WARWICK BUSINESS SCHOOL

Masters Programmes

Assignment Cover Sheet

Submitted by: <2092410>

Date Sent: 30/06/2021

Module Title: Text Analytics

Module Code: IB9CW0

Date/Year of Module: 2021

Submission Deadline: 30/06/2021

Word Count: 2609 words

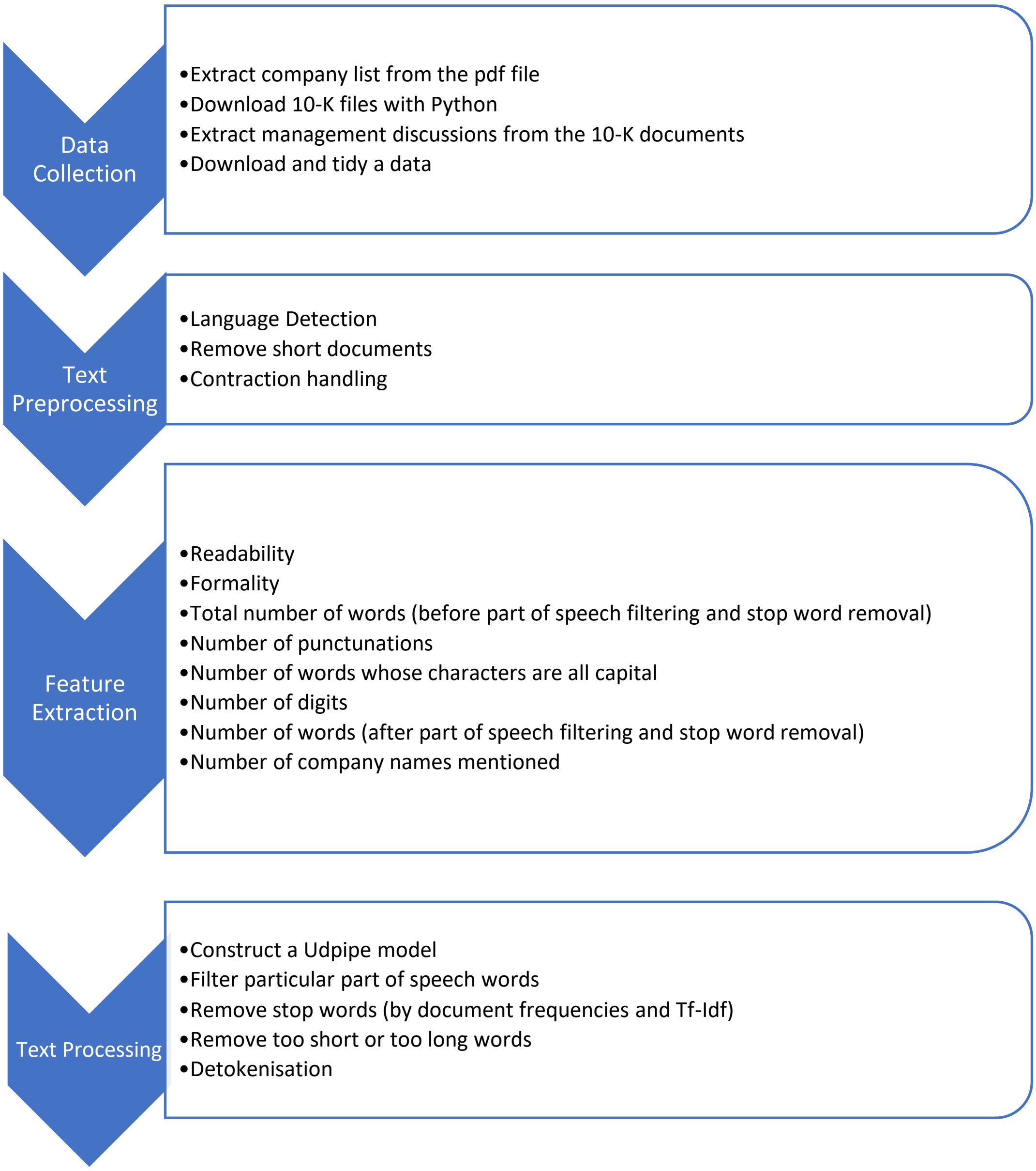
Number of Pages: 59

"I declare that this work is entirely my own in accordance with the University's [Regulation 11](#) and the WBS guidelines on plagiarism and collusion. All external references and sources are clearly acknowledged and identified within the contents.

No substantial part(s) of the work submitted here has also been submitted by me in other assessments for accredited courses of study, and I acknowledge that if this has been done it may result in us being reported for self-plagiarism and an appropriate reduction in marks may be made when marking this piece of work."

Contents

Text Tidying Workflow



Part A: Construction of Corpus – Fetching 10-Q and 10-K forms from EDGAR

Load Required packages

```
library(tidytext)
library(tidyverse)
library(stringr)
library(readxl)
library(lubridate)
library(edgar)
library(BatchGetSymbols)
library(pdftools)
library(SentimentAnalysis)
library(stm)
library(cld2)
library(textclean)
library(udpipe)
library(ggpubr)
library(janitor)
```

Data Collection

Build a company data frame from the assignment description

```
# create the data frame
project_description = pdf_text("../../individual_assignment_de
scription_ib9cw0_2021.pdf")
company_list = unlist(str_split(project_description, "[\\r\\n]
+"))
company_list = str_split_fixed(str_trim(company_list), "\\s
{2,}", 5) %>%
  as.data.frame()
company_list[company_list == ""] = NA
company_list = na.omit(company_list)

# first row to column names
library(janitor)
company_list = company_list %>%
  row_to_names(row_number(1))

# reset row index
row.names(company_list) = NULL
```

```
# clear the memory
rm(project_description)

# check the data frame
str(company_list)

# change CIK into numerical form
company_list$CIK = as.numeric(company_list$CIK)

# check the data frame
str(company_list)
```

Download all required datasets

Get a full list of CIK

```
CIK_list = company_list$CIK

write.csv(company_list, "company_list.csv")
```

Get management discussions

I used Python to download the required datasets because the R API was not working.

```
from sec_edgar_downloader import Downloader
import pandas as pd
company_list = pd.read_csv("company_list.csv")
company_cik_list = [str(i).zfill(10) for i in
list(company_list["CIK"])]
dl = Downloader()
for company in company_cik_list:
    dl.get("10-K", company, after="2010-01-01", before="2020-
12-31")
```

Extract Management Discussions from the filings

```
all_md_texts = data.frame()
i = 1
# all_files = data.frame()
all_ciks = list.files(path = "sec-edgar-filings/")
for (cik in all_ciks){
  print(i)
  cik_dir_path = paste0("sec-edgar-filings/", cik)
  all_filings_directories = list.files(path = paste0(cik_dir_p
ath, "/10-K"))
  for (filing in all_filings_directories){
    filing_path = paste0(cik_dir_path, "/10-K/", filing, "/ful
l-submission.txt")
    filing.text = readLines(filing_path)

    # Get filing dates
    filing.text.combined = paste(filing.text, collapse = "")
```

```r
    date.end = as.numeric(str_locate(filing.text.combined, "DATE AS OF CHANGE:"))[1]-1
    date.filed = ymd(str_sub(filing.text.combined, date.end-7, date.end))

    # Extract data from first <DOCUMENT> to </DOCUMENT>
    tryCatch({
      filing.text = filing.text[(grep("<DOCUMENT>", filing.text, ignore.case = TRUE)[1]):(grep("</DOCUMENT>", filing.text, ignore.case = TRUE)[1])]}, error = function(e) {
      filing.text = filing.text
      ## In case opening and closing DOCUMENT TAG not found, cosnider full web page
    })

    # See if 10-K is in XLBR or old text format
    if (any(grepl(pattern = "<xml>|<type>xml|<html>|10k.htm", filing.text, ignore.case = T))) {
      doc = XML::htmlParse(filing.text, asText = TRUE, useInternalNodes = TRUE, addFinalizer = FALSE)
      f.text = XML::xpathSApply(doc, "//text()[not(ancestor::script)][not(ancestor::style)][not(ancestor::noscript)][not(ancestor::form)]", XML::xmlValue)
      f.text = iconv(f.text, "latin1", "ASCII", sub = " ")
      ## Free up htmlParse document to avoid memory leakage, this calls C function
      #.Call('RS_XML_forceFreeDoc', doc, package= 'XML')
    } else {
      f.text = filing.text
    }

    # Preprocessing the filing text
    f.text = gsub("\\n|\\t|$", " ", f.text)
    f.text = gsub("^\\s{1,}", "", f.text)
    f.text = gsub(" s ", " ", f.text)
    # Check for empty Lines and delete it
    empty.lnumbers = grep("^\\s*$", f.text)
    if (length(empty.lnumbers) > 0) {
      f.text <- f.text[-empty.lnumbers]  ## Remove all lines only with space
    }

    # Get MD&A sections
    startline <- grep("^Item\\s{0,}7[^A]", f.text, ignore.case = TRUE)
    endline <- grep("^Item\\s{0,}7A", f.text, ignore.case = TRUE)
    # if dont have Item 7A, then take upto Item 8
    if (length(endline) == 0) {
```

```
        endline <- grep("^Item\\s{0,}8", f.text, ignore.case =
 TRUE)
    }
    md.dicusssion <- NA
    if (length(startline) != 0 && length(endline) != 0) {

        startline <- startline[length(startline)]
        endline <- endline[length(endline)] - 1

        md.dicusssion <- paste(f.text[startline:endline], coll
apse = " ")
        md.dicusssion <- gsub("\\s{2,}", " ", md.dicusssion)
        words.count <- stringr::str_count(md.dicusssion, patte
rn = "\\S+")
    }
    temp = data.frame(CIK = cik, date.filed = date.filed, md_t
ext = md.dicusssion)
    all_md_texts = all_md_texts %>%
      rbind(temp) %>%
      na.omit()
  }
  i = i + 1
}
rm(temp, all_ciks, all_filings_directories, cik, cik_dir_path,
 date.end, date.filed, doc, empty.lnumbers, endline, f.text, f
iling, filing_path, filing.text, filing.text.combined, md.dicu
sssion, startline, words.count, i)

all_md_texts$CIK = as.numeric(all_md_texts$CIK)

saveRDS(all_md_texts, "all_md_texts.rds")
```

Combine the two datasets

```
all_md_texts = readRDS("all_md_texts.rds")

data = company_list %>%
  select(CIK, Symbol, Security, `GICS Sub Industry`) %>%
  left_join(all_md_texts, by = c("CIK")) %>%
  rename(company.name = Security)

# remove observations without management discussions
data = data %>%
  na.omit()
row.names(data) = NULL

# clear the memory
rm(all_md_texts, company_list, CIK_list)

data$date_before_filing = data$date.filed - 7
data$date_after_filing = data$date.filed +3
```

get financial data

downloading

```r
returns_weekly = data.frame()
error_companies = data.frame()
for (i in 1:nrow(data)){
  temp = tryCatch({BatchGetSymbols::BatchGetSymbols(data$Symbol[i],
                                   freq.data = "weekly",
                                   first.date = data$date_before_filing[i],
                                   last.date = data$date_after_filing[i],
                                   type.return = "log")},
           error = function(e){
             data.frame(Symbol = data$Symbol[i],
                        date_before_filing = data$date_before_filing[i],
                        date_after_filing = data$date_after_filing[i])
           })
  if (class(temp) == "list"){
    temp = temp$df.tickers
    returns_weekly = returns_weekly %>%
      rbind(temp)
  }
  else{
    error_companies = error_companies %>%
      rbind(temp)
  }
}
rm(i, temp)

saveRDS(returns_weekly, "returns_weekly.rds")
saveRDS(error_companies, "error_companies.rds")
```

remove indexes that do not have financial data

```r
returns_weekly = readRDS("returns_weekly.rds")
error_companies = readRDS("error_companies.rds")

data.backup = data

data = data %>%
  anti_join(error_companies, by = c("Symbol", "date_before_filing", "date_after_filing"))
rm(error_companies)
```

calculate price changes and bind with the indexes

```
returns_adjusted = returns_weekly %>%
  mutate(year = year(ref.date)) %>%
  group_by(ticker, year) %>%
  slice(c(1,n())) %>%
  select(-ret.adjusted.prices, -ret.closing.prices) %>%
  mutate(previous.prices = lag(price.adjusted)) %>%
  mutate(log.adjusted.prices = log(price.adjusted) - log(previ
ous.prices)) %>%
  na.omit() %>%
  left_join(returns_weekly, .)

temp = na.omit(returns_adjusted)
data$price_before = as.numeric(temp$previous.prices)
data$price_after = as.numeric(temp$price.adjusted)
data$price_change = as.numeric(temp$log.adjusted.prices)

rm(returns_adjusted, returns_weekly, temp)
```

Add a year column

```
data$year = year(data$date.filed)
```

Text Preprocessing

Create document ID

```
data = data %>%
  mutate(doc_id = row_number())
```

Get company name list

```
name_list = str_trim(gsub("\\bplc|INC|CORP|DE|LTD|CO|CA\\b", "
", unique(data$company.name),
                          ignore.case = TRUE))

part_of_name_list = c("plc","INC","CORP","DE","LTD","CO","CA")
```

Language detection

Check if all management discussions are in English

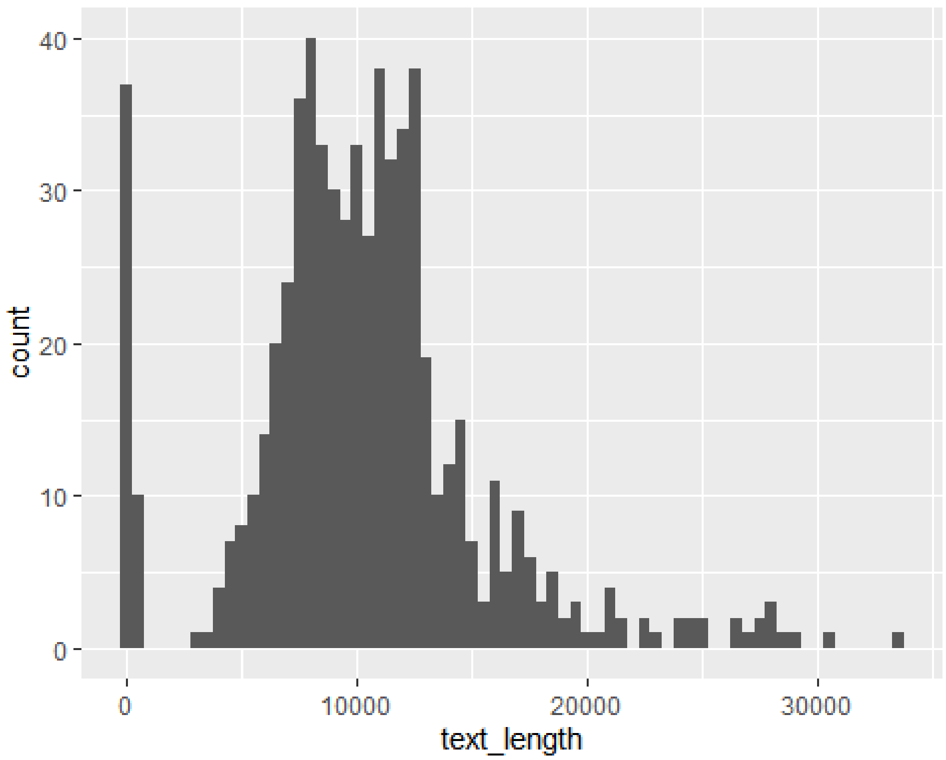
```
data = data %>%
  mutate(language = detect_language(md_text))

which(data$language != "en")

data$language = NULL
```

Management discussions are all in English.

Text Length

Check if all texts are long enough for analyses

```
data$text_length = str_count(data$md_text, "\\b[A-Za-z]+\\b")

min(data$text_length)

## [1] 12

max(data$text_length)

## [1] 33435

mean(data$text_length)

## [1] 10287.63

ggplot(data = data, aes(x = text_length)) + geom_histogram(bin
width = 500)
```




```
# Remove documents which is shorter than 250 words
data = data %>%
  filter(text_length > 2500)

data$text_length = NULL
```

47 documents which have fewer than 2500 words are seen to be abnormal and incomplete, so they are removed.

Contraction Handling

```
data$no_contraction_text = replace_contraction(data$md_text, l
exicon::key_contractions)
```

trim spaces

```
data$trimmed_text = str_trim(data$no_contraction_text)
```

Extract features

Readability

```
readability_df = quanteda.textstats::textstat_readability(data
$md_text, measure = c("ARI", "Flesch.Kincaid", "Linsear.Write
"), remove_hyphens = F)

readability_df$document = NULL

saveRDS(readability_df, "readability_df.rds")

readability_df = readRDS("readability_df.rds")

data = data %>%
  cbind(readability_df)

rm(readability_df)
```

Formality

```
formality_df = qdap::formality(data$md_text, data$doc_id)

saveRDS(formality_df, "formality_df.rds")

formality_df = readRDS("formality_df.rds")
formality_df = formality_df$formality %>%
  mutate(doc_id = as.numeric(doc_id)) %>%
  select(-word.count)

data = data %>%
  left_join(formality_df, by = "doc_id")

rm(formality_df)
```

Number of words

```
data$total_num_of_words = str_count(data$trimmed_text, "\\b[A-
Za-z]+\\b")
```

Number of Punctuations

Punctuations

```
data$num_of_puncts = str_count(data$trimmed_text, "[[:punct:]]
")
```

Dollar Marks

```
data$num_of_dollar = str_count(data$trimmed_text, "[$]")
```

Percentage Marks

```
data$num_of_percents = str_count(data$trimmed_text, "[%]")
```

Number of words whose characters are all capital

```
data$num_of_capitals = str_count(data$trimmed_text,  "\\b[A-Z]
[A-Z][A-Z]+\\b")
```

Number of digits

```
data$num_of_digits = str_count(data$trimmed_text, "[[:digit:]]
+")
```

Save data before the udpipe process

```
saveRDS(data, "data_before_udpipe.rds")

data = readRDS("data_before_udpipe.rds")
```

Text Processing

Tokenisation

Udpipe model

```
ud_model = udpipe_load_model(udpipe::udpipe_download_model("en
glish"))
annotated_MD = udpipe_annotate(object = ud_model,
                               data$trimmed_text,
                               doc_id = data$doc_id,
                               parallel.cores = 8,
                               trace = T) %>% as.data.frame()
rm(ud_model)

saveRDS(annotated_MD, "annotated_MD.rds")

annotated_MD = readRDS("annotated_MD.rds")

annotated_MD$low_lemma = tolower(annotated_MD$lemma)
annotated_MD$doc_id = as.numeric(annotated_MD$doc_id)
```

Add feature: Number of company names mentioned

```
annotated_MD = annotated_MD %>%
  mutate(name_or_not = ifelse(low_lemma %in% tolower(name_lis
t), 1, 0))
```

```
data = annotated_MD %>%
  select(doc_id, name_or_not) %>%
  filter(name_or_not == 1) %>%
  group_by(doc_id) %>%
  summarise(num_of_names = sum(name_or_not)) %>%
  left_join(data, ., by = "doc_id") %>%
  mutate(num_of_names = ifelse(is.na(num_of_names), 0, num_of_
names))

## `summarise()` ungrouping output (override with `.groups` ar
gument)

# check if a word is "plc", "INC", or others
annotated_MD = annotated_MD %>%
  mutate(com_label = ifelse(low_lemma %in% tolower(part_of_nam
e_list), 1, 0))

# clear the memory
rm(name_list, part_of_name_list)
```

Clean tokens start with "-" or "."

Some words start with "-" may be typos, so those "-" have to be removed.

```
annotated_MD$low_lemma = ifelse(str_starts(annotated_MD$low_le
mma, "-"),
                                str_sub(annotated_MD$low_lemm
a, start = 2),
                                annotated_MD$low_lemma)
```

Some words contain "." may be typos, so those "." have to be removed.

```
annotated_MD$low_lemma = gsub("\\.*", "", annotated_MD$low_lem
ma)
```

Inspecting Part of speech

```
pos_list = unique(annotated_MD$upos)

for (pos in pos_list){
  print(paste0(pos, ": ", paste(annotated_MD$low_lemma[which(a
nnotated_MD$upos == pos)[1:10]], collapse = ", ")))
}

## [1] "PROPN: item, note, disclosure, regard, forward, septem
ber, ltd, bermuda, august, august"
## [1] "X: 7, 4, 3, 1, 3, 4, 1, 4, 1, 2"
## [1] "PUNCT: , , , , , ,, ,, ,, ,, ,"
## [1] "ADJ: management, financial, financial, related, annua
l, annual, irish, public, limited, prior"
## [1] "NOUN: s, discussion, analysis, condition, result, oper
```

```
ations, discussion, analysis, conjunction, statement"
## [1] "CCONJ: and, and, and, and, and, and, and, and, and, an
d"
## [1] "ADP: of, of, in, with, in, on, in, with, in, in"
## [1] "DET: the, this, this, the, this, the, the, this, a, a"
## [1] "VERB: follow, read, consolidate, include, contain, loo
k, read, contain, look, use"
## [1] "AUX: should, be, should, be, may, be, be, be, be, be"
## [1] "PRON: we, we, we, we, we, its, that, its, we, which"
## [1] "ADV: elsewhere, also, forward, also, wholly, otherwis
e, otherwise, so, thereby, where"
## [1] "NUM: 10, k, 10, k, 1, 2009, 31, 2010, 12, 31"
## [1] "SYM: , , $, $, %, %, $, $, %, $"
## [1] "PART: to, to, not, to, to, to, to, to, to, to"
## [1] "SCONJ: unless, unless, that, by, although, that, to, a
lthough, by, on"
## [1] "INTJ: non-gaap, n/m, n/m, n/m, n/m, n/m, n/m, n/m, n/
m, n/m"
```

```
rm(pos, pos_list)
```

Calculating Tf-Idf values needs total word counts of each document, so I use all words except digits and punctuation to measure Tf-Idf values. X, PUNCT, NUM, and SYM are digits and punctuation, which are meaningless in terms of both sentiment analysis and topic modelling. Therefore, they are removed here. The rest of words are used for calculating Tf-Idf values.

```
annotated_MD_for_tfidf = annotated_MD %>%
  filter(!upos %in% c("X", "PUNCT", "NUM", "SYM"))
```

NOUN, PROPN, ADJ, VERB, are ADV are meaningful for text mining, so they will be kept. Let's inspect other categories (CCONJ, ADP, DET, AUX, PRON, PART, SCONJ, INTJ).

CCONJ

```
( annotated_MD %>% filter(upos == "CCONJ") %>% select(low_lemm
a) %>% unique() %>% as.character() )
```

```
## [1] "c(\"and\", \"or\", \"&\", \"nor\", \"plus\", \"neither
\", \"either\", \"not\", \"both\", \"but\", \"non-us\", \"yet\
", \"+\", \"minus\", \"libor\", \"microprocessor\", \"md&\", \
"advisor\", \"other\", \"sectors\", \"focus\", \"smg&\", \"nan
d\", \"sg&\", \"g&\", \"bonus\", \"acquisitions\", \"n\", \"vs
\", \"plan\", \"vendor\", \"versus\", \"cd&\", \"voucher\", \"
fleetcor\", \"as\", \"richer\", \"a\", \"us\", \"essn\", \"mg&
\", \"prior\", \"novellus\", \"b&\", \"cost-plus\", \"junior\
", \"segments\", \"taibor\", \"alignd\", \"operations\", \"tog
ether\", \"nexus\", \"gpu\", \"autonomous\", \"labor\", \"surp
lus\", \n\"icann\", \"repurchases\", \"omnibus\", \"uk&\")"
```

All of the CCONJ words do not have polarity scores. Therefore they will be excluded.

ADP

```
( annotated_MD %>% filter(upos == "ADP") %>% select(low_lemma)
 %>% unique() %>% as.character() )
```

```
## [1] "c(\"of\", \"in\", \"with\", \"on\", \"to\", \"for\", \
"without\", \"into\", \"by\", \"before\", \"over\", \"across\
", \"between\", \"around\", \"during\", \"due\", \"through\",
\"than\", \"after\", \"at\", \"against\", \"as\", \"from\", \"
below\", \"per\", \"under\", \"until\", \"since\", \"out\", \"
despite\", \"upon\", \"about\", \"among\", \"within\", \"throu
ghout\", \"except\", \"restructuring\", \"via\", \"off\", \"al
ong\", \"vsoe\", \"towards\", \"up\", \"because\", \"tpe\", \"
whereby\", \"near\", \"gain\", \"outside\", \"52/\", \"x\", \"
versus\", \"arr\", \"down\", \"record\", \"above\", \"gf\", \"
past\", \n\"semi-custom\", \"prior\", \"beyond\", \"radeon\",
\"intra\", \"opteron\", \"accrue\", \"toward\", \"thereto\", \
"onto\", \"ahead\", \"ott\", \"theretoo\", \"like\", \"operato
r\", \"epsilon\", \"revenue\", \"ember\", \"thereafter\", \"ti
er\", \"talf\", \"ratio\", \"pursue\", \"weaker\", \"debt\", \
"exclude\", \"act\", \"wafer\", \"monitor\", \"obsolescence\",
 \"agreement\", \"value\", \"a\", \"minus\", \"forward\", \"f\
", \"vs\", \"hereafter\", \"include\", \"bonus\", \"debout\",
\"unforeseen\", \"ibookstore\", \"throughput\", \"thin\", \"io
n\", \"undrawn\", \"quarter-over\", \"rd&e\", \"gross\", \n\"s
ilicon\", \"ebout\", \"vendor\", \"thereon\", \"save\", \"und\
", \"3d\", \"alike\", \"upgrades\", \"3ds\", \"peo\", \"unlike
\", \"n/\", \"improve\", \"zero-margin\", \"cver\", \"pass\",
\"broadcom\", \"coupon\", \"sfas\", \"penson\", \"optimize\",
\"behind\", \"th\", \"blockchain\", \"back\", \"third\", \"mon
th\", \"therefore\", \"dividend\", \"amongst\", \"weigh\", \"i
sr\", \"transform\", \"asr\", \"ato\", \"bith\", \"abo\", \"em
ea\", \"abandon\", \"assert\", \"fcpa\", \"strengthan\", \"sam
sng\", \"o\", \"w\", \"hrough\", \"asser\", \"contin\", \"poly
silicon\", \"apbo\", \"samsung\", \"fourth\", \n\"rather\", \"
acquirer\", \"overrides\", \"arx\", \"ar\", \"zero\", \"unwind
\", \"term\", \"t&e\", \"breakdown\", \"inside\", \"footnote\
", \"fleetcor\", \"borrower\", \"trades\", \"re\", \"fortiguar
d\", \"amortize\", \"amazon\", \"amortization\", \"board\", \"
amr\", \"defer\", \"anti-takeover\", \"though\", \"packard\",
\"hpe\", \"spun-off\", \"buyout\", \"pointnext\", \"ess\", \"s
trengthen\", \"beneath\", \"ito\", \"indicator\", \"thinner\",
 \"atom\", \"desktop\", \"fin\", \"telecom\", \"eft\", \"fas\
", \"slt\", \"margin\", \"obsolete\", \"ocx\", \"fund\", \"fro
\", \"incom\", \"nith\", \"don\", \n\"withhold\", \"reuse\", \
"rent/\", \"announce\", \"worth\", \"cost-plus\", \"besides\",
 \"amortizaton\", \"although\", \"excludes\", \"increase/\", \
"tha\", \"scrutinize\", \"amplifier\", \"that\", \"micron\", \
```

```
"attribute\", \"safeguard\", \"thereunder\", \"nor\", \"rambus
\", \"includes\", \"ddr\", \"imft\", \"lte\", \"growth\", \"dv
r\", \"tetra\", \"overhead\", \"outweigh\", \"harm\", \"icera\
", \"strength\", \"photo\", \"quadro\", \"be\", \"decode\", \"
drive\", \"unix\", \"vef\", \"upward\", \"alongside\", \"addit
io\", \"xt\", \"fsa\", \"hr\", \"ng\", \"proceed\", \"xoom\",
\"fico\", \"login\", \n\"rf\", \"antenna\", \"iprd\", \"flo\",
 \"qmt\", \"3g/\", \"3gith\", \"use\", \"forcecom\", \"hdds\",
 \"allo\", \"oi&e\", \"om\", \"icann\", \"thawte\", \"plaintif
f\", \"agreemento\", \"non-us\", \"issuer\", \"will\", \"perta
in\", \"takedown\", \"thirteen\", \"observer\", \"/sith\", \"a
llocation\", \"iith\", \"toshiba\", \"asia\", \"non-compute\",
 \"minimize\", \"outstandin\", \"do\", \"impika\", \"concept\
", \"bpo\", \"a4\", \"instill\", \"own\", \"pts\", \"factoring
\", \"proveo\")"
```

Most of the ADP words are meaningful and have a polarity score, so they will be used for further analyses.

DET

```
( annotated_MD %>% filter(upos == "DET") %>% select(low_lemma)
 %>% unique() %>% as.character() )
```

```
## [1] "c(\"the\", \"this\", \"a\", \"that\", \"all\", \"some\
", \"these\", \"no\", \"those\", \"both\", \"which\", \"anothe
r\", \"each\", \"any\", \"emea\", \"either\", \"treasury\", \"
what\", \"neither\", \"every\", \"judgme\", \"segmentthese\",
\"etla\", \"tfi\", \"such\", \"2015the\", \"of\", \"higher\",
\"fullbeauty\", \"ebitda\", \"revolver\", \"acquisitionthese\
", \"agreementto\", \"apache\", \"workbench\", \"millionthe\",
 \"foundry\", \"etch\", \"aca\", \"eda\", \"merchandise\", \"i
nsieme\", \"dta\", \"h\", \"vat\", \"whenever\", \"dutch\", \"
tfis\", \"half\", \"tnon-fis\", \"north\", \"onetime\", \n\"bu
rdensome\", \"2011the\", \"2014the\", \"delivery\", \"fortiswi
tch\", \"research\", \"other\", \"revenueto\", \"ame\", \"hear
tland\", \"explanatory\", \"tmphasis\", \"2016the\", \"attach\
", \"tranche\", \"accompany\", \"fitch\", \"fpga\", \"nand\",
\"nico\", \"ach\", \"strengthe\", \"wich\", \"orbotech\", \"wh
ichever\", \"fy\", \"approach\", \"time\", \"erall\", \"airban
d\", \"wlan\", \"can\", \"aa\", \"they\", \"to\", \"gsa\", \"w
herever\", \"nvidia\", \"whatever\", \"acme\", \"mainframe\",
\"respectivelyto\", \"excise\", \"bluetooth\", \"qchat\", \"tq
is\", \"ofdma\", \"deutsche\", \"deutsch\", \n\"registry\", \"
idefense\", \"lsa\", \"adoptionthe\", \"eea\", \"laca\", \"tme
\")"
```

Most of the DET words are misspelled words or have no polarity score. Excluded.

AUX

```r
( annotated_MD %>% filter(upos == "AUX") %>% select(low_lemma)
 %>% unique() %>% as.character() )
```

```
## [1] "c(\"should\", \"be\", \"may\", \"have\", \"will\", \"w
ere\", \"do\", \"payroll\", \"could\", \"would\", \"can\", \"s
ca\", \"must\", \"goodwill\", \"might\", \"receivable\", \"rep
urchase\", \"'s\", \"decrease\", \"program\", \"make\", \"aem\
", \"simplify\", \"amd\", \"shall\", \"modify\", \"where\", \"
quantify\", \"enhance\", \"overall\", \"ar\", \"expenditure\",
 \"1,383,\", \"1,433,\", \"trust\", \"utilize\", \"score\", \"
inventory\", \"multiple\", \"summarize\", \"2,701,\", \"subsid
iary\", \"weaken\", \"strengthene\", \"infringe\", \"applecare
\", \"americas\", \"accrue\", \"4s\", \"hatsve\", \"report\",
\n\"compare\", \"encompass\", \"describe\", \"inventories\", \
"reserve\", \"non-america\", \"economie\", \"geography\", \"en
tail\", \"increase\", \"assess\", \"eps\", \"bonus\", \"earnin
g\", \"asset\", \"curve\", \"acquisitiond\", \"divestiture\",
\"borrowing\", \"repay\", \"of\", \"v\", \"arise\", \"hedge\",
 \"uncertainty\", \"saas\", \"allowances\", \"determine\", \"r
aise\", \"receivables\", \"switch\", \"comprise\", \"switching
\", \"camera\", \"clould\", \"cloud\", \"lease\", \"unify\", \
"deficiency\", \"methodology\", \"apas\", \"6,121,\", \"repaid
\", \"corning\", \"target\", \"echnology\", \n\"corn\", \"impo
s\", \"hpe\", \"gbs\", \"is&s\", \"gis\", \"hpes\", \"gb\", \"
establish\", \"1,271,\", \"intensify\", \"fis\", \"tfis\", \"p
articipaco\", \"fiserv\", \"vesting\", \"share\", \"believe\",
 \"varies\", \"forticare\", \"bundle\", \"build\", \"serve\",
\"sas\", \"tight\", \"ucs\", \"frauld\", \"willinghbe\", \"dis
agreement\", \"fraud\", \"ep\", \"weight\", \"es\", \"bcs\", \
"fs\", \"investee\", \"mc\", \"subsidiaries\", \"liquid\", \"h
pfs\", \"hpf\", \"balance\", \"imfs\", \"license\", \"engrave\
", \"amortize\", \"overseen\", \"outlay\", \"install\", \"non-
software\", \"firewall\", \"ay\", \n\"vmware\", \"av\", \"incr
e\", \"n/m\", \"prescribe\", \"efficiency\", \"1,173,\", \"1,0
62,\", \"variable\", \"paid\", \"lam\", \"m&\", \"achieve\", \
"hagdve\", \"gev\", \"operate\", \"analyze\", \"withheld\", \"
faes\", \"r&d\", \"eses\", \"outsource\", \"r\", \"security\",
 \"disclosur\", \"intangibles\", \"minimize\", \"thsey\", \"un
certainties\", \"fa\", \"veritas\", \"become\", \"intangible\
", \"wcould\", \"uld\", \"game\", \"nonsoftware\", \"iaas\", \
"patch\", \"kvm\", \"hrs\", \"haderive\", \"operating\", \"hat
pve\", \"t\", \"2,181,\", \"purchase\", \"imod\", \"qe\", \"ms
m\", \"overcome\", \"demandware\", \n\"1,551,\", \"tbe\", \"na
s\", \"clarify\", \"being\", \"specialty\", \"qualifies\", \"d
ebentures\", \"thirteen\", \"sink\", \"authorize\", \"kendall\
", \"announce\", \"indemnify\", \"rely\", \"disclosures\", \"a
re\", \"seek\", \"expire\", \"hgst\", \"payable\", \"payables\
", \"rate\", \"medicaid\", \"as\", \"notice\", \"hagavrielove\
", \"market\", \"evm\", \"mature\", \"zes\", \"ze\", \"fas\")"
```

Most of the AUX words are meaningful, which can contribute to topic modelling solutions. Keep them.

PRON

```
( annotated_MD %>% filter(upos == "PRON") %>% select(low_lemm
a) %>% unique() %>% as.character() )
```

```
## [1] "c(\"we\", \"its\", \"that\", \"which\", \"there\", \"t
his\", \"they\", \"it\", \"these\", \"accenture\", \"i\", \"th
ose\", \"who\", \"what\", \"you\", \"thitey\", \"vsoe\", \"tpe
\", \"yoou\", \"securitieswe\", \"acrobat\", \"our\", \"one\",
 \"ours\", \"2018we\", \"jous\", \"joi\", \"ourselves\", \"he\
", \"yourself\", \"themselves\", \"incur\", \"their\", \"loyal
tyone\", \"air\", \"hmi\", \"me\", \"analysi\", \"she\", \"asu
\", \"itself\", \"redeem\", \"whose\", \"apache\", \"rabbi\",
\"gilti\", \"optis\", \"phine\", \"phone\", \"backlog\", \"si\
", \"whom\", \"basis\", \"10/25/40/50/100g\", \"2011we\", \"20
12we\", \n\"yooccur\", \"2014we\", \"be\", \"streamline\", \"a
nything\", \"furnish\", \"us\", \"ebin\", \"ebit\", \"aig\", \
"analysis\", \"broadcom\", \"pwi\", \"m&o\", \"yoincu\", \"dio
\", \"vce\", \"everything\", \"switching\", \"anyone\", \"clou
dcom\", \"whichever\", \"zenprise\", \"mofcom\", \"investor\",
 \"item\", \"corn\", \"corning\", \"'s\", \"hsg\", \"standalon
e\", \"fis\", \"fnf\", \"vie\", \"assumpto\", \"tfis\", \"defi
\", \"therefrom\", \"ctf\", \"unding\", \"thermography\", \"ut
m\", \"ou\", \"tranche\", \"bpi\", \"undergone\", \"sicom\", \
"tranco\", \"hpe\", \"dso\", \"dpo\", \"mphasis\", \"tmphasis\
", \n\"in\", \"nm\", \"analog\", \"ipg\", \"pccg\", \"dcg\", \
"ccg\", \"iotg\", \"psg\", \"echo\", \"have\", \"exclude\", \"
gfsi\", \"throut\", \"margi\", \"if\", \"whi\", \"occur\", \"d
atesthe\", \"fy\", \"ifrs\", \"mine\", \"maxit\", \"ii\", \"es
to\", \"microsemi\", \"want\", \"someone\", \"engi\", \"hich\
", \"imaging\", \"dsg\", \"nsg\", \"wsg\", \"mit\", \"sbu\", \
"rout\", \"3g\", \"lte\", \"esg\", \"arpu\", \"apru\", \"audi\
", \"4g\", \"forfeiture\", \"my\", \"peo\", \"hrs\", \"time\",
 \"whe\", \"xoom\", \"tqis\", \"qsi\", \"3g/4g\", \"qwi\", \"5
g\", \"survey\", \"everyone\", \"renminbi\", \"wi\", \"yoau\",
 \"webloyaltycom\", \n\"out\", \"activiti\", \"uk&i\", \"iresp
ect\", \"weakening\", \"wdi\", \"hitachi\", \"westernunioncom\
", \"gis\", \"bpo\", \"moshe\", \"tme\", \"ait\", \"spg\", \"s
teady\")"
```

Most of the PRONs are not meaningful and do not have sentiment scores. Exclude them.

PART

```
( annotated_MD %>% filter(upos == "PART") %>% select(low_lemm
a) %>% unique() %>% as.character() )
```

```
## [1] "c(\"to\", \"not\", \"'s\", \"'\", \"revenueto\", \"be\
", \"s\", \"nt\", \"2\", \"dato\", \"data\", \"606'\", \"na\",
 \"act\", \"eft\", \"slt\", \"tot\", \"and\", \"pct\", \"imft\
", \"robot\", \"qct\", \"qmt\", \"iot\", \"ito\")"
```

Almost all PART words are meaningless. Excluded.

SCONJ

```
( annotated_MD %>% filter(upos == "SCONJ") %>% select(low_lemm
a) %>% unique() %>% as.character() )
```

```
## [1] "c(\"unless\", \"that\", \"by\", \"although\", \"to\",
\"on\", \"as\", \"if\", \"because\", \"in\", \"whether\", \"wh
ile\", \"than\", \"of\", \"into\", \"since\", \"with\", \"for\
", \"whereby\", \"mobilize\", \"before\", \"about\", \"after\
", \"from\", \"so\", \"receivable\", \"whereupon\", \"custom\
", \"once\", \"through\", \"whereas\", \"cause\", \"etlas\", \
"acrobat\", \"over\", \"workforce\", \"without\", \"towards\",
 \"cross\", \"until\", \"gf\", \"goodwill\", \"other\", \"awar
d\", \"further\", \"at\", \"except\", \"indebtedness\", \"buil
d\", \"upon\", \"growth\", \"though\", \"against\", \"forward\
", \n\"variable\", \"reportable\", \"excess\", \"mobile\", \"r
d&e\", \"optimize\", \"reduce\", \"m&e\", \"balance\", \"ignit
e\", \"need\", \"autodes\", \"health\", \"force\", \"peo\", \"
include\", \"less\", \"weigh\", \"toward\", \"iaas\", \"portfo
lio\", \"below\", \"it\", \"despite\", \"saas\", \"wireless\",
 \"observable\", \"getgo\", \"face\", \"floater\", \"somewhat\
", \"elebrat\", \"beyond\", \"ratio\", \"stabilize\", \"diamet
er\", \"multi-year\", \"deliverable\", \"fleetcor\", \"uas\",
\"sas\", \"flat\", \"non-revenue\", \"redeemable\", \"notice\
", \"file\", \"authorize\", \"scalable\", \"hyperscale\", \n\"
strengthen\", \"fpgas\", \"intuit\", \"includes\", \"vsoe\", \
"pep\", \"ppt\", \"semiconductor\", \"issuance\", \"amend\", \
"hold\", \"americas\", \"standard\", \"theretoo\", \"trend\",
\"albeit\", \"besides\", \"exceeds\", \"address\", \"veritas\
", \"double\", \"excludes\", \"geforce\", \"fortnite\", \"life
cycle\", \"trends\", \"xoom\", \"like\", \"qorvo\", \"together
\", \"imbalance\", \"weakness\", \"th\", \"convertible\", \"ab
eyance\", \"leave\", \"standstill\", \"unavailable\", \"mofcom
\", \"estimable\", \"westernunioncom\", \"invoco\", \"a\", \"t
reble\")"
```

Most of the SCONJ words are meaningful, which can contribute to topic modelling solutions. Keep them.

INTJ

```
( annotated_MD %>% filter(upos == "INTJ") %>% select(low_lemm
a) %>% unique() %>% as.character() )
```

```
## [1] "c(\"non-gaap\", \"n/m\", \"omniture\", \"lease\", \"de
crease\", \"nano\", \"atmp\", \"defease\", \"ip\", \"portfolio
\", \"sell\", \"loan\", \"amdiss\", \"gf\", \"nm\", \"increase
\", \"nol\", \"oci\", \"well\", \"no\", \"dotz\", \"smg&a\", \
"outlook\", \"please\", \"apache\", \"right\", \"homepod\", \"
h\", \"itunes\", \"ratio\", \"might\", \"transform\", \"arr\",
 \"payroll\", \"revenue\", \"taxes\", \"o\", \"purchase\", \"n
otebook\", \"dso\", \"switches\", \"saas\", \"app\", \"rightsi
gnature\", \"risks\", \"gaap\", \"liquidity\", \"of\", \"hsg\
", \"gbs\", \"asm\", \"aam\", \"fsg\", \"tpe\", \"nty\", \"par
ticipacoe\", \n\"obs\", \"nyse\", \"svs\", \"ot\", \"ots\", \"
non-employee\", \"wh\", \"anti-spam\", \"fortiap\", \"oy\", \"
gpcs\", \"7a\", \"exhibit\", \"ease\", \"treasury\", \"hpe\",
\"hp\", \"pccg\", \"xp\", \"refresh\", \"ccg\", \"imfs\", \"mm
\", \"fms\", \"ems\", \"quickbook\", \"g&a\", \"ocp\", \"insig
ht\", \"z\", \"ao\", \"espp\", \"ars\", \"flash\", \"l\", \"m\
", \"asp\", \"clearwell\", \"iaas\", \"hrs\", \"surepayroll\",
 \"high\", \"ase\", \"qwi\", \"imod\", \"licensee\", \"welcome
\", \"like\", \"pangea\", \"tegsa\", \"hval\", \"us\", \"proto
col\", \"annul\", \"visa\", \"phase\", \"ito\", \"annuity\", \
"xbs\", \"vi\", \n\"prepay\", \"zes\")"
```

Meaningless. Excluded.

Keep important POS words, remove company names and abbreviations

Hence, I only keep words classified as “NOUN”, “PROPN”, “ADJ”, “VERB”, “ADV”, “ADP”, “AUX”, and “SCONJ” for further analyses. Furthermore, I remove all company names and abbreviations such as “plc”, “INC”, “CORP”, etc.

```
clean_annotated_MD = annotated_MD %>%
  filter(upos %in% c("NOUN", "PROPN", "ADJ", "VERB", "ADV", "A
DP", "AUX", "SCONJ")) %>%
  filter(name_or_not != 1) %>%
  filter(com_label != 1)

# clear the memory
rm(annotated_MD)
clean_annotated_MD$name_or_not = NULL
clean_annotated_MD$com_label = NULL
```

Stop words

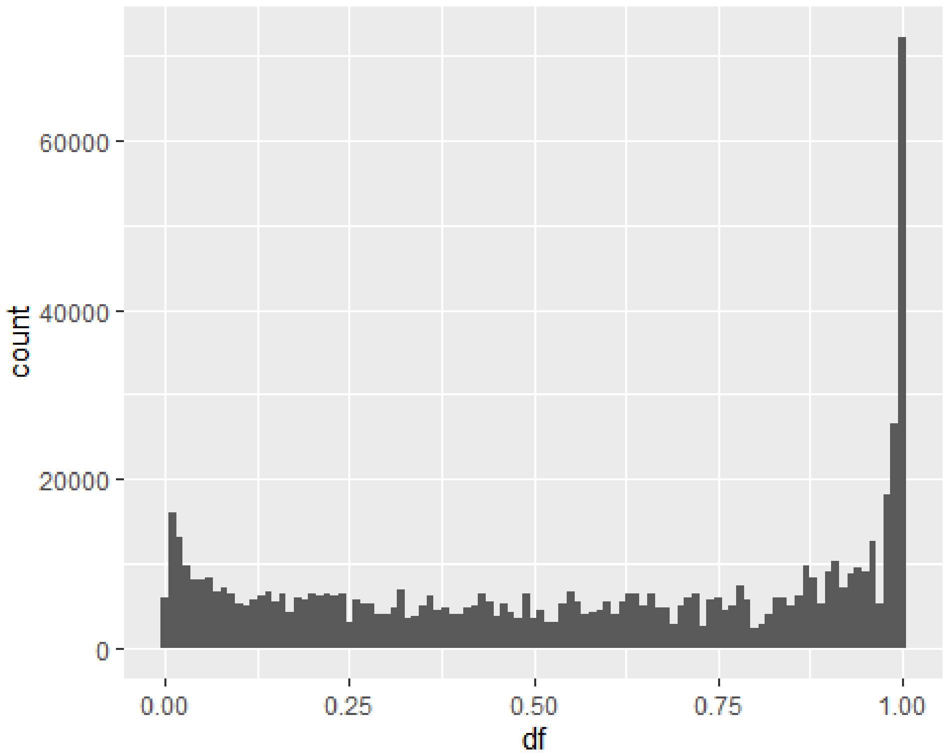
I compute the Tf-Idf values in filing level instead of company level because the stock prices also differ in filing level. Document frequencies are also computed. #### Tf-Idf stop words identification

```
annotated_MD_tfidf = annotated_MD_for_tfidf %>%
  select(doc_id, low_lemma) %>%
  count(doc_id, low_lemma) %>%
```

```
  bind_tf_idf(low_lemma, doc_id, n) %>%
  mutate(df = 1/(exp(idf)))
```

Inspect the distribution of document frequencies I use document frequencies to develop my stop word dictionary instead of using Tf-Idf values because removing words that are very commonly used across the entire corpus can help us focus on more important words.

```
ggplot(annotated_MD_tfidf, aes(df)) +
  geom_histogram(binwidth = 0.01)
```



As can be seen in the histogram, there is a surge of document frequencies at about 0.97. It seems there is a growth of words that appear in 97% of the documents.

```
View( annotated_MD_tfidf %>%
  filter(df > 0.97) %>%
    select(low_lemma, df) %>%
    unique() %>%
    arrange(low_lemma) )
```

There are 203 words whose document frequency is larger than 0.97. It means those words appear in 97% of the documents and hence they are not representative for documents. Moreover, most of them are fairly common in English. They will be treated as stop words.

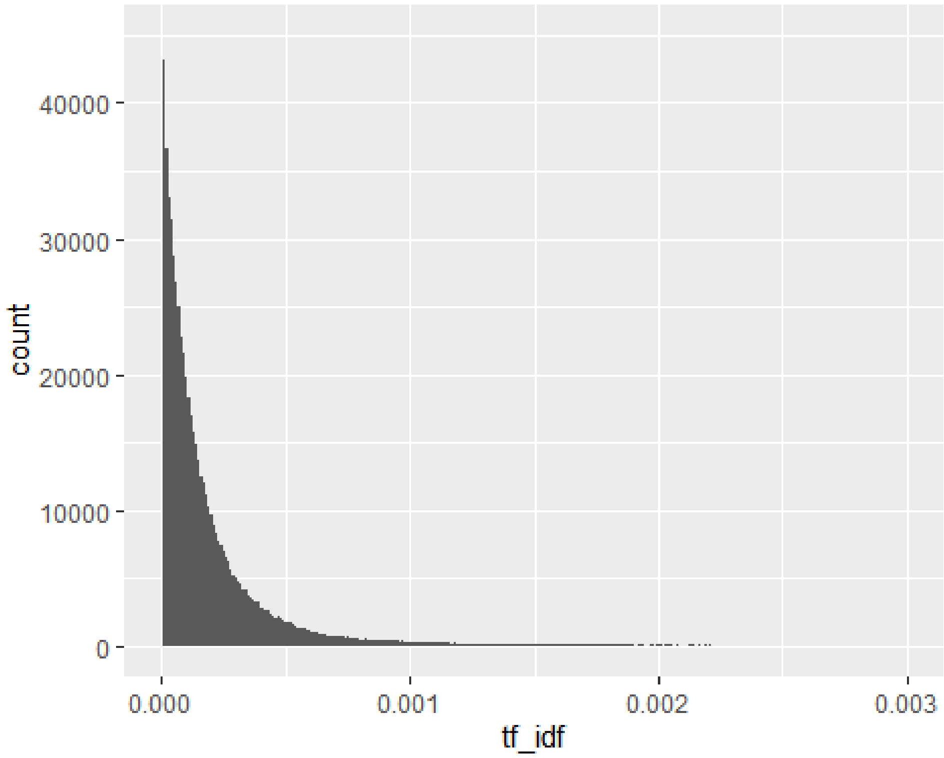
```
# Constructing a stop word dictionary on document frequencies
df_stopwords = annotated_MD_tfidf %>%
  filter(df > 0.97) %>%
  select(low_lemma) %>%
```

```
  mutate(lexicon = "doc_freq") %>%
  rename(word = low_lemma) %>%
  unique()
```

Inspect the distribution of document frequencies

I directly remove words that have a low value of Tf-Idf instead of constructing a stop word lexicon because one word can have multiple Tf-Idf values. For example, assume that there is a word that has a high value of Tf-Idf in one document and has a low Tf-Idf value in another document. If I anti join a dictionary of Tf-Idf stop words, the word in both documents will be dropped.

```
ggplot(annotated_MD_tfidf, aes(tf_idf)) +
  geom_histogram(binwidth = 0.00001) +
  xlim(0, 0.003) +
  ylim(0, 45000)

## Warning: Removed 4273 rows containing non-finite values (st
at_bin).

## Warning: Removed 2 rows containing missing values (geom_ba
r).
```



Examine words that have a low Tf-Idf value

```
annotated_MD_tfidf %>%
  filter(tf_idf < 0.0001) %>%
  select(low_lemma) %>%
  unique()
```

Remove words that have a small value of Tf-Idf

```
clean_annotated_MD = clean_annotated_MD %>%
  left_join(annotated_MD_tfidf %>% select(doc_id, low_lemma, t
f_idf),
            by = c("doc_id", "low_lemma"))

clean_annotated_MD = clean_annotated_MD %>%
  filter(tf_idf < 0.0001) %>%
  select(-tf_idf)
```

The overall stop word dictionary

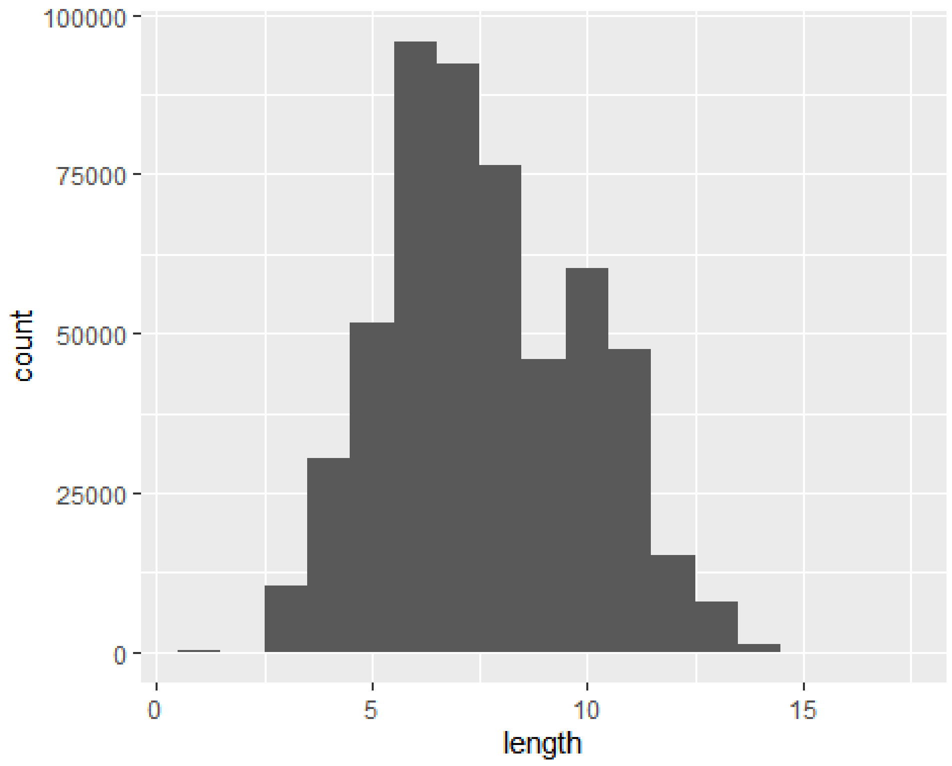
```
my_stopwords = stop_words %>%
  rbind(df_stopwords)
```

Remove stop words

```
no_stop_MD = clean_annotated_MD %>%
  select(doc_id, low_lemma) %>%
  anti_join(my_stopwords, by = c("low_lemma" = "word"))

# clear the memory
rm(df_stopwords, annotated_MD_for_tfidf, annotated_MD_tfidf, m
y_stopwords, clean_annotated_MD)
```

remove short or extremely long tokens

```
no_stop_MD$length = nchar(no_stop_MD$low_lemma)

min(no_stop_MD$length)

## [1] 1

max(no_stop_MD$length)

## [1] 17

ggplot(no_stop_MD, aes(x = length)) + geom_histogram(binwidth
= 1)
```




```
View(no_stop_MD %>% select(low_lemma, length) %>% unique() %>%
 arrange(desc(length)) )
View(no_stop_MD %>% select(low_lemma, length) %>% unique() %>%
 arrange(length) )
```

It seems tokens shorter than 3 are either typos or meaningless. They should all be removed.

```
no_stop_MD = no_stop_MD %>%
  filter(length >= 3)
```

Detokenising clean tokens

```
clean_data = no_stop_MD %>%
  group_by(doc_id) %>%
  summarise(clean_MD_text = paste(low_lemma, collapse = " "))
%>%
  left_join(data, ., by = "doc_id") %>%
  select(-md_text, -no_contraction_text, -trimmed_text)

clean_data$num_of_clean_words = str_count(clean_data$clean_MD_
text, "\\b[A-Za-z]+\\b")

# clear the memory
rm(no_stop_MD, data)

# Save for further uses
saveRDS(clean_data, "clean_data.rds")
```

Task: Important keywords across the industry level (GICS)

```
clean_data = readRDS("clean_data.rds")

View(head(clean_data))

unique(clean_data$`GICS Sub Industry`)

##  [1] "IT Consulting & Other Services"
##  [2] "Application Software"
##  [3] "Semiconductors"
##  [4] "Internet Services & Infrastructure"
##  [5] "Data Processing & Outsourced Services"
##  [6] "Electronic Components"
##  [7] "Technology Hardware, Storage & Peripherals"
##  [8] "Semiconductor Equipment"
##  [9] "Communications Equipment"
## [10] "Technology Distributors"
## [11] "Electronic Equipment & Instruments"
## [12] "Systems Software"
## [13] "Electronic Manufacturing Services"
```

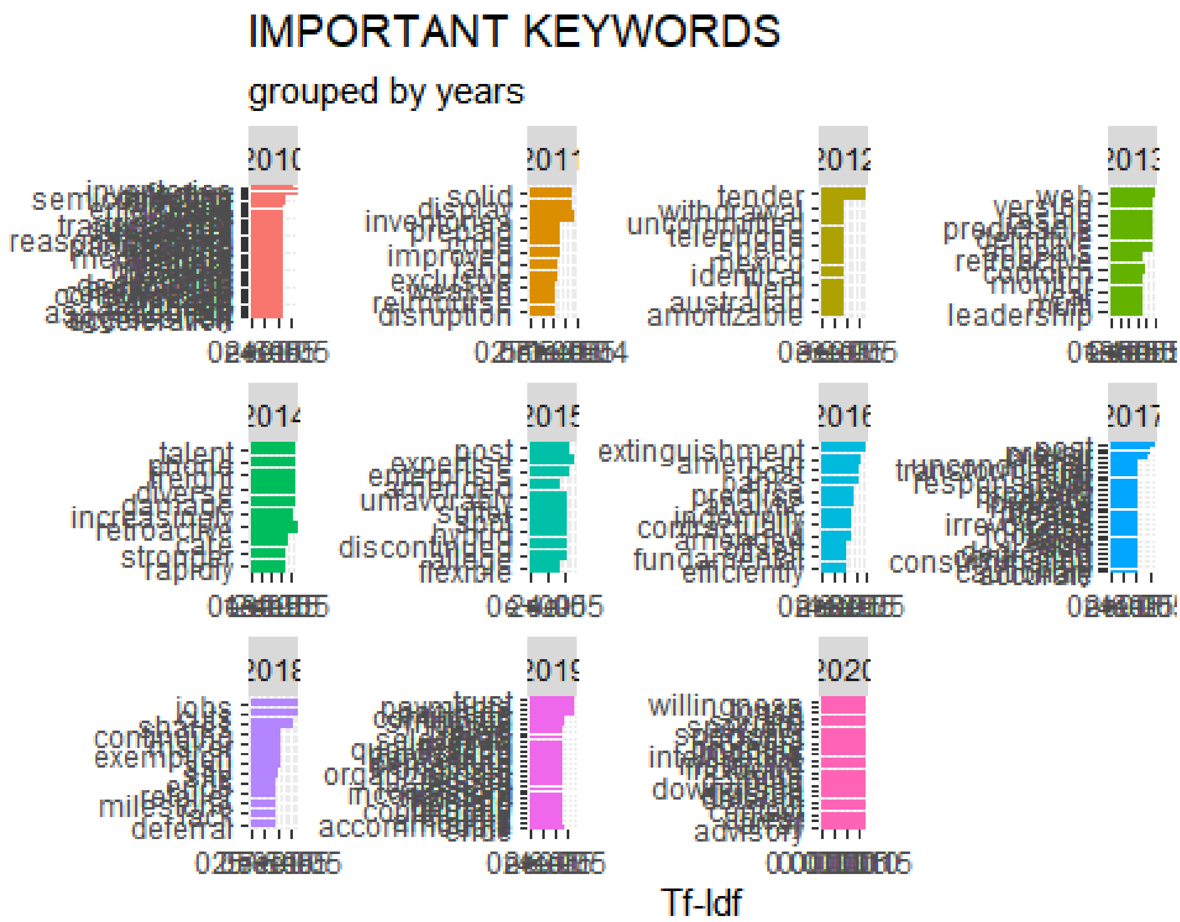
Keywords on the year level

Calculate Tf-Idf values on the year level

```
year_tfidf = clean_data %>%
  unnest_tokens(input = clean_MD_text, output = word) %>%
  count(year, word) %>%
  bind_tf_idf(word, year, n) %>%
  ungroup()
```

Examine if there is any drastic change of keywords during the entire span

```
year_tfidf %>%
  group_by(year) %>%
  arrange(year, desc(tf_idf)) %>%
  top_n(10, tf_idf) %>%
  ungroup() %>%
  mutate(word = fct_reorder(word, tf_idf, .desc = F)) %>%
  ggplot(aes(x = word, y = tf_idf, fill = as.factor(year))) +
  geom_col(show.legend = FALSE) +
  labs(x = NULL, y = "Tf-Idf", title = "IMPORTANT KEYWORDS", s
ubtitle = "grouped by years") +
  facet_wrap(~year, scales = "free", ncol = 4) +
  coord_flip()
```



Keywords on the industry level

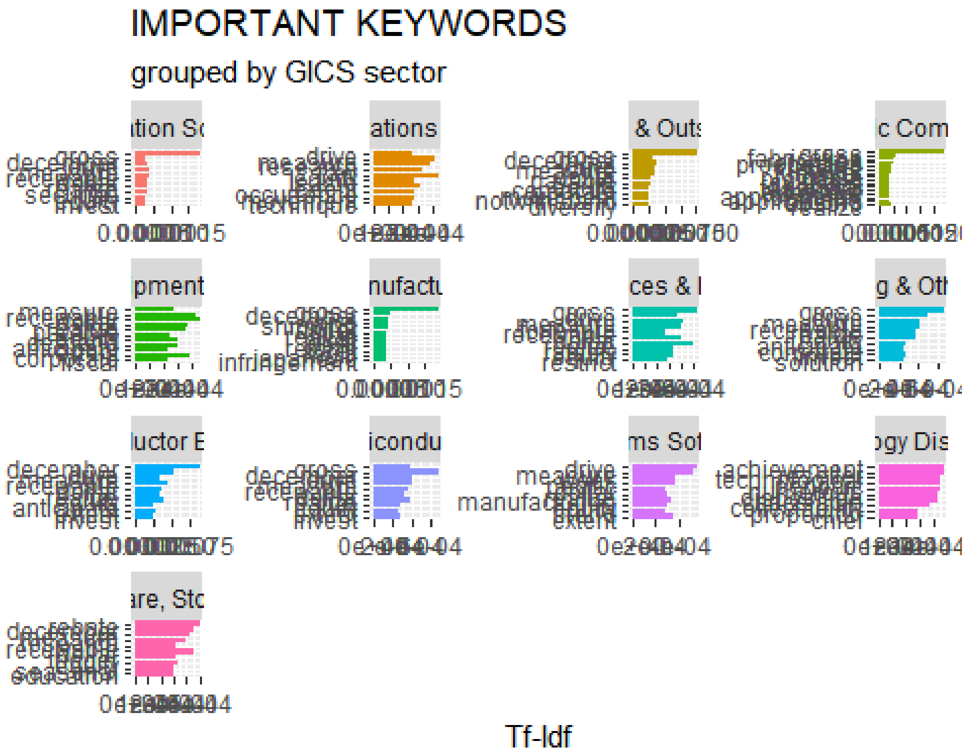
Overall Keywords across the entire span

```
industry_tfidf = clean_data %>%
  unnest_tokens(input = clean_MD_text, output = word) %>%
  count(`GICS Sub Industry`, word) %>%
  bind_tf_idf(word, `GICS Sub Industry`, n) %>%
  ungroup()
```

Examine the differences of keywords in different industries

```
industry_tfidf %>%
  group_by(`GICS Sub Industry`) %>%
  arrange(`GICS Sub Industry`, desc(tf_idf)) %>%
  top_n(10, tf_idf) %>%
  ungroup() %>%
  mutate(word = fct_reorder(word, tf_idf, .desc = F)) %>%
  ggplot(aes(x = word, y = tf_idf, fill = as.factor(`GICS Sub
Industry`))) +
  geom_col(show.legend = FALSE) +
  labs(x = NULL, y = "Tf-Idf", title = "IMPORTANT KEYWORDS", s
ubtitle = "grouped by GICS sector") +
  facet_wrap(~`GICS Sub Industry`, scales = "free", ncol = 4)
```

```
+
  coord_flip()
```



Effects of the pandemic examination

Data after the pandemic (the pandemic was declared by WHO on 11 March 2020)

```
industry_tfidf_after = clean_data %>%
  filter(date.filed > ymd("2020-03-11")) %>%
  unnest_tokens(input = clean_MD_text, output = word) %>%
  count(`GICS Sub Industry`, word) %>%
  bind_tf_idf(word, `GICS Sub Industry`, n) %>%
  ungroup()
```

There are only 3 industries (“Communications Equipment”, “Electronic Equipment & Instruments”, “Semiconductors”) filed after the declaration of the pandemic, so only these 3 industries will be compared.

```
compared_industries = unique(industry_tfidf_after$`GICS Sub In
dustry`)
```

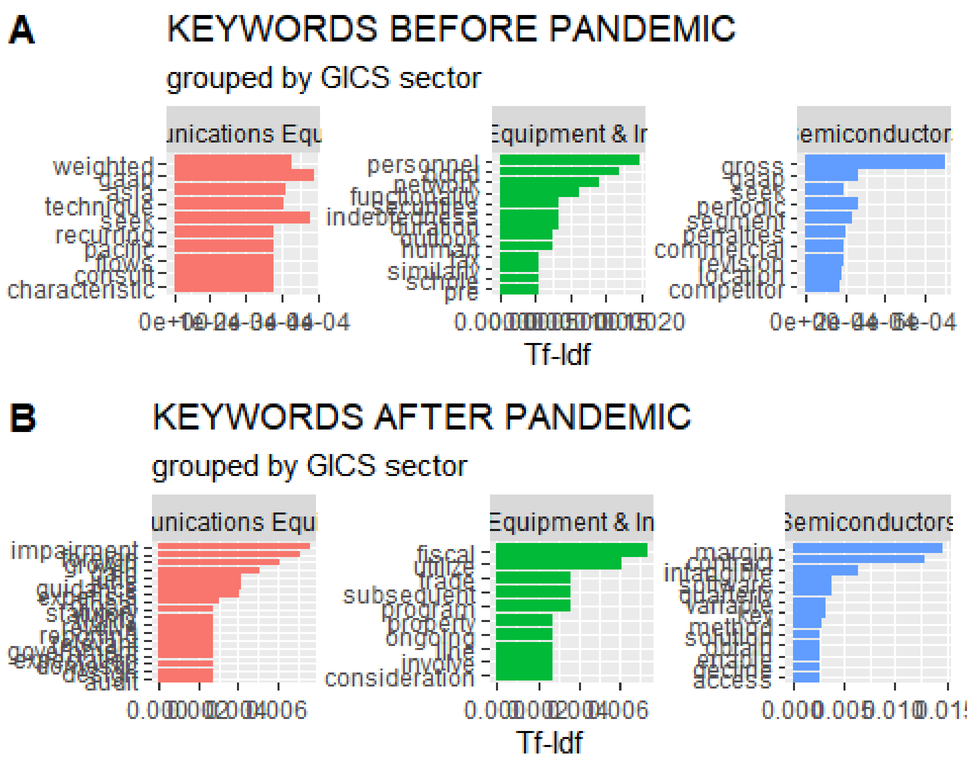
Data before the pandemic

```
industry_tfidf_before = clean_data %>%
  filter(`GICS Sub Industry` %in% compared_industries) %>%
  filter(date.filed <= ymd("2020-03-11")) %>%
  unnest_tokens(input = clean_MD_text, output = word) %>%
  count(`GICS Sub Industry`, word) %>%
```

```
  bind_tf_idf(word, `GICS Sub Industry`, n) %>%
  ungroup()
```

Examine the differences of keywords in different industries

```
# Before the pandemic
g1 = industry_tfidf_before %>%
  group_by(`GICS Sub Industry`) %>%
  arrange(`GICS Sub Industry`, desc(tf_idf)) %>%
  top_n(10, tf_idf) %>%
  ungroup() %>%
  mutate(word = fct_reorder(word, tf_idf, .desc = F)) %>%
  ggplot(aes(x = word, y = tf_idf, fill = as.factor(`GICS Sub
Industry`))) +
  geom_col(show.legend = FALSE) +
  labs(x = NULL, y = "Tf-Idf", title = "KEYWORDS BEFORE PANDEM
IC", subtitle = "grouped by GICS sector") +
  facet_wrap(~`GICS Sub Industry`, scales = "free", ncol = 4)
+
  coord_flip()

# After the pandemic
g2 = industry_tfidf_after %>%
  group_by(`GICS Sub Industry`) %>%
  arrange(`GICS Sub Industry`, desc(tf_idf)) %>%
  top_n(10, tf_idf) %>%
  ungroup() %>%
  mutate(word = fct_reorder(word, tf_idf, .desc = F)) %>%
  ggplot(aes(x = word, y = tf_idf, fill = as.factor(`GICS Sub
Industry`))) +
  geom_col(show.legend = FALSE) +
  labs(x = NULL, y = "Tf-Idf", title = "KEYWORDS AFTER PANDEMI
C", subtitle = "grouped by GICS sector") +
  facet_wrap(~`GICS Sub Industry`, scales = "free", ncol = 4)
+
  coord_flip()

ggarrange(g1, g2, labels = c("A", "B"), nrow = 2)
```




```
# clear the memory
rm(year_tfidf, industry_tfidf_after, industry_tfidf_before, in
dustry_tfidf, compared_industries, g1, g2)
```

Part B: Sentiment association with Financial Indicators

Import data

```
clean_data = readRDS("clean_data.rds")
```

Sentiment Analysis on the entire data

analyzeSentiment function

```
anal_Sent_data = clean_data %>%
  select(doc_id, clean_MD_text) %>%
  cbind(analyzeSentiment(clean_data$clean_MD_text)) %>%
  select(doc_id, SentimentGI, SentimentHE, SentimentLM, RatioU
ncertaintyLM, SentimentQDAP)
```

QDAP polarity

```
qdap_data = data.frame()
for (i in 1:nrow(clean_data)){
  temp = data.frame(doc_id = clean_data$doc_id[i]) %>%
    cbind(qdap::polarity(clean_data$clean_MD_text[i]))
  qdap_data = qdap_data %>%
    rbind(temp)
}
rm(temp, i)

qdap_data = qdap_data %>%
  select(doc_id, all.polarity) %>%
  rename(QDAP_SENT = all.polarity)

saveRDS(qdap_data, "qdap_data.rds")

qdap_data = readRDS("qdap_data.rds")
```

Get_sentiments function

Tokenisation

```
clean_tokens = clean_data %>%
  select(doc_id, clean_MD_text) %>%
  unnest_tokens(input = clean_MD_text, output = word)
```

Bing dictionary (only positive and negative)

```
# get_sentiments("bing") %>%
#   count(sentiment)

Bing_data = clean_tokens %>%
  inner_join(get_sentiments("bing"), by = "word")

Bing_data = Bing_data %>%
  group_by(doc_id) %>%
  count(sentiment) %>%
  pivot_wider(names_from = sentiment, values_from = n, values_
fill = 0) %>%
  mutate(BING_SENT = (positive - negative)/((positive + negati
ve)),
         word.count = negative + positive) %>%
  mutate(BING_negative = negative/word.count,
         BING_positive = positive/word.count) %>%
  select(doc_id, BING_positive, BING_negative, BING_SENT)
```

Loughran dictionary (particularly for financial analysis)

```
# get_sentiments("loughran") %>%
#   count(sentiment)
```

```
Loughran_data = clean_tokens %>%
  inner_join(get_sentiments("loughran"), by = "word")
# There is no superfluous words

Loughran_data = Loughran_data %>%
  group_by(doc_id) %>%
  count(sentiment) %>%
  pivot_wider(names_from = sentiment, values_from = n, values_
fill = 0) %>%
  mutate(LM_SENT = (positive - negative)/((positive + negativ
e)),
         word.count = constraining + litigious + negative + po
sitive + uncertainty) %>%
  mutate(LM_constraining = constraining/word.count,
         LM_litigious = litigious/word.count,
         LM_negative = negative/word.count,
         LM_positive = positive/word.count,
         LM_uncertainty = uncertainty/word.count) %>%
  select(doc_id, LM_constraining, LM_litigious, LM_negative, L
M_positive, LM_uncertainty, LM_SENT)
```

NRC dictionary (feelings)

```
# get_sentiments("nrc") %>%
#   count(sentiment)

NRC_data = clean_tokens %>%
  inner_join(get_sentiments("nrc"), by = "word")

NRC_data = NRC_data %>%
  group_by(doc_id) %>%
  count(sentiment) %>%
  pivot_wider(names_from = sentiment, values_from = n, values_
fill = 0) %>%
  mutate(NRC_SENT = (positive - negative)/((positive + negativ
e)),
         word.count = anger + anticipation + disgust + fear +
joy + negative + positive + sadness + surprise + trust) %>%
  mutate(NRC_anger = anger/word.count,
         NRC_anticipation = anticipation/word.count,
         NRC_disgust = disgust/word.count,
         NRC_fear = fear/word.count,
         NRC_joy = joy/word.count,
         NRC_negative = negative/word.count,
         NRC_positive = positive/word.count,
         NRC_sadness = sadness/word.count,
         NRC_surprise = surprise/word.count,
         NRC_trust = trust/word.count) %>%
  select(doc_id, NRC_anger, NRC_anticipation, NRC_disgust, NRC
```

```
_fear, NRC_joy, NRC_negative, NRC_positive, NRC_sadness, NRC_s
urprise, NRC_trust, NRC_SENT)
```

Afinn dictionary (sentiment scores from -5 to 5)

```
# get_sentiments("afinn") %>%
#   count(value)

Afinn_data = clean_tokens %>%
  inner_join(get_sentiments("afinn"), by = "word")

Afinn_data = Afinn_data %>%
  group_by(doc_id) %>%
  summarise(AFINN_SENT = sum(value))

## `summarise()` ungrouping output (override with `.groups` ar
gument)
```

Combine Bing, NRC, Afinn, Loughran labelled data

```
four_in_one_data = Afinn_data %>%
  left_join(Bing_data) %>%
  left_join(Loughran_data) %>%
  left_join(NRC_data)

## Joining, by = "doc_id"
## Joining, by = "doc_id"
## Joining, by = "doc_id"

rm(Afinn_data, Bing_data, Loughran_data, NRC_data)
```

Combine all sentiment data

```
sentiment_data = four_in_one_data %>%
  left_join(qdap_data) %>%
  left_join(anal_Sent_data)

## Joining, by = "doc_id"
## Joining, by = "doc_id"

rm(four_in_one_data, qdap_data, anal_Sent_data, clean_tokens)

# two documents have NaN LM sentiments because their LM_positi
ve and LM_negative are both zero
sentiment_data[is.na(sentiment_data)] = 0
```

Regressions

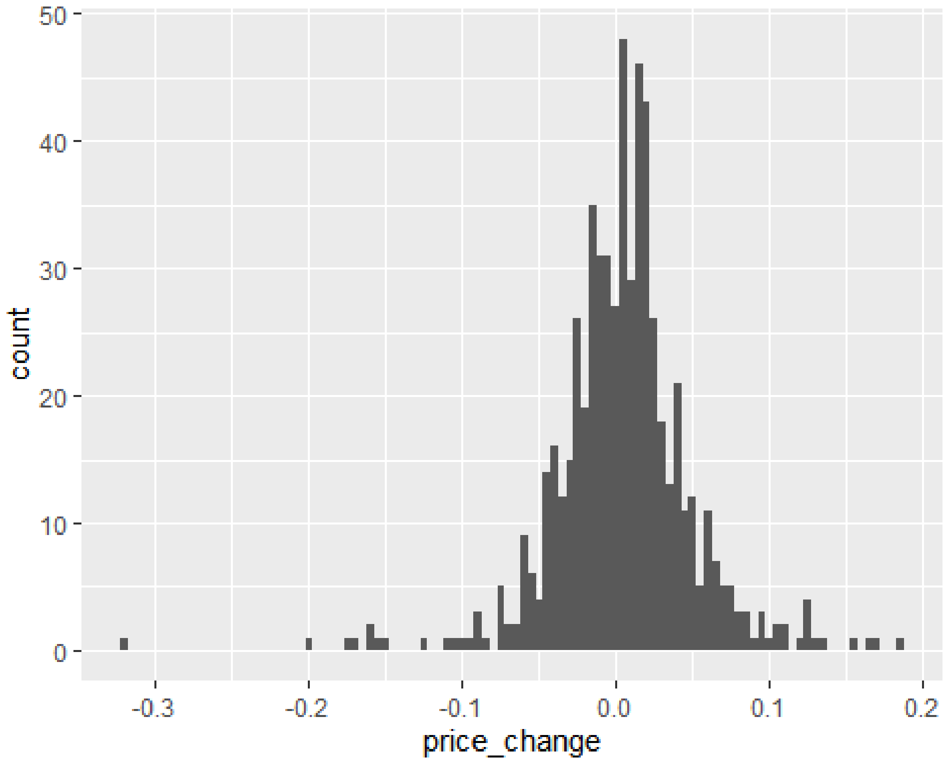
Construct a data frame for regressions

```
regression_data = clean_data %>%
  select(-CIK, -Symbol, -`GICS Sub Industry`, -company.name, -
date.filed, -date_before_filing, -date_after_filing, -price_be
```

```
fore, -price_after, -clean_MD_text) %>%
  left_join(sentiment_data, .) %>%
  select(price_change, everything()) %>%
  na.omit()

## Joining, by = "doc_id"
```

Check the distribution of price changes

```
ggplot(regression_data, aes(price_change)) + geom_histogram(bi
nwidth = 0.005)
```



It is quite a normal distribution so that no transformation will be conducted.

Standardising data

```
z_regression_data = regression_data %>%
  as.matrix() %>%
  scale() %>%
  as.data.frame()
```

Select the best set of variables for predicting stock prices

Build a regression model

```
m_price_by_sent = lm(price_change ~., data = z_regression_data
 %>% select(-doc_id))
```

Use stepAIC function from the MASS package to stepwisely select variables

```r
m_price_by_sent_best = MASS::stepAIC(m_price_by_sent, directio
n = "backward")

summary(m_price_by_sent_best)
```

```
##
## Call:
## lm(formula = price_change ~ NRC_negative + NRC_SENT + Senti
mentHE +
##     RatioUncertaintyLM + num_of_percents + num_of_clean_wor
ds,
##     data = z_regression_data %>% select(-doc_id))
##
## Residuals:
##     Min       1Q   Median       3Q      Max
## -7.1293 -0.4359  0.0307  0.4802  3.7654
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         7.701e-17  4.062e-02   0.000   1.0000
## NRC_negative       -2.851e-01  1.341e-01  -2.125   0.0340 *
## NRC_SENT           -3.004e-01  1.393e-01  -2.156   0.0315 *
## SentimentHE         7.637e-02  4.752e-02   1.607   0.1085
## RatioUncertaintyLM -9.582e-02  4.292e-02  -2.232   0.0260 *
## num_of_percents     1.104e-01  4.804e-02   2.297   0.0219 *
## num_of_clean_words -1.034e-01  4.868e-02  -2.123   0.0341 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
' 1
##
## Residual standard error: 0.9924 on 590 degrees of freedom
## Multiple R-squared:  0.02509,    Adjusted R-squared:  0.015
17
## F-statistic: 2.531 on 6 and 590 DF,  p-value: 0.01993
```

```r
( m_sent_summary = as.data.frame(summary(m_price_by_sent_best)
$coefficient) %>%
    rename(p_value = `Pr(>|t|)`) %>%
    filter(p_value < 0.05) )
```

```
##                       Estimate Std. Error   t value    p_value
## NRC_negative       -0.28506573 0.13412400 -2.125389 0.03396
896
## NRC_SENT           -0.30039134 0.13931717 -2.156169 0.03147
396
## RatioUncertaintyLM -0.09581914 0.04292413 -2.232291 0.02597
030
## num_of_percents     0.11036473 0.04803976  2.297362 0.02194
728
```

```
## num_of_clean_words -0.10337153 0.04868051 -2.123469 0.03413
014
```

```
row.names(m_sent_summary)
```

```
## [1] "NRC_negative"       "NRC_SENT"            "RatioUncerta
intyLM"
## [4] "num_of_percents"    "num_of_clean_words"
```

In this section, I have proven that the correlations between the stock price changes and the sentiment scores are not quite significantly strong. So far, only the NRC negative scores, the NRC sentiment scores, the number of words after tidying, the number of percent signs, and the uncertainty ratios from Loughran are significantly predictive to the changes of stock prices. In fact, only the number of percent signs have a positive effect on the stock price changes. In other words, adding 1 percent sign will lead to 0.1104 units of log stock price changes. All the other variables negatively affect the stock price changes. On the other hand, the overall adjusted R-squared is 0.01517 and the p-value is 0.01993. This shows that the model can only explain 1.52% of variances of the stock price changes. This R squared may still be improved once variables from topic modelling in part C are added.

```
# clear the memory
rm(m_price_by_sent, m_price_by_sent_best, m_sent_final, m_sent
_summary, regression_data, z_regression_data)
```

Inspect Sentiment Changes during the pandemic

Check if there was any dramatic sentiment difference after the pandemic was declared

Construct sentiment data before the pandemic

```
clean_data_before = clean_data %>%
  filter(date.filed <= ymd("2020-03-11"))

clean_tokens_before = clean_data_before %>%
  select(doc_id, clean_MD_text) %>%
  unnest_tokens(input = clean_MD_text, output = word)

# analyzeSentiment
anal_Sent_data_before = clean_data_before %>%
  select(doc_id, clean_MD_text) %>%
  cbind(analyzeSentiment(clean_data_before$clean_MD_text)) %>%
  select(doc_id, SentimentGI, SentimentHE, SentimentLM, RatioU
ncertaintyLM, SentimentQDAP)

# qdap polarity
qdap_data_before = data.frame()
for (i in 1:nrow(clean_data_before)){
  temp = data.frame(doc_id = clean_data_before$doc_id[i]) %>%
```

```r
    cbind(qdap::polarity(clean_data_before$clean_MD_text[i]))
  qdap_data_before = qdap_data_before %>%
    rbind(temp)
}
rm(temp, i)
qdap_data_before = qdap_data_before %>%
  select(doc_id, all.polarity) %>%
  rename(QDAP_SENT = all.polarity)
saveRDS(qdap_data_before, "qdap_data_before.rds")

qdap_data_before = readRDS("qdap_data_before.rds")

# bing
Bing_data_before = clean_tokens_before %>%
  inner_join(get_sentiments("bing"), by = "word") %>%
  group_by(doc_id) %>%
  count(sentiment) %>%
  pivot_wider(names_from = sentiment, values_from = n, values_
fill = 0) %>%
  mutate(BING_SENT = (positive - negative)/((positive + negati
ve))) %>%
  select(doc_id, BING_SENT)

# Loughran
Loughran_data_before = clean_tokens_before %>%
  inner_join(get_sentiments("loughran"), by = "word") %>%
  group_by(doc_id) %>%
  count(sentiment) %>%
  pivot_wider(names_from = sentiment, values_from = n, values_
fill = 0) %>%
  mutate(LM_SENT = (positive - negative)/((positive + negativ
e))) %>%
  select(doc_id, LM_SENT)
Loughran_data_before[is.na(Loughran_data_before)] = 0

# NRC
NRC_data_before = clean_tokens_before %>%
  inner_join(get_sentiments("nrc"), by = "word") %>%
  group_by(doc_id) %>%
  count(sentiment) %>%
  pivot_wider(names_from = sentiment, values_from = n, values_
fill = 0) %>%
  mutate(NRC_SENT = (positive - negative)/((positive + negativ
e))) %>%
  select(doc_id, NRC_SENT)

# combine all sentiment data
sentiment_data_before = anal_Sent_data_before %>%
  left_join(qdap_data_before) %>%
```

```
  left_join(Bing_data_before) %>%
  left_join(Loughran_data_before) %>%
  left_join(NRC_data_before)

## Joining, by = "doc_id"
## Joining, by = "doc_id"
## Joining, by = "doc_id"
## Joining, by = "doc_id"

rm(qdap_data_before, Bing_data_before, Loughran_data_before, N
RC_data_before, clean_tokens_before, anal_Sent_data_before)
```

Construct sentiment data after the pandemic

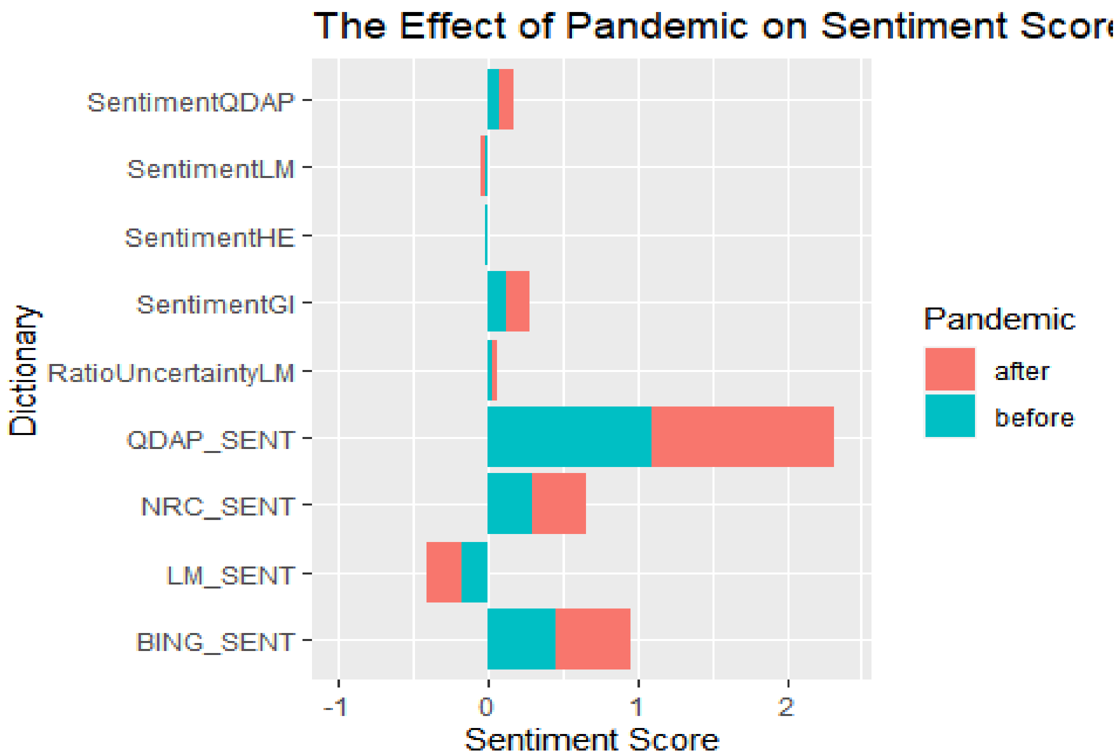
```
clean_data_after = clean_data %>%
  filter(date.filed > ymd("2020-03-11"))

clean_tokens_after = clean_data_after %>%
  select(doc_id, clean_MD_text) %>%
  unnest_tokens(input = clean_MD_text, output = word)

# analyzeSentiment
anal_Sent_data_after = clean_data_after %>%
  select(doc_id, clean_MD_text) %>%
  cbind(analyzeSentiment(clean_data_after$clean_MD_text)) %>%
  select(doc_id, SentimentGI, SentimentHE, SentimentLM, RatioU
ncertaintyLM, SentimentQDAP)

# qdap polarity
qdap_data_after = data.frame()
for (i in 1:nrow(clean_data_after)){
  temp = data.frame(doc_id = clean_data_after$doc_id[i]) %>%
    cbind(qdap::polarity(clean_data_after$clean_MD_text[i]))
  qdap_data_after = qdap_data_after %>%
    rbind(temp)
}
rm(temp, i)
qdap_data_after = qdap_data_after %>%
  select(doc_id, all.polarity) %>%
  rename(QDAP_SENT = all.polarity)
saveRDS(qdap_data_after, "qdap_data_after.rds")

qdap_data_after = readRDS("qdap_data_after.rds")

# bing
Bing_data_after = clean_tokens_after %>%
  inner_join(get_sentiments("bing"), by = "word") %>%
  group_by(doc_id) %>%
  count(sentiment) %>%
  pivot_wider(names_from = sentiment, values_from = n, values_
```

```
fill = 0) %>%
  mutate(BING_SENT = (positive - negative)/((positive + negati
ve))) %>%
  select(doc_id, BING_SENT)

# Loughran
Loughran_data_after = clean_tokens_after %>%
  inner_join(get_sentiments("loughran"), by = "word") %>%
  group_by(doc_id) %>%
  count(sentiment) %>%
  pivot_wider(names_from = sentiment, values_from = n, values_
fill = 0) %>%
  mutate(LM_SENT = (positive - negative)/((positive + negativ
e))) %>%
  select(doc_id, LM_SENT)

# NRC
NRC_data_after = clean_tokens_after %>%
  inner_join(get_sentiments("nrc"), by = "word") %>%
  group_by(doc_id) %>%
  count(sentiment) %>%
  pivot_wider(names_from = sentiment, values_from = n, values_
fill = 0) %>%
  mutate(NRC_SENT = (positive - negative)/((positive + negativ
e))) %>%
  select(doc_id, NRC_SENT)

# combine all sentiment data
sentiment_data_after = anal_Sent_data_after %>%
  left_join(qdap_data_after) %>%
  left_join(Bing_data_after) %>%
  left_join(Loughran_data_after) %>%
  left_join(NRC_data_after)

## Joining, by = "doc_id"
## Joining, by = "doc_id"
## Joining, by = "doc_id"
## Joining, by = "doc_id"

rm(qdap_data_after, Bing_data_after, Loughran_data_after, NRC_
data_after, clean_tokens_after, anal_Sent_data_after)
```

Compare

```
sentiment_data_before %>%
  pivot_longer(-doc_id, names_to = "Dictionary", values_to = "
Sentiment") %>%
  group_by(Dictionary) %>%
  summarise(AVG_Sentiment_before = mean(Sentiment)) %>%
  inner_join(
```

```
    sentiment_data_after %>%
      pivot_longer(-doc_id, names_to = "Dictionary", values_to
 = "Sentiment") %>%
      group_by(Dictionary) %>%
      summarise(AVG_Sentiment_after = mean(Sentiment)), by = "
Dictionary") %>%
  pivot_longer(-Dictionary, names_to = "Pandemic", values_to =
 "Sentiment") %>%
  mutate(Pandemic = gsub("AVG_Sentiment_", "", Pandemic)) %>%
  ggplot(aes(x = Dictionary, y = Sentiment, fill = as.factor(P
andemic))) +
  geom_col() +
  labs(title = "The Effect of Pandemic on Sentiment Scores", x
 = "Dictionary", y = "Sentiment Score", fill = "Pandemic") +
  ylim(-1, 2.4) +
  coord_flip()
```



As shown in the plot, the management discussions became more emotional. Magnitudes of sentiment scores after the pandemic are all significantly larger than those before the pandemic. However, the result is still unstable as only few management discussions are available so far. Therefore, collecting and analysing more data after the pandemic can be further developed in the future research.

```
# clear the memory
rm(clean_data_after, clean_data_before, sentiment_data_after,
sentiment_data_before)
```

Part C: Topic Modelling and Latent Dirichlet allocation

Import the dataset

```
clean_data = readRDS("clean_data.rds")
```

Text processing

```
# Stop words have already been removed
MD_text_processed = textProcessor(clean_data$clean_MD_text,
                                  metadata = clean_data,
                                  stem = F)

# setting the threshold
threshold = round(1/100 * length(MD_text_processed$documents),
0)

MD_text_out = prepDocuments(MD_text_processed$documents,
                            MD_text_processed$vocab,
                            MD_text_processed$meta,
                            lower.thresh = threshold)

rm(threshold)
```

No document is removed.

STM modelling

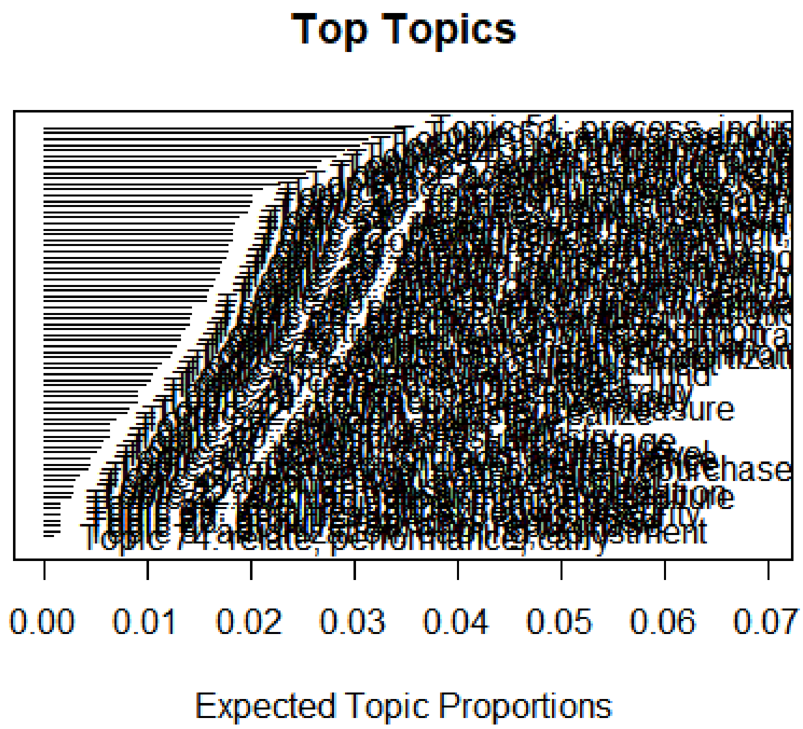
Find the best value of k

Adapt the concept of gradient descent algorithm to find the locally optimal number of topics. It is an unsupervised process. Firstly, setting k = 0 in the stm function is used to derive the initial feasible solution. Later, the searchK function is implemented to search better models close to the initial solution.

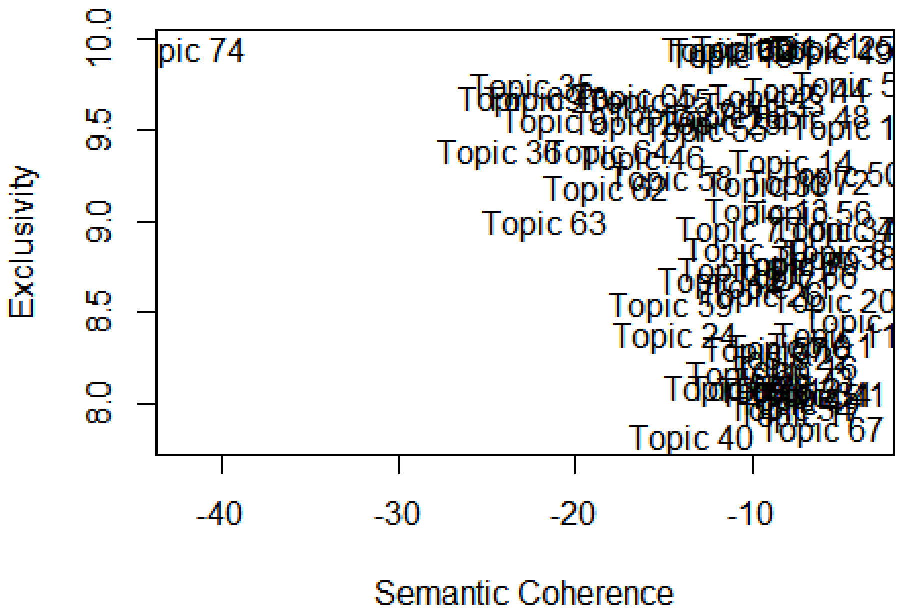
Set K = 0 in the stm function

```
stm_unsupervised = stm(documents = MD_text_out$documents,
                       vocab = MD_text_out$vocab,
                       K = 0,
                       prevalence = ~price_change,
                       max.em.its = 75,
                       data = MD_text_out$meta,
                       reportevery = 5,
                       sigma.prior = 0.7,
                       init.type = "Spectral")
```

```r
saveRDS(stm_unsupervised, "stm_unsupervised.rds")

stm_unsupervised = readRDS("stm_unsupervised.rds")

# Check the topic model and topic proportions
plot(stm_unsupervised)
```




```r
# Inspect the topic distribution
topicQuality(stm_unsupervised, document = MD_text_out$document
s)
```



Choosing a model by exclusivity and Semantic Coherence is a tradeoff. A model with more topics tend to have a lower semantic coherence and a higher exclusivity

Review FREX words for every topic

```
topicProportions = colMeans(stm_unsupervised$theta)
stm_unsupervised_summary = summary(stm_unsupervised)
```

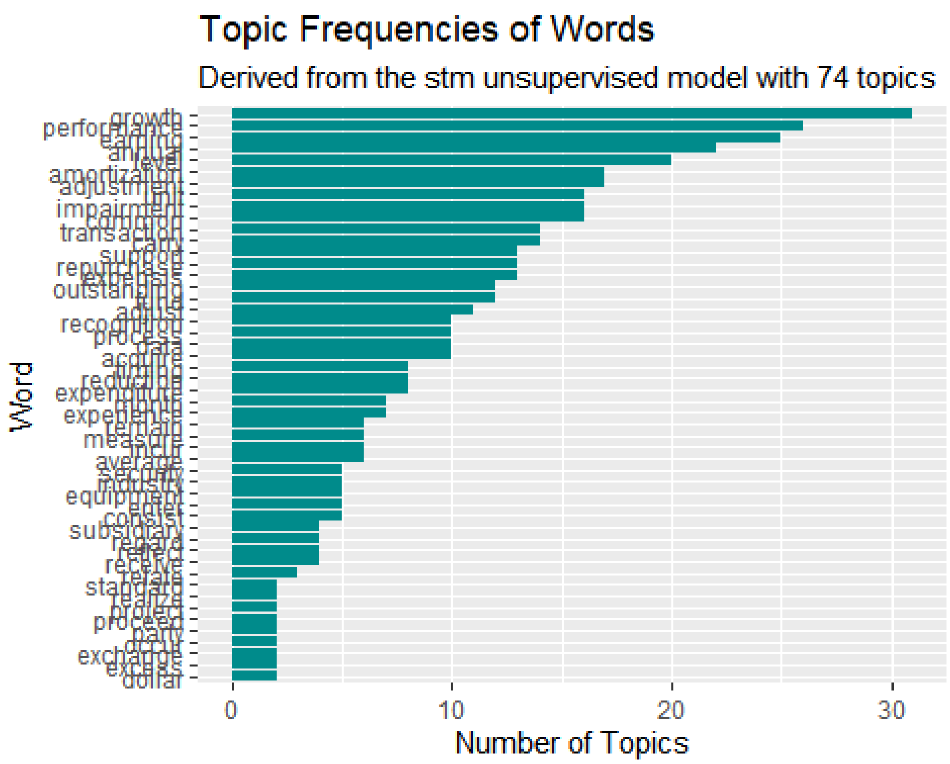
```
## A topic model with 74 topics, 597 documents and a 1366 word
 dictionary.
```

```
unsupervised_frex = data.frame()
for (i in 1:length(stm_unsupervised_summary$topicnums)){
  row_here = data.frame(topic_num = stm_unsupervised_summary$t
opicnums[i],
                        proportion = 100 * round(topicProporti
ons[i], 4),
                        frex_words = paste(stm_unsupervised_su
mmary$frex[i, 1:7], collapse = ", "))
  unsupervised_frex = rbind(row_here, unsupervised_frex)
}
rm(i, row_here)
```

The unsupervised model suggests 74 topics. However, it seems there are many overlaps between topics.

Detect stop words which appear in many topics

```r
stm_unsupervised_high_prob_words = data.frame(stm_unsupervised
_summary$prob)
colnames(stm_unsupervised_high_prob_words) = paste0("word_",
1:7)
stm_unsupervised_high_prob_words$topic = paste0("topic_", 1:le
ngth(stm_unsupervised_summary$topicnums))
stm_unsupervised_high_prob_words = stm_unsupervised_high_prob_
words %>%
  unite("Text", word_1:word_7, remove = TRUE, sep = " ") %>%
  unnest_tokens(input = Text, output = word) %>%
  count(topic, word) %>%
  bind_tf_idf(word, topic, n) %>%
  mutate(df = 1/exp(idf)) %>%
  select(word, df) %>%
  unique() %>%
  mutate(num_of_topics = df * length(stm_unsupervised_summary$
topicnums)) %>%
  arrange(desc(df))

stm_unsupervised_high_prob_words %>%
  top_n(50, df) %>%
  mutate(word = fct_reorder(word, df, .desc = FALSE)) %>%
  ggplot(aes(x = word, y = num_of_topics)) +
  geom_col(show.legend = FALSE, fill = "darkcyan") +
  labs(title = "Topic Frequencies of Words", subtitle = paste0
("Derived from the stm unsupervised model with ", length(stm_u
nsupervised_summary$topicnums), " topics"), y = "Number of Top
ics", x = "Word") +
  coord_flip()
```

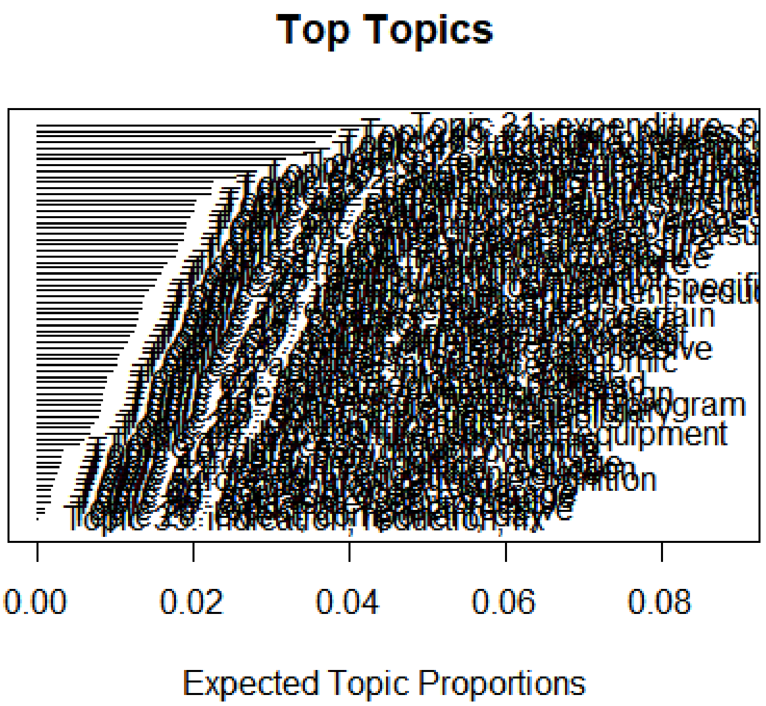



```
topic_stopwords = stm_unsupervised_high_prob_words %>%
  filter(df > 0.15) %>%
  mutate(lexicon = "topic_df") %>%
  select(word, lexicon)
```

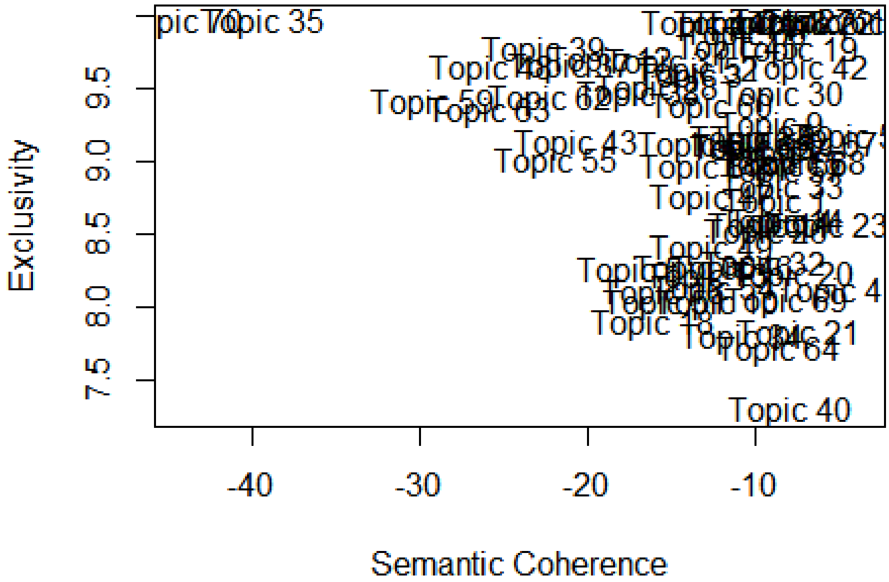
Second Iteration: remove stop words on the topic level

```
MD_text_processed2 = textProcessor(clean_data$clean_MD_text,
                                   metadata = clean_data,
                                   customstopwords = topic_stopwords$wo
rd,
                                   stem = F)

# setting the threshold
threshold = round(1/100 * length(MD_text_processed2$document
s),0)
MD_text_out2 = prepDocuments(MD_text_processed2$documents,
                             MD_text_processed2$vocab,
                             MD_text_processed2$meta,
                             lower.thresh = threshold)

rm(threshold)

# unsupervised
stm_unsupervised2 = stm(documents = MD_text_out2$documents,
                        vocab = MD_text_out2$vocab,
                        K = 0,
                        prevalence = ~price_change,
                        max.em.its = 75,
```

```r
                      data = MD_text_out2$meta,
                      reportevery = 5,
                      sigma.prior = 0.7,
                      init.type = "Spectral")
saveRDS(stm_unsupervised2, "stm_unsupervised2.rds")

stm_unsupervised2 = readRDS("stm_unsupervised2.rds")
# Check the topic model and topic proportions
plot(stm_unsupervised2)
```

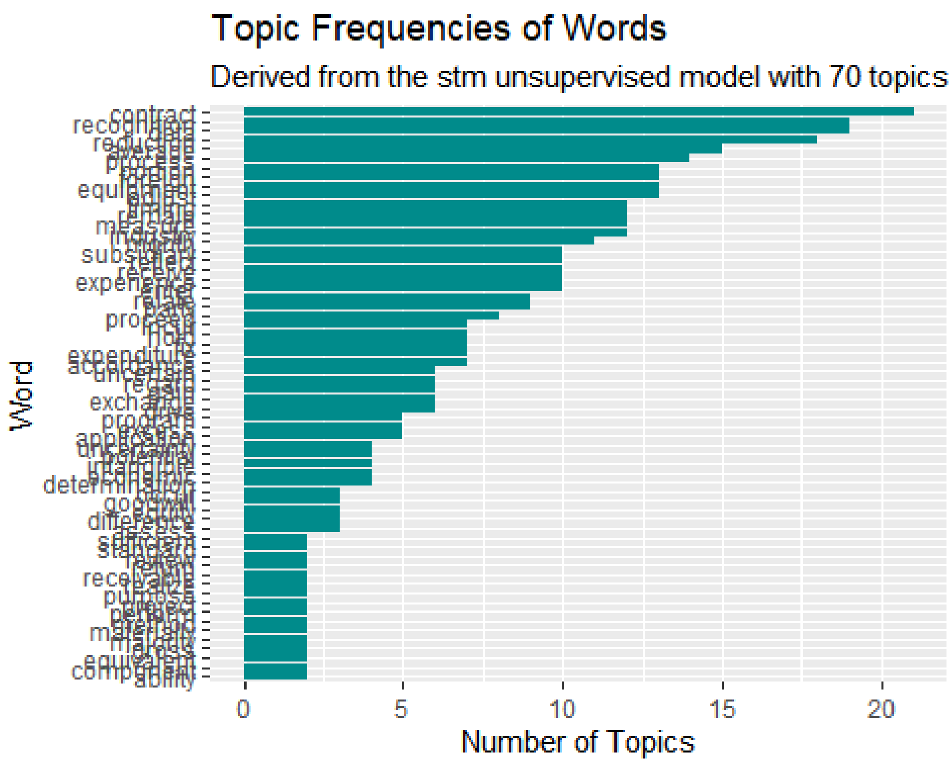



```r
# Inspect the topic distribution
topicQuality(stm_unsupervised2, document = MD_text_out2$docume
nts)
```




```
topicProportions2 = colMeans(stm_unsupervised2$theta)
stm_unsupervised_summary2 = summary(stm_unsupervised2)

unsupervised_frex2 = data.frame()
for (i in 1:length(stm_unsupervised_summary2$topicnums)){
  row_here = data.frame(topic_num = stm_unsupervised_summary2$
topicnums[i],
                        proportion = 100 * round(topicProporti
ons2[i], 4),
                        frex_words = paste(stm_unsupervised_su
mmary2$frex[i, 1:7], collapse = ", "))
  unsupervised_frex2 = rbind(row_here, unsupervised_frex2)
}
rm(i, row_here)

stm_unsupervised_high_prob_words2 = data.frame(stm_unsupervise
d_summary2$prob)
colnames(stm_unsupervised_high_prob_words2) = paste0("word_",
1:7)
stm_unsupervised_high_prob_words2$topic = paste0("topic_", 1:l
ength(stm_unsupervised_summary2$topicnums))
stm_unsupervised_high_prob_words2 = stm_unsupervised_high_prob
_words2 %>%
  unite("Text", word_1:word_7, remove = TRUE, sep = " ") %>%
  unnest_tokens(input = Text, output = word) %>%
  count(topic, word) %>%
  bind_tf_idf(word, topic, n) %>%
  mutate(df = 1/exp(idf)) %>%
```

```
  select(word, df) %>%
  unique() %>%
  mutate(num_of_topics = df * length(stm_unsupervised_summary2
$topicnums)) %>%
  arrange(desc(df))

stm_unsupervised_high_prob_words2 %>%
  top_n(50, df) %>%
  mutate(word = fct_reorder(word, df, .desc = FALSE)) %>%
  ggplot(aes(x = word, y = num_of_topics)) +
  geom_col(show.legend = FALSE, fill = "darkcyan") +
  labs(title = "Topic Frequencies of Words", subtitle = paste0
("Derived from the stm unsupervised model with ", length(stm_u
nsupervised_summary2$topicnums), " topics"), y = "Number of To
pics", x = "Word") +
  coord_flip()
```



It looks better than the previous model as the exclusivity increases. Another, there are fewer overlaps in this model as all topic frequencies of words are not larger than 0.3. Therefore, 70-topic result is the initial solution.

```
# clear the memory
rm(MD_text_out, MD_text_processed, MD_text_out2, MD_text_proce
ssed2, stm_unsupervised, stm_unsupervised_high_prob_words, stm
_unsupervised_high_prob_words2, stm_unsupervised_summary, stm_
unsupervised_summary2, stm_unsupervised2, unsupervised_frex, u
nsupervised_frex2, topicProportions, topicProportions2)
```

SearchK approach

Setting the parametre grid

```
grid = 60:80
```

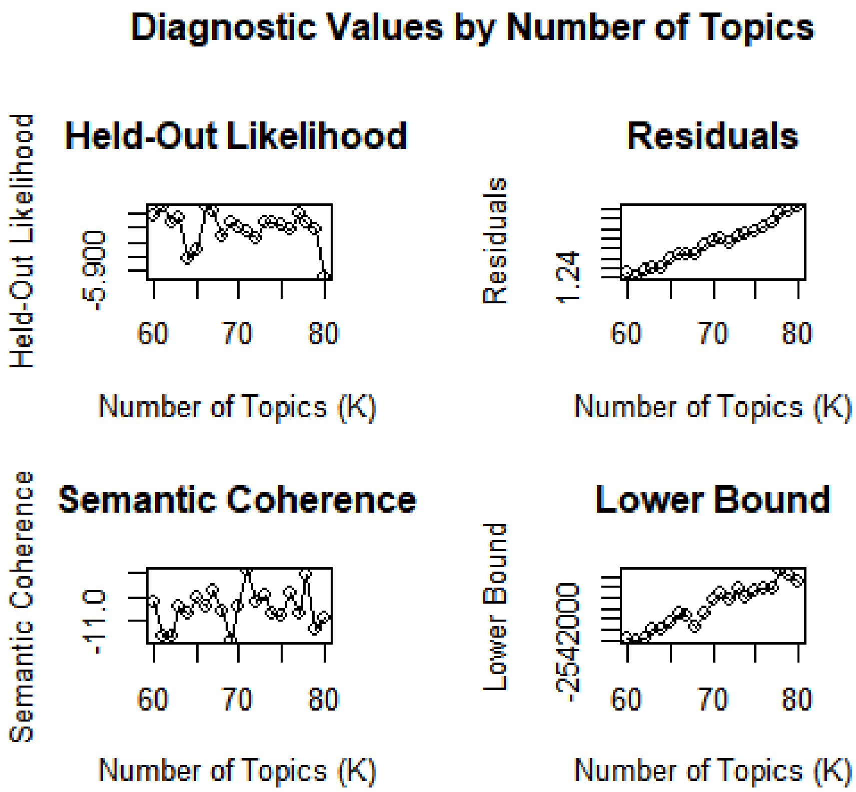
Preprocessing

```
initial_processed = textProcessor(clean_data$clean_MD_text,
                                  metadata = clean_data,
                                  customstopwords = topic_stopwords$wo
rd,
                                  stem = F)

# setting the threshold
threshold = round(1/100 * length(initial_processed$documents),
0)
initial_out = prepDocuments(initial_processed$documents,
                            initial_processed$vocab,
                            initial_processed$meta,
                            lower.thresh = threshold)

rm(threshold)
```

No document was removed.

Start searching local optimal number of topics

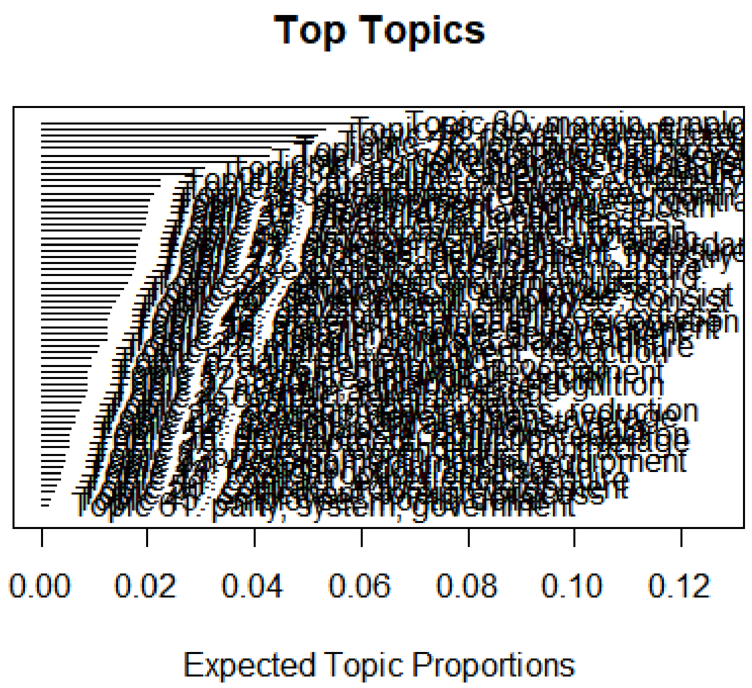
```
num_topics = searchK(initial_out$documents,
                     initial_out$vocab,
                     K = grid)

saveRDS(num_topics, "num_topics.rds")

rm(grid)

num_topics = readRDS("num_topics.rds")

plot(num_topics)
```



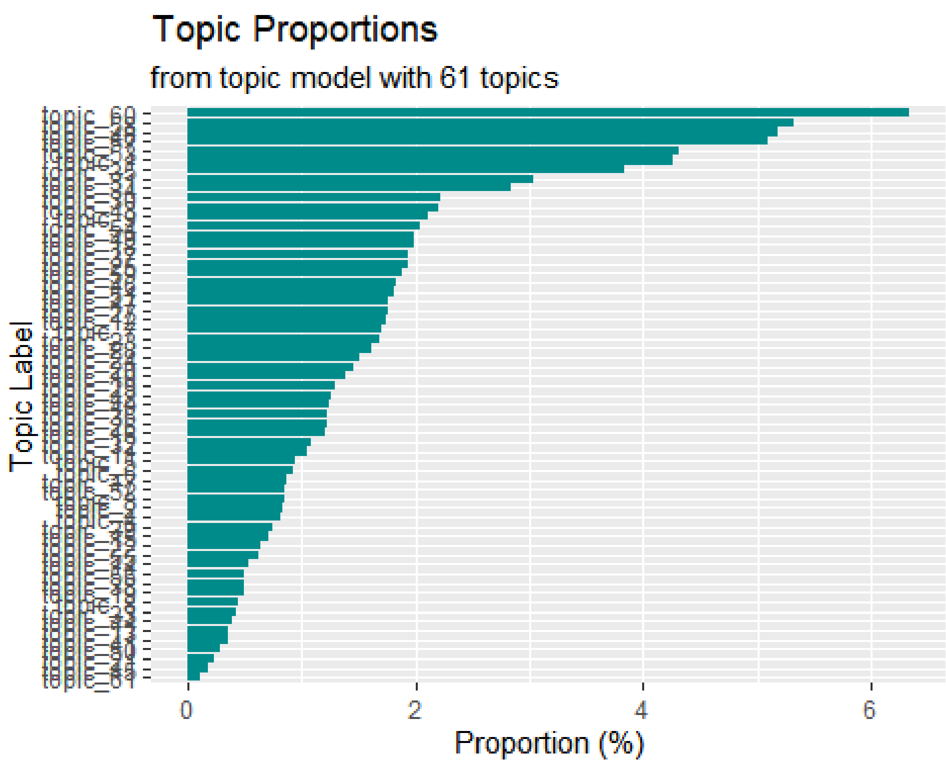
According to the plot, 61 will be set as the final best number of topics because the held-out likelihood is the highest, the semantic coherence and the residuals are the lowest at K = 61.

Final topic model with 61 topics

```
final_processed = initial_processed
final_out = initial_out

rm(initial_out, initial_processed)

final_topic_model = stm(documents = final_out$documents,
                   vocab = final_out$vocab,
                   K = 61,
                   prevalence =~ price_change,
                   max.em.its = 150,
                   data = final_out$meta,
                   reportevery = 5,
                   sigma.prior = 0.7,
                   init.type = "Spectral")
saveRDS(final_topic_model, "final_topic_model.rds")

final_topic_model = readRDS("final_topic_model.rds")

# plot the model
plot(final_topic_model)
```

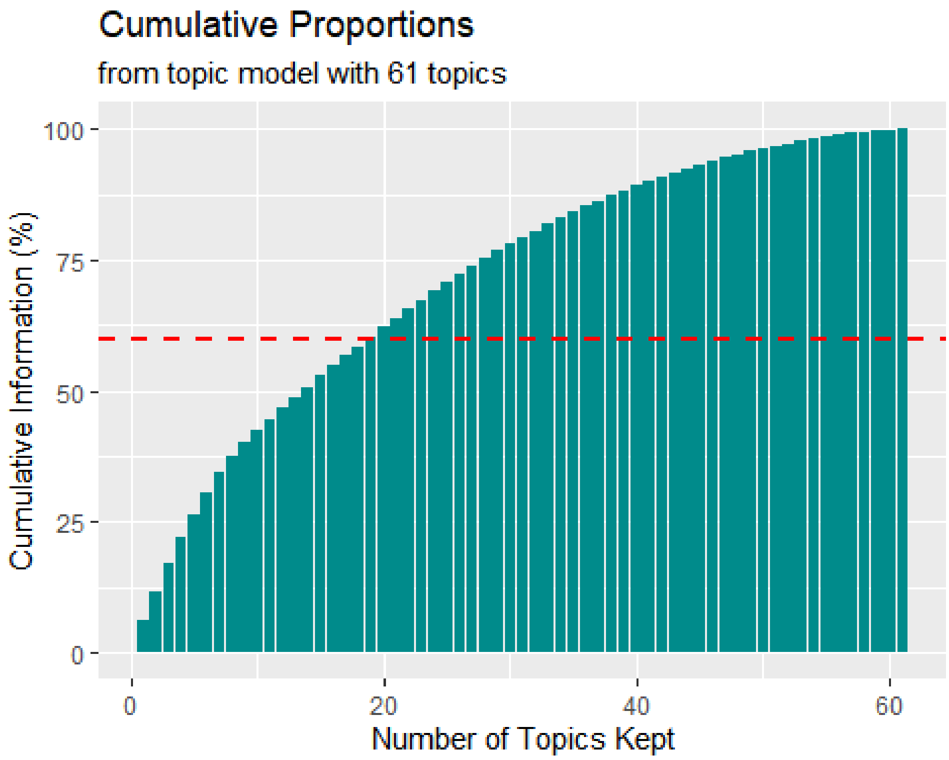



```
# construct a model data frame with frex words and proportions
final_topic_model_summary = summary(final_topic_model)

final_topic_proportions = colMeans(final_topic_model$theta)
topic_labels = paste0("topic_",1:length(final_topic_model_summ
ary$topicnums))
final_topic_labels = data.frame()
for(i in 1:length(final_topic_model_summary$topicnums)){
   row_here = data.frame(topicnum = final_topic_model_summary$
topicnums[i],
                         topic_label = topic_labels[i],
                         proportion = 100*round(final_topic_pr
oportions[i],4),
                         frex_words = paste(final_topic_model_
summary$frex[i,1:7], collapse = ", "))
   final_topic_labels = rbind(row_here, final_topic_labels)
}
rm(row_here, topic_labels, final_topic_proportions, i)


final_topic_labels %>%
  mutate(topic_label = fct_reorder(topic_label, proportion, .d
esc = FALSE)) %>%
  ggplot(aes(x = topic_label, y = proportion)) +
  geom_col(show.legend = FALSE, fill = "darkcyan") +
  labs(title = "Topic Proportions", subtitle = "from topic mod
el with 61 topics", x = "Topic Label", y = "Proportion (%)") +
  coord_flip()
```



It is not easy to interpret all 61 topics. Hence, I have to reduce the number of topics. It is dimension reduction. According to the lecture materials of factor analysis from the module Advanced Data Analysis, I can keep first several factors (topics) that explain about 60% of information contained in the original variables if the purpose is to reduce the data dimension.

```
cumulative_proportions = data.frame()
for (i in 1:61){
  a = final_topic_labels %>% arrange(desc(proportion))
  b = a$topic_label[i]
  cumulative_proportion = sum(a$proportion[1:i])
  temp = data.frame(number_of_topic_kept = i, new_add_topic =
b, cumulative_proportion)
  cumulative_proportions = cumulative_proportions %>%
    rbind(temp)
}
rm(i, a, b, temp, cumulative_proportion)

ggplot(cumulative_proportions, aes(x = number_of_topic_kept, y
 = cumulative_proportion)) +
  geom_col(show.legend = FALSE, fill = "darkcyan") +
  geom_hline(yintercept = 60, linetype="dashed", color = "red
", size = 1) +
  labs(title = "Cumulative Proportions", subtitle = "from topi
c model with 61 topics", x = "Number of Topics Kept", y = "Cum
ulative Information (%)")
```




```
# get label of the top 19 topics
top_19_topic_labels = cumulative_proportions$new_add_topic[1:1
9]
```

As shown in the plot, the top 19 topics can explain 60.38% of the overall information.

Interpretation of the top 19 topics

```
(final_topic_labels %>%
  arrange(desc(proportion)) %>%
  top_n(19, proportion) %>%
  select(-topicnum, -proportion))

##    topic_label
## 1     topic_60
## 2     topic_58
## 3     topic_48
## 4     topic_57
## 5     topic_53
## 6      topic_1
## 7     topic_35
## 8     topic_31
## 9     topic_34
## 10    topic_30
## 11    topic_43
## 12     topic_9
## 13    topic_54
## 14    topic_49
```

```
## 15    topic_18
## 16    topic_37
## 17    topic_25
## 18    topic_59
## 19    topic_46
##
        frex_words
## 1        margin, potential, measure, investing, taxes, quar
terly, recognition
## 2  development, exchange, security, investing, settlement,
december, currency
## 3          fiscal, determination, expenditure, limit, offe
r, prepare, process
## 4             foreign, fee, excess, expenditure, decline, e
quipment, currency
## 5        intangible, integration, goodwill, occur, process,
 development, data
## 6                      contract, process, involve, hold, ti
ming, data, option
## 7         timing, recognition, review, drive, multiple, dev
elopment, solution
## 8        adjust, average, board, proceed, expenditure, inta
ngible, accordance
## 9       december, average, exchange, acquire, forward, unce
rtain, recognition
## 10          segment, resources, volume, measure, fee, deter
mination, industry
## 11           excess, exchange, forward, retirement, foreig
n, currency, entity
## 12      development, month, timing, employee, combination, i
nvesting, resource
## 13      forfeiture, form, consist, receivable, accordance, d
ecline, materially
## 14                        source, month, billing, exchange,
line, regard, data
## 15                  gross, regard, dividend, accordance, mo
nth, form, acquire
## 16       measure, gross, assess, potential, allowance, juris
diction, uncertain
## 17                   consist, model, reflect, december, lin
e, pricing, profit
## 18        intangible, currency, foreign, enter, software, re
ceivable, exercise
## 19    december, uncertain, materially, program, employee, m
argin, application

# clear the memory
rm(final_out, final_processed, final_topic_labels, final_topic
```

```
_model_summary, num_topics, topic_stopwords, cumulative_propor
tions)
```

Regressions

Construct topic features

```
# document-topic matrix (contain 61 topics)
final_model_theta = as.data.frame(final_topic_model$theta)
colnames(final_model_theta) = paste0("topic_",1:61)

top_19_topic_theta = final_model_theta %>%
  select(top_19_topic_labels)

## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(top_19_topic_labels)` instead of `top_19_topi
c_labels` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-
vector.html>.
## This message is displayed once per session.
```

The predictability of topic features on estimating the stock price.

```
topic_regression_data = clean_data %>%
  select(doc_id, price_change) %>%
  cbind(top_19_topic_theta)

m_topic_test = lm(price_change ~., data = topic_regression_dat
a %>% select(-doc_id))
summary(m_topic_test)

##
## Call:
## lm(formula = price_change ~ ., data = topic_regression_data
 %>%
##     select(-doc_id))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.32823 -0.02170  0.00107  0.01940  0.17314
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0027969  0.0033625   0.832   0.4059
## topic_60     0.0059414  0.0154003   0.386   0.6998
## topic_58     0.0089937  0.0159648   0.563   0.5734
## topic_48     0.0183045  0.0141950   1.290   0.1977
## topic_57     0.0005596  0.0149176   0.038   0.9701
## topic_53     0.0122889  0.0149959   0.819   0.4128
## topic_1     -0.0007266  0.0170241  -0.043   0.9660
```

```
## topic_35    -0.0001680  0.0164376  -0.010   0.9918
## topic_31     0.0207667  0.0189708   1.095   0.2741
## topic_34    -0.0723611  0.0332205  -2.178   0.0298 *
## topic_30     0.0021585  0.0153912   0.140   0.8885
## topic_43     0.0243047  0.0161480   1.505   0.1328
## topic_9      0.0101585  0.0166300   0.611   0.5415
## topic_54     0.0076530  0.0166149   0.461   0.6453
## topic_49    -0.0130561  0.0166176  -0.786   0.4324
## topic_18    -0.0142527  0.0154617  -0.922   0.3570
## topic_37    -0.0062416  0.0186866  -0.334   0.7385
## topic_25     0.0375220  0.0172773   2.172   0.0303 *
## topic_59    -0.0220973  0.0216069  -1.023   0.3069
## topic_46    -0.0051576  0.0185753  -0.278   0.7814
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
' 1
##
## Residual standard error: 0.04594 on 577 degrees of freedom
## Multiple R-squared:  0.03571,    Adjusted R-squared:  0.003
953
## F-statistic: 1.124 on 19 and 577 DF,  p-value: 0.3212

This is not a significant model as its p-value (0.3212) is lar
ger than 0.05
```

Predict the stock price changes with all derived features

Combine all features

```
overall_features_regression_data = clean_data %>%
  select(doc_id, price_change, ARI, Flesch.Kincaid, Linsear.Wr
ite, formality, total_num_of_words, num_of_puncts, num_of_doll
ar, num_of_percents, num_of_capitals, num_of_digits, num_of_na
mes, num_of_clean_words) %>%
  left_join(sentiment_data, by = "doc_id") %>%
  cbind(top_19_topic_theta)
```

Standardising data

```
z_overall_features_regression_data = overall_features_regressi
on_data %>%
  as.matrix() %>%
  scale() %>%
  as.data.frame()
```

Modelling

```
m_overall = lm(price_change~., data = z_overall_features_regre
ssion_data %>% select(-doc_id))
summary(m_overall)
```

```
##
## Call:
## lm(formula = price_change ~ ., data = z_overall_features_re
gression_data %>%
##     select(-doc_id))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9352 -0.4574  0.0457  0.4831  3.4056
##
## Coefficients: (4 not defined because of singularities)
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -2.012e-17  4.104e-02   0.000   1.0000
## ARI                   3.227e-02  4.039e-01   0.080   0.9364
## Flesch.Kincaid       -2.073e-01  4.733e-01  -0.438   0.6615
## Linsear.Write         1.713e-01  1.879e-01   0.912   0.3622
## formality             6.066e-02  6.697e-02   0.906   0.3655
## total_num_of_words -9.919e-02  3.635e-01  -0.273   0.7850
## num_of_puncts         5.917e-02  2.591e-01   0.228   0.8195
## num_of_dollar         2.155e-01  1.150e-01   1.874   0.0614 .
## num_of_percents       1.441e-01  5.868e-02   2.455   0.0144 *
## num_of_capitals      -5.188e-02  6.736e-02  -0.770   0.4416
## num_of_digits        -2.177e-01  1.889e-01  -1.153   0.2496
## num_of_names         -4.567e-02  4.841e-02  -0.943   0.3459
## num_of_clean_words -5.106e-02  3.314e-01  -0.154   0.8776
## AFINN_SENT           -6.955e-02  1.368e-01  -0.508   0.6113
## BING_positive        -1.703e-01  1.023e-01  -1.665   0.0966 .
## BING_negative                NA         NA      NA       NA
## BING_SENT                    NA         NA      NA       NA
## LM_constraining       1.899e-02  8.228e-02   0.231   0.8175
## LM_litigious         -7.718e-02  1.079e-01  -0.716   0.4746
## LM_negative          -7.917e-02  1.285e-01  -0.616   0.5381
## LM_positive          -6.267e-02  1.306e-01  -0.480   0.6316
## LM_uncertainty               NA         NA      NA       NA
## LM_SENT               1.457e-02  1.532e-01   0.095   0.9243
## NRC_anger             2.712e-02  5.772e-02   0.470   0.6386
## NRC_anticipation      4.131e-04  5.674e-02   0.007   0.9942
## NRC_disgust          -4.228e-02  6.786e-02  -0.623   0.5335
## NRC_fear             -3.188e-02  6.413e-02  -0.497   0.6193
## NRC_joy               5.658e-02  6.551e-02   0.864   0.3881
## NRC_negative         -5.826e-01  4.676e-01  -1.246   0.2133
## NRC_positive          6.318e-02  2.489e-01   0.254   0.7997
## NRC_sadness           1.195e-01  8.017e-02   1.490   0.1368
## NRC_surprise         -4.533e-03  6.019e-02  -0.075   0.9400
## NRC_trust                    NA         NA      NA       NA
## NRC_SENT             -5.556e-01  6.130e-01  -0.906   0.3652
## QDAP_SENT             1.968e-01  1.266e-01   1.554   0.1207
## SentimentGI          -5.243e-02  7.755e-02  -0.676   0.4993
## SentimentHE           4.697e-02  6.612e-02   0.710   0.4778
```

```
## SentimentLM          7.002e-03  9.008e-02   0.078   0.9381  
## RatioUncertaintyLM -1.193e-01  7.782e-02  -1.533   0.1259  
## SentimentQDAP       5.704e-02  8.532e-02   0.668   0.5041  
## topic_60           -5.155e-02  6.112e-02  -0.843   0.3994  
## topic_58           -1.701e-02  5.338e-02  -0.319   0.7501  
## topic_48            2.957e-02  5.896e-02   0.502   0.6162  
## topic_57            4.692e-02  6.193e-02   0.758   0.4489  
## topic_53            3.387e-02  5.914e-02   0.573   0.5671  
## topic_1            -1.564e-02  5.467e-02  -0.286   0.7750  
## topic_35           -5.248e-02  5.114e-02  -1.026   0.3053  
## topic_31            7.096e-02  5.079e-02   1.397   0.1629  
## topic_34           -6.305e-02  5.058e-02  -1.247   0.2131  
## topic_30            9.721e-03  5.126e-02   0.190   0.8497  
## topic_43            1.211e-01  5.104e-02   2.373   0.0180 *
## topic_9             1.070e-02  4.913e-02   0.218   0.8277  
## topic_54            8.532e-02  4.841e-02   1.763   0.0785 .
## topic_49           -5.715e-02  5.300e-02  -1.078   0.2814  
## topic_18           -3.646e-02  5.677e-02  -0.642   0.5210  
## topic_37           -6.267e-02  4.973e-02  -1.260   0.2081  
## topic_25            7.391e-02  4.974e-02   1.486   0.1379  
## topic_59           -8.568e-02  4.956e-02  -1.729   0.0844 .
## topic_46           -4.571e-02  4.688e-02  -0.975   0.3300  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
' 1
## 
## Residual standard error: 1.003 on 542 degrees of freedom
## Multiple R-squared:  0.08537,    Adjusted R-squared:  -0.00
5761 
## F-statistic: 0.9368 on 54 and 542 DF,  p-value: 0.6045
```

The adjusted R squared is quite small (-0.005761), and the p-value is 0.6045, which indicates that the model is not significantly predictive. Hence, stepAIC must be applied to find the optimal model.

Choose the best model with stepAIC

```
m_overall_best = MASS::stepAIC(m_overall, direction = "backwar
d")

summary(m_overall_best)

## 
## Call:
## lm(formula = price_change ~ total_num_of_words + num_of_per
cents + 
##     NRC_negative + NRC_SENT + RatioUncertaintyLM + topic_34
 + 
##     topic_43 + topic_25, data = z_overall_features_regressi
on_data %>%
```

```
##     select(-doc_id))
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -6.9396 -0.4625  0.0276  0.4844  3.5487 
## 
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)  
## (Intercept)         1.004e-16  4.034e-02   0.000   1.0000  
## total_num_of_words -1.060e-01  4.885e-02  -2.170   0.0304 *
## num_of_percents     1.055e-01  4.879e-02   2.163   0.0309 *
## NRC_negative       -2.602e-01  1.261e-01  -2.064   0.0395 *
## NRC_SENT           -2.617e-01  1.267e-01  -2.065   0.0393 *
## RatioUncertaintyLM -9.123e-02  4.317e-02  -2.113   0.0350 *
## topic_34           -8.722e-02  4.285e-02  -2.035   0.0423 *
## topic_43            7.603e-02  4.091e-02   1.859   0.0636 .
## topic_25            8.975e-02  4.109e-02   2.184   0.0293 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
' 1
## 
## Residual standard error: 0.9857 on 588 degrees of freedom
## Multiple R-squared:  0.04136,    Adjusted R-squared:  0.028
31 
## F-statistic: 3.171 on 8 and 588 DF,  p-value: 0.001579
```

Finally, only 8 variables (topics) are predictive to the stock price changes. The adjusted R squared is 0.02831 and the p-value is 0.001579, which show that this final model can significantly explain 2.83% of variances of the stock price changes. Moreover, the number of percent signs, topic_43 **(excess, exchange, forward, retirement, foreign, currency, entity)**, and topic_25 **(consist, model, reflect, december, line, pricing, profit)** have positive affects to the stock price changes. On the contrary, the total number of words, the negative scores and the sentiment scores from the NRC dictionary, the uncertainty ratio from the Loughran dictionary, and topic_34 **(december, average, exchange, acquire, forward, uncertain, recognition)** negatively affect to the stock price changes. In conclusion, similar to the result from part B, the correlations between the stock price changes and the other features are extremely limited.

END