Assignment 3

M. Nethercott

3/4/2017

## Import Libraries

library(tm)

## Loading required package: NLP

library(SnowballC)  
library(wordcloud)

## Loading required package: RColorBrewer

library(ggplot2)

##   
## Attaching package: 'ggplot2'

## The following object is masked from 'package:NLP':  
##   
## annotate

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(tidyr)  
library(topicmodels)

## Import all document files and the list of weeks file

Calling setlocale() seems to cause issues with importing the data files (quotation marks get replaced with , for example). Removing this function call fixes the issue (**Ask about this**): Sys.setlocale("LC\_ALL", "C")

##### Create a list of all the files

file.list <- list.files(path="./data", pattern=".csv",full.names = TRUE)

##### Loop over file list importing them and binding them together

D1 <- do.call(rbind, lapply(file.list, read.csv, header = TRUE, stringsAsFactors = FALSE, na.strings=c("")))

##### Import the week-list data

D2 <- read.csv("./data/Week-List/week-list.csv", header = TRUE)

## Strip HTML Tags

##### Strip out HTML tags, comments, and the   whitespace command

D1$Notes <- gsub("</?\\w+((\\s+\\w+(\\s\*=\\s\*(?:\".\*?\"|'.\*?'|[\\^'\">\\s]+))?)+\\s\*|\\s\*)/?>|<!--[\\s\\S]\*?-->", "", D1$Notes)  
D1$Notes <- gsub("&nbsp;", " ", D1$Notes)

##### Filter out rows with missing values in either the Title or Notes column (if no notes we don't include it in the analysis anyway, and no title means we can't match it with the week-list data frame)

D1 <- filter(D1, !is.na(Notes) & !is.na(Title))

## Merge with week list

D3 <- merge(D1, D2, by.x = "Title", by.y = "Title", all.x = TRUE)

##### Filter rows where week is NA, since we are only doing the analysis across each week

D3 <- filter(D3, !is.na(week))

## Process text

##### Convert the data frame to the corpus format that the tm package uses

corpus <- Corpus(VectorSource(D3$Notes))

##### Remove spaces

corpus <- tm\_map(corpus, stripWhitespace)

##### Convert to lower case

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

##### Remove pre-defined stop words ('the', 'a', etc)

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

##### Convert words to stems ("education" = "edu") for analysis, for more info see <http://tartarus.org/~martin/PorterStemmer/>

corpus <- tm\_map(corpus, stemDocument, lazy=TRUE)

##### Remove numbers

corpus <- tm\_map(corpus, removeNumbers, lazy=TRUE)

##### remove punctuation

corpus <- tm\_map(corpus, removePunctuation, lazy=TRUE)

## Create a term matrix

##### Convert corpus to a term document matrix - so each word can be analyzed individuallly

tdm.corpus <- TermDocumentMatrix(corpus)

## Sentiment analysis

##### Upload positive and negative word lexicons

positive <- readLines("./data/positive-words.txt")  
negative <- readLines("./data/negative-words.txt")

##### Search for matches between each word and the two lexicons

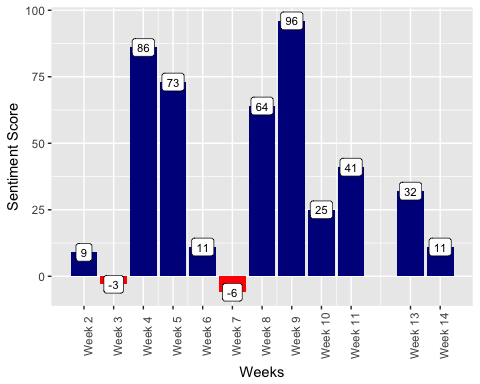
D3$positive <- tm\_term\_score(tdm.corpus, positive)  
D3$negative <- tm\_term\_score(tdm.corpus, negative)

##### Generate an overall pos-neg score for each line

D3$score <- D3$positive - D3$negative

##### SENTIMENT SCORE PLOT

#Group the data by week, then summarize the grouped data and create the computed column for the sentiment score sum  
grouped <- group\_by(D3, week)  
summ <- summarise(grouped, sum\_score = sum(score))  
  
# Add 'pos' column to indicate positive or negative sentiment score (this is just for aesthetic purposes, to set different colors for positive and negative sentiment scores)  
summ$pos <- ifelse(summ$sum\_score >= 0, "positive", "negative")  
  
brk <- distinct(D3, week) #Get the breaks for the sentiment plot by weeks  
brk <- brk[!is.na(brk)] #Remove the NA value as we don't need this in the plot  
ggplot(summ, aes(x = week, y = sum\_score, fill = pos)) +   
 xlab("Weeks") +   
 ylab("Sentiment Score") +   
 scale\_x\_continuous(  
 label=function(x){return(paste("Week", x))},   
 breaks = as.numeric(unlist(brk))) +   
 geom\_bar(stat="identity",position="identity") +   
 scale\_fill\_manual(values = c("positive" = "darkblue", "negative" = "red")) +  
 theme(axis.text.x = element\_text(angle = 90), legend.position="none") +  
 geom\_label(aes(label = sum\_score), size = 3, fill = "white")



## LDA topic modeling

##### Term Frequency Inverse Document Frequency

dtm.tfi <- DocumentTermMatrix(corpus, control = list(weighting = weightTf))

##### Remove very uncommon terms (term freq inverse document freq < 0.1)

dtm.tfi <- dtm.tfi[,dtm.tfi$v >= 0.1]

##### Remove non-zero entries

rowTotals <- apply(dtm.tfi , 1, sum) #Find the sum of words in each Document  
dtm.tfi <- dtm.tfi[rowTotals> 0, ] #Divide by sum across rows  
lda.model = LDA(dtm.tfi, k = 3)

##### Which terms are most common in each topic

terms <- terms(lda.model)

##### Which documents belong to which topic

topics <- topics(lda.model)

##### Assign each assigned topic to every observation in D3 dataset

grouped$topic <- topics  
  
#Recode topic values to match the actual terms created as a result of the LDA modeling  
grouped$topic[grouped$topic==1] <- terms[1]  
grouped$topic[grouped$topic==2] <- terms[2]  
grouped$topic[grouped$topic==3] <- terms[3]

##### Combined visualization

#Determine important topic for each week, first creating a crosstab of weeks and topics, with frequency for each topic grouped by week  
topicSumm <- as.data.frame(table(grouped$topic, grouped$week))  
topicSumm <- rename(topicSumm, topic = Var1, week = Var2, freq = Freq)  
topicGrp <- group\_by(topicSumm, week)  
  
#Sort by frequency (descending) of topic within each week, then slice the top row from each  
topicGrp <- arrange(topicGrp, week, desc(freq))  
topicPerWeek <- slice(topicGrp, 1)  
  
summ <- merge(summ, topicPerWeek, by.x = "week", by.y = "week", all = TRUE)  
ggplot(summ, aes(x = week, y = sum\_score, fill = pos)) +   
 xlab("Weeks") +   
 ylab("Sentiment Score") +   
 scale\_x\_continuous(  
 label=function(x){return(paste("Week", x))},   
 breaks = as.numeric(unlist(brk))) +   
 scale\_y\_continuous(limits = c(-20,110)) +  
 geom\_bar(stat="identity",position="identity") +   
 scale\_fill\_manual(values = c("positive" = "darkblue", "negative" = "red")) +  
 theme(axis.text.x = element\_text(angle = 90), legend.position="none") +  
 geom\_label(aes(label=paste(topic, " (", sum\_score, ")")), parse = TRUE, size = 2.5, fill = "white", vjust=1)

