Brian-Kurniawan-assignment-1.R

briankurniawan

2021-02-10

```
#BRIAN KURNIAWAN
#MSBA 1 - DAT 5317
#02/10/2021
#Business Insight Report

#Selecting Top 5 Technology Companies by Market Cap
#With the transcript call from the Q4 2020 call earning, we will analyze and identify any
#business insight from each company.
#The list of the Companies Apple, Amazon, Google(Alphabet), Facebook

#importing PDF Files
#install.packages("pdftools")
#install.packages("wordcloud")
#install.packages(IRdisplay)
library(pdftools)
```

```
## Using poppler version 0.73.0
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.3
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
library(stringr)
library(tidytext)
library(tm)
```

```
## Loading required package: NLP

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

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

```
library(scales)
```

```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:purrr':
##
##     discard
```

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(ggplot2)
library(wordcloud)
```

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

```
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(plotly)
```

```
##
## Attaching package: 'plotly'
```

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

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```r
library(IRdisplay)


setwd("/Users/briankurniawan/Documents/R/NLP/Assignment 1/pdf")
nm <- list.files(path= "/Users/briankurniawan/Documents/R/NLP/Assignment 1/pdf")

earning_report <- do.call(rbind, lapply(nm, function(x) pdf_text(x)))
```

```
## Warning in (function (..., deparse.level = 1) : number of columns of result is
## not a multiple of vector length (arg 1)
```

```r
#load pdf file
apple_report <- pdf_text("/Users/briankurniawan/Documents/R/NLP/Assignment 1/pdf/APPLE Q1 2021 earning.p
        readr::read_lines()

#first step of cleaning is to remove the header and footer of the website to remove
apple_report_points <- apple_report[18:897] %>%
          strsplit(split = " ")


#create corpus
apple_report_corpus = Corpus(VectorSource(apple_report_points))

#clean corpus
#change all character to lowercase, to reduce the double count on capitalize word
apple_report_corpus = tm_map(apple_report_corpus, tolower)
```

```
## Warning in tm_map.SimpleCorpus(apple_report_corpus, tolower): transformation
## drops documents
```

```r
#remove all punctuations in order to get word by word
apple_report_corpus = tm_map(apple_report_corpus, removePunctuation)
```

```
## Warning in tm_map.SimpleCorpus(apple_report_corpus, removePunctuation):
## transformation drops documents
```

```r
#data(stop_words)
#cleaning stopwords
clean_apple = tm_map(apple_report_corpus, removeWords, stopwords("en"))
```

```
## Warning in tm_map.SimpleCorpus(apple_report_corpus, removeWords,
## stopwords("en")): transformation drops documents
```

```r
#change the to matrix forms
dtm_apple = DocumentTermMatrix(clean_apple)
dtm_apple = as.matrix(dtm_apple)
dtm_apple = t(dtm_apple) #transpose X and Y axis

#Sums each row
dtm_apple_summary = rowSums(dtm_apple)
dtm_apple_summary = sort(dtm_apple_summary, decreasing = TRUE)

wordcloud(head(names(dtm_apple_summary), 10), head(dtm_apple_summary, 10), scale = c(2,1))
```

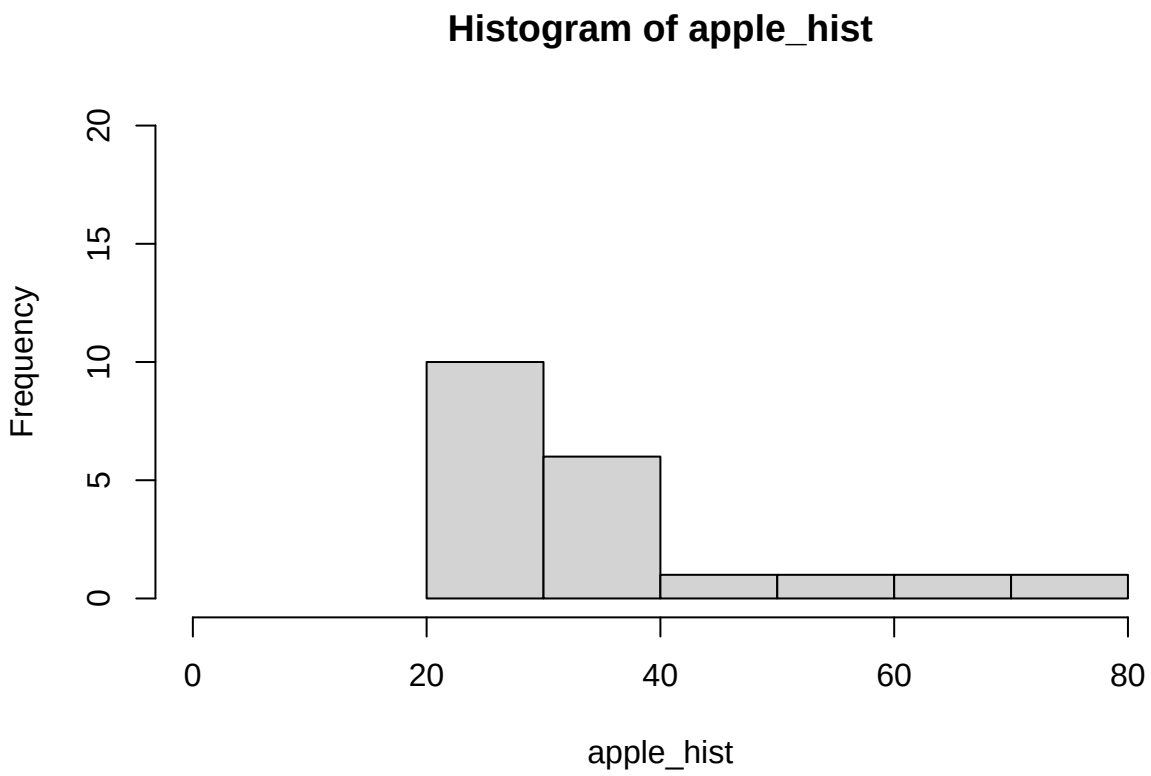

```
#identifying the 20 most frequent words
apple_hist <- data.frame(head(dtm_apple_summary, n = 20))

apple_hist <- data.matrix(apple_hist)
apple_hist
```

```
##          head.dtm_apple_summary..n...20.
## quarter                               78
## apple                                 62
## new                                   54
## services                              46
## year                                  38
## thank                                 37
## well                                  37
## think                                 37
## iphone                                36
## call                                  34
## billion                               30
## tim                                   29
## strong                                28
## growth                                28
## weve                                  28
## 2021                                  27
## can                                   27
## earnings                              26
## look                                  26
```

```
## question                                26
```

```
hist(apple_hist, xlim = c(0,80), ylim= c(0,20))
```




```
######################
##AMAZON

#load pdf file
amazon_report <- pdf_text("/Users/briankurniawan/Documents/R/NLP/Assignment 1/pdf/Amazon Q4 2020 earning
  readr::read_lines()

#first step of cleaning is to remove the header and footer of the website to remove
amazon_report_points <- amazon_report[17:640] %>%
  strsplit(split = " ")
#create corpus
amazon_report_corpus = Corpus(VectorSource(amazon_report_points))

#clean corpus
#change all character to lowercase, to reduce the double count on capitalize word
amazon_report_corpus = tm_map(amazon_report_corpus, tolower)
```

```
## Warning in tm_map.SimpleCorpus(amazon_report_corpus, tolower): transformation
## drops documents
```

```r
#remove all punctions in order to get word by word
amazon_report_corpus = tm_map(amazon_report_corpus, removePunctuation)
```

```
## Warning in tm_map.SimpleCorpus(amazon_report_corpus, removePunctuation):
## transformation drops documents
```

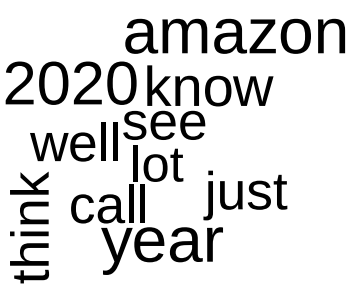
```r
#cleaning stopwords
clean_amazon = tm_map(amazon_report_corpus, removeWords, stopwords("en"))
```

```
## Warning in tm_map.SimpleCorpus(amazon_report_corpus, removeWords,
## stopwords("en")): transformation drops documents
```

```r
dtm_amazon = DocumentTermMatrix(clean_amazon)
dtm_amazon = as.matrix(dtm_amazon)
dtm_amazon = t(dtm_amazon)

dtm_amazon_summary = rowSums(dtm_amazon)
dtm_amazon_summary = sort(dtm_amazon_summary, decreasing = TRUE)

##wordcloud with the top frequent words
wordcloud(head(names(dtm_amazon_summary), 10), head(dtm_amazon_summary, 10), scale = c(2,1))
```

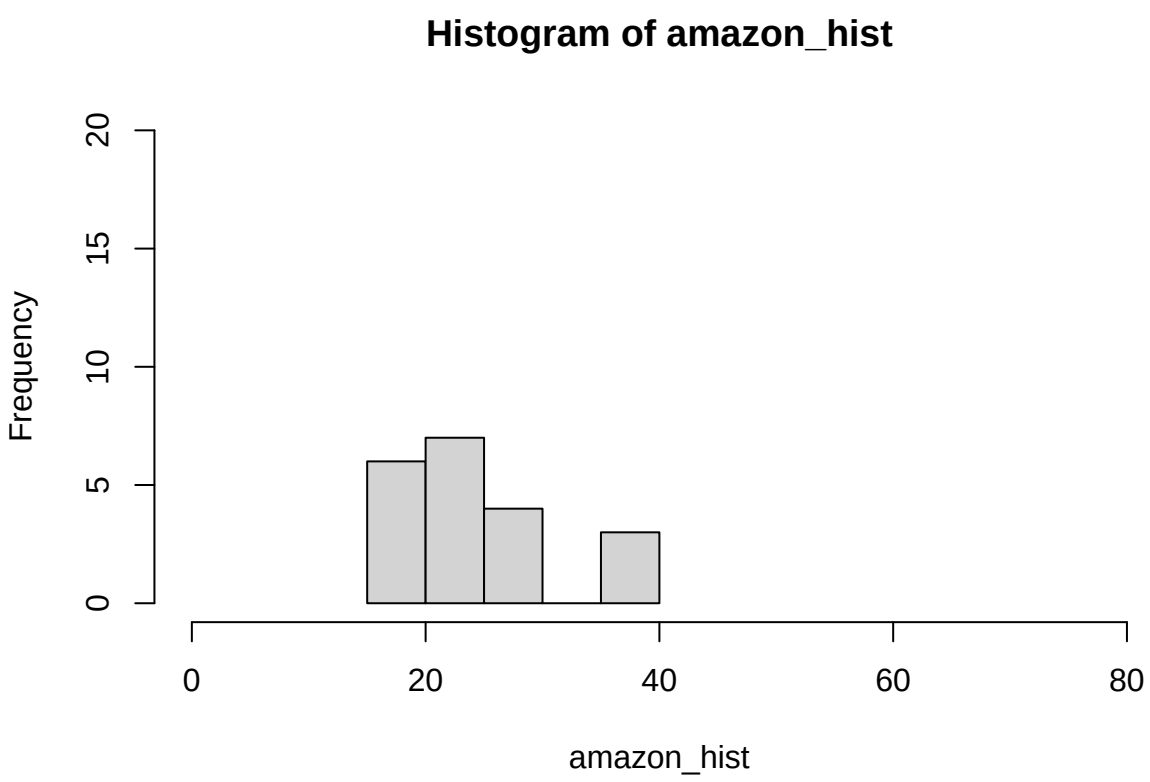


```r
#Histogram
amazon_hist <- data.frame(head(dtm_amazon_summary, n = 20))

amazon_hist <- data.matrix(amazon_hist)
amazon_hist
```

```
##          head.dtm_amazon_summary..n...20.
## year                                   40
## amazon                                 39
## 2020                                   38
## know                                   30
## call                                   27
## well                                   27
## just                                   27
## see                                    25
## think                                  25
## lot                                    24
## quarter                                21
## also                                   21
## question                               21
## going                                  21
## aws                                    20
## prime                                  20
## new                                    19
## saw                                    19
## brian                                  18
## things                                 18
```

```r
hist(amazon_hist, xlim = c(0,80), ylim= c(0,20))
```




```
####################
####Facebook

facebook_report <- pdf_text("/Users/briankurniawan/Documents/R/NLP/Assignment 1/pdf/Facebook Q4 report.p
  readr::read_lines()

#first step of cleaning is to remove the header and footer of the website to remove
facebook_report_points <- facebook_report[17:1050] %>%
  strsplit(split = " ")
#create corpus
facebook_report_corpus = Corpus(VectorSource(facebook_report_points))

#clean corpus
#change all character to lowercase, to reduce the double count on capitalize word
facebook_report_corpus = tm_map(facebook_report_corpus, tolower)
```

```
## Warning in tm_map.SimpleCorpus(facebook_report_corpus, tolower): transformation
## drops documents
```

```
#remove all punctions in order to get word by word
facebook_report_corpus = tm_map(facebook_report_corpus, removePunctuation)
```

```
## Warning in tm_map.SimpleCorpus(facebook_report_corpus, removePunctuation):
## transformation drops documents
```

```
data(stop_words)
#cleaning stopwords
clean_facebook = tm_map(facebook_report_corpus, removeWords, stopwords("en"))
```

```
## Warning in tm_map.SimpleCorpus(facebook_report_corpus, removeWords,
## stopwords("en")): transformation drops documents
```

```
dtm_facebook = DocumentTermMatrix(clean_facebook)
dtm_facebook = as.matrix(dtm_facebook)
dtm_facebook = t(dtm_facebook) #transpose X and Y axis

dtm_facebook_summary = rowSums(dtm_facebook)
dtm_facebook_summary = sort(dtm_facebook_summary, decreasing = TRUE)
head(dtm_facebook_summary, n = 20)
```

```
##     people      think businesses   facebook        now       year        can
##         72         66         65         49         46         44         43
##       call       2020      going   earnings       well       just     motley
##         40         34         34         33         33         33         32
##        one       weve       will       like       fool   services
##         32         32         31         31         31         31
```

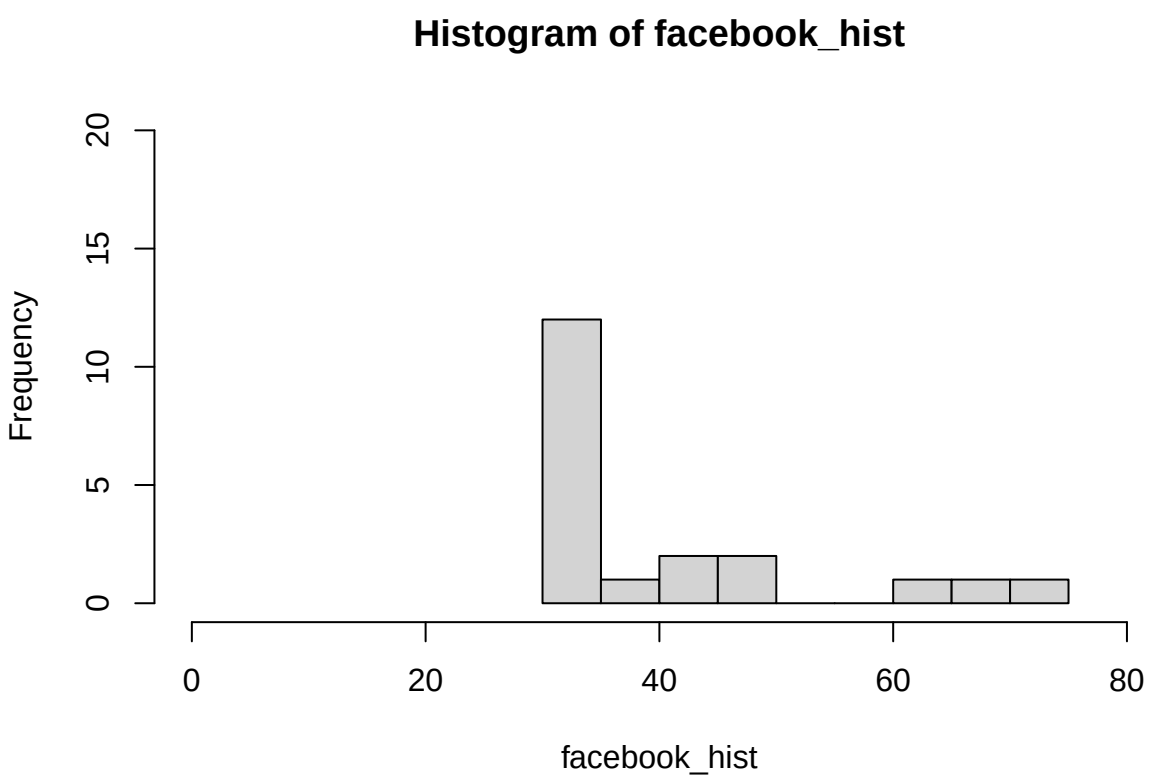
```
##wordcloud with the top frequent words
wordcloud(head(names(dtm_facebook_summary), 10), head(dtm_facebook_summary, 10), scale = c(2,1))
```

```r
#Histogram
facebook_hist <- data.frame(head(dtm_facebook_summary, n = 20))

facebook_hist <- data.matrix(facebook_hist)
facebook_hist
```

```
##            head.dtm_facebook_summary..n...20.
## people                                     72
## think                                      66
## businesses                                 65
## facebook                                   49
## now                                        46
## year                                       44
## can                                        43
## call                                       40
## 2020                                       34
## going                                      34
## earnings                                   33
## well                                       33
## just                                       33
## motley                                     32
## one                                        32
## weve                                       32
## will                                       31
## like                                       31
## fool                                       31
## services                                   31
```

```r
hist(facebook_hist, xlim = c(0,80), ylim= c(0,20))
```




```
################
#ALPHABET(GOOGLE)

google_report <- pdf_text("/Users/briankurniawan/Documents/R/NLP/Assignment 1/pdf/Alphabet Q4 earning.p
  readr::read_lines()

#first step of cleaning is to remove the header and footer of the website to remove
google_report_points <- google_report[17:807] %>%
  strsplit(split = " ")
#create corpus
google_report_corpus = Corpus(VectorSource(google_report_points))

#clean corpus
#change all character to lowercase, to reduce the double count on capitalize word
google_report_corpus = tm_map(google_report_corpus, tolower)
```

```
## Warning in tm_map.SimpleCorpus(google_report_corpus, tolower): transformation
## drops documents
```

```
#remove all punctions in order to get word by word
google_report_corpus = tm_map(google_report_corpus, removePunctuation)
```

```
## Warning in tm_map.SimpleCorpus(google_report_corpus, removePunctuation):
## transformation drops documents
```

```
#cleaning stopwords
clean_google = tm_map(google_report_corpus, removeWords, stopwords("en"))
```

```
## Warning in tm_map.SimpleCorpus(google_report_corpus, removeWords,
## stopwords("en")): transformation drops documents
```

```
dtm_google = DocumentTermMatrix(clean_google)
dtm_google = as.matrix(dtm_google)
dtm_google = t(dtm_google)

dtm_google_summary = rowSums(dtm_google)
dtm_google_summary = sort(dtm_google_summary, decreasing = TRUE)
head(dtm_google_summary, n =20)
```

```
##   google  youtube     2020     call  quarter  billion     year    thank
##       58       55       41       37       36       34       32       31
##   really business     well alphabet    cloud      now     will   across
##       31       30       30       29       29       28       27       27
##    think earnings  philipp   growth
##       26       25       24       24
```

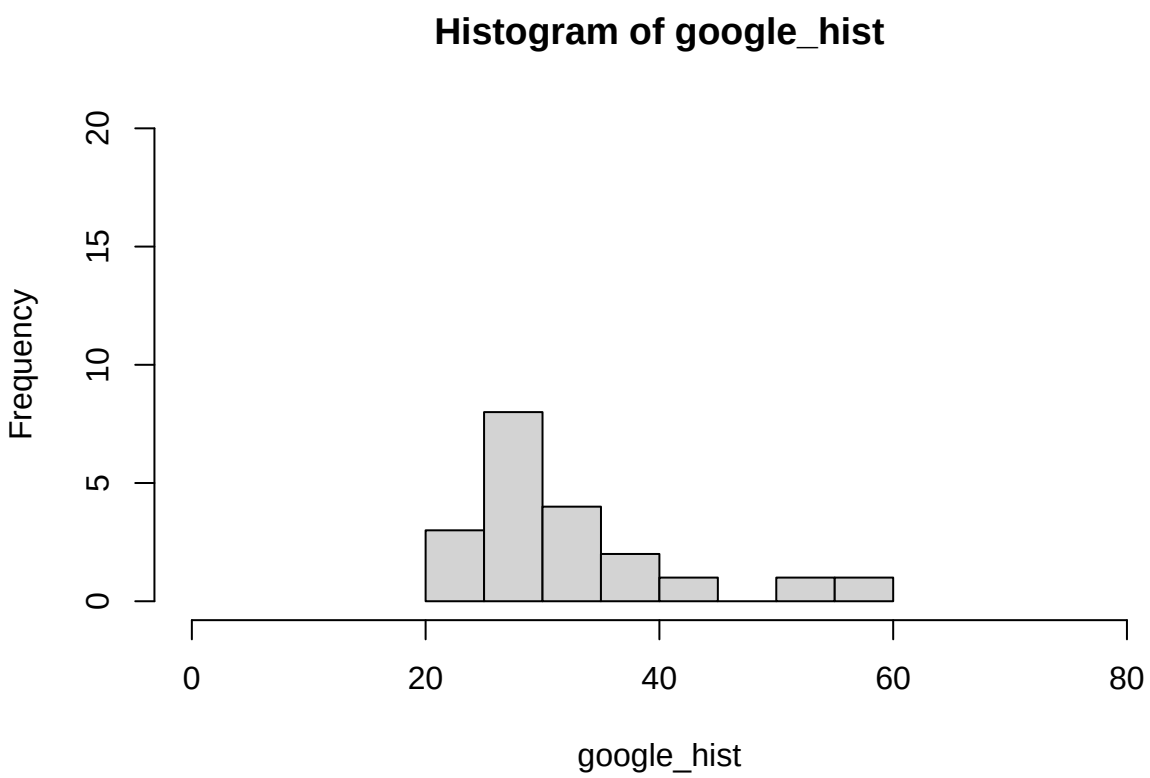
```
##wordcloud with the top frequent words
wordcloud(head(names(dtm_google_summary), 10), head(dtm_google_summary, 10), scale = c(2,1))
```

```r
#Histogram
google_hist <- data.frame(head(dtm_google_summary, n = 20))

google_hist <- data.matrix(google_hist)
google_hist
```

```
##          head.dtm_google_summary..n...20.
## google                                 58
## youtube                                55
## 2020                                   41
## call                                   37
## quarter                                36
## billion                                34
## year                                   32
## thank                                  31
## really                                 31
## business                               30
## well                                   30
## alphabet                               29
## cloud                                  29
## now                                    28
## will                                   27
## across                                 27
## think                                  26
## earnings                               25
## philipp                                24
## growth                                 24
```

```r
hist(google_hist, xlim = c(0,80), ylim= c(0,20))
```


Histogram of google_hist
Frequency
0
5
10
15
20
0
20
40
60
80
google_hist