Project2

Xiaochen Jin

April 30, 2018

```
library(stringr)
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(ggplot2)
library(tm)
```

```
## Warning: package 'tm' was built under R version 3.4.4
```

```
## Loading required package: NLP
```

```
##
## Attaching package: 'NLP'
```

```
## The following object is masked from 'package:ggplot2':
##
##     annotate
```

```
library(wordcloud)
```

```
## Warning: package 'wordcloud' was built under R version 3.4.4
```

```
## Loading required package: RColorBrewer
```

Apple

```
load("C:/Users/liz/Desktop/tweets.AAPL.RData")
```

```
positive=scan('C:/Users/liz/Documents/positive-words.txt',what='character',comment.char=';')
negative=scan('C:/Users/liz/Documents/negative-words.txt',what='character',comment.char=';')


tryTolower = function(x)
{
  y = NA
  # tryCatch error
  try_error = tryCatch(tolower(x), error = function(e) e)
  # if not an error
  if (!inherits(try_error, "error")){
    y = tolower(x)
```

```
  }else{
    y = tolower(iconv(x, "latin1", "ASCII", sub=""))
  }
  return(y)
}

clean=function(t){
  t=gsub('[[:punct:]]','',t)
  t=gsub('[[:cntrl:]]','',t)
  t=gsub('\\d+','',t)
  t=gsub('[[:digit:]]','',t)
  t=gsub('@\\w+','',t)
  t=gsub('http\\w+','',t)
  t=gsub("^\\s+|\\s+$", "", t)
  t=sapply(t,function(x) tryTolower(x))
  t=str_split(t," ")
  t=unlist(t)
  return(t)
}


aaplT=lapply(tweets,function(t) t$getText())
clean.aapl=lapply(aaplT,function(x) clean(tryTolower(x)))


#####################################

score=function(tweet, pos = positive, neg = negative) {
  pos.match <- match(tweet, pos)
  neg.match <- match(tweet, neg)

  ## Scores
  pos.match.s <- !is.na(pos.match)
  neg.match.s <- !is.na(neg.match)

  pos.score <- sum(pos.match.s)
  neg.score <- sum(neg.match.s)

  ## Words
  posw <- pos[pos.match]
  posw <- posw[!is.na(posw)]

  negw <- neg[neg.match]
  negw <- negw[!is.na(negw)]

  return(list(pos.score = pos.score, neg.score = neg.score, pos.words = posw, neg.words = negw))
}

###calculate total number of positive and negative words ###
returnpscore=function(tweet) {
  pos.match=match(tweet,positive)
  pos.match=!is.na(pos.match)
  pos.score=sum(pos.match)
```

```
  return(pos.score)
}

returnnscore=function(tweet) {
  neg.match=match(tweet,negative)
  neg.match=!is.na(neg.match)
  neg.score=sum(neg.match)
  return(neg.score)
}

positive.score=lapply(clean.aapl,function(x) returnpscore(x))
negative.score=lapply(clean.aapl,function(x) returnnscore(x))

pcount=0
for (i in 1:length(positive.score)) {
  pcount=pcount+positive.score[[i]]
}
pcount
```

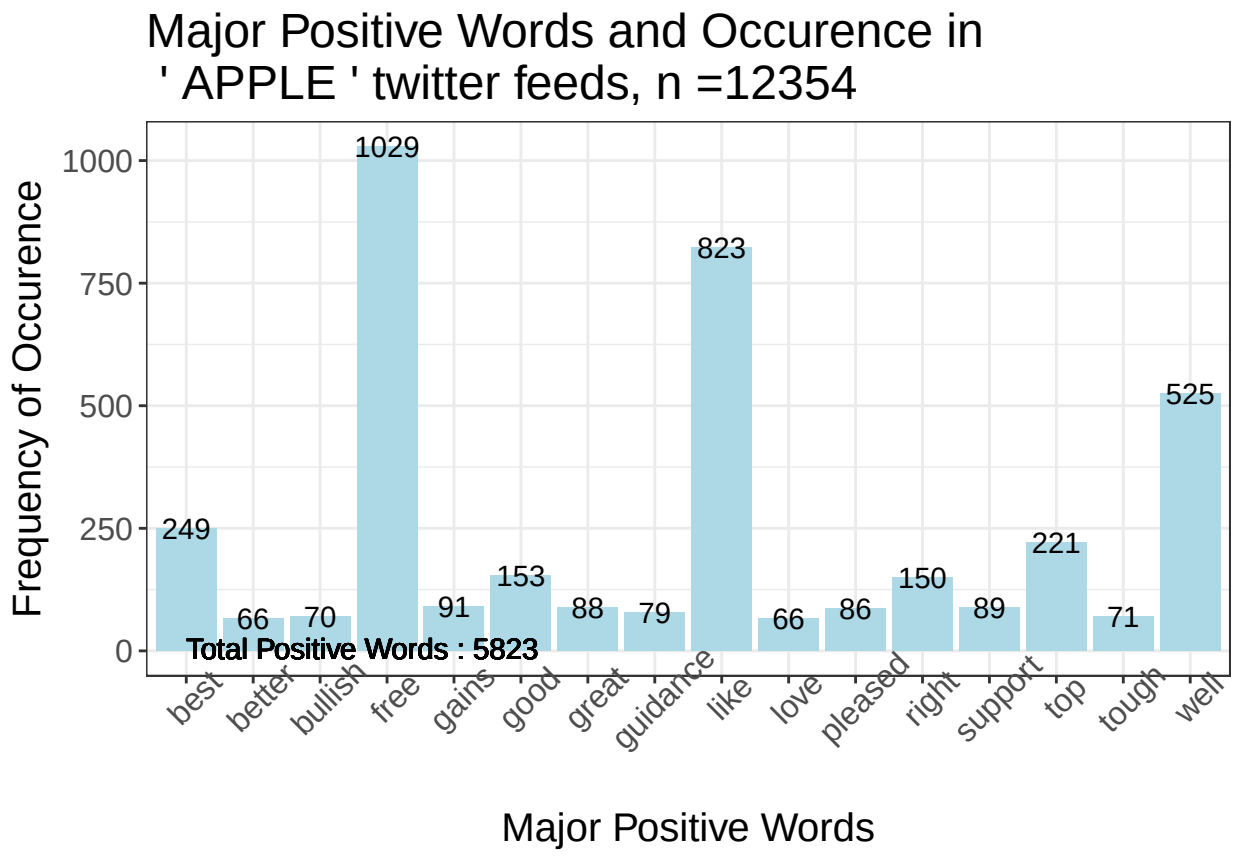
```
## [1] 5823
```

```
ncount=0
for (i in 1:length(negative.score)) {
  ncount=ncount+negative.score[[i]]
}
ncount
```

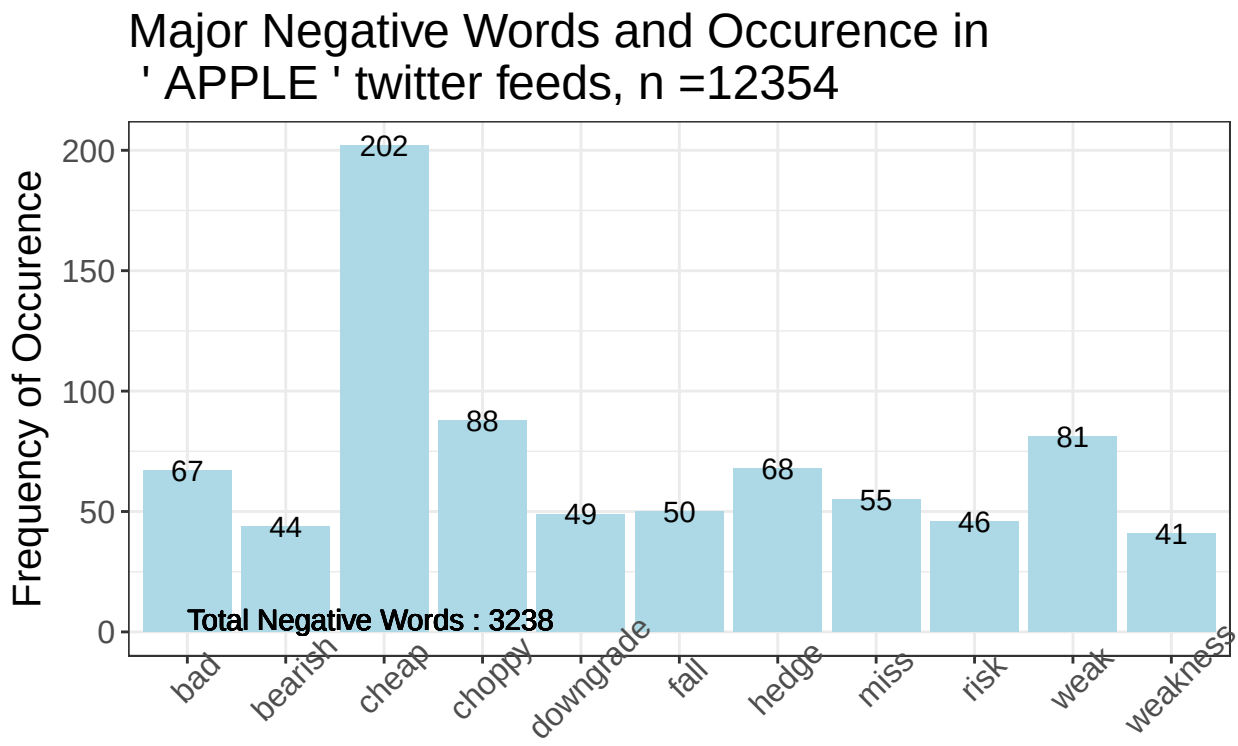
```
## [1] 3238
```

```
###creat graph###

poswords<-lapply(clean.aapl,function(x)score(x)$pos.words)
negwords<-lapply(clean.aapl,function(x)score(x)$neg.words)

pwords <- unlist(poswords)
nwords <- unlist(negwords)
dpwords=data.frame(table(pwords))
dnwords=data.frame(table(nwords))

dpwords=dpwords%>%
  mutate(pwords=as.character(pwords))%>%
  filter(Freq>60)

dnwords=dnwords%>%
  mutate(nwords=as.character(nwords))%>%
  filter(Freq>40)
```

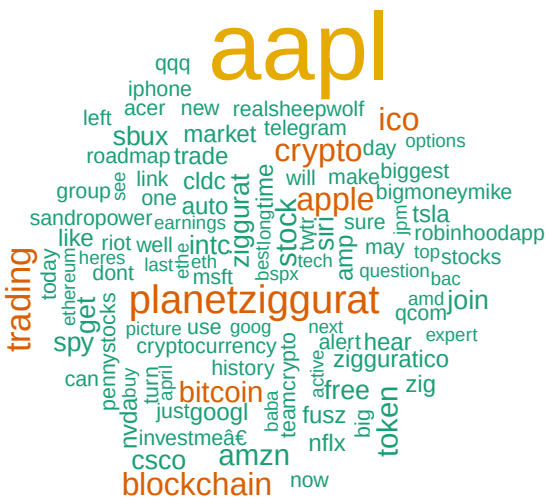
```
##aapl##
ggplot(dpwords,aes(pwords,Freq))+geom_bar(stat="identity",fill="lightblue")+theme_bw()+
  geom_text(aes(pwords,Freq,label=Freq),size=4)+
  labs(x="Major Positive Words", y="Frequency of Occurence",title=paste("Major Positive Words and Occur
  geom_text(aes(1,5,label=paste("Total Positive Words : 5823")),size=4,hjust=0)+theme(axis.text.x=eleme
```




```
ggplot(dnwords,aes(nwords,Freq))+geom_bar(stat="identity",fill="lightblue")+theme_bw()+
  geom_text(aes(nwords,Freq,label=Freq),size=4)+
  labs(x="Major Negative Words", y="Frequency of Occurence",title=paste("Major Negative Words and Occur
  geom_text(aes(1,5,label=paste("Total Negative Words : 3238")),size=4,hjust=0)+theme(axis.text.x=eleme
```




```
###wordcloud###

tweetscorpus=Corpus(VectorSource(clean.aapl))
tweetscorpus=tm_map(tweetscorpus,removeWords,stopwords("english"))
wordcloud(tweetscorpus,scale=c(3,0.5),random.order = TRUE,rot.per = 0.20,use.r.layout = FALSE,colors = 
```



SPY

```
load("C:/Users/liz/Desktop/tweets.SPY.RData")

aaplT=lapply(tweets,function(t) t$getText())
clean.aapl=lapply(aaplT,function(x) clean(tryTolower(x)))



#######################################

###calculate total number of positive and negative words ###

positive.score=lapply(clean.aapl,function(x) returnpscore(x))
negative.score=lapply(clean.aapl,function(x) returnnscore(x))

pcount=0
for (i in 1:length(positive.score)) {
  pcount=pcount+positive.score[[i]]
}
pcount
```

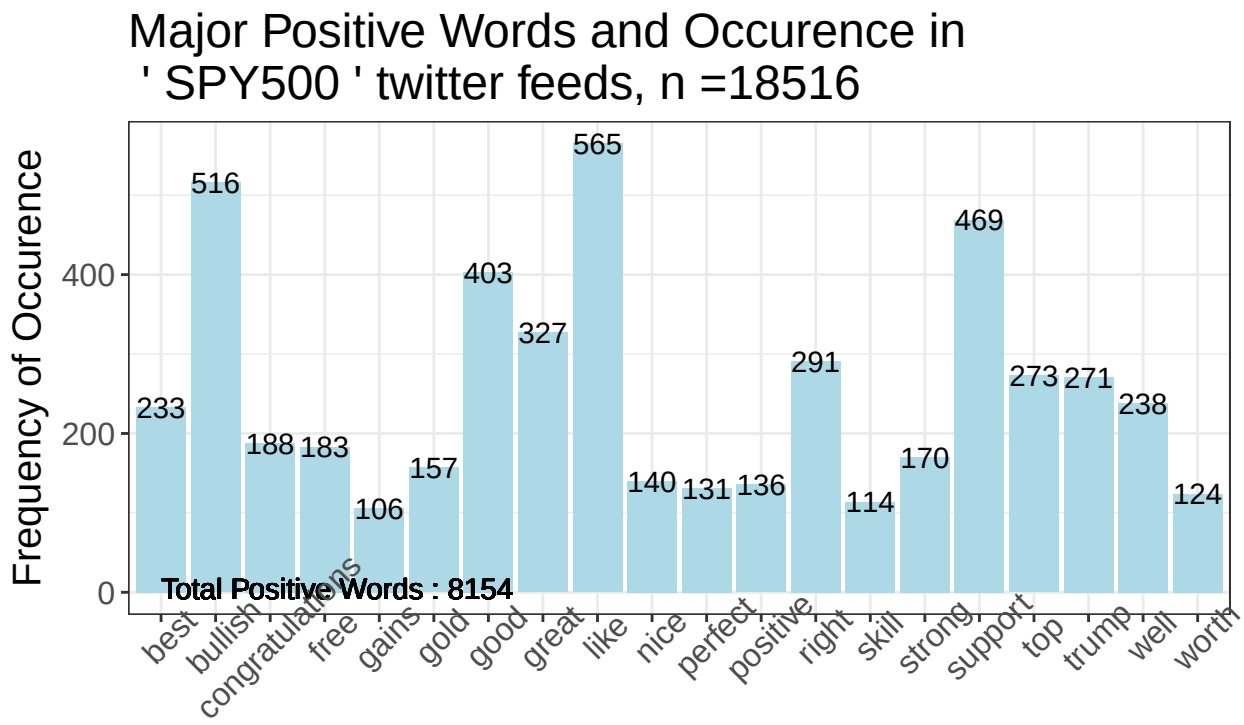
```
## [1] 8154
```

```
ncount=0
for (i in 1:length(negative.score)) {
```

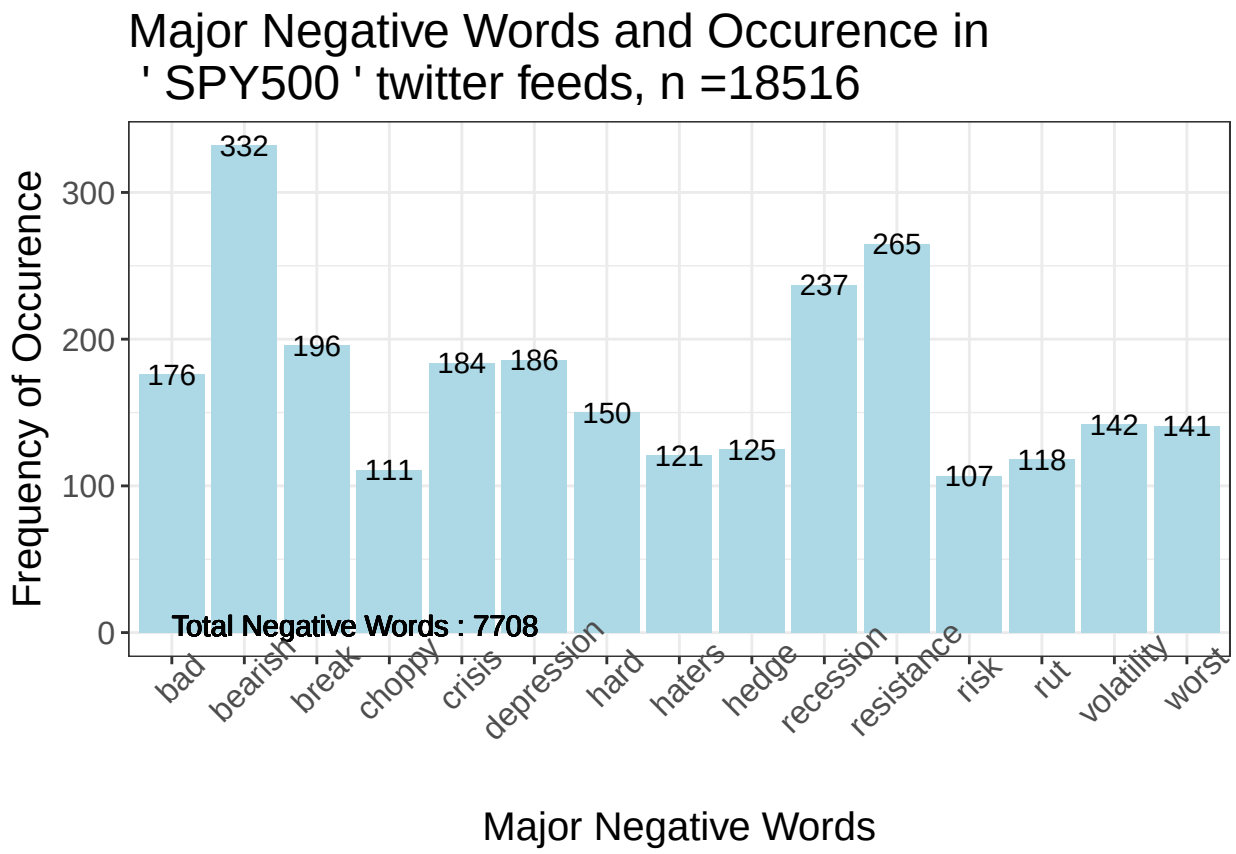
```
  ncount=ncount+negative.score[[i]]
}
ncount
```

```
## [1] 7708
```

```
###creat graph###

poswords<-lapply(clean.aapl,function(x)score(x)$pos.words)
negwords<-lapply(clean.aapl,function(x)score(x)$neg.words)

pwords <- unlist(poswords)
nwords <- unlist(negwords)
dpwords=data.frame(table(pwords))
dnwords=data.frame(table(nwords))

dpwords=dpwords%>%
  mutate(pwords=as.character(pwords))%>%
  filter(Freq>100)

dnwords=dnwords%>%
  mutate(nwords=as.character(nwords))%>%
  filter(Freq>100)

ggplot(dpwords,aes(pwords,Freq))+geom_bar(stat="identity",fill="lightblue")+theme_bw()+
  geom_text(aes(pwords,Freq,label=Freq),size=4)+
  labs(x="Major Positive Words", y="Frequency of Occurence",title=paste("Major Positive Words and Occur
  geom_text(aes(1,5,label=paste("Total Positive Words : 8154")),size=4,hjust=0)+theme(axis.text.x=eleme
```




```
ggplot(dnwords,aes(nwords,Freq))+geom_bar(stat="identity",fill="lightblue")+theme_bw()+
  geom_text(aes(nwords,Freq,label=Freq),size=4)+
  labs(x="Major Negative Words", y="Frequency of Occurence",title=paste("Major Negative Words and Occur
  geom_text(aes(1,5,label=paste("Total Negative Words : 7708")),size=4,hjust=0)+theme(axis.text.x=eleme
```




```
###wordcloud###

tweetscorpus=Corpus(VectorSource(clean.aapl))
tweetscorpus=tm_map(tweetscorpus,removeWords,stopwords("english"))
wordcloud(tweetscorpus,scale=c(3,0.5),random.order = TRUE,rot.per = 0.20,use.r.layout = FALSE,colors = 
```

FB

```
load("C:/Users/liz/Desktop/tweets.FB.RData")

aaplT=lapply(tweets,function(t) t$getText())
clean.aapl=lapply(aaplT,function(x) clean(tryTolower(x)))


#######################################

###calculate total number of positive and negative words ###

positive.score=lapply(clean.aapl,function(x) returnpscore(x))
negative.score=lapply(clean.aapl,function(x) returnnscore(x))

pcount=0
for (i in 1:length(positive.score)) {
  pcount=pcount+positive.score[[i]]
}
pcount
```

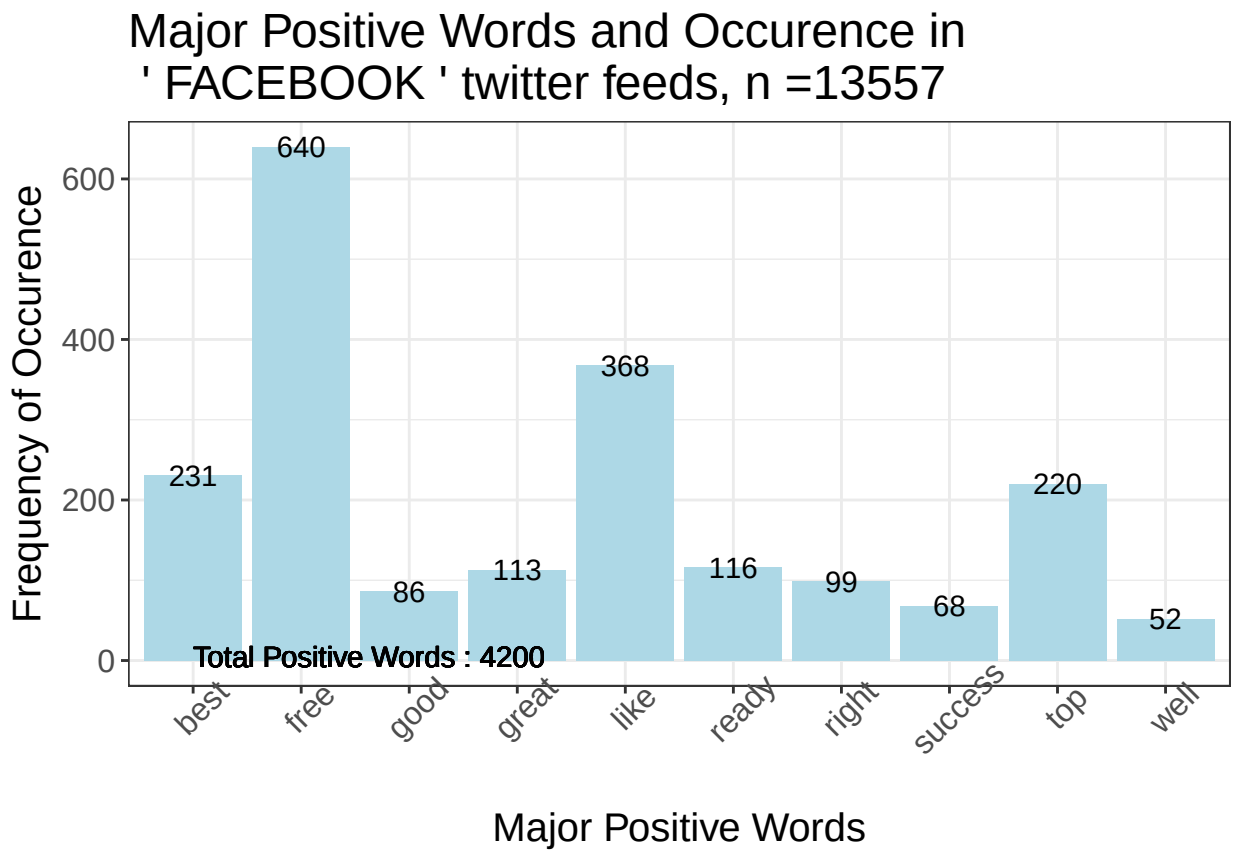
```
## [1] 4200
```

```
ncount=0
for (i in 1:length(negative.score)) {
```

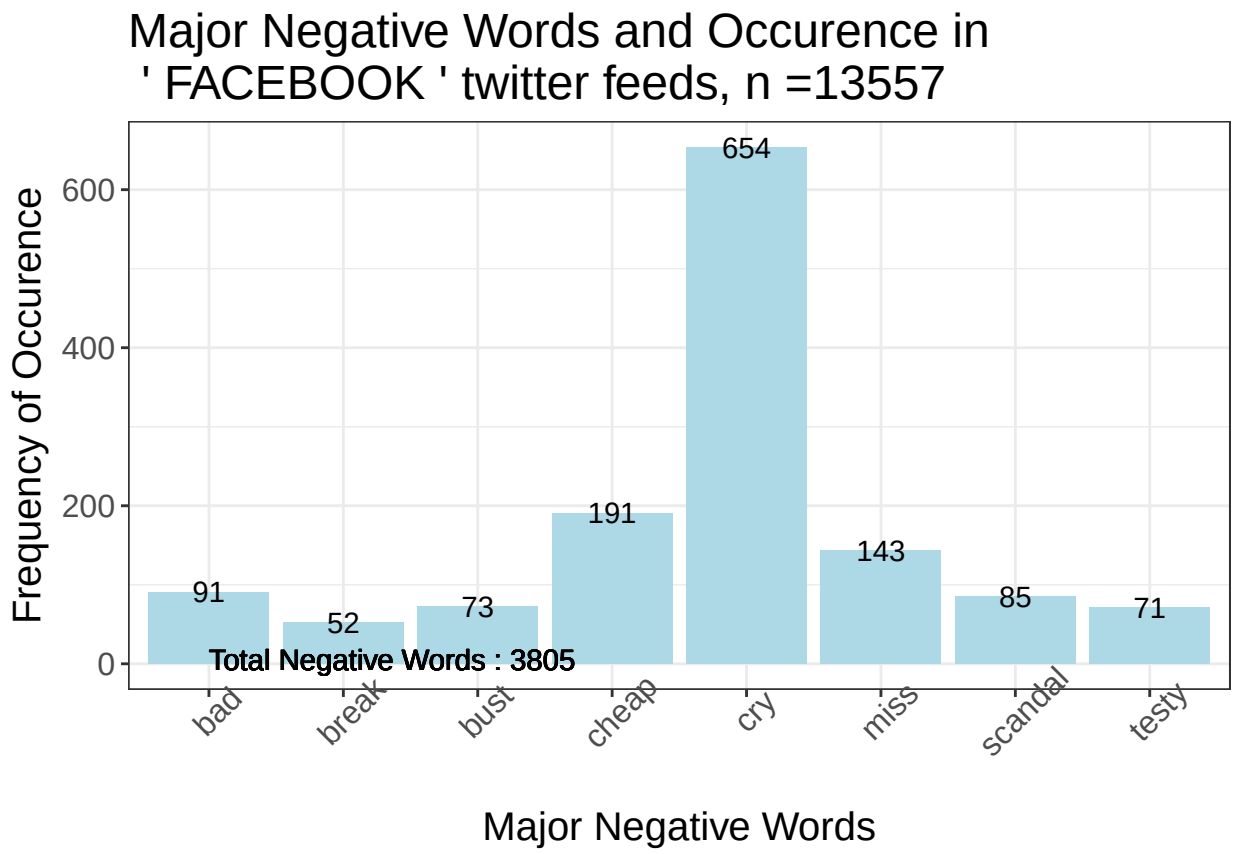
```
  ncount=ncount+negative.score[[i]]
}
ncount
```

```
## [1] 3805
```

```
###creat graph###

poswords<-lapply(clean.aapl,function(x)score(x)$pos.words)
negwords<-lapply(clean.aapl,function(x)score(x)$neg.words)

pwords <- unlist(poswords)
nwords <- unlist(negwords)
dpwords=data.frame(table(pwords))
dnwords=data.frame(table(nwords))

dpwords=dpwords%>%
  mutate(pwords=as.character(pwords))%>%
  filter(Freq>50)

dnwords=dnwords%>%
  mutate(nwords=as.character(nwords))%>%
  filter(Freq>50)

ggplot(dpwords,aes(pwords,Freq))+geom_bar(stat="identity",fill="lightblue")+theme_bw()+
  geom_text(aes(pwords,Freq,label=Freq),size=4)+
  labs(x="Major Positive Words", y="Frequency of Occurence",title=paste("Major Positive Words and Occur
  geom_text(aes(1,5,label=paste("Total Positive Words : 4200")),size=4,hjust=0)+theme(axis.text.x=eleme
```



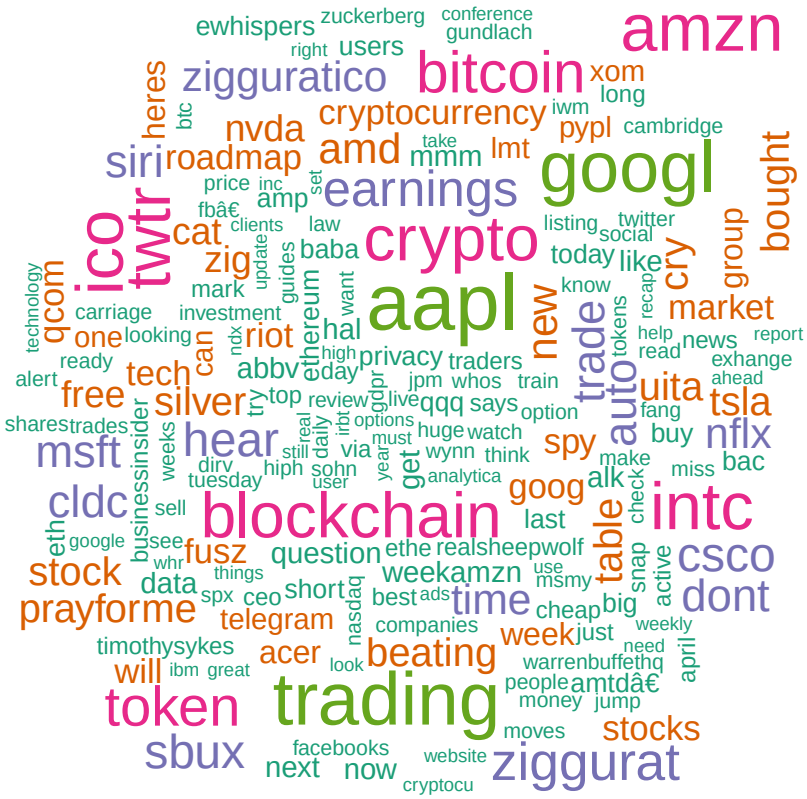

```
ggplot(dnwords,aes(nwords,Freq))+geom_bar(stat="identity",fill="lightblue")+theme_bw()+
  geom_text(aes(nwords,Freq,label=Freq),size=4)+
  labs(x="Major Negative Words", y="Frequency of Occurence",title=paste("Major Negative Words and Occur
  geom_text(aes(1,5,label=paste("Total Negative Words : 3805")),size=4,hjust=0)+theme(axis.text.x=eleme
```




```
###wordcloud###

tweetscorpus=Corpus(VectorSource(clean.aapl))
tweetscorpus=tm_map(tweetscorpus,removeWords,stopwords("english"))
wordcloud(tweetscorpus,scale=c(3,0.5),random.order = TRUE,rot.per = 0.20,use.r.layout = FALSE,colors = 
```

```
## Warning in wordcloud(tweetscorpus, scale = c(3, 0.5), random.order =
## TRUE, : planetziggurat could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(tweetscorpus, scale = c(3, 0.5), random.order =
## TRUE, : facebook could not be fit on page. It will not be plotted.
```


amzn
ewhispers
zuckerberg
conference
gundlach
right
users
bitcoin
xom
long
heres
btc
zigguratico
cryptocurrency
iwm
pypl
cambridge
nvda
amd
take
mmm
lmt
siri
roadmap
googl
bought
price
inc
set
earnings
amp
fbâ€
clients
law
listing
twitter
social
ico
twtr
cat
crypto
group
cry
zig
update
guides
baba
today
like
qcom
technology
mark
know
aapl
new
market
carriage
investment
want
recap
trade
tokens
one
looking
can
ndx
riot
hal
help
news
report
read
ready
high
privacy
traders
exhange
alert
tech
abbv
day
gdpr
jpm
whos
train
ahead
free
silver
try
top
review
live
qqq
says
option
auto
uita
tsla
fang
shares
trades
insider
weeks
hear
real
daily
irbt
options
must
huge
watch
spy
buy
nflx
msft
still
via
wynn
think
make
get
year
dirv
hiph
sohn
tuesday
user
analytica
goog
alk
miss
bac
check
cldc
sell
blockchain
last
intc
eth
google
see
businessinsider
fusz
question
ethe
realsheepwolf
table
snap
active
csco
whr
things
weekamzn
use
msmy
stock
data
spx
ceo
short
best
ads
time
cheap
big
dont
prayforme
telegram
nasdaq
companies
week
just
weekly
timothysykes
acer
look
beating
need
warrenbuffethq
april
will
ibm
great
people
amtdâ€
token
trading
money
jump
stocks
moves
sbux
facebooks
website
ziggurat
next
now
cryptocu