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)

tweets <- na.omit(read.csv(file="C:/Users/praveen/Documents/tweets.csv",stringsAsFactors = FALSE))  
nrow(tweets)

## [1] 71517

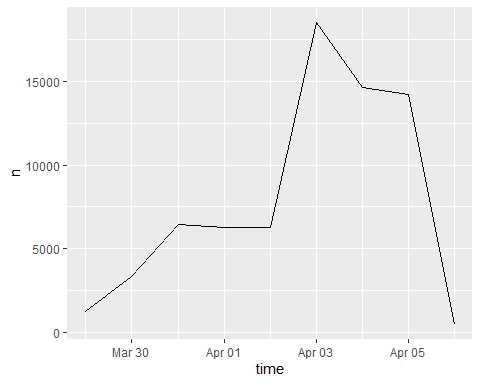
head(tweets)

## 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 ObamaÃ¢â‚¬â„¢s Economy wasnÃ¢â‚¬â„¢t nearly as bad as Limbaugh and others have said. Look it up. TrumpÃ¢â‚¬â„¢s GDP and Stock Market gains arenÃ¢â‚¬â„¢t better in some cases as ObamaÃ¢â‚¬â„¢s 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, IÃ¢â‚¬â„¢ll delete my page today. \n\nIÃ¢â‚¬â„¢m 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\nIÃ¢â‚¬â„¢m 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")



sentiment=sentiment\_by(tweets$text[1:10000])

## 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.

sentiment

## element\_id word\_count sd ave\_sentiment  
## 1: 1 26 NA 0.15689291  
## 2: 2 53 0.15843836 0.11580862  
## 3: 3 23 0.13631132 0.02421788  
## 4: 4 31 0.10022297 -0.07734377  
## 5: 5 53 0.43707986 -0.25856286  
## ---   
## 9996: 9996 53 0.00000000 0.00000000  
## 9997: 9997 5 NA 0.00000000  
## 9998: 9998 49 0.29976472 -0.16007737  
## 9999: 9999 45 0.02165064 0.02267757  
## 10000: 10000 32 0.09502054 0.27129167

summary(sentiment$ave\_sentiment)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -2.20663 -0.13381 0.00000 -0.01617 0.10840 1.03500

res = data.frame(sentiment$ave\_sentiment,tweets$created\_at[1:10000])  
colnames(res)=c("sentiment","created\_at")  
head(res)

## sentiment created\_at  
## 1 0.15689291 2020-04-05 23:59:26  
## 2 0.11580862 2020-04-05 23:59:24  
## 3 0.02421788 2020-04-05 23:58:51  
## 4 -0.07734377 2020-04-05 23:58:05  
## 5 -0.25856286 2020-04-04 09:13:02  
## 6 0.10634020 2020-04-05 23:49:58

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

head(res)

## sentiment date time  
## 1 0.15689291 2020-04-05 23:59:26  
## 2 0.11580862 2020-04-05 23:59:24  
## 3 0.02421788 2020-04-05 23:58:51  
## 4 -0.07734377 2020-04-05 23:58:05  
## 5 -0.25856286 2020-04-04 09:13:02  
## 6 0.10634020 2020-04-05 23:49:58

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

## sentiment date time count  
## 1 -0.2090020 2020-04-03 14:43:50 1455  
## 2 0.2712917 2020-04-04 14:32:55 3091  
## 3 -0.2440383 2020-04-05 00:00:01 5454

sec1 <- sqldf("select sentiment, date, time, count(sentiment) as count\_neg2 from res where sentiment>=-2.20663 and sentiment<=-0.13381 group by date")  
#sum(sec1$count)  
#sec1$count = sec1$count/sum(sec1$count) \*100  
head(sec1)

## sentiment date time count\_neg2  
## 1 -0.2090020 2020-04-03 14:43:50 296  
## 2 -0.1600774 2020-04-04 14:33:28 803  
## 3 -0.2440383 2020-04-05 00:00:01 1401

sec2 <- sqldf("select sentiment, date, time, count(sentiment) as count\_neg1 from res where sentiment>-0.13381 and sentiment<=0 group by date")  
#sum(sec2$count)  
#sec2$count = sec2$count/sum(sec2$count) \*100  
head(sec2)

## sentiment date time count\_neg1  
## 1 -0.09180614 2020-04-03 14:35:22 560  
## 2 0.00000000 2020-04-04 14:33:36 1025  
## 3 0.00000000 2020-04-05 00:00:08 1776

sec3 <- sqldf("select sentiment, date, time, count(sentiment) as count\_pos1 from res where sentiment>0 and sentiment<=0.10840 group by date")  
#sum(sec3$count)  
#sec3$count = sec3$count/sum(sec3$count) \*100  
head(sec3)

## sentiment date time count\_pos1  
## 1 0.062560453 2020-04-03 14:58:47 215  
## 2 0.022677574 2020-04-04 14:33:24 504  
## 3 0.007185795 2020-04-05 00:01:02 920

sec4 <- sqldf("select sentiment, date, time, count(sentiment) as count\_pos2 from res where sentiment>0.10840 and sentiment<=1.03500 group by date")  
#sum(sec4$count)  
#sec4$count = sec4$count/sum(sec4$count) \*100  
head(sec4)

## sentiment date time count\_pos2  
## 1 0.1365968 2020-04-03 07:54:54 384  
## 2 0.2712917 2020-04-04 14:32:55 758  
## 3 0.1218940 2020-04-05 00:00:12 1357

sec = cbind(sec,sec1$count\_neg2,sec2$count\_neg1,sec3$count\_pos1,sec4$count\_pos2)  
colnames(sec)[5]<-"neg2"  
colnames(sec)[6]<-"neg1"  
colnames(sec)[7]<-"pos1"  
colnames(sec)[8]<-"pos2"  
  
head(sec)

## sentiment date time count neg2 neg1 pos1 pos2  
## 1 -0.2090020 2020-04-03 14:43:50 1455 296 560 215 384  
## 2 0.2712917 2020-04-04 14:32:55 3091 803 1025 504 758  
## 3 -0.2440383 2020-04-05 00:00:01 5454 1401 1776 920 1357

sec$neg2 = sec$neg2/sec$count \* 100  
sec$neg1 = sec$neg1/sec$count \* 100  
sec$pos1 = sec$pos1/sec$count \* 100  
sec$pos2 = sec$pos2/sec$count \* 100  
head(sec)

## sentiment date time count neg2 neg1 pos1 pos2  
## 1 -0.2090020 2020-04-03 14:43:50 1455 20.34364 38.48797 14.77663 26.39175  
## 2 0.2712917 2020-04-04 14:32:55 3091 25.97865 33.16079 16.30540 24.52281  
## 3 -0.2440383 2020-04-05 00:00:01 5454 25.68757 32.56326 16.86835 24.88082

sec$date = factor(sec$date)  
head(sec)

## sentiment date time count neg2 neg1 pos1 pos2  
## 1 -0.2090020 2020-04-03 14:43:50 1455 20.34364 38.48797 14.77663 26.39175  
## 2 0.2712917 2020-04-04 14:32:55 3091 25.97865 33.16079 16.30540 24.52281  
## 3 -0.2440383 2020-04-05 00:00:01 5454 25.68757 32.56326 16.86835 24.88082

ggplot(data=sec,aes(x=date,y=neg2,color="red",group=1))+geom\_line()+geom\_line(data = sec,aes(x=date,y=neg1,color="violet"))+geom\_line(data = sec,aes(x=date,y=pos1,color="blue"))+geom\_line(data = sec,aes(x=date,y=pos2,color="green"))

