Intel sentiment analysis

October 9, 2021

[1]:

```r
library("tm")
library("SnowballC")
library("wordcloud")
library("RColorBrewer")
library("syuzhet")
library("ggplot2")
```

```
Loading required package: NLP

Loading required package: RColorBrewer


Attaching package: ‘ggplot2’


The following object is masked from ‘package:NLP’:

    annotate
```

[2]:

```r
text<-readLines('intel.txt')
TextDoc <- Corpus(VectorSource(text))
```

[3]:

```r
toSpace <- content_transformer(function (x , pattern ) gsub(pattern, " ", x))
TextDoc <- tm_map(TextDoc, toSpace, "/")
TextDoc <- tm_map(TextDoc, toSpace, "@")
TextDoc <- tm_map(TextDoc, toSpace, "\\|")
# Convert the text to lower case
TextDoc <- tm_map(TextDoc, content_transformer(tolower))
# Remove numbers
TextDoc <- tm_map(TextDoc, removeNumbers)
# Remove english common stopwords
TextDoc <- tm_map(TextDoc, removeWords, stopwords("english"))
# Remove your own stop word
# specify your custom stopwords as a character vector
TextDoc <- tm_map(TextDoc, removeWords, c("s", "company", "team"))
# Remove punctuations
```

```
TextDoc <- tm_map(TextDoc, removePunctuation)
# Eliminate extra white spaces
TextDoc <- tm_map(TextDoc, stripWhitespace)
# Text stemming - which reduces words to their root form
TextDoc <- tm_map(TextDoc, stemDocument)
```

```
Warning message in tm_map.SimpleCorpus(TextDoc, toSpace, "/"):
"transformation drops documents"
Warning message in tm_map.SimpleCorpus(TextDoc, toSpace, "@"):
"transformation drops documents"
Warning message in tm_map.SimpleCorpus(TextDoc, toSpace, "\\|"):
"transformation drops documents"
Warning message in tm_map.SimpleCorpus(TextDoc, content_transformer(tolower)):
"transformation drops documents"
Warning message in tm_map.SimpleCorpus(TextDoc, removeNumbers):
"transformation drops documents"
Warning message in tm_map.SimpleCorpus(TextDoc, removeWords,
stopwords("english")):
"transformation drops documents"
Warning message in tm_map.SimpleCorpus(TextDoc, removeWords, c("s", "company",
"team")):
"transformation drops documents"
Warning message in tm_map.SimpleCorpus(TextDoc, removePunctuation):
"transformation drops documents"
Warning message in tm_map.SimpleCorpus(TextDoc, stripWhitespace):
"transformation drops documents"
Warning message in tm_map.SimpleCorpus(TextDoc, stemDocument):
"transformation drops documents"
```

[4]:

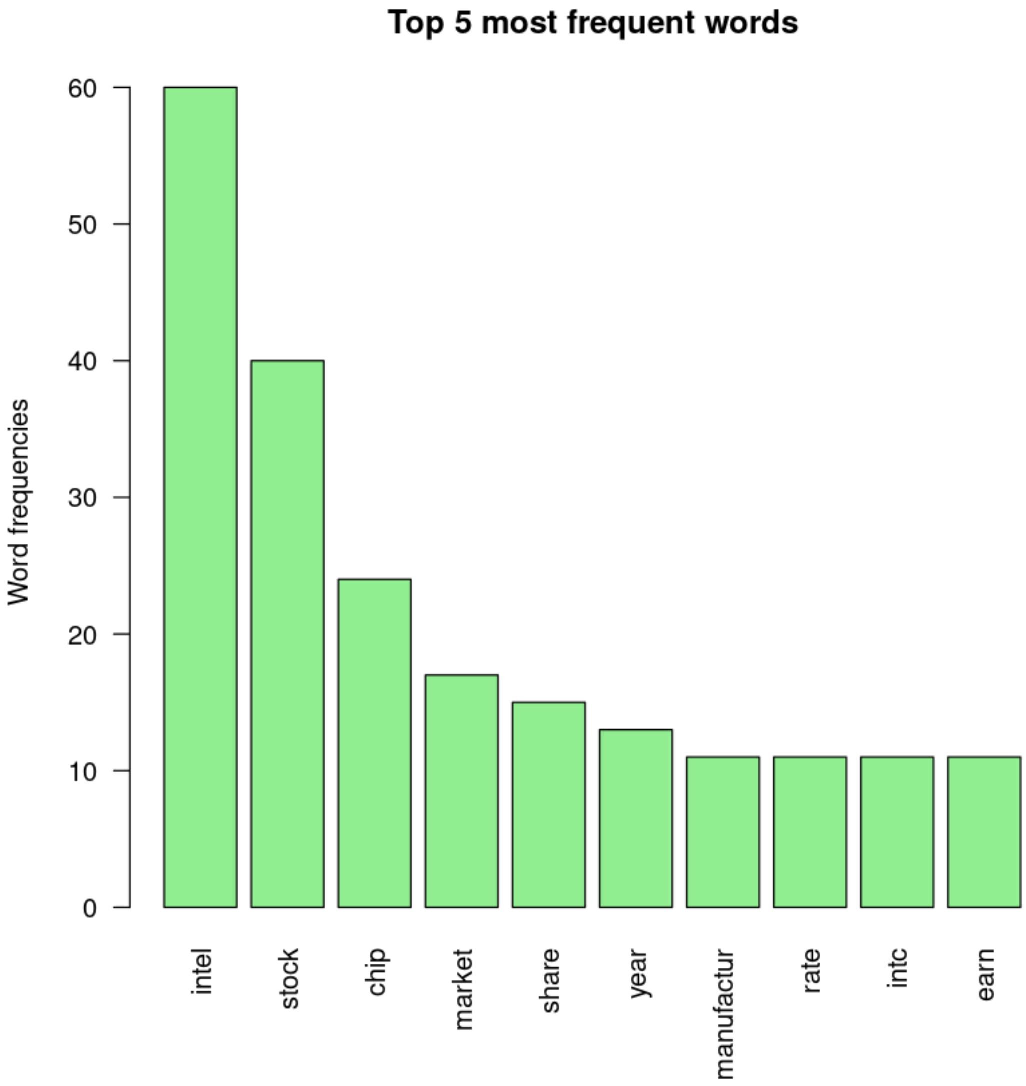
```
# Build a term-document matrix
TextDoc_dtm <- TermDocumentMatrix(TextDoc)
dtm_m <- as.matrix(TextDoc_dtm)
# Sort by descearing value of frequency
dtm_v <- sort(rowSums(dtm_m),decreasing=TRUE)
dtm_d <- data.frame(word = names(dtm_v),freq=dtm_v)
# Display the top 5 most frequent words
head(dtm_d, 10)
```

A data.frame: 10 × 2

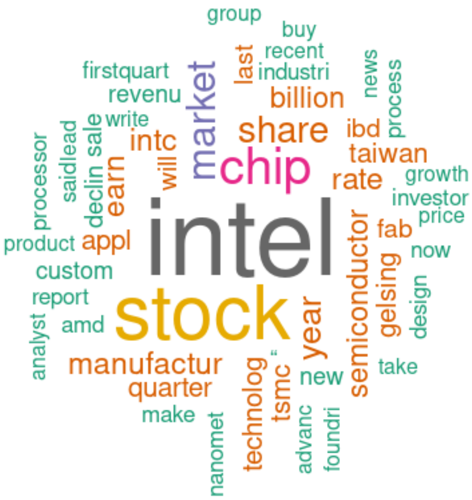
	word	freq
	<chr>	<dbl>
intel	intel	60
stock	stock	40
chip	chip	24
market	market	17
share	share	15
year	year	13
manufactur	manufactur	11
rate	rate	11
intc	intc	11
earn	earn	11

[5]:

```r
barplot(dtm_d[1:10,]$freq, las = 2, names.arg = dtm_d[1:10,]$word,
        col ="lightgreen", main ="Top 5 most frequent words",
        ylab = "Word frequencies")
```



```
[6]: set.seed(1234)
     wordcloud(words = dtm_d$word, freq = dtm_d$freq, min.freq = 5,
               max.words=100, random.order=FALSE, rot.per=0.40,
               colors=brewer.pal(8, "Dark2"))
```



```
[7]: findAssocs(TextDoc_dtm, terms = c("good","work","health"), corlimit = 0.25)
```

```
$good  build 1 develop 1 europ 1 major 1 outlin 1 plus 1 spend 1 plan 0.82 becom 0.7
       global 0.7 arizona 0.7 provid 0.7 capac 0.7 progress 0.7 serv 0.7 two 0.57 march 0.57
       announc 0.57 also 0.49 foundri 0.44 said 0.4 make 0.4 new 0.37 semiconductor 0.3
       custom              0.29 nanomet               0.28 fab              0.25
```

$work

$health

[8]:
```
findAssocs(TextDoc_dtm, terms = findFreqTerms(TextDoc_dtm, lowfreq = 50),␣
↪corlimit = 0.25)
```

$intel = **share** 0.5 **billion** 0.45 **expect** 0.45 **earn** 0.44 **adjust** 0.42 **sale** 0.4 **semiconductor** 0.34 **earlier** 0.33 **period** 0.33 **second** 0.33 **june** 0.33 **ceo** 0.31 **gelsing** 0.31 **pat** 0.31 **financi** 0.31 **quarter** 0.31 **manufactur** 0.3 **dip** 0.3 **predict** 0.3 **yearearli** 0.3 **chief** 0.3 **assum** 0.3 **bob** 0.3 **feb** 0.3 **fiscal** 0.3 **jan** 0.3 **mind** 0.3 **offic** 0.3 **previous** 0.3 **replac** 0.3 **swan** 0.3 **technologyfocus** 0.3 **top** 0.3 **vmw** 0.3 **vmware** 0.3 **chip** 0.29 **year** 0.29 **investor** 0.29 **now** 0.29 **technolog** 0.26 **market** 0.26 **revenu** 0.25

[9]:
```
# regular sentiment score using get_sentiment() function and method of your␣
↪choice
# please note that different methods may have different scales
syuzhet_vector <- get_sentiment(text, method="syuzhet")
# see the first row of the vector
head(syuzhet_vector)
# see summary statistics of the vector
summary(syuzhet_vector)
```

1. -0.6 2. 0 3. -0.5 4. 0 5. 0 6. -1.25

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
-1.9000  0.0000  0.0000  0.3065  0.3750  3.2000 
```

[10]:
```
# bing
bing_vector <- get_sentiment(text, method="bing")
head(bing_vector)
summary(bing_vector)
#affin
afinn_vector <- get_sentiment(text, method="afinn")
head(afinn_vector)
summary(afinn_vector)
```

1. 0 2. 0 3. 0 4. 0 5. 0 6. -2

```
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max. 
-5.000000  0.000000  0.000000 -0.009346  0.000000  2.000000 
```

1. -3 2. 0 3. -2 4. 0 5. 0 6. -3

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 -6.000   0.000   0.000   0.514   0.000   8.000 
```

[11]:

```r
#compare the first row of each vector using sign function
rbind(
  sign(head(syuzhet_vector)),
  sign(head(bing_vector)),
  sign(head(afinn_vector))
)
```

```
                          -1   0   -1   0   0   -1
A matrix: 3 × 6 of type dbl  0   0    0   0   0   -1
                          -1   0   -1   0   0   -1
```

[12]:

```r
# run nrc sentiment analysis to return data frame with each row classified as one of the following
# emotions, rather than a score:
# anger, anticipation, disgust, fear, joy, sadness, surprise, trust
# It also counts the number of positive and negative emotions found in each row
d<-get_nrc_sentiment(text)
# head(d,10) - to see top 10 lines of the get_nrc_sentiment dataframe
head (d,10)
```

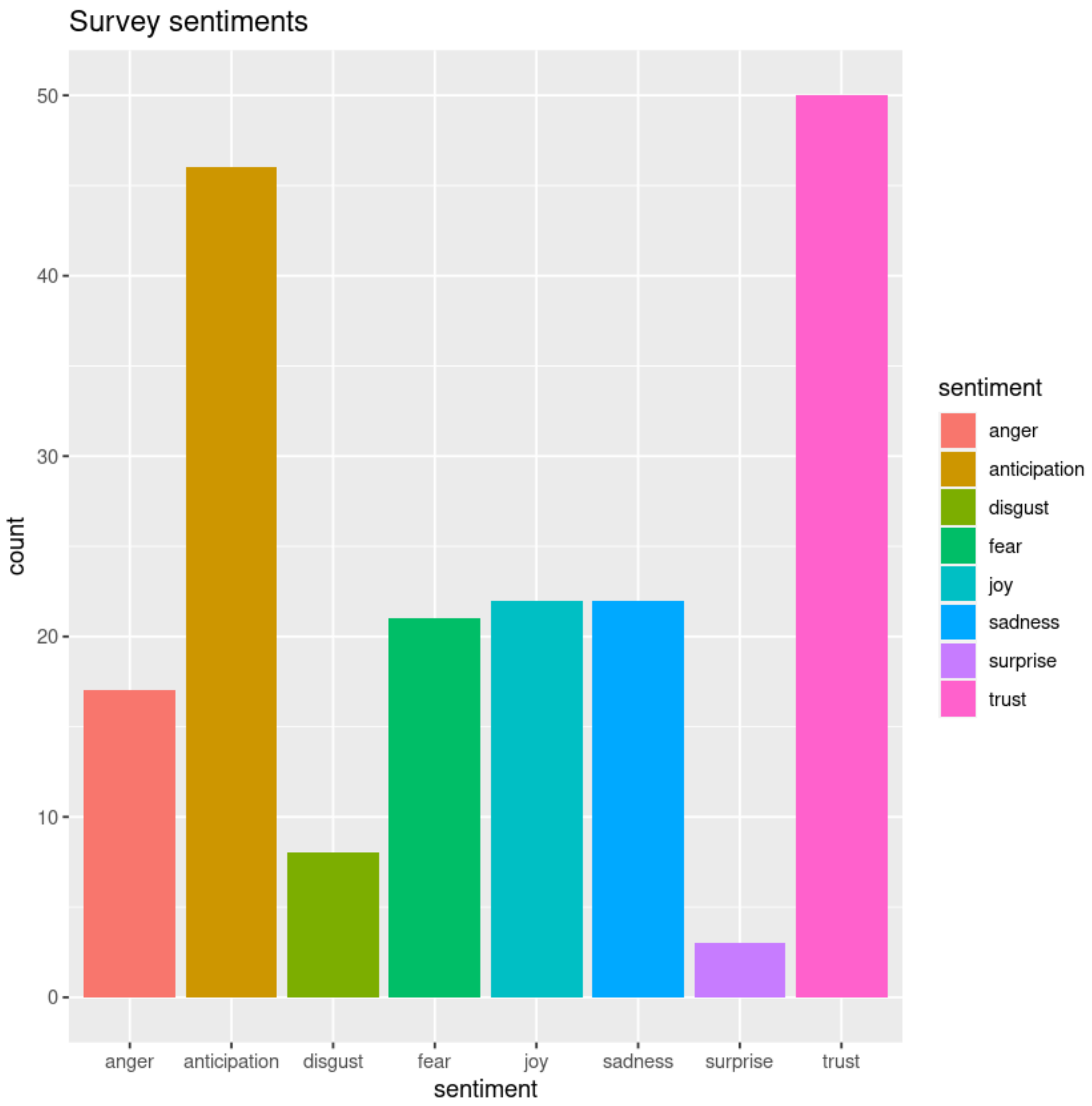
```
Warning message:
"`filter_()` is deprecated as of dplyr 0.7.0.
Please use `filter()` instead.
See vignette('programming') for more help
This warning is displayed once every 8 hours.
Call `lifecycle::last_warnings()` to see where this warning was

generated."
Warning message:
"`group_by_()` is deprecated as of dplyr 0.7.0.
Please use `group_by()` instead.
See vignette('programming') for more help
This warning is displayed once every 8 hours.
Call `lifecycle::last_warnings()` to see where this warning was

generated."
Warning message:
"`data_frame()` is deprecated as of tibble 1.1.0.
Please use `tibble()` instead.
This warning is displayed once every 8 hours.
Call `lifecycle::last_warnings()` to see where this warning was

generated."
```

A data.frame: 10 × 10

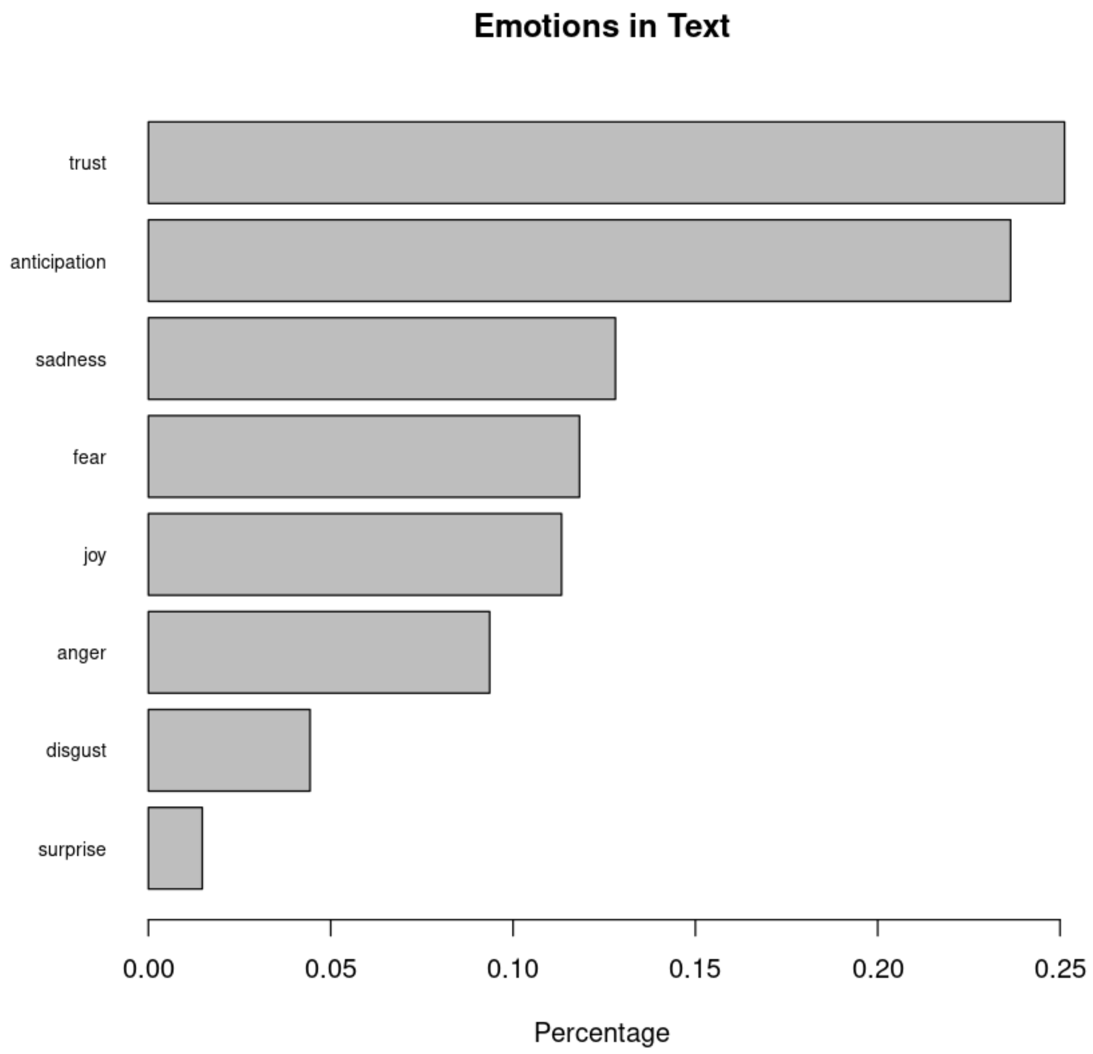
	anger	anticipation	disgust	fear	joy	sadness	surprise	trust	nega
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<db
1	2	2	1	3	1	4	0	1	4
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	2	1	0	4	2	1	0	2	4
7	0	0	0	0	0	0	0	0	0
8	1	2	0	2	0	1	0	1	3
9	0	0	0	0	0	0	0	0	0
10	0	1	0	0	1	0	0	3	0

[13]:

```
#transpose
td<-data.frame(t(d))
#The function rowSums computes column sums across rows for each level of a␣
 ↪grouping variable.
td_new <- data.frame(rowSums(td[2:107]))
#Transformation and cleaning
names(td_new)[1] <- "count"
td_new <- cbind("sentiment" = rownames(td_new), td_new)
rownames(td_new) <- NULL
td_new2<-td_new[1:8,]
#Plot One - count of words associated with each sentiment
quickplot(sentiment, data=td_new2, weight=count, geom="bar", fill=sentiment,␣
 ↪ylab="count")+ggtitle("Survey sentiments")
```



[14]:
```
#Plot two - count of words associated with each sentiment, expressed as a percentage
barplot(
  sort(colSums(prop.table(d[, 1:8]))),
  horiz = TRUE,
  cex.names = 0.7,
  las = 1,
  main = "Emotions in Text", xlab="Percentage"
)
```


Emotions in Text
trust
anticipation
sadness
fear
joy
anger
disgust
surprise
0.00
0.05
0.10
0.15
0.20
0.25
Percentage

```
[ ]:
```

```
[ ]:
```