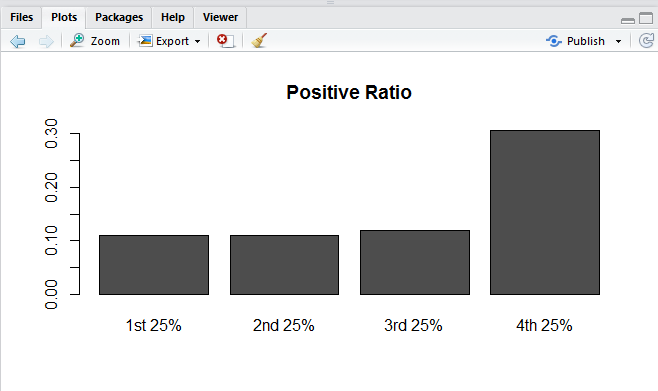
**#ADS IST 687**

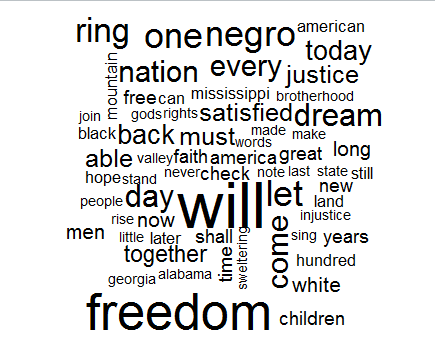
**#Jacob Dineen**

**#Homework 10**

**#Due 9/24/2017**

Plot:





**#################All Calls to Clear Environment and Fetch Packages**

**#CLEAR ENVIRONMENT AND INSTALL INITIAL PACKAGES**

**rm(list = ls(all = TRUE))#Clear Enviroment**

**#specify the packages of interest**

**packages=c("tm","wordcloud")**

**#use this function to check if each package is on the local machine**

**#if a package is installed, it will be loaded**

**#if any are not, the missing package(s) will be installed and loaded**

**package.check <- lapply(packages, FUN = function(x) {**

**if (!require(x, character.only = TRUE)) {**

**install.packages(x, dependencies = TRUE)**

**library(x, character.only = TRUE)**

**}**

**})**

**#Read in the positive and negative keyword files**

**#Load the files**

**pos <- file.choose()**

**neg <- file.choose()**

**#Read the Positive and Negative word Files + clean**

**p <- scan(pos, character(0), sep= "\n")**

**head(p,40)**

**p <- p[-1:-34]**

**tail(p)**

**head(p)**

**n <- scan(neg, character(0), sep= "\n")**

**head(n,40)**

**p <- p[-1:-34]**

**tail(p)**

**head(p)**

**#Process in the MLK Speech**

**mlkFile <- file.choose()**

**mlk <- readLines(mlkFile)**

**str(mlk)**

**#Transformation**

**words.vec <- VectorSource((mlk))**

**words.corpus <- Corpus(words.vec)**

**words.corpus**

**#Munge Data**

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

**words.corpus <- tm\_map(words.corpus, removePunctuation)**

**words.corpus <- tm\_map(words.corpus, removeNumbers)**

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

**#Create TermDocMatrix**

**tdm <- TermDocumentMatrix(words.corpus)**

**tdm**

**#Store as matrix**

**m<- as.matrix(tdm)**

**m**

**#Create list of counts for each word**

**wordCounts <- rowSums(m)**

**wordCounts**

**#Sum total word count**

**totalwords <- sum(wordCounts)**

**totalwords**

**#Sort wordCount by freq of occurrence**

**wordCounts <- sort(wordCounts, decreasing=TRUE)**

**head(wordCounts)**

**wordCounts**

**wordfreq <- data.frame(wordCounts)**

**wordfreq**

**colnames(wordfreq)**

**#Create Test Wordcloud**

**cloudFrame<-data.frame(word=names(wordCounts),freq=wordCounts)**

**cloudFrame[1:10,]**

**wordcloud(cloudFrame$word,cloudFrame$freq)**

**wordcloud(names(wordCounts),wordCounts,min.freq=3,max.words=50,rot.per=.35,colors=brewer.pal(8,"Dark2"))**

**#Determine How many Positive words were in the speech**

**totalWords <- sum(wordCounts)**

**totalWords**

**words <- names(wordCounts)**

**wordCounts**

**matched <- match(words,p, nomatch = 0)**

**head(matched,150)**

**pWords <- words[which(matched !=0)]**

**head(pWords)**

**Pcounts <- wordCounts[which(matched !=0)]**

**head(Pcounts)**

**nPos <- sum(Pcounts)**

**nPos #95 positive words**

**length(Pcounts) #50 unique positive words**

**nPos/totalWords #11.2960% Positive**

**#Determine How many Negative words were in the speech**

**matched <- match(words,n, nomatch = 0)**

**ncounts <- wordCounts[which(matched !=0)]**

**nNeg <- sum(ncounts)**

**nNeg #63 negative words**

**length(ncounts) #54 unique negative words**

**nWords <- names(ncounts)**

**nWords**

**nNeg/totalWords #7.4910% negative**

**#Calculating Ratio of positive to negative words**

**sentimentratio <- nPos/nNeg**

**sentimentratio #There were 1.5 positive words to every negative word within the MLK speech.**

**# Sentiment Affinity Score**

**#Read in affinity project**

**AFINN<-read.delim(file.choose(),sep="\t",header = FALSE)**

**str(AFINN)**

**head(AFINN)**

**colnames(AFINN)<-c("Word", "Score")**

**AFINN[1:10,]**

**#Merge two dfs - cloudframe and Afinn**

**mergedTable<-merge(cloudFrame,AFINN,by.x="word",by.y="Word")**

**mergedTable[1:10,]**

**str(mergedTable)**

**#Compute total Score and scaled total score**

**overallScore<-sum(mergedTable$freq\*mergedTable$Score)**

**overallScore #Overall score = 113**

**overallScore/totalWords #Scaled Score = 13.43%**

**# Splitting speech into quarters**

**cutpoint <- round(length(words.corpus)/4) #Create cutpoint for MLK speech**

**# first 25%**

**# create word corpus for the first quarter using cutpoints**

**words.corpus1 <- words.corpus[1:cutpoint]**

**# create term document matrix for the first quarter**

**tdm1 <- TermDocumentMatrix(words.corpus1)**

**# convert tdm1 into a matrix called "m1"**

**m1 <- as.matrix(tdm1)**

**# create a list of word counts for the first quarter and sort the list**

**wordCounts1 <- rowSums(m1)**

**wordCounts1 <- sort(wordCounts1, decreasing=TRUE)**

**# calculate total words of the first 25%**

**totalWords1 <- sum(wordCounts1)**

**# create a vector that contains all the words in "wordCounts1"**

**words1 <- names(wordCounts1)**

**# locate which words in first quarter were positive (appeared in positive-word list)**

**matchedP1 <- match(words1, p, nomatch = 0)**

**# calculate the number of positive words in first quarter**

**ptotalNumber1 <- sum(wordCounts1[which(matchedP1 != 0)])**

**# calculate the ratio of positive words in first quarter**

**ratiop1 <- ptotalNumber1/totalWords1**

**# locate which words in first quarter were negative (appeared in negative-word list)**

**matchedN1 <- match(words1, n, nomatch = 0)**

**# calculate the number of negative words in first quarter**

**ntotalNumber1 <- sum(wordCounts1[which(matchedN1 != 0)])**

**# calculate the ratio of negative words in first quarter**

**ration1 <- ntotalNumber1/totalWords1**

**ration1**

**cloudFrame1<-data.frame(word=names(wordCounts1),freq=wordCounts1)**

**mergedTable1<-merge(cloudFrame1,AFINN,by.x="word",by.y="Word")**

**overallScore1<-sum(mergedTable1$freq\*mergedTable1$Score)**

**overallScore1**

**overallscore1scaled <-overallScore1/totalWords1**

**# Total Score Q1 = .09589**

**# second 25%**

**# create word corpus for the second quarter using cutpoints**

**words.corpus2 <- words.corpus[(cutpoint+1):(2\*cutpoint)]**

**# create term document matrix for the second quarter**

**tdm2 <- TermDocumentMatrix(words.corpus2)**

**m2 <- as.matrix(tdm2)**

**# create a list of word counts for the second quarter and sort the list**

**wordCounts2 <- rowSums(m2)**

**wordCounts2<- sort(wordCounts2, decreasing=TRUE)**

**# calculate total words of the second 25%**

**totalWords2 <- sum(wordCounts2)**

**# create a vector that contains all the words in "wordCounts2"**

**words2 <- names(wordCounts2)**

**# locate which words in second quarter were positive (appeared in positive-word list)**

**matchedP2 <- match(words2, p, nomatch = 0)**

**# calculate the number of positive words in second quarter**

**ptotalNumber2 <- sum(wordCounts2[which(matchedP2 != 0)])**

**# calculate the ratio of positive words in second quarter**

**ratiop2 <- ptotalNumber2/totalWords2**

**# locate which words in second quarter were negative (appeared in negative-word list)**

**matchedN2 <- match(words2, n, nomatch = 0)**

**# calculate the number of negative words in second quarter**

**ntotalNumber2 <- sum(wordCounts2[which(matchedN2 != 0)])**

**# calculate the ratio of negative words in second quarter**

**ration2 <- ntotalNumber2/totalWords2**

**ration2**

**cloudFrame2<-data.frame(word=names(wordCounts2),freq=wordCounts2)**

**mergedTable2<-merge(cloudFrame2,AFINN,by.x="word",by.y="Word")**

**overallScore2<-sum(mergedTable2$freq\*mergedTable2$Score)**

**overallScore2**

**overallscore2scaled <-overallScore2/totalWords2**

**# third 25%**

**# create word corpus for the third quarter using cutpoints**

**words.corpus3 <- words.corpus[(2\*cutpoint+1):cutpoint]**

**# create term document matrix for the third quarter**

**tdm3 <- TermDocumentMatrix(words.corpus3)**

**m3 <- as.matrix(tdm3)**

**# create a list of word counts for the third quarter and sort the list**

**wordCounts3 <- rowSums(m3)**

**wordCounts3<- sort(wordCounts3, decreasing=TRUE)**

**# calculate total words of the third 25%**

**totalWords3 <- sum(wordCounts3)**

**# create a vector that contains all the words in "wordCounts3"**

**words3 <- names(wordCounts3)**

**# locate which words in third quarter were positive (appeared in positive-word list)**

**matchedP3 <- match(words3, p, nomatch = 0)**

**# calculate the number of positive words in third quarter**

**ptotalNumber3 <- sum(wordCounts3[which(matchedP3 != 0)])**

**# calculate the ratio of positive words in third quarter**

**ratiop3 <- ptotalNumber3/totalWords3**

**# locate which words in third quarter were negative (appeared in negative-word list)**

**matchedN3 <- match(words3, n, nomatch = 0)**

**# calculate the number of negative words in third quarter**

**ntotalNumber3 <- sum(wordCounts3[which(matchedN3 != 0)])**

**# calculate the ratio of negative words in third quarter**

**ration3 <- ntotalNumber3/totalWords3**

**ration3**

**cloudFrame3<-data.frame(word=names(wordCounts3),freq=wordCounts3)**

**mergedTable3<-merge(cloudFrame3,AFINN,by.x="word",by.y="Word")**

**overallScore3<-sum(mergedTable3$freq\*mergedTable3$Score)**

**overallScore3**

**overallscore3scaled <-overallScore3/totalWords3**

**# forth 25%**

**# create word corpus for the forth quarter using cutpoints**

**words.corpus4 <- words.corpus[(3\*cutpoint+1):length(words.corpus)]**

**# create term document matrix for the forth quarter**

**tdm4 <- TermDocumentMatrix(words.corpus4)**

**tdm4**

**m4 <- as.matrix(tdm4)**

**m4**

**# create a list of word counts for the forth quarter and sort the list**

**wordCounts4 <- rowSums(m4)**

**wordCounts4<- sort(wordCounts4, decreasing=TRUE)**

**# calculate total words of the forth 25%**

**totalWords4 <- sum(wordCounts4)**

**# create a vector that contains all the words in "wordCounts4"**

**words4 <- names(wordCounts4)**

**# locate which words in forth quarter were positive (appeared in positive-word list)**

**matchedP4 <- match(words4, p, nomatch = 0)**

**# calculate the number of positive words in forth quarter**

**ptotalNumber4 <- sum(wordCounts4[which(matchedP4 != 0)])**

**# calculate the ratio of positive words in forth quarter**

**ratiop4 <- ptotalNumber4/totalWords4**

**# locate which words in forth quarter were negative (appeared in negative-word list)**

**matchedN4 <- match(words4, n, nomatch = 0)**

**# calculate the number of negative words in forth quarter**

**ntotalNumber4 <- sum(wordCounts4[which(matchedN4 != 0)])**

**# calculate the ratio of negative words in forth quarter**

**ration4 <- ntotalNumber4/totalWords4**

**ration4**

**cloudFrame4<-data.frame(word=names(wordCounts4),freq=wordCounts4)**

**mergedTable4<-merge(cloudFrame4,AFINN,by.x="word",by.y="Word")**

**overallScore4 <-sum(mergedTable4$freq\*mergedTable4$Score)**

**overallScore4**

**overallscore4scaled <- overallScore4/totalWords4**

**ratioP <- cbind(overallscore1scaled, overallscore2scaled, overallscore3scaled, overallscore4scaled)**

**barplot(ratioP, names.arg = c("1st 25%","2nd 25%","3rd 25%","4th 25%"), main = "Positive Ratio")**

**Executed:**

#ADS IST 687

> #Jacob Dineen

> #Homework 10

> #Due 9/24/2017

>

> #################All Calls to Clear Environment and Fetch Packages

> #CLEAR ENVIRONMENT AND INSTALL INITIAL PACKAGES

> rm(list = ls(all = TRUE))#Clear Enviroment

>

> #specify the packages of interest

> packages=c("tm","wordcloud")

>

> #use this function to check if each package is on the local machine

> #if a package is installed, it will be loaded

> #if any are not, the missing package(s) will be installed and loaded

> package.check <- lapply(packages, FUN = function(x) {

+ if (!require(x, character.only = TRUE)) {

+ install.packages(x, dependencies = TRUE)

+ library(x, character.only = TRUE)

+ }

+ })

>

>

> #Read in the positive and negative keyword files

>

> #Load the files

> pos <- file.choose()

> neg <- file.choose()

>

> #Read the Positive and Negative word Files + clean

> p <- scan(pos, character(0), sep= "\n")

Read 2040 items

> head(p,40)

[1] ";;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;"

[2] "; "

[3] "; Opinion Lexicon: Positive"

[4] ";"

[5] "; This file contains a list of POSITIVE opinion words (or sentiment words)."

[6] ";"

[7] "; This file and the papers can all be downloaded from "

[8] "; http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html"

[9] ";"

[10] "; If you use this list, please cite one of the following two papers:"

[11] ";"

[12] "; Minqing Hu and Bing Liu. \"Mining and Summarizing Customer Reviews.\" "

[13] "; Proceedings of the ACM SIGKDD International Conference on Knowledge "

[14] "; Discovery and Data Mining (KDD-2004), Aug 22-25, 2004, Seattle, "

[15] "; Washington, USA, "

[16] "; Bing Liu, Minqing Hu and Junsheng Cheng. \"Opinion Observer: Analyzing "

[17] "; and Comparing Opinions on the Web.\" Proceedings of the 14th "

[18] "; International World Wide Web conference (WWW-2005), May 10-14, "

[19] "; 2005, Chiba, Japan."

[20] ";"

[21] "; Notes: "

[22] "; 1. The appearance of an opinion word in a sentence does not necessarily "

[23] "; mean that the sentence expresses a positive or negative opinion. "

[24] "; See the paper below:"

[25] ";"

[26] "; Bing Liu. \"Sentiment Analysis and Subjectivity.\" An chapter in "

[27] "; Handbook of Natural Language Processing, Second Edition, "

[28] "; (editors: N. Indurkhya and F. J. Damerau), 2010."

[29] ";"

[30] "; 2. You will notice many misspelled words in the list. They are not "

[31] "; mistakes. They are included as these misspelled words appear "

[32] "; frequently in social media content. "

[33] ";"

[34] ";;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;"

[35] "a+"

[36] "abound"

[37] "abounds"

[38] "abundance"

[39] "abundant"

[40] "accessable"

> p <- p[-1:-34]

> tail(p)

[1] "yay" "youthful" "zeal" "zenith" "zest" "zippy"

> head(p)

[1] "a+" "abound" "abounds" "abundance" "abundant" "accessable"

>

>

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

Read 4817 items

> head(n,40)

[1] ";;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;"

[2] "; "

[3] "; Opinion Lexicon: Negative"

[4] ";"

[5] "; This file contains a list of NEGATIVE opinion words (or sentiment words)."

[6] ";"

[7] "; This file and the papers can all be downloaded from "

[8] "; http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html"

[9] ";"

[10] "; If you use this list, please cite one of the following two papers:"

[11] ";"

[12] "; Minqing Hu and Bing Liu. \"Mining and Summarizing Customer Reviews.\" "

[13] "; Proceedings of the ACM SIGKDD International Conference on Knowledge "

[14] "; Discovery and Data Mining (KDD-2004), Aug 22-25, 2004, Seattle, "

[15] "; Washington, USA, "

[16] "; Bing Liu, Minqing Hu and Junsheng Cheng. \"Opinion Observer: Analyzing "

[17] "; and Comparing Opinions on the Web.\" Proceedings of the 14th "

[18] "; International World Wide Web conference (WWW-2005), May 10-14, "

[19] "; 2005, Chiba, Japan."

[20] ";"

[21] "; Notes: "

[22] "; 1. The appearance of an opinion word in a sentence does not necessarily "

[23] "; mean that the sentence expresses a positive or negative opinion. "

[24] "; See the paper below:"

[25] ";"

[26] "; Bing Liu. \"Sentiment Analysis and Subjectivity.\" An chapter in "

[27] "; Handbook of Natural Language Processing, Second Edition, "

[28] "; (editors: N. Indurkhya and F. J. Damerau), 2010."

[29] ";"

[30] "; 2. You will notice many misspelled words in the list. They are not "

[31] "; mistakes. They are included as these misspelled words appear "

[32] "; frequently in social media content. "

[33] ";"

[34] ";;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;"

[35] "2-faced"

[36] "2-faces"

[37] "abnormal"

[38] "abolish"

[39] "abominable"

[40] "abominably"

> p <- p[-1:-34]

> tail(p)

[1] "yay" "youthful" "zeal" "zenith" "zest" "zippy"

> head(p)

[1] "admiring" "admiringly" "adorable" "adore" "adored" "adorer"

>

> #Process in the MLK Speech

> mlkFile <- file.choose()

> mlk <- readLines(mlkFile)

>

> #Transformation

> words.vec <- VectorSource((mlk))

> words.corpus <- Corpus(words.vec)

>

> #Munge Data

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

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

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

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

>

> #Create TermDocMatrix

> tdm <- TermDocumentMatrix(words.corpus)

>

> #Store as matrix

> m<- as.matrix(tdm)

>

> #Create list of counts for each word

> wordCounts <- rowSums(m)

>

> #Sum total word count

> totalwords <- sum(wordCounts)

>

> #Sort wordCount by freq of occurrence

> wordCounts <- sort(wordCounts, decreasing=TRUE)

> head(wordCounts)

will freedom negro one let ring

26 20 13 13 13 12

>

> wordfreq <- data.frame(wordCounts)

> colnames(wordfreq)

[1] "wordCounts"

>

> #Create Test Wordcloud

> cloudFrame<-data.frame(word=names(wordCounts),freq=wordCounts)

> cloudFrame[1:10,]

word freq

will will 26

freedom freedom 20

negro negro 13

one one 13

let let 13

ring ring 12

day day 11

dream dream 11

nation nation 10

come come 10

> wordcloud(cloudFrame$word,cloudFrame$freq)

> wordcloud(names(wordCounts),wordCounts,min.freq=3,max.words=50,rot.per=.35,colors=brewer.pal(8,"Dark2"))

>

>

>

> #Determine How many Positive words were in the speech

> totalWords <- sum(wordCounts)

> words <- names(wordCounts)

> matched <- match(words,p, nomatch = 0)

> head(matched,150)

[1] 0 732 0 0 0 0 0 0 0 0 0 0 0 0 0 1532 0 0 0 0 0 0 823 0 730 0 0 0

[29] 0 0 0 639 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

[57] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1049 0 0 0 0 0 0 0 0 0 0 0 0 0

[85] 0 345 397 405 0 0 0 0 0 0 0 1054 1145 0 0 0 0 0 1953 0 0 0 0 0 0 0 0 0

[113] 0 0 0 0 0 0 0 0 0 0 824 850 0 0 0 0 0 0 0 0 1012 0 0 1155 0 0 0 0

[141] 0 0 0 0 0 0 0 0 0 0

> pWords <- words[which(matched !=0)]

> head(pWords)

[1] "freedom" "satisfied" "great" "free" "faith" "liberty"

> Pcounts <- wordCounts[which(matched !=0)]

> head(Pcounts)

freedom satisfied great free faith liberty

20 8 5 5 5 2

> nPos <- sum(Pcounts)

> nPos #95 positive words

[1] 95

> length(Pcounts) #50 unique positive words

[1] 50

> nPos/totalWords #11.2960% Positive

[1] 0.1129608

>

>

> #Determine How many Negative words were in the speech

> matched <- match(words,n, nomatch = 0)

> ncounts <- wordCounts[which(matched !=0)]

> nNeg <- sum(ncounts)

> nNeg #63 negative words

[1] 63

> length(ncounts) #54 unique negative words

[1] 54

> nWords <- names(ncounts)

> nNeg/totalWords #7.4910% negative

[1] 0.07491082

>

> #Calculating Ratio of positive to negative words

> sentimentratio <- nPos/nNeg

> sentimentratio #There were 1.5 positive words to every negative word within the MLK speech.

[1] 1.507937

>

> # Sentiment Affinity Score

>

> #Read in affinity project

> AFINN<-read.delim(file.choose(),sep="\t",header = FALSE)

> str(AFINN)

'data.frame': 2477 obs. of 2 variables:

$ V1: Factor w/ 2477 levels "abandon","abandoned",..: 1 2 3 4 5 6 7 8 9 10 ...

$ V2: int -2 -2 -2 -2 -2 -2 -3 -3 -3 -3 ...

> head(AFINN)

V1 V2

1 abandon -2

2 abandoned -2

3 abandons -2

4 abducted -2

5 abduction -2

6 abductions -2

> colnames(AFINN)<-c("Word", "Score")

> AFINN[1:10,]

Word Score

1 abandon -2

2 abandoned -2

3 abandons -2

4 abducted -2

5 abduction -2

6 abductions -2

7 abhor -3

8 abhorred -3

9 abhorrent -3

10 abhors -3

>

> #Merge two dfs - cloudframe and Afinn

> mergedTable<-merge(cloudFrame,AFINN,by.x="word",by.y="Word")

> mergedTable[1:10,]

word freq Score

1 allow 2 1

2 alone 1 -2

3 bad 1 -3

4 bankrupt 1 -3

5 beautiful 1 3

6 bright 1 1

7 creative 2 2

8 demand 1 -1

9 demonstration 1 -1

10 despair 2 -3

> str(mergedTable)

'data.frame': 61 obs. of 3 variables:

$ word : Factor w/ 463 levels "able","ago","ahead",..: 6 8 19 21 25 36 74 89 91 93 ...

$ freq : num 2 1 1 1 1 1 2 1 1 2 ...

$ Score: int 1 -2 -3 -3 3 1 2 -1 -1 -3 ...

>

> #Compute total Score and scaled total score

> overallScore<-sum(mergedTable$freq\*mergedTable$Score)

> overallScore #Overall score = 113

[1] 113

> overallScore/totalWords #Scaled Score = 13.43%

[1] 0.1343639

>

> # Splitting speech into quarters

> cutpoint <- round(length(words.corpus)/4) #Create cutpoint for MLK speech

>

>

> # first 25%

> # create word corpus for the first quarter using cutpoints

> words.corpus1 <- words.corpus[1:cutpoint]

> # create term document matrix for the first quarter

> tdm1 <- TermDocumentMatrix(words.corpus1)

> # convert tdm1 into a matrix called "m1"

> m1 <- as.matrix(tdm1)

> # create a list of word counts for the first quarter and sort the list

> wordCounts1 <- rowSums(m1)

> wordCounts1 <- sort(wordCounts1, decreasing=TRUE)

> # calculate total words of the first 25%

> totalWords1 <- sum(wordCounts1)

> # create a vector that contains all the words in "wordCounts1"

> words1 <- names(wordCounts1)

> # locate which words in first quarter were positive (appeared in positive-word list)

> matchedP1 <- match(words1, p, nomatch = 0)

> # calculate the number of positive words in first quarter

> ptotalNumber1 <- sum(wordCounts1[which(matchedP1 != 0)])

> # calculate the ratio of positive words in first quarter

> ratiop1 <- ptotalNumber1/totalWords1

> # locate which words in first quarter were negative (appeared in negative-word list)

> matchedN1 <- match(words1, n, nomatch = 0)

> # calculate the number of negative words in first quarter

> ntotalNumber1 <- sum(wordCounts1[which(matchedN1 != 0)])

> # calculate the ratio of negative words in first quarter

> ration1 <- ntotalNumber1/totalWords1

>

> cloudFrame1<-data.frame(word=names(wordCounts1),freq=wordCounts1)

> mergedTable1<-merge(cloudFrame1,AFINN,by.x="word",by.y="Word")

> overallScore1<-sum(mergedTable1$freq\*mergedTable1$Score)

> overallscore1scaled <-overallScore1/totalWords1

>

>

> # Total Score Q1 = .09589

>

>

> # second 25%

> # create word corpus for the second quarter using cutpoints

> words.corpus2 <- words.corpus[(cutpoint+1):(2\*cutpoint)]

> # create term document matrix for the second quarter

> tdm2 <- TermDocumentMatrix(words.corpus2)

> m2 <- as.matrix(tdm2)

> # create a list of word counts for the second quarter and sort the list

> wordCounts2 <- rowSums(m2)

> wordCounts2<- sort(wordCounts2, decreasing=TRUE)

> # calculate total words of the second 25%

> totalWords2 <- sum(wordCounts2)

> # create a vector that contains all the words in "wordCounts2"

> words2 <- names(wordCounts2)

> # locate which words in second quarter were positive (appeared in positive-word list)

> matchedP2 <- match(words2, p, nomatch = 0)

> # calculate the number of positive words in second quarter

> ptotalNumber2 <- sum(wordCounts2[which(matchedP2 != 0)])

> # calculate the ratio of positive words in second quarter

> ratiop2 <- ptotalNumber2/totalWords2

> # locate which words in second quarter were negative (appeared in negative-word list)

> matchedN2 <- match(words2, n, nomatch = 0)

> # calculate the number of negative words in second quarter

> ntotalNumber2 <- sum(wordCounts2[which(matchedN2 != 0)])

> # calculate the ratio of negative words in second quarter

> ration2 <- ntotalNumber2/totalWords2

>

> cloudFrame2<-data.frame(word=names(wordCounts2),freq=wordCounts2)

> mergedTable2<-merge(cloudFrame2,AFINN,by.x="word",by.y="Word")

> overallScore2<-sum(mergedTable2$freq\*mergedTable2$Score)

> overallscore2scaled <-overallScore2/totalWords2

>

> # third 25%

> # create word corpus for the third quarter using cutpoints

> words.corpus3 <- words.corpus[(2\*cutpoint+1):cutpoint]

> # create term document matrix for the third quarter

> tdm3 <- TermDocumentMatrix(words.corpus3)

> m3 <- as.matrix(tdm3)

> # create a list of word counts for the third quarter and sort the list

> wordCounts3 <- rowSums(m3)

> wordCounts3<- sort(wordCounts3, decreasing=TRUE)

> # calculate total words of the third 25%

> totalWords3 <- sum(wordCounts3)

> # create a vector that contains all the words in "wordCounts3"

> words3 <- names(wordCounts3)

> # locate which words in third quarter were positive (appeared in positive-word list)

> matchedP3 <- match(words3, p, nomatch = 0)

> # calculate the number of positive words in third quarter

> ptotalNumber3 <- sum(wordCounts3[which(matchedP3 != 0)])

> # calculate the ratio of positive words in third quarter

> ratiop3 <- ptotalNumber3/totalWords3

> # locate which words in third quarter were negative (appeared in negative-word list)

> matchedN3 <- match(words3, n, nomatch = 0)

> # calculate the number of negative words in third quarter

> ntotalNumber3 <- sum(wordCounts3[which(matchedN3 != 0)])

> # calculate the ratio of negative words in third quarter

> ration3 <- ntotalNumber3/totalWords3

>

> cloudFrame3<-data.frame(word=names(wordCounts3),freq=wordCounts3)

> mergedTable3<-merge(cloudFrame3,AFINN,by.x="word",by.y="Word")

> overallScore3<-sum(mergedTable3$freq\*mergedTable3$Score)

> overallscore3scaled <-overallScore3/totalWords3

>

> # forth 25%

> # create word corpus for the forth quarter using cutpoints

> words.corpus4 <- words.corpus[(3\*cutpoint+1):length(words.corpus)]

> # create term document matrix for the forth quarter

> tdm4 <- TermDocumentMatrix(words.corpus4)

> m4 <- as.matrix(tdm4)

>

> # create a list of word counts for the forth quarter and sort the list

> wordCounts4 <- rowSums(m4)

> wordCounts4<- sort(wordCounts4, decreasing=TRUE)

> # calculate total words of the forth 25%

> totalWords4 <- sum(wordCounts4)

> # create a vector that contains all the words in "wordCounts4"

> words4 <- names(wordCounts4)

> # locate which words in forth quarter were positive (appeared in positive-word list)

> matchedP4 <- match(words4, p, nomatch = 0)

> # calculate the number of positive words in forth quarter

> ptotalNumber4 <- sum(wordCounts4[which(matchedP4 != 0)])

> # calculate the ratio of positive words in forth quarter

> ratiop4 <- ptotalNumber4/totalWords4

> # locate which words in forth quarter were negative (appeared in negative-word list)

> matchedN4 <- match(words4, n, nomatch = 0)

> # calculate the number of negative words in forth quarter

> ntotalNumber4 <- sum(wordCounts4[which(matchedN4 != 0)])

> # calculate the ratio of negative words in forth quarter

> ration4 <- ntotalNumber4/totalWords4

>

>

> cloudFrame4<-data.frame(word=names(wordCounts4),freq=wordCounts4)

> mergedTable4<-merge(cloudFrame4,AFINN,by.x="word",by.y="Word")

> overallScore4 <-sum(mergedTable4$freq\*mergedTable4$Score)

> overallscore4scaled <- overallScore4/totalWords4

>

> ratioP <- cbind(overallscore1scaled, overallscore2scaled, overallscore3scaled, overallscore4scaled)

> barplot(ratioP, names.arg = c("1st 25%","2nd 25%","3rd 25%","4th 25%"), main = "Positive Ratio")