Sentiment Analysis of Tweets over Stock Market prices

From: RStudio Markdown  
Purpose: assignment for *R for Data Science*Author: yours truly =]

## Import libraries

library(rtweet)  
library(ggplot2)  
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(maps)  
library(SentimentAnalysis)

##   
## Attaching package: 'SentimentAnalysis'

## The following object is masked from 'package:base':  
##   
## write

library(tidyquant)

## Loading required package: lubridate

##   
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':  
##   
## date

## Loading required package: PerformanceAnalytics

## Loading required package: xts

## Loading required package: zoo

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

##   
## Attaching package: 'xts'

## The following objects are masked from 'package:dplyr':  
##   
## first, last

##   
## Attaching package: 'PerformanceAnalytics'

## The following object is masked from 'package:graphics':  
##   
## legend

## Loading required package: quantmod

## Loading required package: TTR

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo

## Version 0.4-0 included new data defaults. See ?getSymbols.

## ══ Need to Learn tidyquant? ═════════════════════════════════════════════════  
## Business Science offers a 1-hour course - Learning Lab #9: Performance Analysis & Portfolio Optimization with tidyquant!  
## </> Learn more at: https://university.business-science.io/p/learning-labs-pro </>

## Get tweets

Setup search tweet parameters and search for tweets, this might take some time.

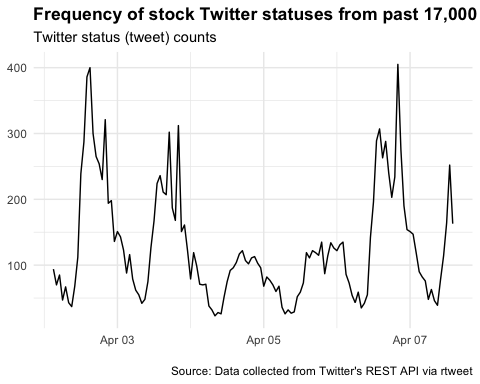
geocode="usa"  
q <- "stock market"  
n <- 17000  
  
rt <- search\_tweets(q=q, n=n, lang="en",   
 include\_rts=FALSE, geocode=lookup\_coords(geocode))

remove weblinks and #

rt$stripped\_text <- gsub("http\\S+","",rt$text)  
rt$stripped\_text <- gsub("#","",rt$stripped\_text)

Creates a ggplot2 plot of the frequency of tweets over a specified interval of time. plot time series of tweets.

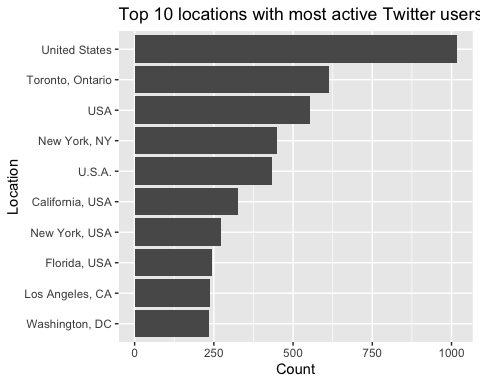
rt %>% ts\_plot("1 hours") +  
 ggplot2::theme\_minimal() +  
 ggplot2::theme(plot.title = ggplot2::element\_text(face = "bold")) +  
 ggplot2::labs(  
 x = NULL, y = NULL,  
 title = "Frequency of stock Twitter statuses from past 17,000 tweets",  
 subtitle = "Twitter status (tweet) counts",  
 caption = "\nSource: Data collected from Twitter's REST API via rtweet"  
 )



Let’s sort by count and just plot the top locations. To do this, you use top\_n(). Note that in this case you are grouping your data by user.

rt %>%  
 count(location, sort = TRUE) %>%  
 mutate(location = reorder(location, n)) %>%  
 top\_n(10) %>%  
 ggplot(aes(x = location, y = n)) +  
 geom\_col() +  
 coord\_flip() +  
 labs(x = "Location",  
 y = "Count",  
 title = "Top 10 locations with most active Twitter users")

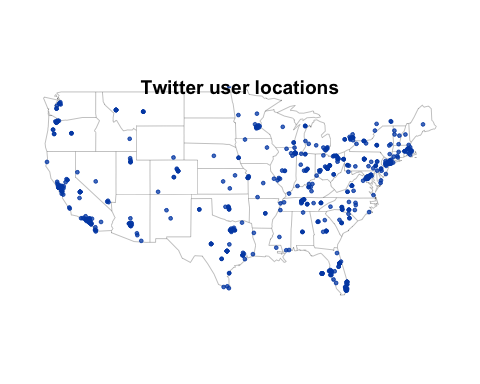
## Selecting by n



We attemped to show twitter user locations on US map. However, many users did not enable location sharing. So we can only show locations of the limited users who have shared.

# get missing geolocations  
map\_ref <- rtweet:::citycoords  
rt$location\_ref <- gsub(",","",tolower(rt$location))  
rt1 <- left\_join(rt,map\_ref,by=c("location\_ref"="city"))  
  
rtll <- lat\_lng(rt1)

## plot state boundaries  
par(mar = c(0, 0, 0, 0))  
maps::map("state", lwd = .25)  
title(main="Twitter user locations")  
  
## plot lat and lng points onto state map  
with(rtll, points(lng, lat, pch = 20, cex = .75, col = rgb(0, .3, .7, .75)))

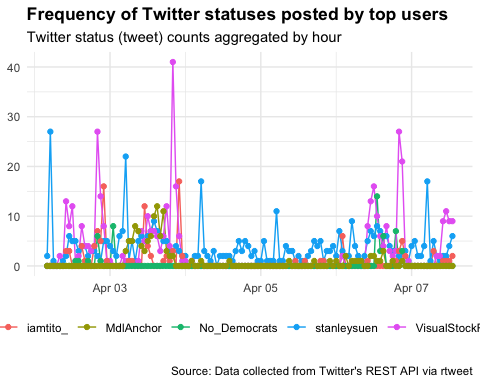


plot the frequency of tweets for each top 5 users over time.

top5 <- rt %>%  
 count(screen\_name, sort = TRUE) %>%  
 mutate(screen\_name = reorder(screen\_name, n)) %>%  
 top\_n(5)

## Selecting by n

rt %>%  
 dplyr::filter(screen\_name %in% top5$screen\_name) %>%  
 dplyr::group\_by(screen\_name) %>%  
 ts\_plot("1 hours", trim = 1L) +  
 ggplot2::geom\_point() +  
 ggplot2::theme\_minimal() +  
 ggplot2::theme(  
 legend.title = ggplot2::element\_blank(),  
 legend.position = "bottom",  
 plot.title = ggplot2::element\_text(face = "bold")) +  
 ggplot2::labs(  
 x = NULL, y = NULL,  
 title = "Frequency of Twitter statuses posted by top users",  
 subtitle = "Twitter status (tweet) counts aggregated by hour",  
 caption = "\nSource: Data collected from Twitter's REST API via rtweet"  
 )



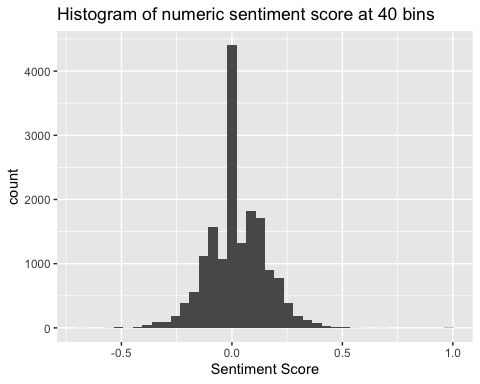
Use SentimentAnalysis default function to analyse the whole tweet. Then, add QDAP scores (both numeric and categorical) to our data.

rx1 <- analyzeSentiment(rt$stripped\_text)  
rx3 <- convertToDirection(rx1)

rx4 <- rt %>% select(stripped\_text,created\_at,favorite\_count,reply\_count,  
 retweet\_count,followers\_count,verified,screen\_name)  
rx4$sentimentQDAP <- rx1$SentimentQDAP  
rx4$WordCount <- rx1$WordCount  
rx4$sentimentQDAPgroup <- rx3$SentimentQDAP  
  
rx4 <- rx4[complete.cases(rx4[, c('sentimentQDAP', 'sentimentQDAPgroup')]),]

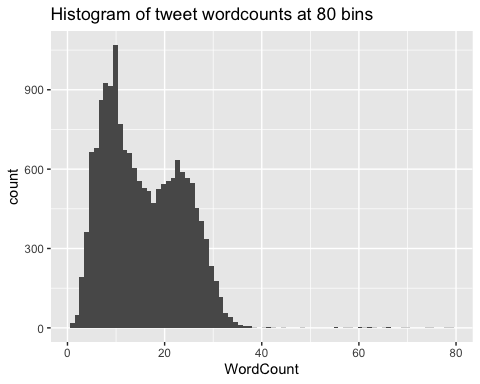
Look at histogram of numeric sentiment score

ggplot(rx4, aes(x=sentimentQDAP)) + geom\_histogram(bins=40) + ggtitle("Histogram of numeric sentiment score at 40 bins") +   
 xlab("Sentiment Score")



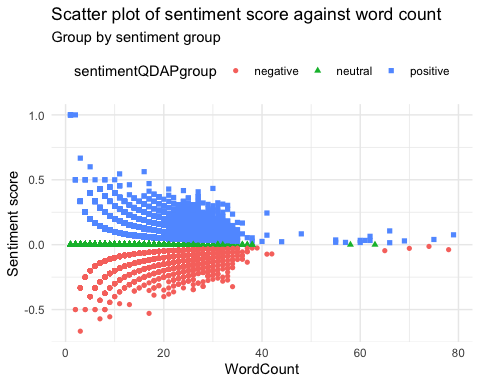
Look at histogram of tweet wordcounts

ggplot(rx4, aes(x=WordCount)) + geom\_histogram(bins=80) + ggtitle("Histogram of tweet wordcounts at 80 bins")



Look at scatter graph of sentiment score against word count, and group by sentiment group

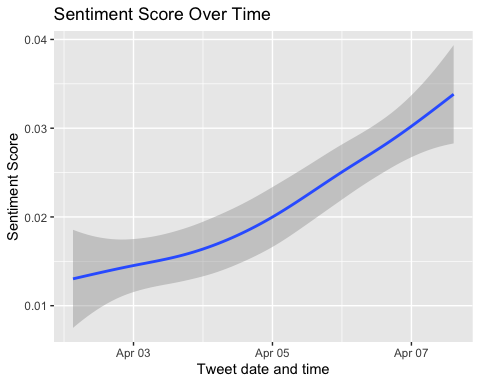
rx4 %>% ggplot(aes(x=WordCount, y=sentimentQDAP, shape=sentimentQDAPgroup,  
 color=sentimentQDAPgroup)) + geom\_point() + theme\_minimal() +  
 theme(legend.position = "top") +  
 labs(title="Scatter plot of sentiment score against word count",  
 subtitle="Group by sentiment group", y="Sentiment score",  
 fill="group")



plot of sentiment over time & automatically choose a method to model the change.

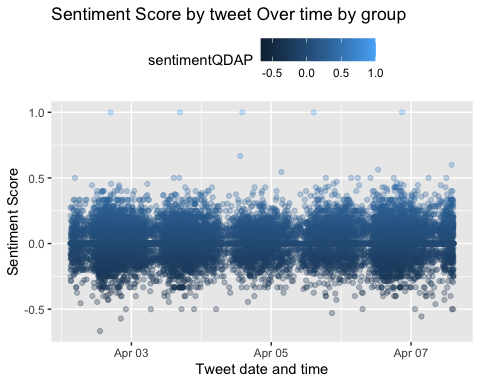
ggplot(rx4, aes(x = created\_at, y = sentimentQDAP)) +   
 geom\_smooth(method = "auto") + ggtitle("Sentiment Score Over Time") +  
 ylab("Sentiment Score") + xlab("Tweet date and time")

## `geom\_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'

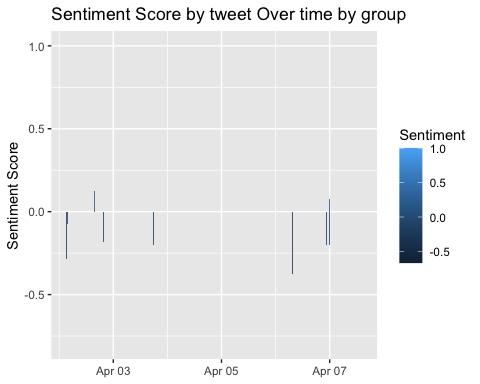


Sentiment Score by tweet Over time.

ggplot(rx4, aes(x = created\_at, y = sentimentQDAP, color=sentimentQDAP)) +  
 geom\_point(alpha = 0.3) +  
 theme(legend.position = "top") +  
 labs(y = "Sentiment Score", x = "Tweet date and time") +  
 ggtitle("Sentiment Score by tweet Over time by group")

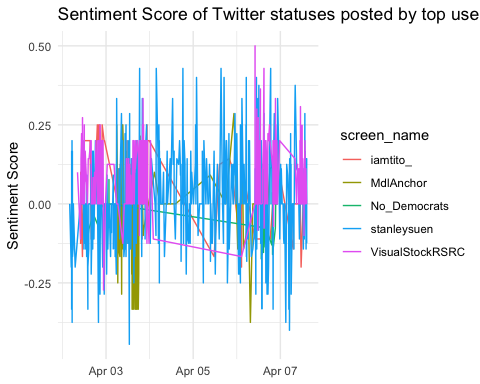


ggplot(rx4, aes(x = created\_at, y = sentimentQDAP, fill = sentimentQDAP)) + geom\_bar(alpha = 0.8, stat = "identity") +  
 labs(y = "Sentiment Score", x = NULL, fill = "Sentiment") +  
 ggtitle("Sentiment Score by tweet Over time by group")



plot the sentiment of tweets for each top 5 users over time.

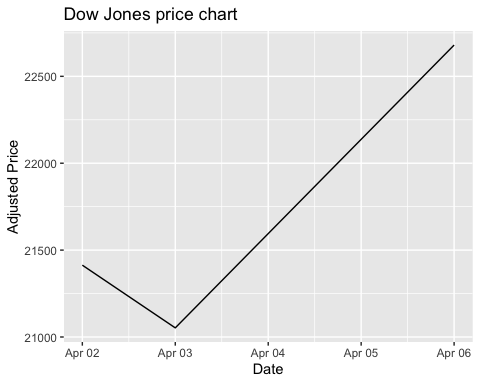
rx4 %>%  
 dplyr::filter(screen\_name %in% top5$screen\_name) %>%  
 ggplot(aes(x = created\_at, y = sentimentQDAP, color = screen\_name)) +   
 ggplot2::geom\_line() +  
 ggplot2::theme\_minimal() +  
 ggplot2::labs(  
 x = NULL, y = "Sentiment Score",  
 title = "Sentiment Score of Twitter statuses posted by top users")



# get oldest tweet date  
from <- substr(toString(min(rt$created\_at, na.rm=TRUE)),1,10)  
#from <- "2020-04-01"

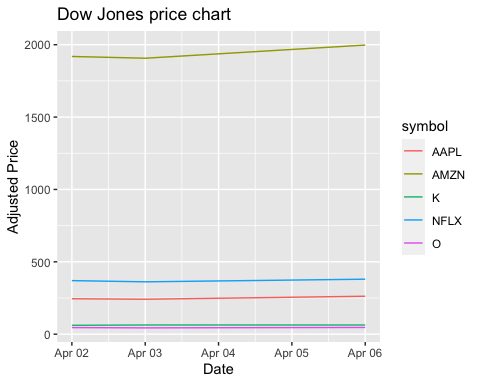
Download DOW Jones price data during since first tweet collected

DJI <- tq\_get('DJI',from=from,get='stock.prices')  
  
DJI %>%  
 ggplot(aes(x = date, y = adjusted)) +  
 geom\_line() +  
 labs(x = 'Date',  
 y = "Adjusted Price",  
 title = "Dow Jones price chart")



Look at stock prices of other popular stocks over the same period

tickers = c("AAPL", "NFLX", "AMZN", "K", "O")  
  
prices <- tq\_get(tickers,  
 from = from,  
 get = "stock.prices")  
  
prices %>%  
 ggplot(aes(x = date, y = adjusted, color = symbol)) +  
 geom\_line() + labs(x = 'Date',  
 y = "Adjusted Price",  
 title = "Other Stocks price chart")



Other Stocks price chart



## Analysis

We see that the overall sentiment score of tweets showed the same patter as the stock market. But the sentiment of top users do not. Therefore, we conclude that when the stock market goes up, the overall sentiment score is positive. And vice versa. Furthermore, the sentiment score does not seem to relate to word count of a tweet.