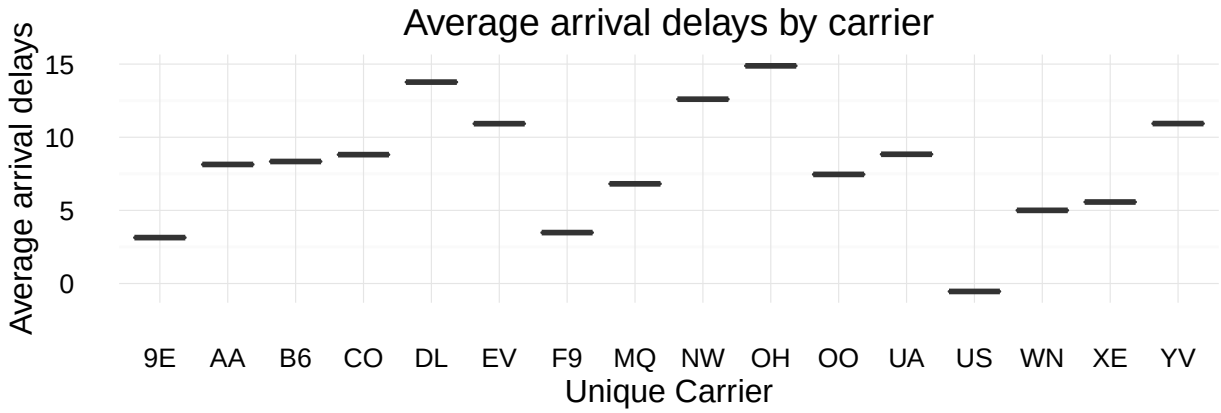
How are arrival and departure delays distributed across carriers?

```
deptdelays <- airlines %>%
  group_by(UniqueCarrier) %>%
  summarise(
    avgdelay = mean(DepDelay, na.rm = TRUE),
    mediandelay = median(DepDelay, na.rm = TRUE),
    maxdelay = max(DepDelay, na.rm = TRUE)
    )
arrdelays <- airlines %>%
  group_by(UniqueCarrier) %>%
  summarise(
    avgdelay = mean(ArrDelay,na.rm = TRUE),
    mediandelay = median(ArrDelay, na.rm = TRUE),
    maxdelay = max(ArrDelay, na.rm = TRUE)
    )


dept_delay <- ggplot(deptdelays, aes(x=UniqueCarrier, y = avgdelay)) + geom_boxplot() + theme_minimal()
arr_delay <- ggplot(arrdelays, aes(x=UniqueCarrier, y = avgdelay)) + geom_boxplot() + theme_minimal() +

grid.arrange(dept_delay,arr_delay,ncol = 1)
```



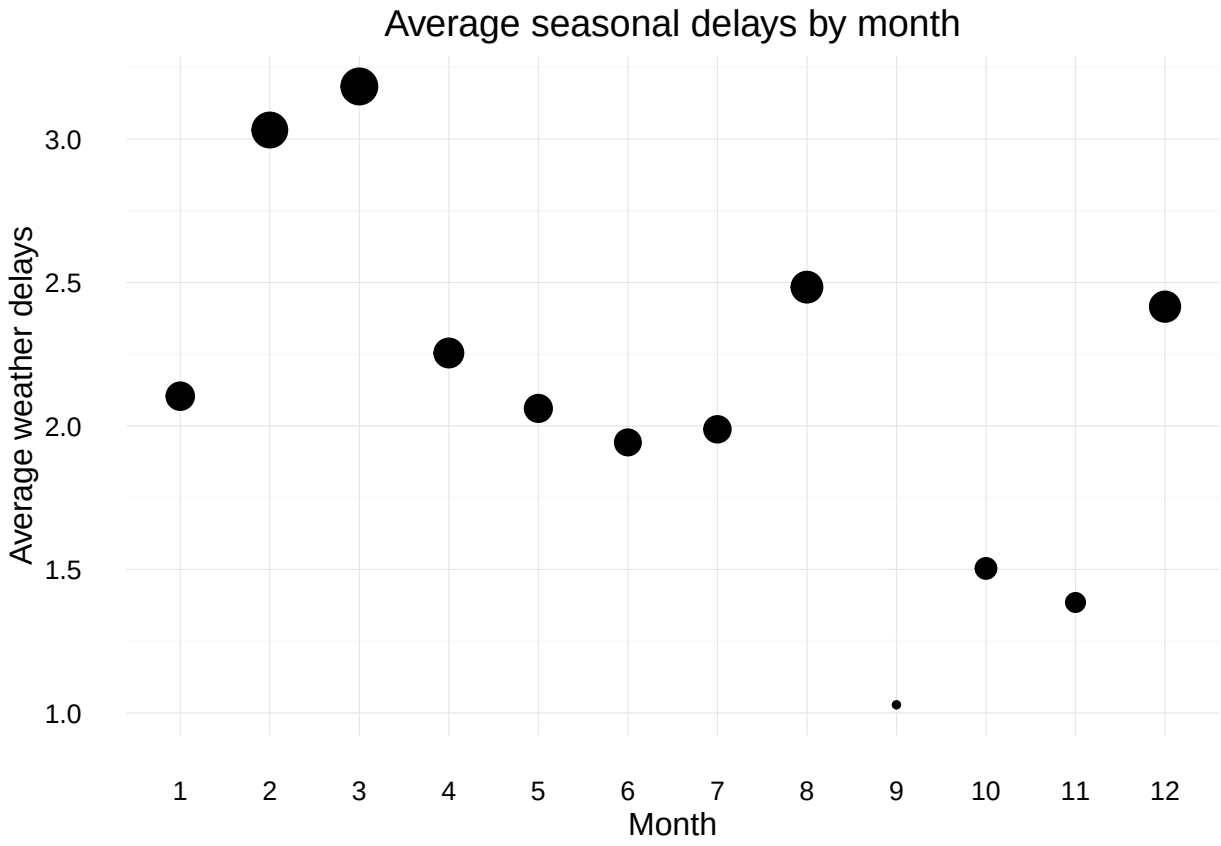


From the above graphs, we can see that B6,EV,F9,UA,WN are flights that have, on average, greater departure delays than arrival delays. Overall, EV,OH and YV are flights that have greatest average delays. The best flights to fly with minimum delays are US, 9E and F9.

Seasonal delays are heavily influenced by the weather conditions. So, it would be good to look at weather delays by month to understand which months to avoid traveling.

```
seasondelays <- airlines %>%
  group_by(Month) %>%
  summarise(
    avgdelay = mean(WeatherDelay, na.rm = TRUE),
    mediandelay = median(WeatherDelay, na.rm = TRUE),
    maxdelay = max(WeatherDelay, na.rm = TRUE)
    )

ggplot(seasondelays, aes(x=Month, y = avgdelay)) + geom_point(aes(size = avgdelay),show.legend = FALSE)
```



March seems to be the worst month to make air travel plans. Why? Is it the destinations? Let's explore

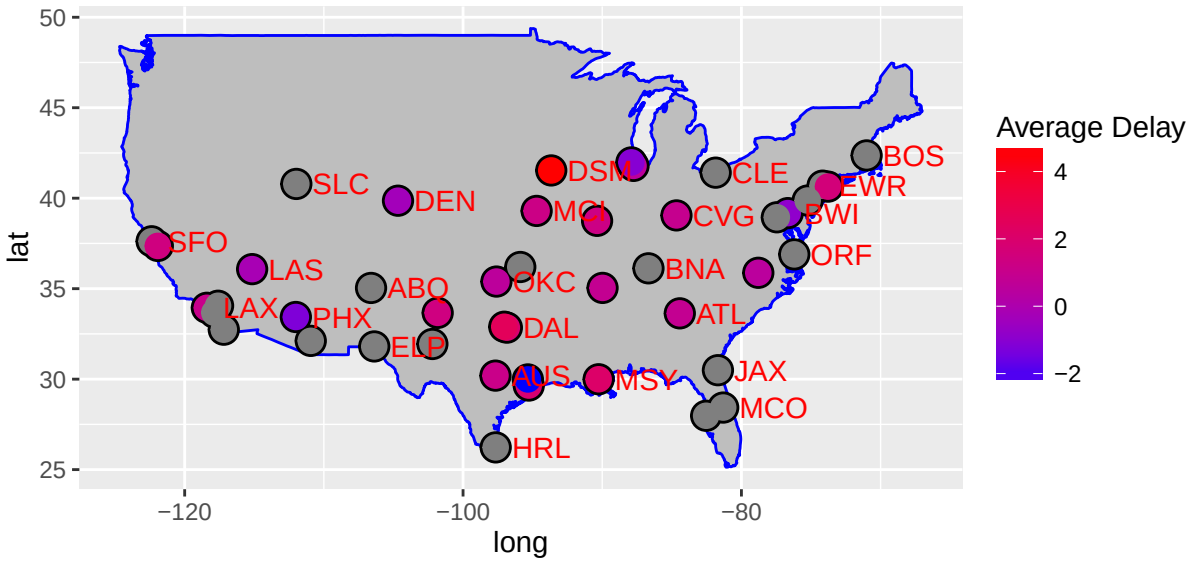
```
airlines_march <- subset(airlines, airlines$Month == 3)

wdelay = airlines_march %>% select(Dest,Lat,Long,WeatherDelay) %>%
    group_by(Dest,Lat,Long) %>%
    summarise_each(funs(mean(., na.rm=TRUE)),avg_delay = WeatherDelay)

usa <- map_data("usa") # we already did this, but we can do it again

usa2 = ggplot() + geom_polygon(data = usa, aes(x=long, y = lat, group = group),fill="grey",color="blue"

usa2 +
    geom_point(data = wdelay, aes(x = wdelay$Long, y = wdelay$Lat ),color="black", size = 5,show.legend
    geom_point(data = wdelay, aes(x = wdelay$Long, y = wdelay$Lat,color= log(wdelay$avg_delay)), size =
    geom_text(data = wdelay, aes(x = wdelay$Long, y = wdelay$Lat , label = paste("  ", as.character(wdel
    scale_colour_gradient(name = "Average Delay",
                          low = "blue", high = "red")
```

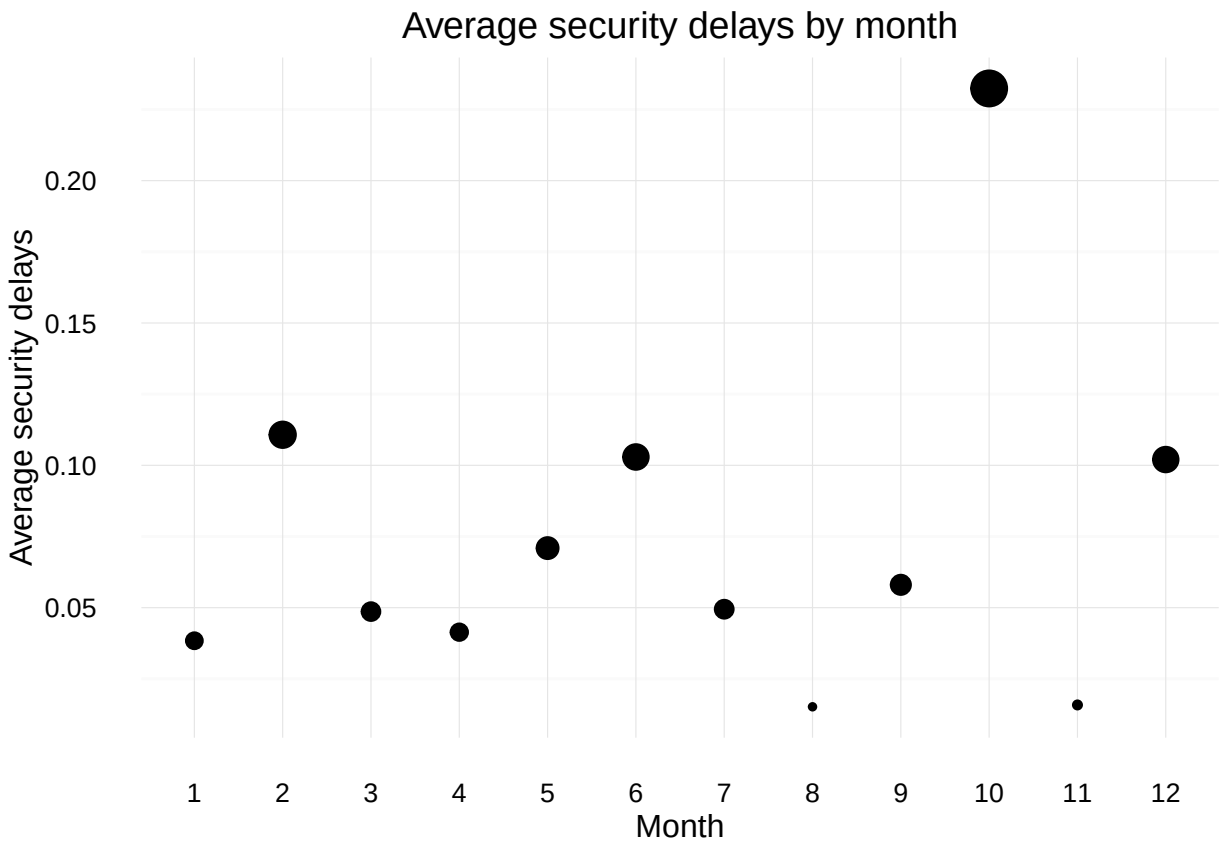


Des Moines Intl. airport has the worst weather related delays for all flights departing from it. Must be something to do with the bad weather in Iowa in March.

Monthly delays are also influenced by security checks - during holiday seasonal months, there would be comparitively heavier checks on security. Let's take a look.

```
securitydelays <- airlines %>%
  group_by(Month) %>%
  summarise(
    avgdelay = mean(SecurityDelay, na.rm = TRUE),
    mediandelay = median(SecurityDelay, na.rm = TRUE),
    maxdelay = max(SecurityDelay, na.rm = TRUE)
    )

ggplot(securitydelays, aes(x=Month, y = avgdelay)) + geom_point(aes(size = avgdelay),show.legend = FALSE
```

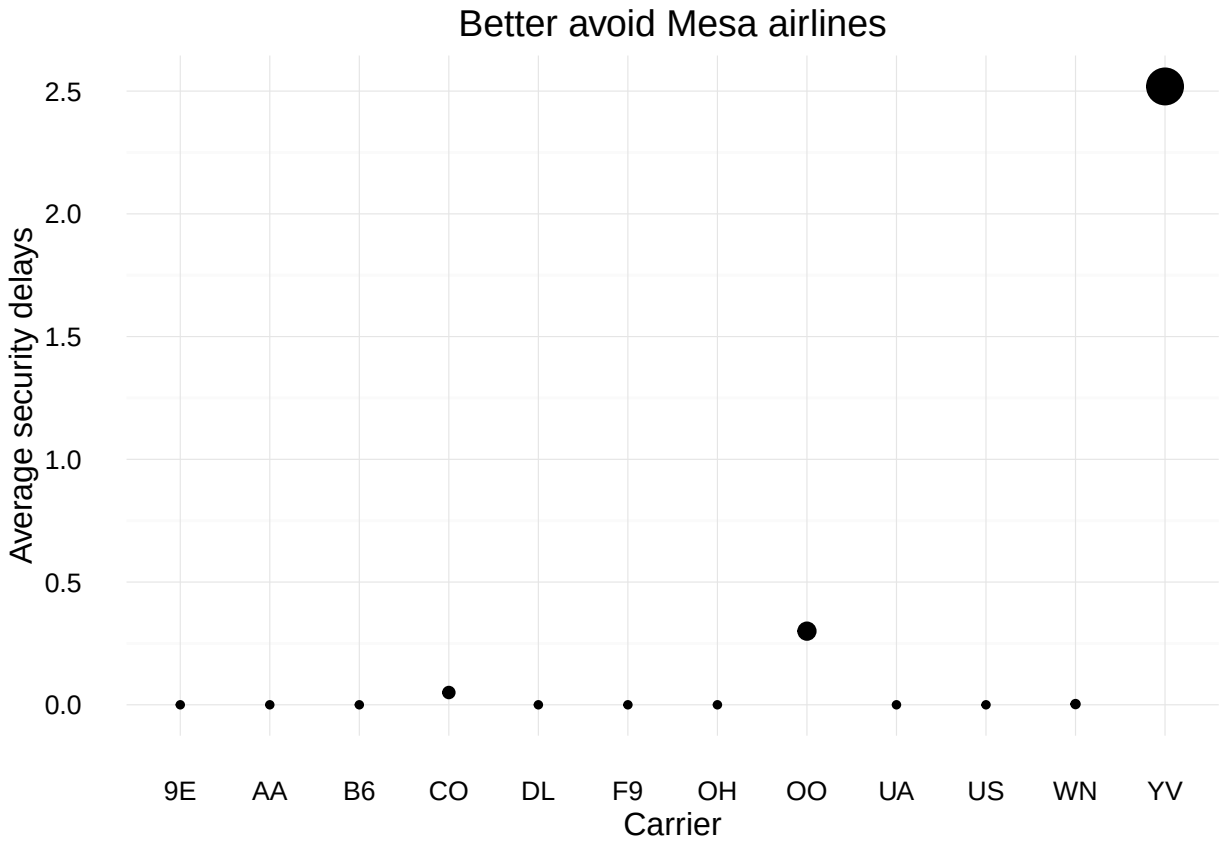


October seems to have the highest number of security delays by month.

Let's explore which flights have the most security delays in October

```
security_delays = subset(airlines, airlines$Month == 10)
carrier_security_delays <- security_delays %>%
  group_by(UniqueCarrier) %>%
  summarise(
    avgdelay = mean(SecurityDelay, na.rm = TRUE),
    mediandelay = median(SecurityDelay, na.rm = TRUE),
    maxdelay = max(SecurityDelay, na.rm = TRUE)
    )

ggplot(carrier_security_delays, aes(x=UniqueCarrier, y = avgdelay)) + geom_point(aes(size = avgdelay),sh
```



Interesting results - MESA airlines seems to have the highest security related delays in Oct. Why so? Let's explore where MESA airlines usually flies to in this month. We will also look at where SKYWEST (the second highest ranked airline that has delays) flies to

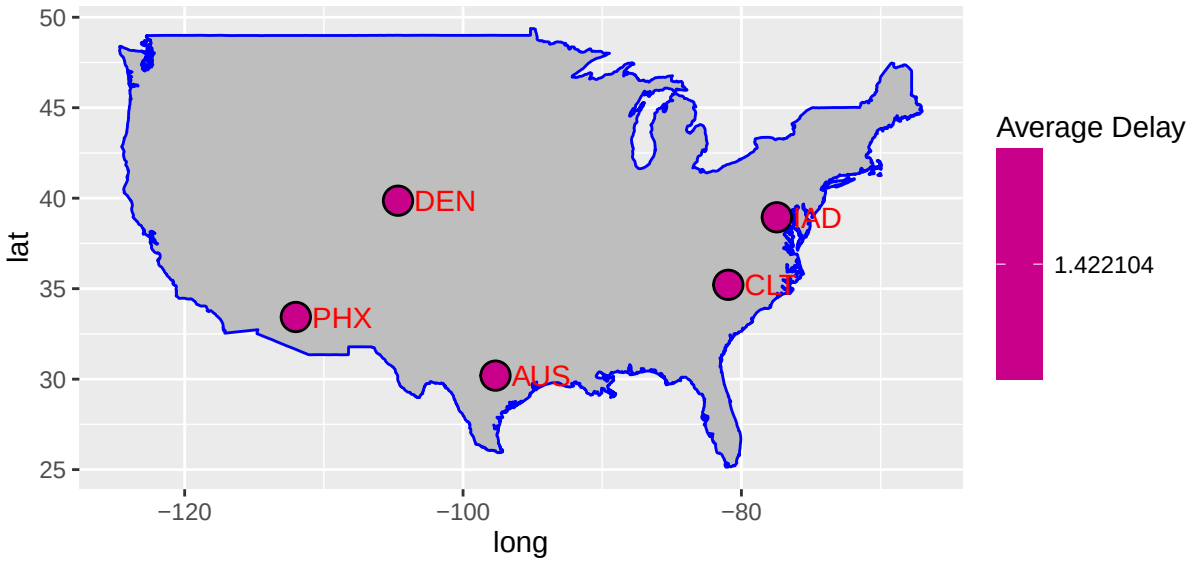
```
mesa_airlines = subset(airlines, airlines$UniqueCarrier == "YV" & airlines$Month == 10)

mesadelay = mesa_airlines %>% select(Dest,Lat,Long,SecurityDelay) %>%
    group_by(Dest,Lat,Long) %>%
    summarise_each(funs(mean(., na.rm=TRUE)),avg_delay = SecurityDelay)

usa <- map_data("usa") # we already did this, but we can do it again

usa2 = ggplot() + geom_polygon(data = usa, aes(x=long, y = lat, group = group),fill="grey",color="blue"

usa2 +
    geom_point(data = mesadelay, aes(x = mesadelay$Long, y = mesadelay$Lat ),color="black", size = 5,sh
    geom_point(data = mesadelay, aes(x = mesadelay$Long, y = mesadelay$Lat,color= log(mesadelay$avg_del
    geom_text(data = mesadelay, aes(x = mesadelay$Long, y = mesadelay$Lat , label = paste("  ", as.chara
    scale_colour_gradient(name = "Average Delay",
                          low = "blue", high = "red")
```



All these destinations represented on the map have the most delays with Mesa airlines. Phoenix could be explained because MESA is a carrier based out of Arizona so a large number of delays are expected due to high volume of MESA airlines in Phoenix. Likewise, since we are looking at data from Austin primarily, there are more Austin related records which explains why we see Austin on the map.

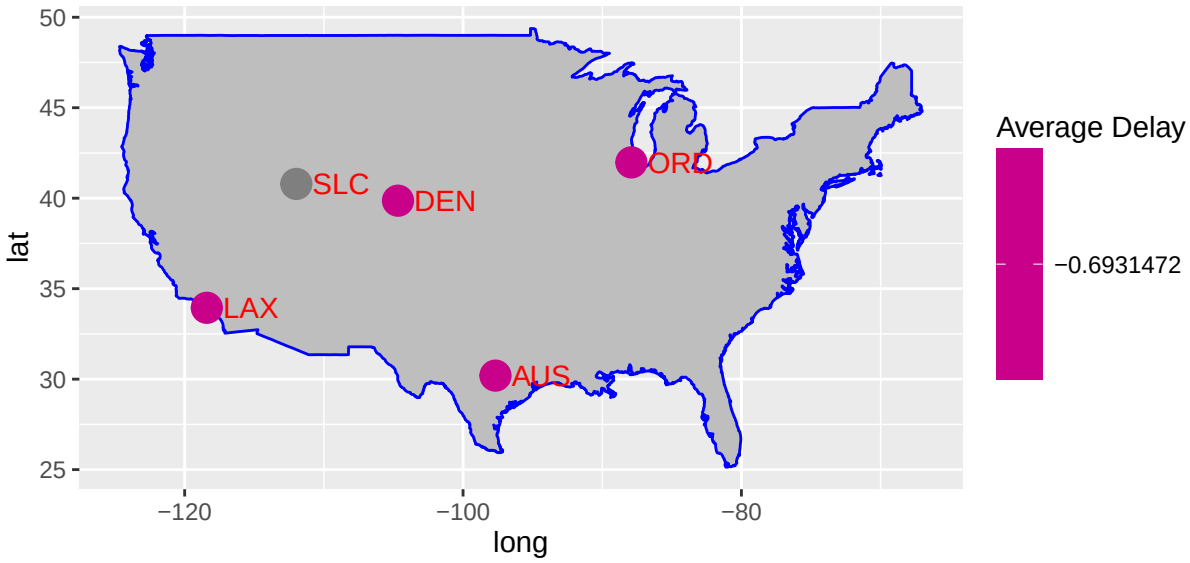
Let's now look at SKYWEST

```
skywest_airlines = subset(airlines, airlines$UniqueCarrier == "OO" & airlines$Month == 10)

skywestdelay = skywest_airlines %>% select(Dest,Lat,Long,SecurityDelay) %>%
    group_by(Dest,Lat,Long) %>%
    summarise_each(funs(mean(., na.rm=TRUE)),avg_delay = SecurityDelay)

usa <- map_data("usa") # we already did this, but we can do it again

usa2 = ggplot() + geom_polygon(data = usa, aes(x=long, y = lat, group = group),fill="grey",color="blue"
usa2 + geom_point(data = skywestdelay, aes(x = skywestdelay$Long, y = skywestdelay$Lat,color = log(skyw
    geom_text(data = skywestdelay, aes(x = skywestdelay$Long, y = skywestdelay$Lat , label = paste("  "
    scale_colour_gradient(name = "Average Delay",
                          low = "blue", high = "red")
```



These are interesting results. SKYWEST experiences the most destination related delays in the points plotted on the map. SLC is easily explained because since the carrier is based out of Utah, there is a high volume of these aircrafts in Salt Lake City. Austin is on the map because like explained previously, this data set has a lot of Austin related records So why does it experience more delays in Denver, Chicago and Los Angeles? It would be meaningful to look at the history of SKYWEST in the other airports from previous years to understand this trend.

Problem 2 :

```
suppressMessages(library(tm))
suppressMessages(library(caret))
suppressMessages(library(glmnet))
suppressMessages(library(SnowballC))
```

Creating the corpus - We preprocess the data to remove numbers, remove stopwords, make everything lowercase, remove punctuation and remove extra white spaces. We are going to create the document term matrix for both the train and test together to avoid the possibility of seeing some terms that are there in the test but not in training and vice versa.

```
readerPlain = function(fname){
  readPlain(elem=list(content=readLines(fname)),
            id=fname, language='en') }

authors_train <- Sys.glob('ReutersC50/C50train/*')
file_list_train = NULL
labels_train = NULL
authors_test = Sys.glob('ReutersC50/C50test/*')
file_list_test = NULL
labels_test = NULL

for(i in authors_train) {
  author_name_train = substring(i, first = 21)
  files_to_add_train = Sys.glob(paste0(i, '/*.txt'))
  file_list_train = append(file_list_train, files_to_add_train)
```

```
  labels_train = append(labels_train, rep(author_name_train, length(files_to_add_train)))
}

for(i in authors_test) {
  author_name_test = substring(i, first = 20)
  files_to_add_test = Sys.glob(paste0(i, '/*.txt'))
  file_list_test = append(file_list_test, files_to_add_test)
  labels_test = append(labels_test, rep(author_name_test, length(files_to_add_test)))
}

file_list <- append(file_list_train,file_list_test)
labels <- unique(append(labels_train,labels_test))

all_docs = lapply(file_list, readerPlain)
names(all_docs) = file_list
names(all_docs) = sub('.txt', '', names(all_docs))
all_corpus = Corpus(VectorSource(all_docs))
names(all_corpus) = labels

all_corpus = tm_map(all_corpus, content_transformer(tolower)) # make everything lowercase
all_corpus = tm_map(all_corpus, content_transformer(removeNumbers)) # remove numbers
all_corpus = tm_map(all_corpus, content_transformer(removePunctuation)) # remove punctuation
all_corpus = tm_map(all_corpus, content_transformer(stripWhitespace)) ## remove excess white-space
all_corpus = tm_map(all_corpus, content_transformer(removeWords), stopwords("en"))  #remove stop words
all_corpus <- tm_map(all_corpus, stemDocument)    #stemming the document to reduce the words to their w

#creating a document term matrix with tf idf scores
dtm <- DocumentTermMatrix(all_corpus,control = list(weighting = function(x) weightTfIdf(x, normalize =
dtm <- removeSparseTerms(dtm, 0.975)
```

Let now apply multinomial bayes model to the training data set. We calculate the weight vector for each author - the weight vector is a weight of each token or word for that author.In order to avoid zero probabilities for those words that could be in the test set but not in training set, we use Lidstone smoothing -

```
x <- as.matrix(dtm)
x_train <- x[1:2500,]
x_test <- x[2501:5000,]
smooth_count = 1/nrow(x_train)

author_sums <- rowsum(x_train +smooth_count, labels_train)
wt <- rowSums(author_sums)
author_wt <- log(author_sums/wt)
```

Now, we use the x_test to multiply the log probabilities calculated from the weights of the training set authors

```
predicted_proba <- x_test%*%t(author_wt)
```

Now, into a new list predicted_authors we assign for each of the 2500 test points, the author that has the max sum of probabilities.

```
predicted_authors = NULL
for( i in 1:2500) {
  predicted_authors = c(predicted_authors,which.max(predicted_proba[i,]))
  }

predicted_authors <- as.data.frame(predicted_authors)
predicted_authors$actual <- rep(1:50,each = 50)
```

Now that we have predicted the authors, let's see the confusion matrix to see how we have performed.

```
confusionMatrix(predicted_authors$predicted_authors, predicted_authors$actual)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21
##         1  45  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##         2   0 27  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  1  0  0  0
##         3   0  0 25  0  2  0  0  0  2  0  0  0  0  0  0  0  5  1  1  9  0
##         4   0  0  0 11  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0
##         5   0  0  0  0 27  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##         6   2  0  0  0  2 38  0  3  0  0  0  0  0  0  0  0  0  0  0  0  0
##         7   0  0  0  0  0  0 12  0  0  0  0  0  6  0  0  0  0  0  0  0  0
##         8   0  0  0  0  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0  0  0
##         9   0  0  0  0  0  0  0  0 14  0  0  0  0  0  0  0  0  0  2  0  0
##         10  0  0  0  0  0  1  0  0  0 33  0  0  0  0  0  0  0  0  0  0  0
##         11  0  0  0  0  0  0  0  0  0  0 35  0  0  0  0  0  0  0  0  0  0
##         12  0  0  0  2  0  0  0  0  0  0  0 33  0  2  0  0  0  0  0  0  0
##         13  0  0  0  0  4  0 37  0  0  0  0  0 17  0  0  0  0  0  0  0  0
##         14  0  0  0 15  0  0  0  1  0  0  1  4  8 21  0  0  0  1  0  0  0
##         15  0  8  0  0  0  0  0  0  1  0  0  0  0  0 30  0  0 16  0  0  0
##         16  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0 48  0  0  0  0  0
##         17  0  0 19  0  1  1  0  1  0  0  4  0  0  0  0  0 33  0  0  1  0
##         18  0 14  0  0  1  0  0  0  0  0  0  0  1  0 17  0  0 27  0  0  0
##         19  0  0  1  0  0  0  0  0  5  0  0  0  0  0  0  0  0  0 32  0  0
##         20  0  0  0  0  0  0  0  0  3  0  0  0  0  0  0  0  1  0  1 28  0
##         21  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0 47
##         22  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##         23  0  0  0  0  0  3  0  9  0  0  0  0  0  0  0  0  0  0  2  0  0
##         24  0  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##         25  0  0  0  0  0  0  0  0  6  2  0  0  0  0  0  0  2  0  3  6  0
##         26  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##         27  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##         28  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##         29  0  0  0  0  1  0  0  0  0  0  0  0  1  0  1  0  0  0  1  0  1
##         30  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0
##         31  0  0  0  0  2  0  0  0  0  0  0  0  4  0  0  0  0  0  0  0  0
##         32  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##         33  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##         34  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##         35  0  0  0  1  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0
##         36  0  0  0  0  0  0  0  0  0  6  0  0  0  0  0  0  0  0  0  0  0
##         37  0  0  0  0  0  0  0  0  1  1  0  0  0  0  0  0  0  0  0  0  0
```

```
##         38  0  0  0  3  0  0  0  0  0  0  0  0  4  0  0  0  0  0  0  0  1
##         39  0  0  0  0  0  0  0  0  7  0  0  0  0  0  0  2  0  0  0  0  0
##         40  0  0  0  0  0  1  0  1  0  1  0  0  0  0  0  0  0  0  0  0  0
##         41  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##         42  2  0  0  0  0  1  0  1  0  2  0  0  0  0  0  0  0  0  0  0  0
##         43  0  0  1  0  2  0  0  1  0  0 10 10  2  0  0  0  3  0  1  0  0
##         44  0  0  0 11  0  0  0  0  0  0  0  1  6 23  0  0  0  0  0  0  1
##         45  0  0  3  0  1  0  0  0  0  0  0  0  0  0  0  0  5  0  2  0  0
##         46  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  1  0  0  0
##         47  1  0  1  0  0  2  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0
##         48  0  0  0  0  2  0  0  0  4  0  0  0  0  0  0  0  1  0  5  6  0
##         49  0  0  0  0  0  3  0 28  6  0  0  0  0  0  1  0  0  1  0  0  0
##         50  0  0  0  7  0  0  0  0  0  0  0  1  0  2  0  0  0  0  0  0  0
##           Reference
## Prediction 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42
##         1   0  0  0  2  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  2  1
##         2   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##         3   2  0  0  1  0  0  0  0  0  2  0  0  4  0  0  0  0  3  1  1  0
##         4   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0
##         5   0  0  8  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0
##         6   0  3  0  0  2  3  0  0  0  1  4  0  0  0  0  5  0  0  2  0  1
##         7   2  1  0  0  0  6  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2
##         8   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##         9   2  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  1  0  0  0
##         10  0  1  0  1  7  2  0  0  2  0  4  0  1  0  6  3  0  0  1  0 11
##         11  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##         12  0  0  0  0  0  0  1  0  0  0  0  0  0  2  0  0  0  0  0  0  0
##         13  0  0  0  0  0  0  0  0  0  2  0  3  0  0  0  0  0  0  0  0  0
##         14  0  0  1  0  0  0  5  1  0  1  0  0  0  8  0  0  2  0  0  0  0
##         15  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##         16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0
##         17  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  2  0  0  0
##         18  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##         19  0  1  0  1  0  0  0  0  5  0  1  0  0  0  0  1  0  1  0  0  0
##         20  1  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##         21  0  0  0  0  0  0  0  0  1  0  0  1  0  0  0  0  0  0  0  1  0
##         22 36  0  2  9  0  0  0  0  4  0  0  0  2  0  1  0  0  0  0  1  0
##         23  0 25  0  0  0  2  0  0  1  1  0  0  1  0  1 12  0  0  3  0  0
##         24  0  0 23  0  0  0  0  0  0  8  0  0  1  0  0  0  0  0  1  0  0
##         25  5  0  1 33  0  0  0  0  2  0  0  0  0  0  2  0  0  1  0  0  0
##         26  0  0  0  0 35  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
##         27  0  0  0  0  0 31  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##         28  0  0  0  0  0  0 38  0  0  0  0  0  0  1  0  0  0  0  0  0  0
##         29  0  0  0  0  0  0  0 49  3  3  0  0  0  0  0  0  0  0  0  0  0
##         30  0  0  0  0  0  0  0  0 29  0  0  0  0  0  0  0  0  7  0  0  0
##         31  0  0  4  0  0  0  1  0  0 18  0  0  0  0  0  1  0  0  0  0  0
##         32  0  2  0  0  1  1  0  0  0  0 28  0  0  0  1  0  0  0  0  0  1
##         33  0  0  0  0  0  0  0  0  0  0  0 43  0  0  0  0  0  0  0  0  0
##         34  0  1  0  0  0  0  0  0  0  1  1  0 36  0  0  0  0  0  1  0  0
##         35  0  0  0  0  0  0  1  0  0  2  0  0  0 18  0  0  0  0  0  0  0
##         36  0  1  0  0  1  0  0  0  0  0  0  0  0  0 38  0  0  0  0  2  0
##         37  0 12  0  0  0  1  0  0  0  0  1  0  0  0  1 23  0  0  0  0  0
##         38  1  0  0  0  0  0  0  0  0  0  0  0  0  4  0  0 24  0  0  0  0
##         39  0  0  0  0  0  0  0  0  1  0  2  0  0  1  0  0  0 34  0  0  0
```

```
##         40  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 40  0  0
##         41  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 39  0
##         42  0  1  0  1  3  3  0  0  1  0  6  0  3  1  0  1  0  0  0  1 27
##         43  0  1 10  1  0  1  1  0  0  6  0  0  0  2  0  0  4  0  1  0  0
##         44  0  0  0  0  0  0  0  0  0  1  0  1  0  6  0  0  0  0  0  3  0
##         45  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0
##         46  0  0  0  0  0  0  1  0  0  0  0  2  0  0  0  0 18  0  0  0  0
##         47  0  0  0  0  1  0  0  0  1  0  1  0  0  0  0  0  0  0  0  0  6
##         48  1  0  1  0  0  0  0  0  0  0  1  0  0  0  0  1  0  1  0  0  0
##         49  0  0  0  0  0  0  1  0  0  2  0  0  0  0  0  1  0  0  0  0  0
##         50  0  0  0  0  0  0  1  0  0  0  0  0  0  7  0  0  0  0  0  0  0
##           Reference
## Prediction 43 44 45 46 47 48 49 50
##         1   0  0  0  0  2  0  0  0
##         2   0  0  0  0  0  0  0  0
##         3   2  0 10  0  0  1  0  0
##         4   0  0  0  0  0  0  0  5
##         5   0  0  0  1  0  0  0  0
##         6   0  0  0  0  0  0  0  0
##         7   0  0  0  0  0  0  0  0
##         8   0  0  0  0  0  0  9  0
##         9   1  0  0  0  0  0  0  0
##         10  0  0  0  0 13  0  0  0
##         11  0  0  0  0  0  0  0  1
##         12  0  1  0  0  0  0  0  2
##         13  0  0  0  0  0  0  0  0
##         14  0 21  0  0  0  0  1  7
##         15  0  0  0  0  0  0  0  0
##         16  0  0  0  0  0  0  3  0
##         17  0  0  9  0  0  3  0  0
##         18  0  0  0  0  0  0  0  0
##         19  0  0  0  0  0  2  0  0
##         20  0  0  0  0  0  0  0  0
##         21  0  1  0  0  0  0  2  0
##         22  0  0  0  0  0  3  0  0
##         23  0  0  0  0  1  0  5  0
##         24  0  0  0  1  0  0  0  0
##         25  0  0  0  0  0  0  0  0
##         26  0  0  0  0  1  0  0  0
##         27  0  0  0  0  0  0  0  0
##         28  0  0  0  0  0  0  0  1
##         29  0  0  0  0  0  1  0  0
##         30  0  0  0  0  0  0  0  0
##         31  0  0  0  0  0  0  0  0
##         32  0  0  0  0  1  1  0  1
##         33  0  0  0  0  0  0  0  0
##         34  0  0  1  0  0  1  4  0
##         35  1  1  0  1  0  0  0  4
##         36  0  0  0  0  1  0  0  0
##         37  0  0  0  0  0  0  0  0
##         38 10  4  0 11  0  0  0  2
##         39  0  0  0  0  0  0  0  0
##         40  0  0  0  0  0  0  0  0
##         41  0  0  0  0  0  0  0  0
```

```
##         42  0  0  0  0  8  0  0  0
##         43 29  0  1 11  0  0  0  5
##         44  0 21  0  4  0  0  1  9
##         45  0  0 28  0  0  3  0  0
##         46  7  1  0 20  0  0  0  1
##         47  0  0  0  0 23  1  0  0
##         48  0  0  1  0  0 34  0  0
##         49  0  0  0  0  0  0 25  0
##         50  0  0  0  1  0  0  0 12
##
## Overall Statistics
##
##                Accuracy : 0.5788
##                  95% CI : (0.5592, 0.5983)
##     No Information Rate : 0.02
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5702
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5 Class: 6
## Sensitivity            0.9000   0.5400   0.5000   0.2200   0.5400   0.7600
## Specificity            0.9967   0.9992   0.9808   0.9971   0.9955   0.9886
## Pos Pred Value         0.8491   0.9310   0.3472   0.6111   0.7105   0.5758
## Neg Pred Value         0.9980   0.9907   0.9897   0.9843   0.9907   0.9951
## Prevalence             0.0200   0.0200   0.0200   0.0200   0.0200   0.0200
## Detection Rate         0.0180   0.0108   0.0100   0.0044   0.0108   0.0152
## Detection Prevalence   0.0212   0.0116   0.0288   0.0072   0.0152   0.0264
## Balanced Accuracy      0.9484   0.7696   0.7404   0.6086   0.7678   0.8743
##                      Class: 7 Class: 8 Class: 9 Class: 10 Class: 11
## Sensitivity            0.2400   0.1000   0.2800    0.6600    0.7000
## Specificity            0.9931   0.9963   0.9971    0.9784    0.9996
## Pos Pred Value         0.4138   0.3571   0.6667    0.3837    0.9722
## Neg Pred Value         0.9846   0.9819   0.9855    0.9930    0.9939
## Prevalence             0.0200   0.0200   0.0200    0.0200    0.0200
## Detection Rate         0.0048   0.0020   0.0056    0.0132    0.0140
## Detection Prevalence   0.0116   0.0056   0.0084    0.0344    0.0144
## Balanced Accuracy      0.6165   0.5482   0.6386    0.8192    0.8498
##                      Class: 12 Class: 13 Class: 14 Class: 15 Class: 16
## Sensitivity             0.6600    0.3400    0.4200    0.6000    0.9600
## Specificity             0.9959    0.9812    0.9686    0.9898    0.9980
## Pos Pred Value          0.7674    0.2698    0.2143    0.5455    0.9057
## Neg Pred Value          0.9931    0.9865    0.9879    0.9918    0.9992
## Prevalence              0.0200    0.0200    0.0200    0.0200    0.0200
## Detection Rate          0.0132    0.0068    0.0084    0.0120    0.0192
## Detection Prevalence    0.0172    0.0252    0.0392    0.0220    0.0212
## Balanced Accuracy       0.8280    0.6606    0.6943    0.7949    0.9790
##                      Class: 17 Class: 18 Class: 19 Class: 20 Class: 21
## Sensitivity             0.6600    0.5400    0.6400    0.5600    0.9400
## Specificity             0.9829    0.9865    0.9927    0.9971    0.9967
## Pos Pred Value          0.4400    0.4500    0.6400    0.8000    0.8545
## Neg Pred Value          0.9930    0.9906    0.9927    0.9911    0.9988
```

```
## Prevalence              0.0200    0.0200    0.0200    0.0200    0.0200
## Detection Rate          0.0132    0.0108    0.0128    0.0112    0.0188
## Detection Prevalence    0.0300    0.0240    0.0200    0.0140    0.0220
## Balanced Accuracy       0.8214    0.7633    0.8163    0.7786    0.9684
##                      Class: 22 Class: 23 Class: 24 Class: 25 Class: 26
## Sensitivity             0.7200    0.5000    0.4600    0.6600    0.7000
## Specificity             0.9906    0.9833    0.9935    0.9878    0.9992
## Pos Pred Value          0.6102    0.3788    0.5897    0.5238    0.9459
## Neg Pred Value          0.9943    0.9897    0.9890    0.9930    0.9939
## Prevalence              0.0200    0.0200    0.0200    0.0200    0.0200
## Detection Rate          0.0144    0.0100    0.0092    0.0132    0.0140
## Detection Prevalence    0.0236    0.0264    0.0156    0.0252    0.0148
## Balanced Accuracy       0.8553    0.7416    0.7267    0.8239    0.8496
##                      Class: 27 Class: 28 Class: 29 Class: 30 Class: 31
## Sensitivity             0.6200    0.7600    0.9800    0.5800    0.3600
## Specificity             1.0000    0.9992    0.9951    0.9967    0.9951
## Pos Pred Value          1.0000    0.9500    0.8033    0.7838    0.6000
## Neg Pred Value          0.9923    0.9951    0.9996    0.9915    0.9870
## Prevalence              0.0200    0.0200    0.0200    0.0200    0.0200
## Detection Rate          0.0124    0.0152    0.0196    0.0116    0.0072
## Detection Prevalence    0.0124    0.0160    0.0244    0.0148    0.0120
## Balanced Accuracy       0.8100    0.8796    0.9876    0.7884    0.6776
##                      Class: 32 Class: 33 Class: 34 Class: 35 Class: 36
## Sensitivity             0.5600    0.8600    0.7200    0.3600    0.7600
## Specificity             0.9963    1.0000    0.9955    0.9951    0.9955
## Pos Pred Value          0.7568    1.0000    0.7660    0.6000    0.7755
## Neg Pred Value          0.9911    0.9972    0.9943    0.9870    0.9951
## Prevalence              0.0200    0.0200    0.0200    0.0200    0.0200
## Detection Rate          0.0112    0.0172    0.0144    0.0072    0.0152
## Detection Prevalence    0.0148    0.0172    0.0188    0.0120    0.0196
## Balanced Accuracy       0.7782    0.9300    0.8578    0.6776    0.8778
##                      Class: 37 Class: 38 Class: 39 Class: 40 Class: 41
## Sensitivity             0.4600    0.4800    0.6800    0.8000    0.7800
## Specificity             0.9931    0.9837    0.9947    0.9984    1.0000
## Pos Pred Value          0.5750    0.3750    0.7234    0.9091    1.0000
## Neg Pred Value          0.9890    0.9893    0.9935    0.9959    0.9955
## Prevalence              0.0200    0.0200    0.0200    0.0200    0.0200
## Detection Rate          0.0092    0.0096    0.0136    0.0160    0.0156
## Detection Prevalence    0.0160    0.0256    0.0188    0.0176    0.0156
## Balanced Accuracy       0.7265    0.7318    0.8373    0.8992    0.8900
##                      Class: 42 Class: 43 Class: 44 Class: 45 Class: 46
## Sensitivity             0.5400    0.5800    0.4200    0.5600    0.4000
## Specificity             0.9857    0.9698    0.9727    0.9935    0.9869
## Pos Pred Value          0.4355    0.2816    0.2386    0.6364    0.3846
## Neg Pred Value          0.9906    0.9912    0.9880    0.9910    0.9877
## Prevalence              0.0200    0.0200    0.0200    0.0200    0.0200
## Detection Rate          0.0108    0.0116    0.0084    0.0112    0.0080
## Detection Prevalence    0.0248    0.0412    0.0352    0.0176    0.0208
## Balanced Accuracy       0.7629    0.7749    0.6963    0.7767    0.6935
##                      Class: 47 Class: 48 Class: 49 Class: 50
## Sensitivity             0.4600    0.6800    0.5000    0.2400
## Specificity             0.9922    0.9902    0.9824    0.9922
## Pos Pred Value          0.5476    0.5862    0.3676    0.3871
## Neg Pred Value          0.9890    0.9934    0.9897    0.9846
```

```
## Prevalence              0.0200    0.0200    0.0200    0.0200
## Detection Rate          0.0092    0.0136    0.0100    0.0048
## Detection Prevalence    0.0168    0.0232    0.0272    0.0124
## Balanced Accuracy       0.7261    0.8351    0.7412    0.6161
```

The naive bayes model is giving an accuracy of 57.88% on test set. Let's look at another regression model - perhaps the multinomial logistic regression model?

```
pc_x <- prcomp(x)
#The first 300 components explains 0.06% of variance, so let's use the first 300 components
transformed_x <- pc_x$x[,1:300]
train_transformed <- transformed_x[1:2500,]
test_transformed <- transformed_x[2501:5000,]
logit_model <- glmnet(y = rep(1:50, each = 50),x = train_transformed,  family = "multinomial", alpha = 
predicted_authors_logit <- as.data.frame(predict(logit_model, newx = test_transformed,type = "class", s
predicted_authors_logit$actual_authors <- rep(1:50, each = 50)
confusionMatrix(predicted_authors_logit$`1`,predicted_authors_logit$actual_authors)
```

```
## Warning in confusionMatrix.default(predicted_authors_logit$`1`,
## predicted_authors_logit$actual_authors): Levels are not in the same order
## for reference and data. Refactoring data to match.
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21
##         1  40  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##         2   0 31  0  0  0  0  0  0  0  0  0  0  0  0  3  0  0  0  0  0  0
##         3   0  0 16  0  0  0  0  0  1  0  2  0  1  0  0  0  3  1  1  3  0
##         4   0  0  0 11  0  0  0  0  0  0  0  0  0  9  0  0  0  0  0  0  0
##         5   0  0  0  0 30  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0
##         6   1  0  0  0  1 34  0 13  0  2  0  0  0  0  0  0  0  0  0  0  0
##         7   0  0  0  0  0  0 14  0  0  0  0  0  7  0  0  0  2  0  0  0  0
##         8   0  0  0  0  0  0  0 12  0  0  0  0  0  0  0  0  0  0  0  0  0
##         9   0  0  0  0  0  0  0  0 20  1 14  1  0  0  0  0  3  1  3  1  0
##         10  2  0  0  0  0  0  0  0  0 27  0  0  0  0  0  0  0  0  0  0  0
##         11  0  0  0  0  0  0  0  0  0  1 26  0  0  0  0  0  0  0  0  0  0
##         12  0  0  0  2  0  0  0  0  0  0  0 36  0  1  0  0  0  0  0  0  0
##         13  0  0  0  0  2  0 12  0  0  0  0  0 16  0  0  0  0  0  0  0  0
##         14  0  0  0 13  0  0  2  0  0  1  0  3  4 17  0  0  0  0  0  0  0
##         15  0  3  0  0  0  0  1  0  0  0  0  0  3  0 28  0  0 13  0  0  0
##         16  0  0  0  0  0  0  0  1  0  0  0  1  0  0  0 50  0  0  0  0  0
##         17  0  0 20  0  1  0  1  0  0  0  1  1  0  0  0  0 26  0  0  0  0
##         18  0 16  0  0  0  0  0  0  0  0  0  0  0  0 13  0  0 34  0  0  0
##         19  0  0  1  0  0  0  0  0  8  0  0  0  0  0  0  0  1  0 39  3  0
##         20  0  0  0  0  0  0  0  0  0  2  0  0  0  0  1  0  0  0  0 28  0
##         21  2  0  0  3  0  0  3  0  0  0  0  1  1  1  2  0  0  0  0  0 50
##         22  0  0  2  0  0  0  1  0  2  0  0  0  0  0  0  0  0  0  0  3  0
##         23  0  0  0  0  0  7  2  7  0  3  0  1  2  0  0  0  1  0  1  0  0
##         24  0  0  0  0  7  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##         25  0  0  0  0  1  0  0  0  5  0  0  0  0  0  0  0  0  0  1  6  0
##         26  0  0  1  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0
##         27  0  0  0  0  0  0  1  0  0  0  0  0  3  0  0  0  0  0  0  0  0
```

```
##          28  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          29  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0
##          30  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0
##          31  0  0  0  0  3  0  0  0  0  0  1  0  5  0  0  0  0  0  0  0  0
##          32  0  0  1  0  0  0  2  1  3  1  1  0  0  0  0  0  0  0  0  0  0
##          33  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          34  0  0  0  0  2  2  0  0  1  0  1  0  0  1  0  0  1  0  1  0  0
##          35  0  0  0  3  0  0  1  0  0  0  0  4  0  2  1  0  0  0  0  0  0
##          36  0  0  0  0  0  0  0  0  0  8  0  0  0  0  0  0  0  0  0  0  0
##          37  0  0  1  0  0  3  1  4  0  0  1  0  0  0  0  0  0  0  0  0  0
##          38  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          39  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0
##          40  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0
##          41  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0
##          42  2  0  0  0  0  1  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0
##          43  1  0  1  0  0  0  0  1  0  0  0  0  1  0  0  0  0  0  0  0  0
##          44  1  0  0  6  0  0  6  0  0  0  0  0  4 11  0  0  0  1  0  0  0
##          45  0  0  5  0  0  0  1  0  2  0  0  0  1  0  0  0  6  0  1  3  0
##          46  1  0  0  2  0  0  0  0  0  0  1  0  1  1  0  0  1  0  0  0  0
##          47  0  0  1  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  1  0
##          48  0  0  1  0  0  1  0  0  3  0  0  0  0  0  0  0  5  0  2  1  0
##          49  0  0  0  0  0  1  0  9  3  0  0  0  0  0  0  0  0  0  0  0  0
##          50  0  0  0  8  1  1  0  0  1  0  1  2  1  7  0  0  1  0  0  0  0
##            Reference
## Prediction 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42
##          1  0  0  0  2  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  4  1
##          2  2  0  1  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0
##          3  0  0  0  4  0  0  0  0  0  0  0  0  0  0  1  2  0  1  1  0  0
##          4  0  0  0  0  0  0  0  0  0  0  0  0  0  3  0  0  2  0  0  0  0
##          5  0  0  3  0  0  0  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0
##          6  0  4  0  0  1  1  0  0  0  1  1  0  1  0  0  0  0  0  1  0  0
##          7  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          8  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          9  1  0  2  1  0  0  0  0  0  1  1  0  0  0  0  0  0  0  0  0  0
##          10 0  0  0  0  5  0  0  0  0  0  0  0  0  0  2  0  0  0  0  4  1
##          11 0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          12 0  0  0  0  0  0  5  0  0  0  0  0  0  1  0  0  0  0  0  0  0
##          13 0  0  0  0  0  0  0  2  0  0  0  3  0  0  0  0  0  0  0  0  0
##          14 0  0  0  0  0  0  0  0  0  0  0  0  0  3  0  0  1  0  0  0  0
##          15 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          16 0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  1  0  0  2  0
##          17 0  0  0  0  0  0  0  1  0  2  0  0  0  0  1  0  0  0  0  2  0
##          18 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0
##          19 0  0  0  2  0  0  0  1  2  0  0  0  1  0  1  1  0  0  0  0  0
##          20 0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0
##          21 0  0  0  0  0  0  0  0  2  0  0  2  1  0  0  0  0  0  0  1  0
##          22 37  0  0 10  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  4  1
##          23  0 34  2  0  0  0  0  0  0  0  3  0  0  1  0  6  0  1  1  0  3
##          24  0  0 25  0  0  0  1  0  1  6  0  0  0  0  0  0  0  0  2  0  0
##          25  3  0  1 25  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  1  0
##          26  0  0  0  0 36  2  0  0  1  0  3  0  1  0  1  0  0  0  0  1  5
##          27  0  0  0  0  0 40  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##          28  0  0  0  0  0  0 37  0  0  0  0  0  0  1  0  0  0  0  0  0  0
##          29  0  0  0  0  0  0  0 45  0  0  0  0  0  0  0  0  0  0  0  0  0
```

```
##          30  0  0  0  1  0  0  0  0 34  0  0  0  0  0  0  0  0 14  0  0  0
##          31  1  0  3  0  0  2  1  0  0 28  0  0  4  0  1  0  0  0  0  0  0
##          32  0  0  0  0  2  0  0  0  0  1 32  0  0  1  0  0  0  1  0  0  6
##          33  0  0  0  0  0  0  0  0  0  0  0 40  0  0  0  0  0  0  0  0  0
##          34  2  1  2  1  1  2  0  1  0  0  2  0 41  0  3  2  0  0  1  1  1
##          35  0  0  0  0  0  0  0  0  1  2  0  0  0 16  0  0  1  0  0  0  0
##          36  0  0  4  1  1  0  0  0  0  0  0  0  0  0 34  0  0  0  0  2  1
##          37  0  5  0  0  0  0  0  0  0  0  1  4  0  0  1 36  0  0  3  0  0
##          38  0  0  0  0  0  0  0  0  0  0  0  0  1  1  0  0 11  0  0  0  0
##          39  0  0  0  0  0  0  0  0  8  0  0  0  0  0  0  0  0 33  0  0  0
##          40  1  3  0  0  0  1  0  0  0  0  1  0  0  0  0  2  0  0 41  0  0
##          41  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 23  0
##          42  0  2  0  0  4  0  1  0  1  0  2  0  0  0  1  0  0  0  0  1 23
##          43  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  3  0  0  0  0
##          44  0  0  2  0  0  0  0  0  0  0  0  0  0  8  0  0  0  0  0  1  0
##          45  1  0  0  2  0  0  0  0  0  3  1  0  0  0  0  1  0  0  0  0  0
##          46  0  0  0  0  0  0  3  0  0  0  1  0  0  3  0  0 29  0  0  0  0
##          47  1  0  0  0  0  0  0  0  0  0  0  0  0  1  1  0  0  0  0  0  6
##          48  1  0  3  1  0  0  0  0  0  0  1  1  0  0  0  0  0  0  0  0  0
##          49  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  1  2
##          50  0  1  1  0  0  0  1  0  0  0  0  0  0 11  1  0  0  0  0  0  0
##            Reference
## Prediction 43 44 45 46 47 48 49 50
##          1   0  0  0  0  0  0  1  0
##          2   0  0  0  0  0  0  0  0
##          3   0  0  2  0  0  0  0  0
##          4   0  5  0  0  0  0  0  8
##          5   0  0  0  0  0  0  0  0
##          6   0  0  0  0  2  0  2  0
##          7   0  0  0  0  0  0  0  0
##          8   0  0  0  0  0  0 13  0
##          9   0  0  0  0  0  2  0  0
##          10  0  0  0  0  3  2  0  1
##          11  0  0  0  0  0  0  0  0
##          12  0  0  0  0  0  0  0  2
##          13  0  0  0  0  0  0  1  0
##          14  0 11  0  2  0  0  0  3
##          15  0  0  0  2  0  0  0  0
##          16  0  0  0  0  0  0  5  0
##          17  0  0  6  1  0  0  0  0
##          18  0  0  0  1  0  0  0  0
##          19  0  0  1  0  0  5  0  0
##          20  0  0  0  0  0  0  0  0
##          21  1  2  0  0  0  0  2  1
##          22  0  0  0  0  0  2  0  0
##          23  0  0  0  0  0  0  5  0
##          24  0  0  0  1  0  0  0  0
##          25  0  0  0  0  1  0  0  0
##          26  0  0  0  0  8  0  0  1
##          27  0  0  0  0  0  0  0  0
##          28  0  0  0  0  0  0  0  0
##          29  0  0  0  0  0  0  0  0
##          30  0  0  0  0  0  0  0  0
##          31  0  0  0  0  0  0  0  1
```

```
##         32  0  0  0  0  2  1  0  1
##         33  0  0  0  0  0  0  0  0
##         34  0  0  0  2  0  0  6  0
##         35  1  7  0  1  0  0  0  5
##         36  0  0  0  0  0  0  0  0
##         37  0  0  0  0  1  0  0  0
##         38  6  0  0  6  0  0  0  1
##         39  0  0  0  0  0  0  0  0
##         40  0  0  0  0  0  0  0  0
##         41  0  0  0  0  0  0  0  0
##         42  0  0  0  0  3  0  0  0
##         43 26  0  0  6  0  0  1  0
##         44  1 10  0  0  0  0  1  5
##         45  0  0 40  1  0  5  0  0
##         46 13  6  0 25  0  1  0  4
##         47  0  0  0  0 30  0  0  0
##         48  0  0  1  0  0 32  0  0
##         49  0  0  0  0  0  0 13  0
##         50  2  9  0  2  0  0  0 17
##
## Overall Statistics
##
##                Accuracy : 0.5796
##                  95% CI : (0.56, 0.599)
##     No Information Rate : 0.02
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.571
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5 Class: 6
## Sensitivity            0.8000   0.6200   0.3200   0.2200   0.6000   0.6800
## Specificity            0.9955   0.9971   0.9906   0.9890   0.9963   0.9873
## Pos Pred Value         0.7843   0.8158   0.4103   0.2895   0.7692   0.5231
## Neg Pred Value         0.9959   0.9923   0.9862   0.9842   0.9919   0.9934
## Prevalence             0.0200   0.0200   0.0200   0.0200   0.0200   0.0200
## Detection Rate         0.0160   0.0124   0.0064   0.0044   0.0120   0.0136
## Detection Prevalence   0.0204   0.0152   0.0156   0.0152   0.0156   0.0260
## Balanced Accuracy      0.8978   0.8086   0.6553   0.6045   0.7982   0.8337
##                      Class: 7 Class: 8 Class: 9 Class: 10 Class: 11
## Sensitivity            0.2800   0.2400   0.4000    0.5400    0.5200
## Specificity            0.9959   0.9947   0.9869    0.9918    0.9992
## Pos Pred Value         0.5833   0.4800   0.3846    0.5745    0.9286
## Neg Pred Value         0.9855   0.9846   0.9877    0.9906    0.9903
## Prevalence             0.0200   0.0200   0.0200    0.0200    0.0200
## Detection Rate         0.0056   0.0048   0.0080    0.0108    0.0104
## Detection Prevalence   0.0096   0.0100   0.0208    0.0188    0.0112
## Balanced Accuracy      0.6380   0.6173   0.6935    0.7659    0.7596
##                      Class: 12 Class: 13 Class: 14 Class: 15 Class: 16
## Sensitivity             0.7200    0.3200    0.3400    0.5600    1.0000
## Specificity             0.9955    0.9918    0.9824    0.9910    0.9955
## Pos Pred Value          0.7660    0.4444    0.2833    0.5600    0.8197
```

```
## Neg Pred Value          0.9943    0.9862    0.9865    0.9910    1.0000
## Prevalence              0.0200    0.0200    0.0200    0.0200    0.0200
## Detection Rate          0.0144    0.0064    0.0068    0.0112    0.0200
## Detection Prevalence    0.0188    0.0144    0.0240    0.0200    0.0244
## Balanced Accuracy       0.8578    0.6559    0.6612    0.7755    0.9978
##                      Class: 17 Class: 18 Class: 19 Class: 20 Class: 21
## Sensitivity             0.5200    0.6800    0.7800    0.5600    1.0000
## Specificity             0.9849    0.9869    0.9890    0.9984    0.9898
## Pos Pred Value          0.4127    0.5152    0.5909    0.8750    0.6667
## Neg Pred Value          0.9902    0.9934    0.9955    0.9911    1.0000
## Prevalence              0.0200    0.0200    0.0200    0.0200    0.0200
## Detection Rate          0.0104    0.0136    0.0156    0.0112    0.0200
## Detection Prevalence    0.0252    0.0264    0.0264    0.0128    0.0300
## Balanced Accuracy       0.7524    0.8335    0.8845    0.7792    0.9949
##                      Class: 22 Class: 23 Class: 24 Class: 25 Class: 26
## Sensitivity             0.7400    0.6800    0.5000    0.5000    0.7200
## Specificity             0.9894    0.9812    0.9918    0.9914    0.9898
## Pos Pred Value          0.5873    0.4250    0.5556    0.5435    0.5902
## Neg Pred Value          0.9947    0.9934    0.9898    0.9898    0.9943
## Prevalence              0.0200    0.0200    0.0200    0.0200    0.0200
## Detection Rate          0.0148    0.0136    0.0100    0.0100    0.0144
## Detection Prevalence    0.0252    0.0320    0.0180    0.0184    0.0244
## Balanced Accuracy       0.8647    0.8306    0.7459    0.7457    0.8549
##                      Class: 27 Class: 28 Class: 29 Class: 30 Class: 31
## Sensitivity             0.8000    0.7400    0.9000    0.6800    0.5600
## Specificity             0.9984    0.9992    0.9996    0.9931    0.9910
## Pos Pred Value          0.9091    0.9487    0.9783    0.6667    0.5600
## Neg Pred Value          0.9959    0.9947    0.9980    0.9935    0.9910
## Prevalence              0.0200    0.0200    0.0200    0.0200    0.0200
## Detection Rate          0.0160    0.0148    0.0180    0.0136    0.0112
## Detection Prevalence    0.0176    0.0156    0.0184    0.0204    0.0200
## Balanced Accuracy       0.8992    0.8696    0.9498    0.8365    0.7755
##                      Class: 32 Class: 33 Class: 34 Class: 35 Class: 36
## Sensitivity             0.6400    0.8000    0.8200    0.3200    0.6800
## Specificity             0.9902    1.0000    0.9849    0.9882    0.9931
## Pos Pred Value          0.5714    1.0000    0.5256    0.3556    0.6667
## Neg Pred Value          0.9926    0.9959    0.9963    0.9862    0.9935
## Prevalence              0.0200    0.0200    0.0200    0.0200    0.0200
## Detection Rate          0.0128    0.0160    0.0164    0.0064    0.0136
## Detection Prevalence    0.0224    0.0160    0.0312    0.0180    0.0204
## Balanced Accuracy       0.8151    0.9000    0.9024    0.6541    0.8365
##                      Class: 37 Class: 38 Class: 39 Class: 40 Class: 41
## Sensitivity             0.7200    0.2200    0.6600    0.8200    0.4600
## Specificity             0.9898    0.9935    0.9963    0.9963    0.9996
## Pos Pred Value          0.5902    0.4074    0.7857    0.8200    0.9583
## Neg Pred Value          0.9943    0.9842    0.9931    0.9963    0.9891
## Prevalence              0.0200    0.0200    0.0200    0.0200    0.0200
## Detection Rate          0.0144    0.0044    0.0132    0.0164    0.0092
## Detection Prevalence    0.0244    0.0108    0.0168    0.0200    0.0096
## Balanced Accuracy       0.8549    0.6067    0.8282    0.9082    0.7298
##                      Class: 42 Class: 43 Class: 44 Class: 45 Class: 46
## Sensitivity             0.4600    0.5200    0.2000    0.8000    0.5000
## Specificity             0.9918    0.9943    0.9808    0.9865    0.9727
## Pos Pred Value          0.5349    0.6500    0.1754    0.5479    0.2717
```

```
## Neg Pred Value          0.9890    0.9902    0.9836    0.9959    0.9896
## Prevalence              0.0200    0.0200    0.0200    0.0200    0.0200
## Detection Rate          0.0092    0.0104    0.0040    0.0160    0.0100
## Detection Prevalence    0.0172    0.0160    0.0228    0.0292    0.0368
## Balanced Accuracy       0.7259    0.7571    0.5904    0.8933    0.7363
##                      Class: 47 Class: 48 Class: 49 Class: 50
## Sensitivity             0.6000    0.6400    0.2600    0.3400
## Specificity             0.9947    0.9914    0.9931    0.9792
## Pos Pred Value          0.6977    0.6038    0.4333    0.2500
## Neg Pred Value          0.9919    0.9926    0.9850    0.9864
## Prevalence              0.0200    0.0200    0.0200    0.0200
## Detection Rate          0.0120    0.0128    0.0052    0.0068
## Detection Prevalence    0.0172    0.0212    0.0120    0.0272
## Balanced Accuracy       0.7973    0.8157    0.6265    0.6596
```

An accuracy of 57.96% after performing pca and then doing multinomial logistic regression. Even though the Naive Bayes approach does not perform any better than the multinomial logistic regression, for ease of interpretability, Naive Bayes classification model would be preferred.

Which authors are hardest to predict?

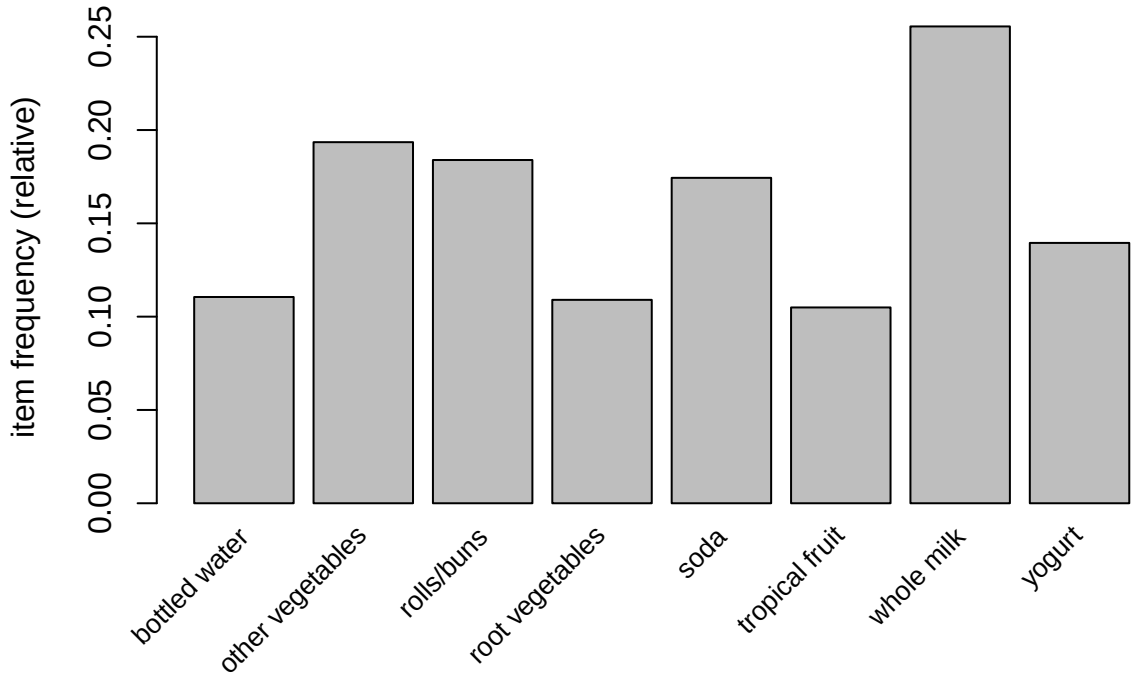
From the confusion matrix, we can see that Author 8 (David Lawder) and Author 44 (Sarah Davison) have a balanced accuracy of about 60% in both the regression models. After a little bit of Googling about these authors, it is hard not to see why they are the hardest authors to predict - both of them write on contemporary issues and are modernist authors so they use a lot of varied words which does not belong to any particular category per se.

Problem 3

```
suppressMessages(library(arules))
detach(package:tm, unload = TRUE)
groceries <- read.transactions("groceries.txt", format = "basket", sep = ",")
```

To see which items are important in the transcations we use itemFrequencePlot() to plot those items with frequency greater than 10%. Looks like “whole milk” is the most frequently occuring item.

```
itemFrequencyPlot(groceries, support = 0.1, cex.names=0.8)
```



We will create a set of rules with a low support value of 0.01 and low confidence of 0.2. We are not concerned with eliminating "uninteresting" rules at this point.

```
grocery_rules <- apriori(groceries, parameter = list(support = 0.01, confidence = 0.2, maxlen = 4))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport support minlen maxlen
##         0.2    0.1    1 none FALSE            TRUE    0.01      1      4
##  target   ext
##   rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 98
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[169 item(s), 9835 transaction(s)] done [0.01s].
## sorting and recoding items ... [88 item(s)] done [0.00s].
## creating transaction tree ... done [0.01s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [232 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

Let's look at interesting rules. A rule that has a low support but high confidence is quite unexpected.

```
inspect(subset(grocery_rules, subset=support < 0.02 & confidence > 0.5))
```

```
##    lhs                    rhs                  support confidence     lift
```

```
## 1  {curd,
##     yogurt}             => {whole milk}       0.01006609  0.5823529 2.279125
## 2  {butter,
##     other vegetables}   => {whole milk}       0.01148958  0.5736041 2.244885
## 3  {domestic eggs,
##     other vegetables}   => {whole milk}       0.01230300  0.5525114 2.162336
## 4  {whipped/sour cream,
##     yogurt}             => {whole milk}       0.01087951  0.5245098 2.052747
## 5  {other vegetables,
##     whipped/sour cream} => {whole milk}       0.01464159  0.5070423 1.984385
## 6  {other vegetables,
##     pip fruit}          => {whole milk}       0.01352313  0.5175097 2.025351
## 7  {citrus fruit,
##     root vegetables}    => {other vegetables} 0.01037112  0.5862069 3.029608
## 8  {root vegetables,
##     tropical fruit}     => {other vegetables} 0.01230300  0.5845411 3.020999
## 9  {root vegetables,
##     tropical fruit}     => {whole milk}       0.01199797  0.5700483 2.230969
## 10 {tropical fruit,
##     yogurt}             => {whole milk}       0.01514997  0.5173611 2.024770
## 11 {root vegetables,
##     yogurt}             => {whole milk}       0.01453991  0.5629921 2.203354
## 12 {rolls/buns,
##     root vegetables}    => {other vegetables} 0.01220132  0.5020921 2.594890
## 13 {rolls/buns,
##     root vegetables}    => {whole milk}       0.01270971  0.5230126 2.046888
```

The values of lift for all these transactions are greater than 1 => the antecedent and precedent in the rules are positively correlated,i.e, there is a higher chance of seeing the antecedent if the precedent occurs. For ex, in transaction 9 {root vegetables,tropical fruit} => {whole milk}, the customer has a 2.23 times more likely to buy whole milk if he/she buys root vegetables and tropical fruit.

A classic case of misunderstanding rules arises when we look at only confidence values. High support, high confidence and low lift is not an interesting result. In fact it could be a misleading result.

```
inspect(subset(grocery_rules, subset=support < 0.02 & confidence > 0.5 & lift < 1))
```

There doesn't seem to be any transactions. Let's look at those transaction with low lift values

```
inspect(subset(grocery_rules, subset=lift < 1))
```

```
##     lhs                rhs          support    confidence lift
## 69  {bottled beer}  => {whole milk} 0.02043721 0.2537879  0.9932367
## 117 {shopping bags} => {whole milk} 0.02450432 0.2487100  0.9733637
## 141 {soda}          => {whole milk} 0.04006101 0.2297376  0.8991124
```

These three transactions have negative correlation between the antecedent and precedent. If you buy bottled beer, you are less likely to buy whole milk. This makes a lot of sense because these could be customers who just wanted a few things from the store and are not making deliberate shopping trips with long term goals in mind.

Let's now look at those transactions with high lift values (unlikely to be a chance rule) and see the support and confidence values for those transactions.

```
inspect(subset(grocery_rules, subset=lift > 1))
```

```
##     lhs                            rhs                    support confidence     lift
## 1   {hard cheese}               => {whole milk}          0.01006609  0.4107884 1.607682
## 2   {butter milk}               => {other vegetables}    0.01037112  0.3709091 1.916916
## 3   {butter milk}               => {whole milk}          0.01159126  0.4145455 1.622385
## 4   {ham}                       => {whole milk}          0.01148958  0.4414062 1.727509
## 5   {sliced cheese}             => {whole milk}          0.01077783  0.4398340 1.721356
## 6   {oil}                       => {whole milk}          0.01128622  0.4021739 1.573968
## 7   {onions}                    => {other vegetables}    0.01423488  0.4590164 2.372268
## 8   {onions}                    => {whole milk}          0.01209964  0.3901639 1.526965
## 9   {berries}                   => {yogurt}              0.01057448  0.3180428 2.279848
## 10  {berries}                   => {other vegetables}    0.01026945  0.3088685 1.596280
## 11  {berries}                   => {whole milk}          0.01179461  0.3547401 1.388328
## 12  {hamburger meat}            => {other vegetables}    0.01382816  0.4159021 2.149447
## 13  {hamburger meat}            => {whole milk}          0.01474326  0.4434251 1.735410
## 14  {hygiene articles}          => {whole milk}          0.01281139  0.3888889 1.521975
## 15  {salty snack}               => {other vegetables}    0.01077783  0.2849462 1.472646
## 16  {salty snack}               => {whole milk}          0.01118454  0.2956989 1.157262
## 17  {sugar}                     => {other vegetables}    0.01077783  0.3183183 1.645119
## 18  {sugar}                     => {whole milk}          0.01504830  0.4444444 1.739400
## 19  {waffles}                   => {other vegetables}    0.01006609  0.2619048 1.353565
## 20  {waffles}                   => {whole milk}          0.01270971  0.3306878 1.294196
## 21  {long life bakery product} => {other vegetables}    0.01067616  0.2853261 1.474610
## 22  {long life bakery product} => {whole milk}          0.01352313  0.3614130 1.414444
## 23  {dessert}                   => {other vegetables}    0.01159126  0.3123288 1.614164
## 24  {dessert}                   => {whole milk}          0.01372649  0.3698630 1.447514
## 25  {cream cheese}              => {yogurt}              0.01240468  0.3128205 2.242412
## 26  {cream cheese}              => {other vegetables}    0.01372649  0.3461538 1.788977
## 27  {cream cheese}              => {whole milk}          0.01647178  0.4153846 1.625670
## 28  {chicken}                   => {root vegetables}     0.01087951  0.2535545 2.326221
## 29  {chicken}                   => {other vegetables}    0.01789527  0.4170616 2.155439
## 30  {chicken}                   => {whole milk}          0.01759024  0.4099526 1.604411
## 31  {white bread}               => {soda}                0.01026945  0.2439614 1.399044
## 32  {white bread}               => {other vegetables}    0.01372649  0.3260870 1.685268
## 33  {white bread}               => {whole milk}          0.01708185  0.4057971 1.588147
## 34  {chocolate}                 => {soda}                0.01352313  0.2725410 1.562939
## 35  {chocolate}                 => {rolls/buns}          0.01179461  0.2377049 1.292332
## 36  {chocolate}                 => {other vegetables}    0.01270971  0.2561475 1.323810
## 37  {chocolate}                 => {whole milk}          0.01667514  0.3360656 1.315243
## 38  {coffee}                    => {other vegetables}    0.01342145  0.2311734 1.194740
## 39  {coffee}                    => {whole milk}          0.01870869  0.3222417 1.261141
## 40  {frozen vegetables}         => {root vegetables}     0.01159126  0.2410148 2.211176
## 41  {frozen vegetables}         => {yogurt}              0.01240468  0.2579281 1.848924
## 42  {frozen vegetables}         => {rolls/buns}          0.01016777  0.2114165 1.149409
## 43  {frozen vegetables}         => {other vegetables}    0.01779359  0.3699789 1.912108
## 44  {frozen vegetables}         => {whole milk}          0.02043721  0.4249471 1.663094
## 45  {beef}                      => {root vegetables}     0.01738688  0.3313953 3.040367
## 46  {beef}                      => {yogurt}              0.01169293  0.2228682 1.597601
## 47  {beef}                      => {rolls/buns}          0.01362481  0.2596899 1.411858
## 48  {beef}                      => {other vegetables}    0.01972547  0.3759690 1.943066
## 49  {beef}                      => {whole milk}          0.02125064  0.4050388 1.585180
## 50  {curd}                      => {root vegetables}     0.01087951  0.2041985 1.873407
```

```
##  51  {curd}                     => {yogurt}            0.01728521  0.3244275 2.325615
##  52  {curd}                     => {other vegetables}  0.01718353  0.3225191 1.666829
##  53  {curd}                     => {whole milk}        0.02613116  0.4904580 1.919481
##  54  {napkins}                  => {soda}              0.01199797  0.2291262 1.313969
##  55  {napkins}                  => {yogurt}            0.01230300  0.2349515 1.684218
##  56  {napkins}                  => {rolls/buns}        0.01169293  0.2233010 1.214022
##  57  {napkins}                  => {other vegetables}  0.01443823  0.2757282 1.425006
##  58  {napkins}                  => {whole milk}        0.01972547  0.3766990 1.474268
##  59  {pork}                     => {root vegetables}   0.01362481  0.2363316 2.168210
##  60  {pork}                     => {soda}              0.01189629  0.2063492 1.183350
##  61  {pork}                     => {other vegetables}  0.02165735  0.3756614 1.941476
##  62  {pork}                     => {whole milk}        0.02216573  0.3844797 1.504719
##  63  {frankfurter}              => {rolls/buns}        0.01921708  0.3258621 1.771616
##  64  {frankfurter}              => {other vegetables}  0.01647178  0.2793103 1.443519
##  65  {frankfurter}              => {whole milk}        0.02053889  0.3482759 1.363029
##  66  {bottled beer}             => {soda}              0.01698017  0.2108586 1.209209
##  67  {bottled beer}             => {other vegetables}  0.01616675  0.2007576 1.037546
##  68  {brown bread}              => {yogurt}            0.01453991  0.2241379 1.606703
##  69  {brown bread}              => {other vegetables}  0.01870869  0.2884013 1.490503
##  70  {brown bread}              => {whole milk}        0.02521607  0.3887147 1.521293
##  71  {margarine}                => {yogurt}            0.01423488  0.2430556 1.742312
##  72  {margarine}                => {rolls/buns}        0.01474326  0.2517361 1.368615
##  73  {margarine}                => {other vegetables}  0.01972547  0.3368056 1.740663
##  74  {margarine}                => {whole milk}        0.02419929  0.4131944 1.617098
##  75  {butter}                   => {root vegetables}   0.01291307  0.2330275 2.137897
##  76  {butter}                   => {yogurt}            0.01464159  0.2642202 1.894027
##  77  {butter}                   => {rolls/buns}        0.01342145  0.2422018 1.316780
##  78  {butter}                   => {other vegetables}  0.02003050  0.3614679 1.868122
##  79  {butter}                   => {whole milk}        0.02755465  0.4972477 1.946053
##  80  {newspapers}               => {rolls/buns}        0.01972547  0.2471338 1.343593
##  81  {newspapers}               => {other vegetables}  0.01931876  0.2420382 1.250891
##  82  {newspapers}               => {whole milk}        0.02735130  0.3426752 1.341110
##  83  {domestic eggs}            => {root vegetables}   0.01433655  0.2259615 2.073071
##  84  {domestic eggs}            => {yogurt}            0.01433655  0.2259615 1.619775
##  85  {domestic eggs}            => {rolls/buns}        0.01565836  0.2467949 1.341751
##  86  {domestic eggs}            => {other vegetables}  0.02226741  0.3509615 1.813824
##  87  {domestic eggs}            => {whole milk}        0.02999492  0.4727564 1.850203
##  88  {fruit/vegetable juice}    => {soda}              0.01840366  0.2545710 1.459887
##  89  {fruit/vegetable juice}    => {yogurt}            0.01870869  0.2587904 1.855105
##  90  {fruit/vegetable juice}    => {rolls/buns}        0.01453991  0.2011252 1.093458
##  91  {fruit/vegetable juice}    => {other vegetables}  0.02104728  0.2911392 1.504653
##  92  {fruit/vegetable juice}    => {whole milk}        0.02663955  0.3684951 1.442160
##  93  {whipped/sour cream}       => {root vegetables}   0.01708185  0.2382979 2.186250
##  94  {whipped/sour cream}       => {yogurt}            0.02074225  0.2893617 2.074251
##  95  {whipped/sour cream}       => {rolls/buns}        0.01464159  0.2042553 1.110476
##  96  {whipped/sour cream}       => {other vegetables}  0.02887646  0.4028369 2.081924
##  97  {whipped/sour cream}       => {whole milk}        0.03223183  0.4496454 1.759754
##  98  {pip fruit}                => {tropical fruit}    0.02043721  0.2701613 2.574648
##  99  {pip fruit}                => {root vegetables}   0.01555669  0.2056452 1.886679
## 100  {pip fruit}                => {yogurt}            0.01799695  0.2379032 1.705378
## 101  {pip fruit}                => {other vegetables}  0.02613116  0.3454301 1.785237
## 102  {pip fruit}                => {whole milk}        0.03009659  0.3978495 1.557043
## 103  {pastry}                   => {soda}              0.02104728  0.2365714 1.356665
## 104  {pastry}                   => {rolls/buns}        0.02094560  0.2354286 1.279956
```

```
## 105 {pastry}                  => {other vegetables}   0.02257245  0.2537143 1.311235
## 106 {pastry}                  => {whole milk}         0.03324860  0.3737143 1.462587
## 107 {citrus fruit}            => {tropical fruit}     0.01992883  0.2407862 2.294702
## 108 {citrus fruit}            => {root vegetables}    0.01769192  0.2137592 1.961121
## 109 {citrus fruit}            => {yogurt}             0.02165735  0.2616708 1.875752
## 110 {citrus fruit}            => {rolls/buns}         0.01677682  0.2027027 1.102035
## 111 {citrus fruit}            => {other vegetables}   0.02887646  0.3488943 1.803140
## 112 {citrus fruit}            => {whole milk}         0.03050330  0.3685504 1.442377
## 113 {shopping bags}           => {soda}               0.02460600  0.2497420 1.432194
## 114 {shopping bags}           => {other vegetables}   0.02318251  0.2352941 1.216037
## 115 {sausage}                 => {soda}               0.02430097  0.2586580 1.483324
## 116 {sausage}                 => {yogurt}             0.01962379  0.2088745 1.497289
## 117 {sausage}                 => {rolls/buns}         0.03060498  0.3257576 1.771048
## 118 {sausage}                 => {other vegetables}   0.02694459  0.2867965 1.482209
## 119 {sausage}                 => {whole milk}         0.02989324  0.3181818 1.245252
## 120 {bottled water}           => {soda}               0.02897814  0.2621895 1.503577
## 121 {bottled water}           => {yogurt}             0.02297916  0.2079117 1.490387
## 122 {bottled water}           => {rolls/buns}         0.02419929  0.2189512 1.190373
## 123 {bottled water}           => {other vegetables}   0.02480935  0.2244710 1.160101
## 124 {bottled water}           => {whole milk}         0.03436706  0.3109476 1.216940
## 125 {tropical fruit}          => {root vegetables}    0.02104728  0.2005814 1.840222
## 126 {tropical fruit}          => {yogurt}             0.02928317  0.2790698 2.000475
## 127 {yogurt}                  => {tropical fruit}     0.02928317  0.2099125 2.000475
## 128 {tropical fruit}          => {rolls/buns}         0.02460600  0.2344961 1.274886
## 129 {tropical fruit}          => {other vegetables}   0.03589222  0.3420543 1.767790
## 130 {tropical fruit}          => {whole milk}         0.04229792  0.4031008 1.577595
## 131 {root vegetables}         => {yogurt}             0.02582613  0.2369403 1.698475
## 132 {root vegetables}         => {rolls/buns}         0.02430097  0.2229478 1.212101
## 133 {root vegetables}         => {other vegetables}   0.04738180  0.4347015 2.246605
## 134 {other vegetables}        => {root vegetables}    0.04738180  0.2448765 2.246605
## 135 {root vegetables}         => {whole milk}         0.04890696  0.4486940 1.756031
## 136 {soda}                    => {rolls/buns}         0.03833249  0.2198251 1.195124
## 137 {rolls/buns}              => {soda}               0.03833249  0.2084024 1.195124
## 138 {yogurt}                  => {rolls/buns}         0.03436706  0.2463557 1.339363
## 139 {yogurt}                  => {other vegetables}   0.04341637  0.3112245 1.608457
## 140 {other vegetables}        => {yogurt}             0.04341637  0.2243826 1.608457
## 141 {yogurt}                  => {whole milk}         0.05602440  0.4016035 1.571735
## 142 {whole milk}              => {yogurt}             0.05602440  0.2192598 1.571735
## 143 {rolls/buns}              => {other vegetables}   0.04260295  0.2316197 1.197047
## 144 {other vegetables}        => {rolls/buns}         0.04260295  0.2201787 1.197047
## 145 {rolls/buns}              => {whole milk}         0.05663447  0.3079049 1.205032
## 146 {whole milk}              => {rolls/buns}         0.05663447  0.2216474 1.205032
## 147 {other vegetables}        => {whole milk}         0.07483477  0.3867578 1.513634
## 148 {whole milk}              => {other vegetables}   0.07483477  0.2928770 1.513634
## 149 {curd,
##      yogurt}                  => {whole milk}         0.01006609  0.5823529 2.279125
## 150 {curd,
##      whole milk}              => {yogurt}             0.01006609  0.3852140 2.761356
## 151 {other vegetables,
##      pork}                    => {whole milk}         0.01016777  0.4694836 1.837394
## 152 {pork,
##      whole milk}              => {other vegetables}   0.01016777  0.4587156 2.370714
## 153 {butter,
##      other vegetables}        => {whole milk}         0.01148958  0.5736041 2.244885
```

```
## 154 {butter,
##      whole milk}              => {other vegetables}   0.01148958  0.4169742 2.154987
## 155 {domestic eggs,
##      other vegetables}        => {whole milk}         0.01230300  0.5525114 2.162336
## 156 {domestic eggs,
##      whole milk}              => {other vegetables}   0.01230300  0.4101695 2.119820
## 157 {fruit/vegetable juice,
##      other vegetables}        => {whole milk}         0.01047280  0.4975845 1.947371
## 158 {fruit/vegetable juice,
##      whole milk}              => {other vegetables}   0.01047280  0.3931298 2.031756
## 159 {whipped/sour cream,
##      yogurt}                  => {other vegetables}   0.01016777  0.4901961 2.533410
## 160 {other vegetables,
##      whipped/sour cream}      => {yogurt}             0.01016777  0.3521127 2.524073
## 161 {other vegetables,
##      yogurt}                  => {whipped/sour cream} 0.01016777  0.2341920 3.267062
## 162 {whipped/sour cream,
##      yogurt}                  => {whole milk}         0.01087951  0.5245098 2.052747
## 163 {whipped/sour cream,
##      whole milk}              => {yogurt}             0.01087951  0.3375394 2.419607
## 164 {other vegetables,
##      whipped/sour cream}      => {whole milk}         0.01464159  0.5070423 1.984385
## 165 {whipped/sour cream,
##      whole milk}              => {other vegetables}   0.01464159  0.4542587 2.347679
## 166 {other vegetables,
##      pip fruit}               => {whole milk}         0.01352313  0.5175097 2.025351
## 167 {pip fruit,
##      whole milk}              => {other vegetables}   0.01352313  0.4493243 2.322178
## 168 {other vegetables,
##      pastry}                  => {whole milk}         0.01057448  0.4684685 1.833421
## 169 {pastry,
##      whole milk}              => {other vegetables}   0.01057448  0.3180428 1.643695
## 170 {citrus fruit,
##      root vegetables}         => {other vegetables}   0.01037112  0.5862069 3.029608
## 171 {citrus fruit,
##      other vegetables}        => {root vegetables}    0.01037112  0.3591549 3.295045
## 172 {other vegetables,
##      root vegetables}         => {citrus fruit}       0.01037112  0.2188841 2.644626
## 173 {citrus fruit,
##      yogurt}                  => {whole milk}         0.01026945  0.4741784 1.855768
## 174 {citrus fruit,
##      whole milk}              => {yogurt}             0.01026945  0.3366667 2.413350
## 175 {citrus fruit,
##      other vegetables}        => {whole milk}         0.01301474  0.4507042 1.763898
## 176 {citrus fruit,
##      whole milk}              => {other vegetables}   0.01301474  0.4266667 2.205080
## 177 {other vegetables,
##      sausage}                 => {whole milk}         0.01016777  0.3773585 1.476849
## 178 {sausage,
##      whole milk}              => {other vegetables}   0.01016777  0.3401361 1.757876
## 179 {bottled water,
##      other vegetables}        => {whole milk}         0.01077783  0.4344262 1.700192
## 180 {bottled water,
##      whole milk}              => {other vegetables}   0.01077783  0.3136095 1.620783
```

```
## 181 {root vegetables,
##      tropical fruit}          => {other vegetables}   0.01230300  0.5845411 3.020999
## 182 {other vegetables,
##      tropical fruit}          => {root vegetables}    0.01230300  0.3427762 3.144780
## 183 {other vegetables,
##      root vegetables}         => {tropical fruit}     0.01230300  0.2596567 2.474538
## 184 {root vegetables,
##      tropical fruit}          => {whole milk}         0.01199797  0.5700483 2.230969
## 185 {tropical fruit,
##      whole milk}              => {root vegetables}    0.01199797  0.2836538 2.602365
## 186 {root vegetables,
##      whole milk}              => {tropical fruit}     0.01199797  0.2453222 2.337931
## 187 {tropical fruit,
##      yogurt}                  => {other vegetables}   0.01230300  0.4201389 2.171343
## 188 {other vegetables,
##      tropical fruit}          => {yogurt}             0.01230300  0.3427762 2.457146
## 189 {other vegetables,
##      yogurt}                  => {tropical fruit}     0.01230300  0.2833724 2.700550
## 190 {tropical fruit,
##      yogurt}                  => {whole milk}         0.01514997  0.5173611 2.024770
## 191 {tropical fruit,
##      whole milk}              => {yogurt}             0.01514997  0.3581731 2.567516
## 192 {whole milk,
##      yogurt}                  => {tropical fruit}     0.01514997  0.2704174 2.577089
## 193 {rolls/buns,
##      tropical fruit}          => {whole milk}         0.01098119  0.4462810 1.746587
## 194 {tropical fruit,
##      whole milk}              => {rolls/buns}         0.01098119  0.2596154 1.411452
## 195 {other vegetables,
##      tropical fruit}          => {whole milk}         0.01708185  0.4759207 1.862587
## 196 {tropical fruit,
##      whole milk}              => {other vegetables}   0.01708185  0.4038462 2.087140
## 197 {other vegetables,
##      whole milk}              => {tropical fruit}     0.01708185  0.2282609 2.175335
## 198 {root vegetables,
##      yogurt}                  => {other vegetables}   0.01291307  0.5000000 2.584078
## 199 {other vegetables,
##      root vegetables}         => {yogurt}             0.01291307  0.2725322 1.953611
## 200 {other vegetables,
##      yogurt}                  => {root vegetables}    0.01291307  0.2974239 2.728698
## 201 {root vegetables,
##      yogurt}                  => {whole milk}         0.01453991  0.5629921 2.203354
## 202 {root vegetables,
##      whole milk}              => {yogurt}             0.01453991  0.2972973 2.131136
## 203 {whole milk,
##      yogurt}                  => {root vegetables}    0.01453991  0.2595281 2.381025
## 204 {rolls/buns,
##      root vegetables}         => {other vegetables}   0.01220132  0.5020921 2.594890
## 205 {other vegetables,
##      root vegetables}         => {rolls/buns}         0.01220132  0.2575107 1.400010
## 206 {other vegetables,
##      rolls/buns}              => {root vegetables}    0.01220132  0.2863962 2.627525
## 207 {rolls/buns,
##      root vegetables}         => {whole milk}         0.01270971  0.5230126 2.046888
```

```
## 208 {root vegetables,
##      whole milk}              => {rolls/buns}         0.01270971  0.2598753 1.412865
## 209 {rolls/buns,
##      whole milk}              => {root vegetables}    0.01270971  0.2244165 2.058896
## 210 {other vegetables,
##      root vegetables}         => {whole milk}         0.02318251  0.4892704 1.914833
## 211 {root vegetables,
##      whole milk}              => {other vegetables}   0.02318251  0.4740125 2.449770
## 212 {other vegetables,
##      whole milk}              => {root vegetables}    0.02318251  0.3097826 2.842082
## 213 {soda,
##      yogurt}                  => {whole milk}         0.01047280  0.3828996 1.498535
## 214 {soda,
##      whole milk}              => {yogurt}             0.01047280  0.2614213 1.873964
## 215 {other vegetables,
##      soda}                    => {whole milk}         0.01392984  0.4254658 1.665124
## 216 {soda,
##      whole milk}              => {other vegetables}   0.01392984  0.3477157 1.797049
## 217 {rolls/buns,
##      yogurt}                  => {other vegetables}   0.01148958  0.3343195 1.727815
## 218 {other vegetables,
##      yogurt}                  => {rolls/buns}         0.01148958  0.2646370 1.438753
## 219 {other vegetables,
##      rolls/buns}              => {yogurt}             0.01148958  0.2696897 1.933235
## 220 {rolls/buns,
##      yogurt}                  => {whole milk}         0.01555669  0.4526627 1.771563
## 221 {whole milk,
##      yogurt}                  => {rolls/buns}         0.01555669  0.2776770 1.509648
## 222 {rolls/buns,
##      whole milk}              => {yogurt}             0.01555669  0.2746858 1.969049
## 223 {other vegetables,
##      yogurt}                  => {whole milk}         0.02226741  0.5128806 2.007235
## 224 {whole milk,
##      yogurt}                  => {other vegetables}   0.02226741  0.3974592 2.054131
## 225 {other vegetables,
##      whole milk}              => {yogurt}             0.02226741  0.2975543 2.132979
## 226 {other vegetables,
##      rolls/buns}              => {whole milk}         0.01789527  0.4200477 1.643919
## 227 {rolls/buns,
##      whole milk}              => {other vegetables}   0.01789527  0.3159785 1.633026
## 228 {other vegetables,
##      whole milk}              => {rolls/buns}         0.01789527  0.2391304 1.300082
```

These results tell us that - if you have a low support and low confidence, we must be cautious about disregarding the rules as uninteresting. The high lift values tell a different story. One possible reason for low confidence and low support values is that these transactions do not occur often (low support) and may not have high correctness or reliability (low confidence), but still, the high lift value means that the rules are not a coincidence. They do not occur by chance and it is worth the effort paying attention to them.

For example in transaction 183 {root vegetables,tropical fruit} => {other vegetables} , we are 58% confident that a customer is going to buy other vegetables if he/she buys root vegetables and tropical fruit and they are 3.02 times more likely to buy other vegetables given that they bought root vegetables and tropical fruit. These are not shoppers who grab one or two items out of necessity.