library(rtweet)

## Warning: package 'rtweet' was built under R version 3.5.3

library(tidytext)

## Warning: package 'tidytext' was built under R version 3.5.3

library(igraph)

## Warning: package 'igraph' was built under R version 3.5.3

##   
## Attaching package: 'igraph'

## The following objects are masked from 'package:stats':  
##   
## decompose, spectrum

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

library(dplyr)

## Warning: package 'dplyr' was built under R version 3.5.3

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:igraph':  
##   
## as\_data\_frame, groups, union

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

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

library(ggraph)

## Warning: package 'ggraph' was built under R version 3.5.3

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 3.5.3

library(sentimentr)

## Warning: package 'sentimentr' was built under R version 3.5.3

library(sqldf)

## Warning: package 'sqldf' was built under R version 3.5.3

## Loading required package: gsubfn

## Loading required package: proto

## Loading required package: RSQLite

library(tidyr)

## Warning: package 'tidyr' was built under R version 3.5.3

##   
## Attaching package: 'tidyr'

## The following object is masked from 'package:igraph':  
##   
## crossing

library(ggplot2)

require(curl)

## Loading required package: curl

## Warning: package 'curl' was built under R version 3.5.3

tweets <- na.omit(read.csv(curl("https://raw.githubusercontent.com/PravGitHub/Sentiment-Analysis-with-R/master/tweets.csv"),stringsAsFactors = FALSE))  
nrow(tweets)

## [1] 71517

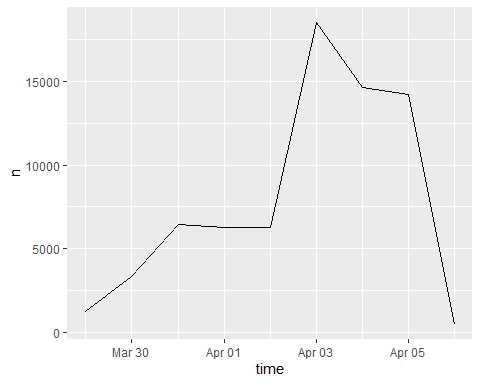
head(tweets)

## created\_at  
## 1 2020-04-05 23:59:26  
## 2 2020-04-05 23:59:24  
## 3 2020-04-05 23:58:51  
## 4 2020-04-05 23:58:05  
## 5 2020-04-04 09:13:02  
## 6 2020-04-05 23:49:58  
## text  
## 1 @MollyJongFast You realize that em the same info he sold his stock based on, was also public way before the stock market started going down, right?  
## 2 @Clayconboy1 I was Shocked when I checked official Government Statistics ( NOT from Liberal Sources, but the actual Stats) that Obamaâ\200\231s Economy wasnâ\200\231t nearly as bad as Limbaugh and others have said. Look it up. Trumpâ\200\231s GDP and Stock Market gains arenâ\200\231t better in some cases as Obamaâ\200\231s was.  
## 3 This Weeks Stock Market Outlook:\n\nIndexes still Blue White Red Downtrends.\nLackluster FTD Thursday\nNot many promising setups\n\nhttps://t.co/yKmGD3zJze  
## 4 @camjesse88 Show me one other person that predicted a sub 20K crash on the stock market (before it happenedH\n\nIf you do, Iâ\200\231ll delete my page today. \n\nIâ\200\231m waiting.  
## 5 Everyone who doubted my calls has lost badly!\n\nI warned people of doom months ago. \n\nI said the stock market would plummet hard, and same with Bitcoin/Alts! \n\nIâ\200\231m so glad I could save people from losing their money, and will continue to do so in the future! https://t.co/UOkijYRuaP  
## 6 The Stock Market will gain new fears later this year when entering Phase 2. \n\nI was the only person who predicted the massive market crash, and many people made money from those calls.\n\nIf you shorted at 29K+ when I said, keep them open for 1+ years! I'll share more soon. https://t.co/tjSylfMwYH

nrow(unique(tweets))

## [1] 70578

ts\_plot(tweets,by="days")



## Seperating into date and time:

tweets = separate(tweets,"created\_at",c("date","time"),sep=" ")  
head(tweets)

## date time  
## 1 2020-04-05 23:59:26  
## 2 2020-04-05 23:59:24  
## 3 2020-04-05 23:58:51  
## 4 2020-04-05 23:58:05  
## 5 2020-04-04 09:13:02  
## 6 2020-04-05 23:49:58  
## text  
## 1 @MollyJongFast You realize that em the same info he sold his stock based on, was also public way before the stock market started going down, right?  
## 2 @Clayconboy1 I was Shocked when I checked official Government Statistics ( NOT from Liberal Sources, but the actual Stats) that Obamaâ\200\231s Economy wasnâ\200\231t nearly as bad as Limbaugh and others have said. Look it up. Trumpâ\200\231s GDP and Stock Market gains arenâ\200\231t better in some cases as Obamaâ\200\231s was.  
## 3 This Weeks Stock Market Outlook:\n\nIndexes still Blue White Red Downtrends.\nLackluster FTD Thursday\nNot many promising setups\n\nhttps://t.co/yKmGD3zJze  
## 4 @camjesse88 Show me one other person that predicted a sub 20K crash on the stock market (before it happenedH\n\nIf you do, Iâ\200\231ll delete my page today. \n\nIâ\200\231m waiting.  
## 5 Everyone who doubted my calls has lost badly!\n\nI warned people of doom months ago. \n\nI said the stock market would plummet hard, and same with Bitcoin/Alts! \n\nIâ\200\231m so glad I could save people from losing their money, and will continue to do so in the future! https://t.co/UOkijYRuaP  
## 6 The Stock Market will gain new fears later this year when entering Phase 2. \n\nI was the only person who predicted the massive market crash, and many people made money from those calls.\n\nIf you shorted at 29K+ when I said, keep them open for 1+ years! I'll share more soon. https://t.co/tjSylfMwYH

## Getting the number tweets for each day

sqldf("select date, count(date) from tweets group by date")

## date count(date)  
## 1 2020-03-29 1246  
## 2 2020-03-30 3347  
## 3 2020-03-31 6456  
## 4 2020-04-01 6288  
## 5 2020-04-02 6274  
## 6 2020-04-03 18519  
## 7 2020-04-04 14658  
## 8 2020-04-05 14239  
## 9 2020-04-06 490

### Since 2020-03-29 and 2020-04-06 have a lot fewer tweets compared to the others, we will eliminate these two.

tweets <- subset(tweets,date!= '2020-03-29')  
tweets <- subset(tweets,date!= '2020-04-06')

## Verifying the elimination

sqldf("select date, count(date) from tweets group by date")

## date count(date)  
## 1 2020-03-30 3347  
## 2 2020-03-31 6456  
## 3 2020-04-01 6288  
## 4 2020-04-02 6274  
## 5 2020-04-03 18519  
## 6 2020-04-04 14658  
## 7 2020-04-05 14239

## Finding the sentiments

sentiment=sentiment\_by(tweets$text)

## Warning: Each time `sentiment\_by` is run it has to do sentence boundary disambiguation when a  
## raw `character` vector is passed to `text.var`. This may be costly of time and  
## memory. It is highly recommended that the user first runs the raw `character`  
## vector through the `get\_sentences` function.

head(sentiment)

## element\_id word\_count sd ave\_sentiment  
## 1: 1 26 NA 0.15689291  
## 2: 2 53 0.1584384 0.11580862  
## 3: 3 23 0.1363113 0.02421788  
## 4: 4 31 0.1002230 -0.07734377  
## 5: 5 53 0.4370799 -0.25856286  
## 6: 6 53 0.2336192 0.10634020

nrow(sentiment)

## [1] 69781

nrow(tweets)

## [1] 69781

## Finding the quartiles of the sentiments

summary(sentiment$ave\_sentiment)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -2.20663 -0.02839 0.06389 0.06466 0.16599 2.09752

res = data.frame(sentiment$ave\_sentiment,tweets$date)  
colnames(res)=c("sentiment","date")  
head(res)

## sentiment date  
## 1 0.15689291 2020-04-05  
## 2 0.11580862 2020-04-05  
## 3 0.02421788 2020-04-05  
## 4 -0.07734377 2020-04-05  
## 5 -0.25856286 2020-04-04  
## 6 0.10634020 2020-04-05

## Getting the day-wise count

sec <- sqldf("select date, count(sentiment) as count from res group by date")  
sec

## date count  
## 1 2020-03-30 3347  
## 2 2020-03-31 6456  
## 3 2020-04-01 6288  
## 4 2020-04-02 6274  
## 5 2020-04-03 18519  
## 6 2020-04-04 14658  
## 7 2020-04-05 14239

# Partitioning based on quartiles

## Min to First quartile

sec1 <- sqldf("select date, count(sentiment) as first\_q from res where sentiment>=-2.20663 and sentiment<=-0.02839 group by date")  
sec1

## date first\_q  
## 1 2020-03-30 717  
## 2 2020-03-31 1317  
## 3 2020-04-01 1390  
## 4 2020-04-02 1507  
## 5 2020-04-03 4851  
## 6 2020-04-04 3883  
## 7 2020-04-05 3774

## First quartile to Second quartile (median)

sec2 <- sqldf("select date, count(sentiment) as second\_q from res where sentiment>-0.02839 and sentiment<=0.06389 group by date")  
sec2

## date second\_q  
## 1 2020-03-30 956  
## 2 2020-03-31 1684  
## 3 2020-04-01 1620  
## 4 2020-04-02 1637  
## 5 2020-04-03 4489  
## 6 2020-04-04 3625  
## 7 2020-04-05 3440

## Second quartile to third quartile

sec3 <- sqldf("select date, count(sentiment) as third\_q from res where sentiment>0.06389 and sentiment<=0.16599 group by date")  
sec3

## date third\_q  
## 1 2020-03-30 762  
## 2 2020-03-31 2035  
## 3 2020-04-01 1914  
## 4 2020-04-02 1790  
## 5 2020-04-03 4453  
## 6 2020-04-04 3408  
## 7 2020-04-05 3269

## Third quartile to Max

sec4 <- sqldf("select date, count(sentiment) as fourth\_q from res where sentiment>0.16599 and sentiment<=2.09752 group by date")  
sec4

## date fourth\_q  
## 1 2020-03-30 912  
## 2 2020-03-31 1420  
## 3 2020-04-01 1364  
## 4 2020-04-02 1340  
## 5 2020-04-03 4726  
## 6 2020-04-04 3742  
## 7 2020-04-05 3755

## Consolidating all the data

sec = cbind(sec,sec1$first\_q,sec2$second\_q,sec3$third\_q,sec4$fourth\_q)  
colnames(sec)[3]<-"first\_q"  
colnames(sec)[4]<-"second\_q"  
colnames(sec)[5]<-"third\_q"  
colnames(sec)[6]<-"fourth\_q"  
  
sec

## date count first\_q second\_q third\_q fourth\_q  
## 1 2020-03-30 3347 717 956 762 912  
## 2 2020-03-31 6456 1317 1684 2035 1420  
## 3 2020-04-01 6288 1390 1620 1914 1364  
## 4 2020-04-02 6274 1507 1637 1790 1340  
## 5 2020-04-03 18519 4851 4489 4453 4726  
## 6 2020-04-04 14658 3883 3625 3408 3742  
## 7 2020-04-05 14239 3774 3440 3269 3755

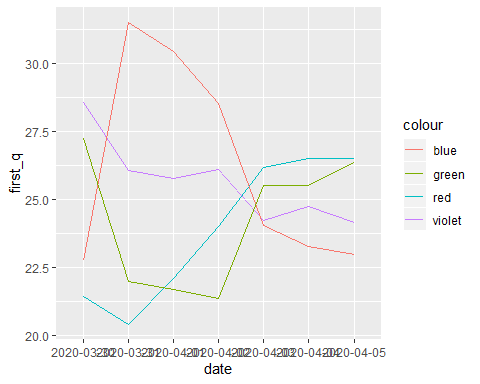
## Converting the count values to percentage values

sec$first\_q = sec$first\_q/sec$count \* 100  
sec$second\_q = sec$second\_q/sec$count \* 100  
sec$third\_q = sec$third\_q/sec$count \* 100  
sec$fourth\_q = sec$fourth\_q/sec$count \* 100  
sec

## date count first\_q second\_q third\_q fourth\_q  
## 1 2020-03-30 3347 21.42217 28.56289 22.76666 27.24828  
## 2 2020-03-31 6456 20.39963 26.08426 31.52107 21.99504  
## 3 2020-04-01 6288 22.10560 25.76336 30.43893 21.69211  
## 4 2020-04-02 6274 24.01976 26.09181 28.53044 21.35799  
## 5 2020-04-03 18519 26.19472 24.23997 24.04557 25.51974  
## 6 2020-04-04 14658 26.49065 24.73052 23.25010 25.52872  
## 7 2020-04-05 14239 26.50467 24.15900 22.95807 26.37123

# Plotting the results

ggplot(data=sec,aes(x=date,y=first\_q,color="red",group=1))+geom\_line()+geom\_line(data = sec,aes(x=date,y=second\_q,color="violet"))+geom\_line(data = sec,aes(x=date,y=third\_q,color="blue"))+geom\_line(data = sec,aes(x=date,y=fourth\_q,color="green"))



require(corrplot)

## Loading required package: corrplot

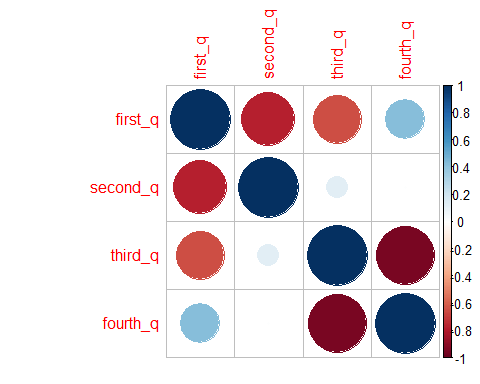
## Warning: package 'corrplot' was built under R version 3.5.3

## corrplot 0.84 loaded

correlation <- cor(sec[3:6])  
correlation

## first\_q second\_q third\_q fourth\_q  
## first\_q 1.0000000 -0.77504838 -0.6477604 0.42734352  
## second\_q -0.7750484 1.00000000 0.1261743 0.00257701  
## third\_q -0.6477604 0.12617426 1.0000000 -0.94028296  
## fourth\_q 0.4273435 0.00257701 -0.9402830 1.00000000

corrplot(correlation,method = "circle")



# Conclusions

### The percentage of highly negative tweets (shown by red) sees a sharp increase.

### The percentage of moderately negative tweets (shown by violet) sees a steady decrease.

### The percentage of moderately positive tweets (shown by blue) peaks early on and then drops.

### The percentage of highly positive tweets (shown by green) is low at the beginning and then increased.