**Unexecuted Code**

## Homework 10 - 28 Aug 2018

#------------------------------------

#Step 1 - load the data

#------------------------------------

library(tm)

library(stringr)

library(wordcloud)

##Step 1: Read in AFINN words and speech

#File name for sentiment words

wordfile <- "AFINN-111.txt"

#Reading the file, and separating each row by the \n

words <- scan(wordfile, character(0), sep = "\n")

#Regular expression to delete everything but the numbers

val <- gsub("(['ïa-z0-9-]+[[:space:]]?)\*\t", "", words)

#Converting the numbers to a numeric format

val <- as.numeric(val)

#Deleting the \t and the numbers after

words <- gsub("(\t)-?[0-9]", "", words)

#Creating df for the word and the sentiment

sentdf <- data.frame(words, sentiment = val)

#Checking for null values

colSums(is.na(sentdf))

mlk <- scan("mlk.txt", character(0), sep = "\n")

#Vectorizing the paragraphs

words.vec <- VectorSource(mlk)

#Creating a corpus

corpus <- Corpus(words.vec)

#Converting to lowercase

corpus <- tm\_map(corpus, content\_transformer(tolower))

#Removing punctuation

corpus <- tm\_map(corpus, removePunctuation)

#Removing numbers

corpus <- tm\_map(corpus, removeNumbers)

#Removing stopwords

corpus <- tm\_map(corpus, removeWords, stopwords("english"))

#Convert to TermDocumentMatrix

tdm <- TermDocumentMatrix(corpus)

#Making tdm as a matrix

word.count <- as.matrix(tdm)

##Step 2: Get overall score for speech

#Creating function to get sentiment score

sentiment\_analysis <- function(m){

#Getting the number of times the word was said in the matrix

m <- rowSums(m)

#Indexing words with sentiment

sindex <- match(names(m), sentdf$words)

#multiplying the values in the indexes that had sentiment by the

#sentiment value given to said word

swords <- m[which(!is.na(sindex))] \* sentdf[sindex[!is.na(sindex)], "sentiment"]

#Getting sum of all the negative words weighted

n <- sum(swords[swords < 0])

#Getting sum of all the positive words weighted

p <- sum(swords[swords > 0])

#Returning the total score, the negative to positive ratio, the negative score

#and the positive score

return(list("total" = n + p, "N/P" = abs(n)/p, "negative" = n, "positive" = p))

}

#Running sentiment on full speech

fullspeech <- sentiment\_analysis(word.count)

sent\_score <- numeric()

for (i in 0:3){

quarter <- word.count[,((i\*7)+1):((i+1)\*7)]

sent\_score[i+1] <- sentiment\_analysis(quarter)[[1]]

}

sent<- data.frame(quarter = c("Q1", "Q2", "Q3", "Q4"), score = sent\_score)

library(ggplot2)

p <- ggplot(sent, aes(x = quarter, y = score)) + geom\_bar(stat = "identity")

p <- p + ggtitle("I Have a Dream Sentiment by Quarter")

p <- p + ylab("Score") + xlab("Quarter")

p

**Console log w/plot**

**Executed code**

> ## Homework 10 - 28 Aug 2018

> #------------------------------------

> #Step 1 - load the data

> #------------------------------------

> library(tm)

> library(stringr)

> library(wordcloud)

>

> ##Step 1: Read in AFINN words and speech

> #File name for sentiment words

> wordfile <- "AFINN-111.txt"

> #Reading the file, and separating each row by the \n

> words <- scan(wordfile, character(0), sep = "\n")

Read 2477 items

> #Regular expression to delete everything but the numbers

> val <- gsub("(['ïa-z0-9-]+[[:space:]]?)\*\t", "", words)

> #Converting the numbers to a numeric format

> val <- as.numeric(val)

> #Deleting the \t and the numbers after

> words <- gsub("(\t)-?[0-9]", "", words)

> #Creating df for the word and the sentiment

> sentdf <- data.frame(words, sentiment = val)

> #Checking for null values

> colSums(is.na(sentdf))

words sentiment

0 0

>

> mlk <- scan("mlk.txt", character(0), sep = "\n")

Read 28 items

> #Vectorizing the paragraphs

> words.vec <- VectorSource(mlk)

> #Creating a corpus

> corpus <- Corpus(words.vec)

> #Converting to lowercase

> corpus <- tm\_map(corpus, content\_transformer(tolower))

Warning message:

In tm\_map.SimpleCorpus(corpus, content\_transformer(tolower)) :

transformation drops documents

> #Removing punctuation

> corpus <- tm\_map(corpus, removePunctuation)

Warning message:

In tm\_map.SimpleCorpus(corpus, removePunctuation) :

transformation drops documents

> #Removing numbers

> corpus <- tm\_map(corpus, removeNumbers)

Warning message:

In tm\_map.SimpleCorpus(corpus, removeNumbers) :

transformation drops documents

> #Removing stopwords

> corpus <- tm\_map(corpus, removeWords, stopwords("english"))

Warning message:

In tm\_map.SimpleCorpus(corpus, removeWords, stopwords("english")) :

transformation drops documents

> #Convert to TermDocumentMatrix

> tdm <- TermDocumentMatrix(corpus)

> #Making tdm as a matrix

> word.count <- as.matrix(tdm)

>

> ##Step 2: Get overall score for speech

> #Creating function to get sentiment score

> sentiment\_analysis <- function(m){

+ #Getting the number of times the word was said in the matrix

+ m <- rowSums(m)

+ #Indexing words with sentiment

+ sindex <- match(names(m), sentdf$words)

+ #multiplying the values in the indexes that had sentiment by the

+ #sentiment value given to said word

+ swords <- m[which(!is.na(sindex))] \* sentdf[sindex[!is.na(sindex)], "sentiment"]

+ #Getting sum of all the negative words weighted

+ n <- sum(swords[swords < 0])

+ #Getting sum of all the positive words weighted

+ p <- sum(swords[swords > 0])

+ #Returning the total score, the negative to positive ratio, the negative score

+ #and the positive score

+ return(list("total" = n + p, "N/P" = abs(n)/p, "negative" = n, "positive" = p))

+ }

> #Running sentiment on full speech

> fullspeech <- sentiment\_analysis(word.count)

>

> sent\_score <- numeric()

> for (i in 0:3){

+ quarter <- word.count[,((i\*7)+1):((i+1)\*7)]

+ sent\_score[i+1] <- sentiment\_analysis(quarter)[[1]]

+ }

>

> sent<- data.frame(quarter = c("Q1", "Q2", "Q3", "Q4"), score = sent\_score)

>

> library(ggplot2)

> p <- ggplot(sent, aes(x = quarter, y = score)) + geom\_bar(stat = "identity")

> p <- p + ggtitle("I Have a Dream Sentiment by Quarter")

> p <- p + ylab("Score") + xlab("Quarter")

