Political Text Analysis

Huang Lin, Chun

2023-05-26

```
Sys.setlocale("LC_ALL", 'en_GB.UTF-8')
Sys.setenv(LANG = "en_GB.UTF-8")
getwd()
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(readtext)
library(quanteda)
```

```
## Package version: 3.3.0
## Unicode version: 14.0
## ICU version: 71.1
## Parallel computing: 8 of 8 threads used.
## See https://quanteda.io for tutorials and examples.
##
## Attaching package: 'quanteda'
##
## The following objects are masked from 'package:readtext':
##
##     docnames, docvars, texts
```

```
library(quanteda.textstats)
library(quanteda.textplots)
```

Check the files in our target folder.

```
list.files(path = './transcripts')
```

Import each file into R

```
earn.call <- readtext("./transcripts/*.txt",
                      docvarsfrom = "filenames",
                      dvsep = "_",
                      docvarnames = c("name", "year", "quarter", "date"))
earn.call
```

```
## readtext object consisting of 42 documents and 4 docvars.
## # A data frame: 42 x 6
##   doc_id                              text              name   year quarter date
##   <chr>                               <chr>             <chr> <int> <chr>   <chr>
## 1 transcript_2013_Q-1_2013-01-24.txt "\"\nGood day,\"~ tran~  2013 Q-1     2013~
## 2 transcript_2013_Q-2_2013-04-23.txt "\"\nPlease st\"~ tran~  2013 Q-2     2013~
## 3 transcript_2013_Q-3_2013-07-23.txt "\"\nPlease st\"~ tran~  2013 Q-3     2013~
## 4 transcript_2013_Q-4_2013-10-29.txt "\"\nGood day,\"~ tran~  2013 Q-4     2013~
## 5 transcript_2014_Q-1_2014-01-28.txt "\"\nGood day,\"~ tran~  2014 Q-1     2014~
## 6 transcript_2014_Q-2_2014-04-24.txt "\"\nGood day,\"~ tran~  2014 Q-2     2014~
## # i 36 more rows
```

Turn the dataframe into Corpus

```
earn.call.corpus <- corpus(earn.call)
summary(earn.call.corpus, n = 10)
```

Tokenize the data

```
earn.call_tokens <- tokens(earn.call.corpus,
                      remove_punct = TRUE,
                      remove_symbols = TRUE,
                      remove_numbers = TRUE,
                      remove_separators = TRUE)
head(earn.call_tokens, n = 3)
```

```
## Tokens consisting of 3 documents and 4 docvars.
## transcript_2013_Q-1_2013-01-24.txt :
##  [1] "Good"         "day"          "everyone"     "and"          "welcome"
##  [6] "to"           "this"         "Apple"        "Incorporated" "First"
## [11] "Quarter"      "Fiscal"
## [ ... and 8,683 more ]
##
## transcript_2013_Q-2_2013-04-23.txt :
```

```
##  [1] "Please"    "standby"   "we"        "are"       "about"     "to"
##  [7] "begin"     "Good"      "day"       "ladies"    "and"       "gentlemen"
## [ ... and 8,273 more ]
##
## transcript_2013_Q-3_2013-07-23.txt :
##  [1] "Please"   "standby"  "we"       "are"      "about"    "to"
##  [7] "begin"    "Good"     "day"      "everyone" "and"      "welcome"
## [ ... and 7,389 more ]
```

Clean up stopwords and lemmatization

```
earn.call_tokens <- tokens_remove(earn.call_tokens,
                            stopwords(language = "en", source = "marimo"))
earn.call_tokens <- tokens_wordstem(earn.call_tokens, language = 'en')
head(earn.call_tokens, n = 3)
```

Turn the big corpus into Document Feature Matrix

```
earn.call.dfm <- dfm(earn.call_tokens)
earn.call.dfm
```

```
# Top features
topfeatures(earn.call.dfm, 20)
```

```
##  quarter    iphon    apple      new    think  billion   revenu    thank
##     2907     1915     1621     1431     1417     1380     1303     1301
##  product   servic   custom   market question      can     ipad      see
##     1291     1257     1145     1089     1040     1029      971      903
##       go     also   growth     just
##      878      877      869      841
```

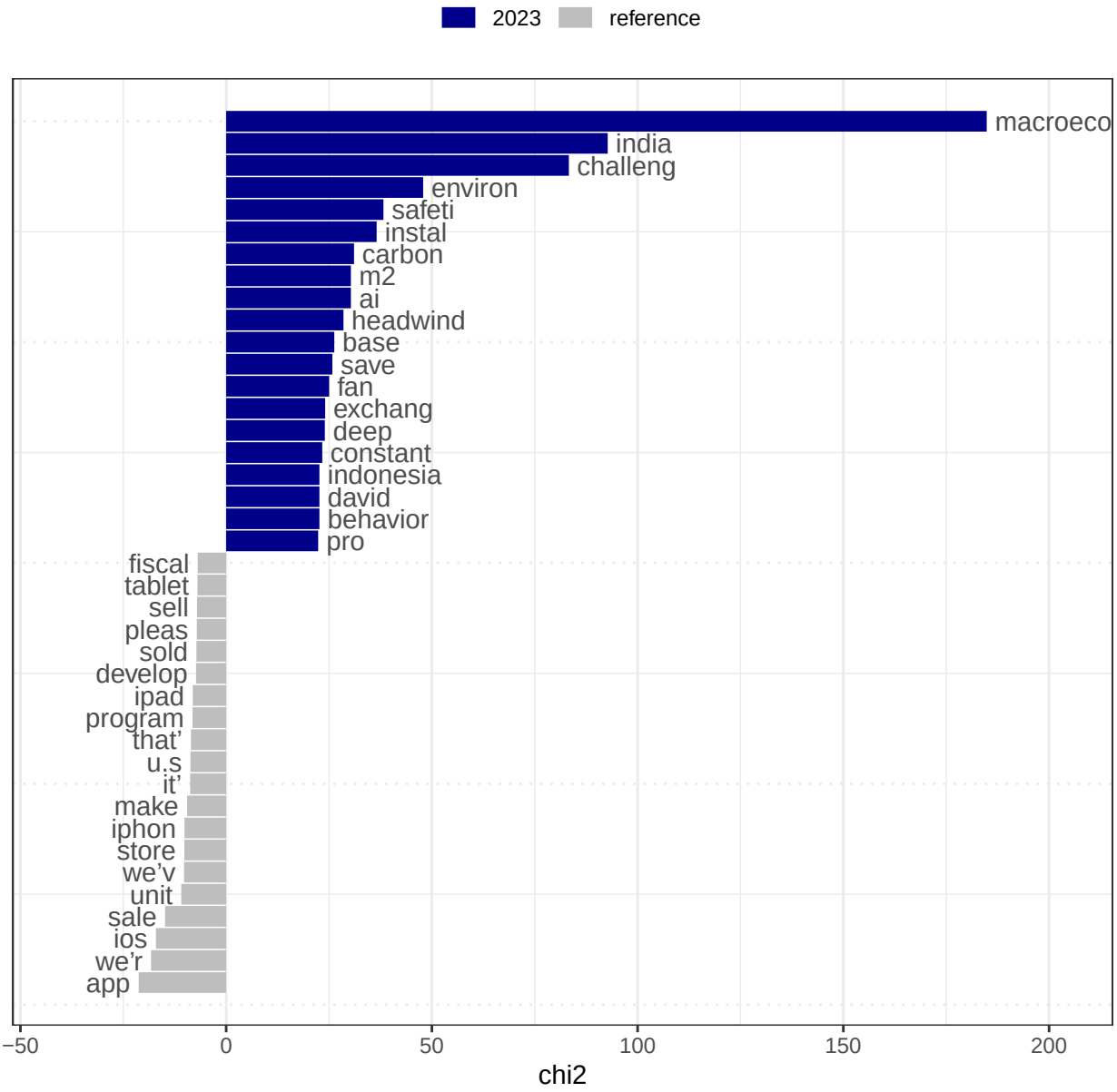
Trimming the DFM

```
earn.call.dfm <- dfm_trim(earn.call.dfm, min_termfreq = 10, min_docfreq = 1)
topfeatures(earn.call.dfm, 20)
```

Keyness (Take 2023 and 2022 as examples)

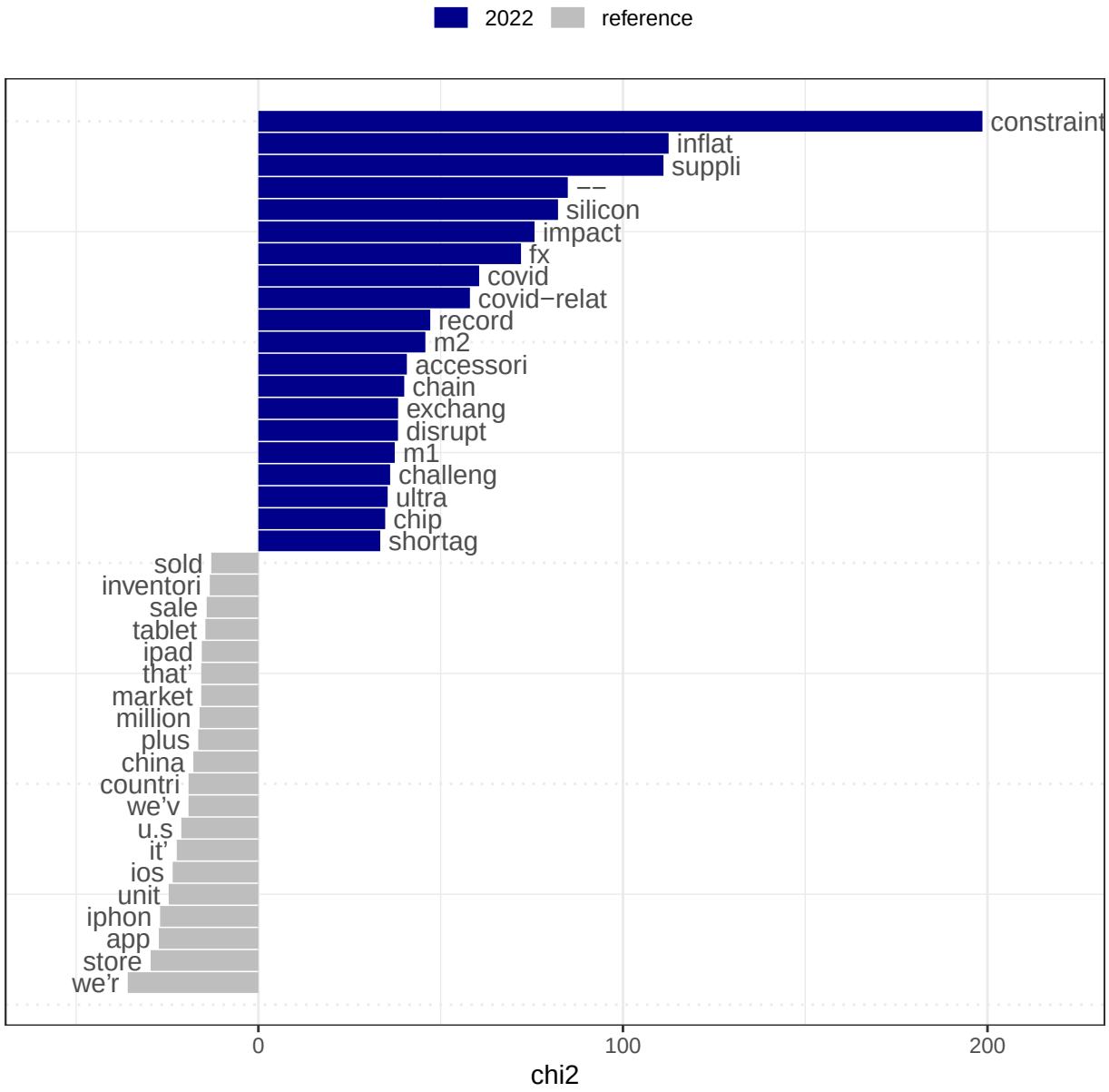
keyness of 2023

```
year_data <- dfm_group(earn.call.dfm, group = year)
keyness.2023 <- textstat_keyness(year_data, target = '2023')
keyness.2023 <- keyness.2023[ which(keyness.2023$p<=0.05), ]
textplot_keyness(keyness.2023)+ theme(legend.position = "top")
```



keyness of 2022

```
year_data <- dfm_group(earn.call.dfm, group = year)
keyness.2022 <- textstat_keyness(year_data, target = '2022')
keyness.2022 <- keyness.2022[ which(keyness.2022$p<=0.05), ]
textplot_keyness(keyness.2022) + theme(legend.position = "top")
```



Event Study

Use CAPM to Calculate the Abnormal Return

```
variables = docvars(earn.call.corpus)
variables = variables[, -1]
colnames(variables)
```

```
## [1] "year"    "quarter" "date"
```

```r
# The market is expected to respond after the date of earnings call
# (Because often, the earnings calls are held after the close time of stock market.)
variables$date = as.Date(variables$date) + 1
variables
```

```
##    year quarter       date
## 1  2013     Q-1 2013-01-25
## 2  2013     Q-2 2013-04-24
## 3  2013     Q-3 2013-07-24
## 4  2013     Q-4 2013-10-30
## 5  2014     Q-1 2014-01-29
## 6  2014     Q-2 2014-04-25
## 7  2014     Q-3 2014-07-24
## 8  2014     Q-4 2014-10-22
## 9  2015     Q-1 2015-01-29
## 10 2015     Q-2 2015-04-29
## 11 2015     Q-3 2015-07-23
## 12 2015     Q-4 2015-10-29
## 13 2016     Q-1 2016-01-28
## 14 2016     Q-2 2016-04-28
## 15 2016     Q-3 2016-07-28
## 16 2016     Q-4 2016-10-27
## 17 2017     Q-1 2017-02-02
## 18 2017     Q-2 2017-05-04
## 19 2017     Q-3 2017-08-03
## 20 2017     Q-4 2017-11-04
## 21 2018     Q-1 2018-02-03
## 22 2018     Q-2 2018-05-03
## 23 2018     Q-3 2018-08-02
## 24 2018     Q-4 2018-11-03
## 25 2019     Q-1 2019-01-31
## 26 2019     Q-2 2019-05-01
## 27 2019     Q-3 2019-08-01
## 28 2019     Q-4 2019-11-01
## 29 2020     Q-1 2020-01-30
## 30 2020     Q-2 2020-05-02
## 31 2020     Q-3 2020-07-31
## 32 2020     Q-4 2020-10-30
## 33 2021     Q-1 2021-01-28
## 34 2021     Q-2 2021-04-29
## 35 2021     Q-3 2021-07-28
## 36 2021     Q-4 2021-10-29
## 37 2022     Q-1 2022-01-28
## 38 2022     Q-2 2022-04-29
## 39 2022     Q-3 2022-07-29
## 40 2022     Q-4 2022-10-28
## 41 2023     Q-1 2023-02-03
## 42 2023     Q-2 2023-05-05
```

```r
library('zoo')
library('xts')
library('TTR')
library('quantmod')
```

```r
library('dplyr')
```

```r
# Get the stock price of Apple Inc.
getSymbols("AAPL", from = '2013-01-01', to = '2023-04-28')
```

```
## [1] "AAPL"
```

```r
# We only need the close price
AAPL = AAPL$AAPL.Close

# Calculate the daily return
returns <- (diff(AAPL) / lag(AAPL, 1))*100
colnames(returns) <- c('AAPL.Ret')

# Import the data set of RM-RF data and RF(risk free return)
# Data Source:
# http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html#Research
MKT <- read.csv('Market_return_data/F-F_Research_Data_Factors_daily2.csv')
MKT$X = as.Date(MKT$X, format = "%Y%m%d")

# started by 2013 because chose only earnings call transcripts from 2013 Q1
MKT = MKT[MKT$X >= as.Date('2013-01-01'),]
MKT = MKT[MKT$X <= as.Date('2023-04-28'),]

# Because the last three rows are NA, so delete
last_three = (nrow(MKT) - 2):nrow(MKT)
MKT = MKT[-last_three,]

# Combine the two dataframe
AAPL_MKT = bind_cols(MKT,returns)

# Calculate the excess return of AAPL (Return - Risk Free)
AAPL_MKT$excess.return = AAPL_MKT$AAPL.Ret - AAPL_MKT$RF

head(AAPL_MKT, n = 5)
```

```
##                X Mkt.RF   SMB   HML RF   AAPL.Ret excess.return
## 22882 2013-01-02   2.62  0.14  0.38  0         NA            NA
## 22883 2013-01-03  -0.14  0.11  0.04  0 -1.2622234    -1.2622234
## 22884 2013-01-04   0.55  0.12  0.36  0 -2.7854637    -2.7854637
## 22885 2013-01-07  -0.31 -0.10 -0.35  0 -0.5882336    -0.5882336
## 22886 2013-01-08  -0.27  0.05  0.00  0  0.2691287     0.2691287
```

```r
# Run the regression of AAPL excess return on Market excess return
# to get the beta of Apple stock
model <- lm(AAPL_MKT$excess.return ~ AAPL_MKT$Mkt.RF)
beta <- coef(model)[2]
beta
```

```
## AAPL_MKT$Mkt.RF
##        1.134838
```

```r
# According to CAMP, the normal (expected) return is given by RF + beta*(RM - RF)
AAPL_MKT$normal.return = AAPL_MKT$RF + beta * AAPL_MKT$Mkt.RF
AAPL_MKT$abnormal.return = AAPL_MKT$AAPL.Ret - AAPL_MKT$normal.return

# Generate a dummy to check whether the day is after any earnings call
AAPL_MKT$After.call = ifelse(AAPL_MKT$X %in% variables$date, 1, 0)

# Run the regression of abnormal return on the dummy variable
summary(lm(AAPL_MKT$abnormal.return ~ AAPL_MKT$After.call))
```

```
##
## Call:
## lm(formula = AAPL_MKT$abnormal.return ~ AAPL_MKT$After.call)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.4874  -0.6455  -0.0275   0.6380   9.0111
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.02943    0.02551   1.154 0.248675
## AAPL_MKT$After.call  0.73508    0.21372   3.439 0.000592 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.291 on 2595 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.004538,   Adjusted R-squared:  0.004154
## F-statistic: 11.83 on 1 and 2595 DF,  p-value: 0.0005923
```