Twitter-Sentiment-Analysis-A.R

shannenserrano

2020-11-16

#install.packages("tidyr")  
#install.packages("dplyr")  
#install.packages("ggplot2")  
#install.packages("tidytext")  
#install.packages("textdata")  
#install.packages("devtools")  
#install.packages("widyr")  
#install.packages("tidyverse")  
#install.packages("wordcloud")  
#install.packages("RColorBrewer")  
library(tidyr)

## Warning: package 'tidyr' was built under R version 4.0.2

library(dplyr)

## Warning: package 'dplyr' was built under R version 4.0.2

##   
## 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(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.0.2

library(tidytext)

## Warning: package 'tidytext' was built under R version 4.0.2

library(textdata)

## Warning: package 'textdata' was built under R version 4.0.2

get\_sentiments("afinn")

## # A tibble: 2,477 x 2  
## word value  
## <chr> <dbl>  
## 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  
## # … with 2,467 more rows

get\_sentiments("bing")

## # A tibble: 6,786 x 2  
## word sentiment  
## <chr> <chr>   
## 1 2-faces negative   
## 2 abnormal negative   
## 3 abolish negative   
## 4 abominable negative   
## 5 abominably negative   
## 6 abominate negative   
## 7 abomination negative   
## 8 abort negative   
## 9 aborted negative   
## 10 aborts negative   
## # … with 6,776 more rows

get\_sentiments("nrc")

## # A tibble: 13,901 x 2  
## word sentiment  
## <chr> <chr>   
## 1 abacus trust   
## 2 abandon fear   
## 3 abandon negative   
## 4 abandon sadness   
## 5 abandoned anger   
## 6 abandoned fear   
## 7 abandoned negative   
## 8 abandoned sadness   
## 9 abandonment anger   
## 10 abandonment fear   
## # … with 13,891 more rows

library(widyr)

## Warning: package 'widyr' was built under R version 4.0.2

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.0.2

## ── Attaching packages ─────────────────────────────────────────────────────────── tidyverse 1.3.0 ──

## ✓ tibble 3.0.1 ✓ stringr 1.4.0  
## ✓ readr 1.4.0 ✓ forcats 0.5.0  
## ✓ purrr 0.3.4

## Warning: package 'readr' was built under R version 4.0.2

## Warning: package 'forcats' was built under R version 4.0.2

## ── Conflicts ────────────────────────────────────────────────────────────── tidyverse\_conflicts() ──  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(wordcloud)

## Warning: package 'wordcloud' was built under R version 4.0.2

## Loading required package: RColorBrewer

## Warning: package 'RColorBrewer' was built under R version 4.0.2

library(RColorBrewer)  
library(reshape2)

##   
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':  
##   
## smiths

#read tweets.csv data set   
rawtweets <- read.csv("/Users/shannenserrano/Google Drive/tweets.csv", header = T, sep = ",")  
  
#review data set  
head(rawtweets)

## X  
## 1 1  
## 2 2  
## 3 3  
## 4 4  
## 5 5  
## 6 6  
## text  
## 1 RT @mrvelstan: literally nobody:\nme:\n\n#AvengersEndgame https://t.co/LR9kFwfD5c  
## 2 RT @agntecarter: i\x92m emotional, sorry!!\n\n2014 x 2019\n#blackwidow\n#captainamerica https://t.co/xcwkCMw18w  
## 3 saving these bingo cards for tomorrow \n\xa9\n #AvengersEndgame https://t.co/d6For0jwRb  
## 4 RT @HelloBoon: Man these #AvengersEndgame ads are everywhere https://t.co/Q0lNf5eJsX  
## 5 RT @Marvel: We salute you, @ChrisEvans! #CaptainAmerica #AvengersEndgame https://t.co/VlPEpnXYgm  
## 6 RT @MCU\_Direct: The first NON-SPOILER #AvengersEndgame critic reactions are here and nearly all are exceptionally positive, with many prais\x85  
## favorited favoriteCount replyToSN created truncated replyToSID  
## 1 FALSE 0 <NA> 2019-04-23 10:43:30 FALSE NA  
## 2 FALSE 0 <NA> 2019-04-23 10:43:30 FALSE NA  
## 3 FALSE 0 <NA> 2019-04-23 10:43:30 FALSE NA  
## 4 FALSE 0 <NA> 2019-04-23 10:43:29 FALSE NA  
## 5 FALSE 0 <NA> 2019-04-23 10:43:29 FALSE NA  
## 6 FALSE 0 <NA> 2019-04-23 10:43:29 FALSE NA  
## id replyToUID  
## 1 1.120639e+18 NA  
## 2 1.120639e+18 NA  
## 3 1.120639e+18 NA  
## 4 1.120639e+18 NA  
## 5 1.120639e+18 NA  
## 6 1.120639e+18 NA  
## statusSource  
## 1 <a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android</a>  
## 2 <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>  
## 3 <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>  
## 4 <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>  
## 5 <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>  
## 6 <a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android</a>  
## screenName retweetCount isRetweet retweeted longitude latitude  
## 1 DavidAc96 637 TRUE FALSE NA NA  
## 2 NRmalaa 302 TRUE FALSE NA NA  
## 3 jijitsuu 0 FALSE FALSE NA NA  
## 4 SahapunB 23781 TRUE FALSE NA NA  
## 5 stella22\_97 13067 TRUE FALSE NA NA  
## 6 Legend662 3122 TRUE FALSE NA NA

str(rawtweets)

## 'data.frame': 15000 obs. of 17 variables:  
## $ X : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ text : chr "RT @mrvelstan: literally nobody:\nme:\n\n#AvengersEndgame https://t.co/LR9kFwfD5c" "RT @agntecarter: i\x92m emotional, sorry!!\n\n2014 x 2019\n#blackwidow\n#captainamerica https://t.co/xcwkCMw18w" "saving these bingo cards for tomorrow \n\xa9\n #AvengersEndgame https://t.co/d6For0jwRb" "RT @HelloBoon: Man these #AvengersEndgame ads are everywhere https://t.co/Q0lNf5eJsX" ...  
## $ favorited : logi FALSE FALSE FALSE FALSE FALSE FALSE ...  
## $ favoriteCount: int 0 0 0 0 0 0 0 0 0 0 ...  
## $ replyToSN : chr NA NA NA NA ...  
## $ created : chr "2019-04-23 10:43:30" "2019-04-23 10:43:30" "2019-04-23 10:43:30" "2019-04-23 10:43:29" ...  
## $ truncated : logi FALSE FALSE FALSE FALSE FALSE FALSE ...  
## $ replyToSID : num NA NA NA NA NA NA NA NA NA NA ...  
## $ id : num 1.12e+18 1.12e+18 1.12e+18 1.12e+18 1.12e+18 ...  
## $ replyToUID : num NA NA NA NA NA NA NA NA NA NA ...  
## $ statusSource : chr "<a href=\"http://twitter.com/download/android\" rel=\"nofollow\">Twitter for Android</a>" "<a href=\"http://twitter.com/download/iphone\" rel=\"nofollow\">Twitter for iPhone</a>" "<a href=\"http://twitter.com/download/iphone\" rel=\"nofollow\">Twitter for iPhone</a>" "<a href=\"http://twitter.com/download/iphone\" rel=\"nofollow\">Twitter for iPhone</a>" ...  
## $ screenName : chr "DavidAc96" "NRmalaa" "jijitsuu" "SahapunB" ...  
## $ retweetCount : int 637 302 0 23781 13067 3122 269 5687 349 23781 ...  
## $ isRetweet : logi TRUE TRUE FALSE TRUE TRUE TRUE ...  
## $ retweeted : logi FALSE FALSE FALSE FALSE FALSE FALSE ...  
## $ longitude : num NA NA NA NA NA NA NA NA NA NA ...  
## $ latitude : num NA NA NA NA NA NA NA NA NA NA ...

summary(rawtweets)

## X text favorited favoriteCount   
## Min. : 1 Length:15000 Mode :logical Min. : 0.000   
## 1st Qu.: 3751 Class :character FALSE:15000 1st Qu.: 0.000   
## Median : 7500 Mode :character Median : 0.000   
## Mean : 7500 Mean : 0.194   
## 3rd Qu.:11250 3rd Qu.: 0.000   
## Max. :15000 Max. :184.000   
##   
## replyToSN created truncated replyToSID   
## Length:15000 Length:15000 Mode :logical Min. :1.107e+18   
## Class :character Class :character FALSE:14433 1st Qu.:1.121e+18   
## Mode :character Mode :character TRUE :567 Median :1.121e+18   
## Mean :1.121e+18   
## 3rd Qu.:1.121e+18   
## Max. :1.121e+18   
## NA's :14631   
## id replyToUID statusSource screenName   
## Min. :1.121e+18 Min. :1.065e+07 Length:15000 Length:15000   
## 1st Qu.:1.121e+18 1st Qu.:1.167e+08 Class :character Class :character   
## Median :1.121e+18 Median :2.112e+08 Mode :character Mode :character   
## Mean :1.121e+18 Mean :8.608e+16   
## 3rd Qu.:1.121e+18 3rd Qu.:2.313e+09   
## Max. :1.121e+18 Max. :1.121e+18   
## NA's :14603   
## retweetCount isRetweet retweeted longitude   
## Min. : 0 Mode :logical Mode :logical Min. :-120.951   
## 1st Qu.: 186 FALSE:1499 FALSE:15000 1st Qu.: -9.238   
## Median : 1755 TRUE :13501 Median : 71.076   
## Mean : 5912 Mean : 35.558   
## 3rd Qu.: 9290 3rd Qu.: 115.872   
## Max. :114850 Max. : 121.029   
## NA's :14996   
## latitude   
## Min. :-26.000   
## 1st Qu.: 4.418   
## Median : 18.420   
## Mean : 12.143   
## 3rd Qu.: 26.145   
## Max. : 37.733   
## NA's :14996

#change text data type from character to string

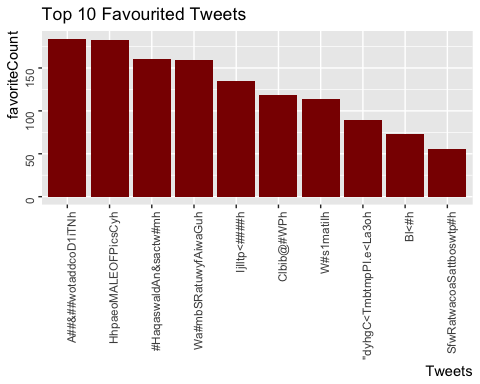
#EXPLORATORY ANALYSIS w/ text | favoriteCount | retweetCount

#Tweets & FavoriteCount  
   
tweetsfav <- rawtweets[c("text", "favoriteCount")]   
  
favorite <- tweetsfav %>%  
 top\_n(10) %>%  
 arrange(desc(favoriteCount)) %>%  
 select(text, favoriteCount)

## Selecting by favoriteCount

#Top 10 Favourited Tweets  
favorite\_bp <- ggplot(data=favorite, aes(x=reorder(text, -favoriteCount), y=favoriteCount))+  
 geom\_bar(stat="identity", fill="darkred") +   
 labs(title= "Top 10 Favourited Tweets") +  
 scale\_x\_discrete(label=abbreviate) +   
 xlab("Tweets") +  
 theme(text = element\_text(angle=90, vjust =0.5, hjust=1)) +  
 theme(plot.title = element\_text(angle=0)) +  
 theme(axis.title.x = element\_text(angle=0, vjust=0.5, hjust=1))  
  
favorite\_bp

## Warning in f(...): abbreviate used with non-ASCII chars

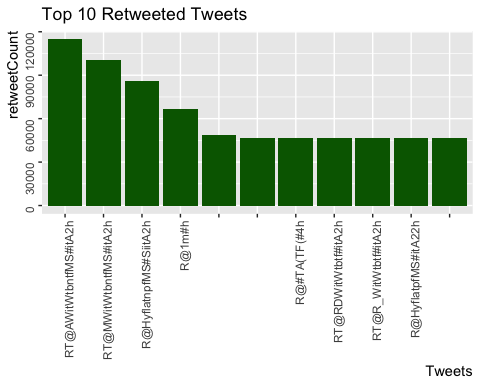


#Tweets & RetweetCount   
  
#for retweetCount, I realized that the dataset contained multiple duplicate tweets as they were   
#retweeted. Moreover, all these retweet shared the same amount of retweetCount so I decided to delete  
#all the duplicate tweets.  
#read tweet dataset with no duplicates   
   
tweetsrt\_raw <- read.csv("/Users/shannenserrano/Google Drive/tweets\_noduplicates.csv", header = T, sep = ",")  
tweetsrt <-tweetsrt\_raw[c("text", "retweetCount")]  
  
#Top 10 Retweeted Tweets  
retweet <- tweetsrt %>%  
 arrange(desc(retweetCount)) %>%  
 top\_n(10) %>%  
 select(text, retweetCount)

## Selecting by retweetCount

retweet\_bp <- ggplot(data=retweet, aes(x=reorder(text, -retweetCount), y=retweetCount)) +  
 geom\_bar(stat="identity", fill="darkgreen") +   
 labs(title= "Top 10 Retweeted Tweets") +  
 scale\_x\_discrete(label=abbreviate) +   
 xlab("Tweets") +  
 theme(text = element\_text(angle=90, vjust =0.5, hjust=1)) +  
 theme(plot.title = element\_text(angle=0))+  
 theme(axis.title.x = element\_text(angle=0, vjust=0.5, hjust=1))  
  
retweet\_bp

## Warning in f(...): abbreviate used with non-ASCII chars  
  
## Warning in f(...): abbreviate used with non-ASCII chars  
  
## Warning in f(...): abbreviate used with non-ASCII chars  
  
## Warning in f(...): abbreviate used with non-ASCII chars  
  
## Warning in f(...): abbreviate used with non-ASCII chars  
  
## Warning in f(...): abbreviate used with non-ASCII chars  
  
## Warning in f(...): abbreviate used with non-ASCII chars  
  
## Warning in f(...): abbreviate used with non-ASCII chars  
  
## Warning in f(...): abbreviate used with non-ASCII chars  
  
## Warning in f(...): abbreviate used with non-ASCII chars  
  
## Warning in f(...): abbreviate used with non-ASCII chars  
  
## Warning in f(...): abbreviate used with non-ASCII chars  
  
## Warning in f(...): abbreviate used with non-ASCII chars  
  
## Warning in f(...): abbreviate used with non-ASCII chars  
  
## Warning in f(...): abbreviate used with non-ASCII chars  
  
## Warning in f(...): abbreviate used with non-ASCII chars  
  
## Warning in f(...): abbreviate used with non-ASCII chars  
  
## Warning in f(...): abbreviate used with non-ASCII chars



#PREPROCESSING w/ text

#tweetsdf <- data.frame(rawtweets$text)  
#names(tweetsdf)[names(tweetsdf) == "rawtweets.text"] <- "tweets"  
  
tweetsdf <- read.csv("/Users/shannenserrano/Google Drive/tweetstext.csv", header = T, sep = ",")  
names(tweetsdf)[names(tweetsdf) == "text"] <- "tweets"  
  
#remove URLs  
tweetsdf$stripped\_text <- gsub("http.\*","", tweetsdf$tweets)  
tweetsdf$stripped\_text <- gsub("https.\*","", tweetsdf$stripped\_text)  
  
#remove punctuation, convert to lowercase, add id for each word   
clntweets0 <- tweetsdf %>%  
 dplyr::select(stripped\_text) %>%  
 unnest\_tokens(word, stripped\_text)  
  
#count words  
clntweets0 %>% count(word, sort = T) %>% head #top words: RT, the, u, a, to...

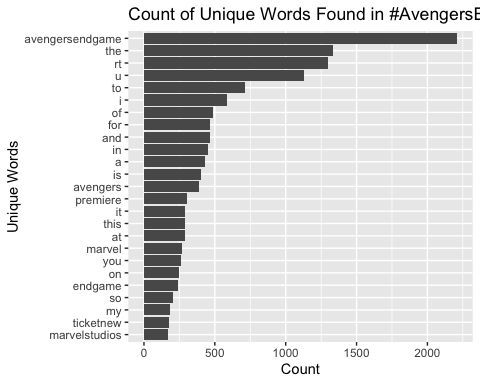
## word n  
## 1 avengersendgame 2209  
## 2 the 1335  
## 3 rt 1295  
## 4 u 1131  
## 5 to 710  
## 6 i 585

nrow(clntweets0) #212636 #39964

## [1] 39964

#graph clntweets0  
clntweets0 %>%  
 count(word, sort = TRUE) %>%  
 top\_n(25) %>%  
 mutate(word = reorder(word, n)) %>%  
 ggplot(aes(x=word, y=n)) +  
 geom\_col()+  
 xlab(NULL)+  
 coord\_flip()+  
 labs(x = "Unique Words",  
 y = "Count",  
 title = "Count of Unique Words Found in #AvengersEndgame - Top 25")

## Selecting by n



#remove stop words: the, y, a, to...  
data("stop\_words") #load list of stop words  
clntweets1 <- clntweets0 %>% anti\_join(stop\_words) #apply stop words to tweets

## Joining, by = "word"

nrow(clntweets1) #124287 #22929

## [1] 22929

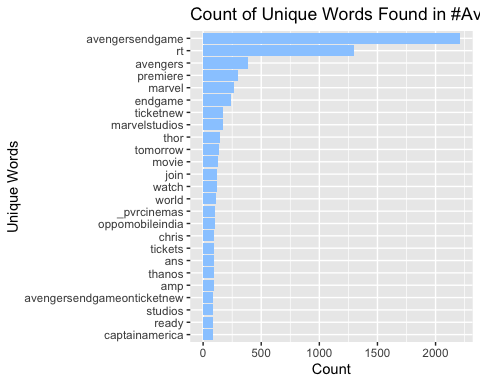
#count words  
clntweets1 %>% count(word, sort = T) %>% head

## word n  
## 1 avengersendgame 2209  
## 2 rt 1295  
## 3 avengers 387  
## 4 premiere 303  
## 5 marvel 267  
## 6 endgame 242

#UNIVARIATE ANALYSIS

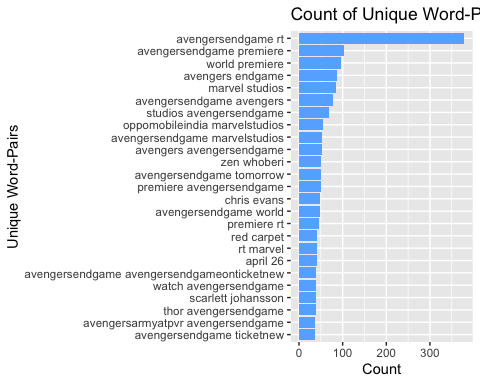
#Unique word count - Unigram   
clntweets1 %>%  
 count(word, sort = TRUE) %>%  
 top\_n(25) %>%  
 mutate(word = reorder(word, n)) %>%  
 ggplot(aes(x=word, y=n)) +  
 geom\_col(fill = "#99CCFF")+  
 xlab(NULL)+  
 coord\_flip()+  
 labs(x = "Unique Words",  
 y = "Count",  
 title = "Count of Unique Words Found in #AvengersEndgame - Top 25")

## Selecting by n



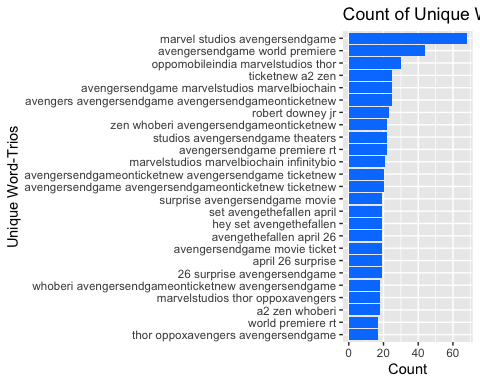
#Unique word count - Bigram   
clntweets2 <- clntweets1 %>%  
 dplyr::select(word) %>%  
 unnest\_tokens(word\_pair, word, token = "ngrams", n=2)  
  
clntweets2 %>%  
 count(word\_pair, sort = TRUE) %>%  
 top\_n(25) %>%  
 mutate(word = reorder(word\_pair, n)) %>%  
 ggplot(aes(x=word, y=n)) +  
 geom\_col(fill = "#66B2FF")+  
 xlab(NULL)+  
 coord\_flip()+  
 labs(x = "Unique Word-Pairs",  
 y = "Count",  
 title = "Count of Unique Word-Pairs Found in #AvengersEndgame - Top 25")

## Selecting by n



#Unique word count - Trigram   
clntweets3 <- clntweets1 %>%  
 dplyr::select(word) %>%  
 unnest\_tokens(word\_trio, word, token = "ngrams", n=3)  
  
clntweets3 %>%  
 count(word\_trio, sort = TRUE) %>%  
 top\_n(25) %>%  
 mutate(word = reorder(word\_trio, n)) %>%  
 ggplot(aes(x=word, y=n)) +  
 geom\_col(fill = "#0080FF")+  
 xlab(NULL)+  
 coord\_flip()+  
 labs(x = "Unique Word-Trios",  
 y = "Count",  
 title = "Count of Unique Word-Trios Found in #AvengersEndgame - Top 25")

## Selecting by n



#SENTIMENT LEXICON ANALYSIS - labelling the tokens with sentiment

#BING AND NRC SENTIMENT LEXICON  
  
get\_sentiments("bing") %>%  
 count(sentiment)

## # A tibble: 2 x 2  
## sentiment n  
## <chr> <int>  
## 1 negative 4781  
## 2 positive 2005

get\_sentiments("nrc") %>%  
 filter(sentiment %in% c(  
 "positive",  
 "negative"  
 )) %>%  
 count(sentiment)

## # A tibble: 2 x 2  
## sentiment n  
## <chr> <int>  
## 1 negative 3324  
## 2 positive 2312

#BING | #NRC  
#negative 4781 | #negative 3324  
#positive 2005 | #positive 2312  
  
#both sentiment lexicons have more negative sentiment than positive.

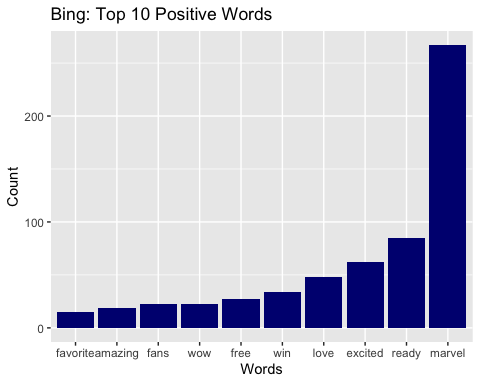
#BING  
#positive  
bing\_positive <- get\_sentiments("bing") %>%   
 filter(sentiment == "positive")  
  
tweet\_bing\_positive <- clntweets1 %>%  
 inner\_join(bing\_positive) %>%  
 count(word, sort = TRUE)

## Joining, by = "word"

#positive - distribution of words  
tweet\_bing\_positive\_graph <- tweet\_bing\_positive %>%  
 top\_n(10) %>%  
 mutate(word = reorder(word, n)) %>%  
 ggplot(aes(x=word, y=n)) +  
 geom\_col(fill = "Navy") +  
 labs(x = "Words",  
 y = "Count",  
 title = "Bing: Top 10 Positive Words")

## Selecting by n

tweet\_bing\_positive\_graph



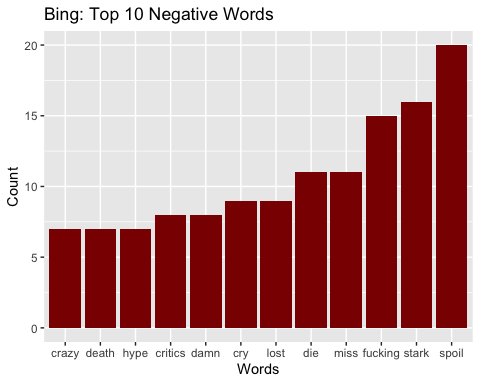
#negative  
bing\_negative <- get\_sentiments("bing") %>%   
 filter(sentiment == "negative")  
  
tweet\_bing\_negative <- clntweets1 %>%  
 inner\_join(bing\_negative) %>%  
 count(word, sort = TRUE)

## Joining, by = "word"

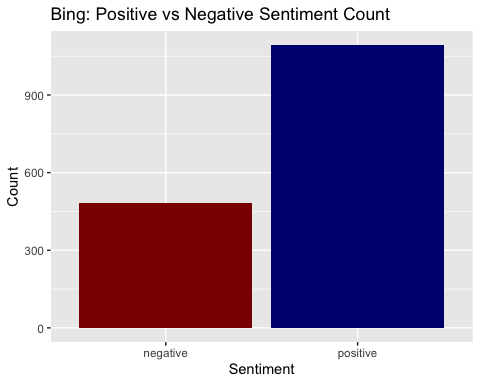
#negative - distribution of words  
tweet\_bing\_negative\_graph <- tweet\_bing\_negative %>%  
 top\_n(10) %>%  
 mutate(word = reorder(word, n)) %>%  
 ggplot(aes(x=word, y=n)) +  
 geom\_col(fill = "Dark Red")+  
 labs(x = "Words",  
 y = "Count",  
 title = "Bing: Top 10 Negative Words")

## Selecting by n

tweet\_bing\_negative\_graph



#full list of positive and negative words w/ sentiment  
bing <- rbind(tweet\_bing\_positive, tweet\_bing\_negative)  
  
#bing consolidated  
bing\_poscnt <- sum(tweet\_bing\_positive$n)  
bing\_negcnt <- sum(tweet\_bing\_negative$n)  
  
bing\_pos\_neg <- c("positive", "negative")  
bing\_pos\_neg\_cnt <- c(bing\_poscnt, bing\_negcnt)  
  
bing\_pos\_neg\_count <- data.frame(sentiment=bing\_pos\_neg,  
 count=bing\_pos\_neg\_cnt)  
  
#bing plot  
bing\_plot <- ggplot(data=bing\_pos\_neg\_count, aes(x=sentiment, y=count)) +  
 geom\_bar(stat="identity", fill=c("navy", "dark red")) +  
 labs(title="Bing: Positive vs Negative Sentiment Count") +  
 xlab("Sentiment") +  
 ylab("Count")   
  
bing\_plot



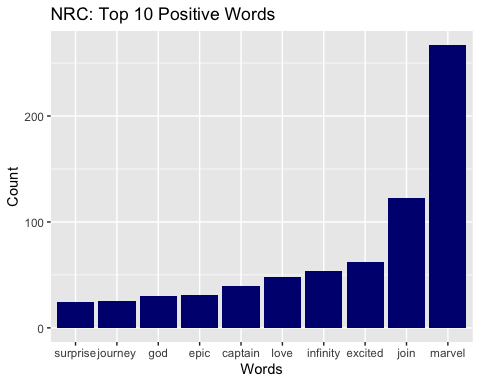
#NRC  
#positive  
nrc\_positive <- get\_sentiments("nrc") %>%   
 filter(sentiment == "positive")  
  
tweet\_nrc\_positive <- clntweets1 %>%  
 inner\_join(nrc\_positive) %>%  
 count(word, sort = TRUE)

## Joining, by = "word"

#positive - distribution of words  
tweet\_nrc\_positive\_graph <- tweet\_nrc\_positive %>%  
 top\_n(10) %>%  
 mutate(word = reorder(word, n)) %>%  
 ggplot(aes(x=word, y=n)) +  
 geom\_col(fill = "Navy")+  
 labs(x = "Words",  
 y = "Count",  
 title = "NRC: Top 10 Positive Words")

## Selecting by n

tweet\_nrc\_positive\_graph



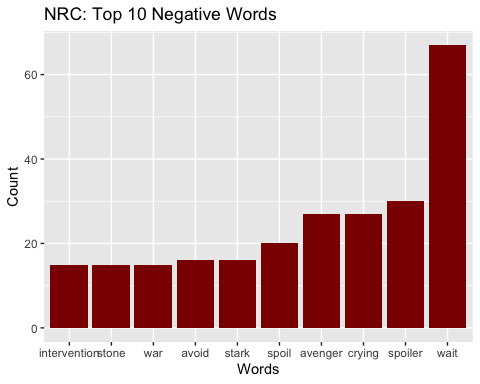
#negative  
nrc\_negative <- get\_sentiments("nrc") %>%   
 filter(sentiment == "negative")  
  
tweet\_nrc\_negative <- clntweets1 %>%  
 inner\_join(nrc\_negative) %>%  
 count(word, sort = TRUE)

## Joining, by = "word"

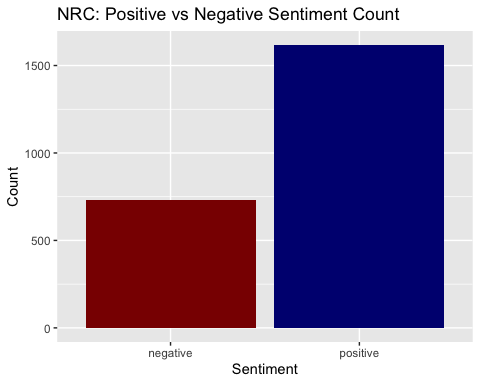
#negative - distribution of words  
tweet\_nrc\_negative\_graph <- tweet\_nrc\_negative %>%  
 top\_n(10) %>%  
 mutate(word = reorder(word, n)) %>%  
 ggplot(aes(x=word, y=n)) +  
 geom\_col(fill = "Dark Red")+  
 labs(x = "Words",  
 y = "Count",  
 title = "NRC: Top 10 Negative Words")

## Selecting by n

tweet\_nrc\_negative\_graph



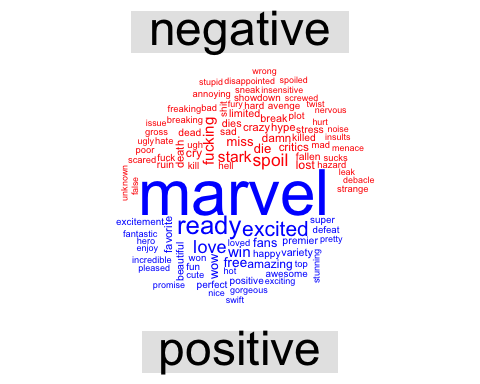
#full list of positive and negative words w/ sentiment  
nrc <- rbind(tweet\_nrc\_positive, tweet\_nrc\_negative)  
   
#nrc consolidated  
nrc\_poscnt <- sum(tweet\_nrc\_positive$n)  
nrc\_negcnt <- sum(tweet\_nrc\_negative$n)  
  
nrc\_pos\_neg <- c("positive", "negative")  
nrc\_pos\_neg\_cnt <- c(nrc\_poscnt, nrc\_negcnt)  
  
nrc\_pos\_neg\_count <- data.frame(sentiment=nrc\_pos\_neg,  
 count=nrc\_pos\_neg\_cnt)  
  
#nrc plot  
nrc\_plot <- ggplot(data=nrc\_pos\_neg\_count, aes(x=sentiment, y=count)) +  
 geom\_bar(stat="identity", fill=c("navy", "dark red")) +  
 labs(title="NRC: Positive vs Negative Sentiment Count") +  
 xlab("Sentiment") +  
 ylab("Count")   
  
nrc\_plot



#WORDCLOUD: BING VS NRC

#most common positive and negative words  
  
bing\_wordcloud <- clntweets1 %>%  
 inner\_join(get\_sentiments("bing")) %>%  
 count(word, sentiment, sort = TRUE) %>%  
 acast(word ~ sentiment, value.var = "n", fill = 0) %>%  
 comparison.cloud(  
 colors = c("red", "blue"),  
 max.words = 100  
 )

## Joining, by = "word"

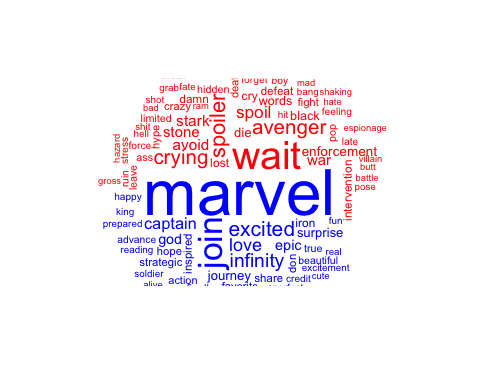


bing\_wordcloud

## NULL

nrc\_wordcloud <- clntweets1 %>%  
 inner\_join(get\_sentiments("nrc")) %>%  
 filter(sentiment %in% c("positive", "negative")) %>%  
 count(word, sentiment, sort = TRUE) %>%  
 acast(word ~ sentiment, value.var = "n", fill = 0) %>%  
 comparison.cloud(  
 colors = c("red", "blue"),  
 max.words = 100  
 )

## Joining, by = "word"



nrc\_wordcloud

## NULL

#FLAWS  
#there are a couple of flaws in the sentiment lexicon labels  
# 1. the main word in dataset is "Marvel" which shouldn't have a negative or positive sentiment because it is the title  
# of the film and should have remained a neutral word in the analysis  
# 2. cry/crying could be both negative and positive as people say ...  
# 3. The sentiment lexicon misreads slang: "hype" is a positive term for excitement  
# Through this analysis, I have learned that unigram sentiment analysis is not as accurate as it does not analyze  
# the context of the words and the sentence structure for example wait is seen as a negative word but if in the context  
# of "I can't wait..." then its sentiment is excitement/positive