Text frequency analysis

Replication for "The Role of Pilot Studies in Financial Regulation"

2025-03-10

Purpose: This code reproduces the calculations in the paper "The Role of Pilot Studies in Financial Regulation". The code here reads the 10,000+ files downloaded from the SEC and computes the results in the paper's appendix.

Most files have extensions, but there are numerous files in the `pr` subdirectory with no extension. These files begin with `<!DOCTYPE html>` and are treated as HTML files. They are readable in most browsers when an htm/html extension is appended.

The code works as follows:

1. The chunk `terms` defines the subdirectories containing the downloaded files as well as the terms to search for.
2. The main data source is the CSV file `search_results.csv`. In order to reproduce this csv file, set the variable `scanfiles` to `TRUE`. This will then build a complete list of files by looping over the five subdirectories defined in the `terms` chunk. The code for constructing the list of files is in the chunk `construct_list_of_files`, and the list is stored in the data frame `df`.
3. The chunk `search file` contains the function `search_file`, which take `df` as input and returns a modified version of `df` which contains file-by-file counts of the search terms (e.g. pilot).
4. The chunk `scan` calls the `search_files` function, which populates the data frame with search results, and which saves the data frame as `search_results.csv`. Note that this error message will occur frequently: "PDF error: Expected the default config, but wasn't able to find it, or it isn't a Dictionary". Based on spot checking it appears to be benign.
5. The rest of the code reproduces the tables and figures in the appendix.

Generate the data

Set `scanfiles=TRUE` to reproduce the data frame `search_results.csv`, or `scanfiles=FALSE` to read the data frame and go from there.

```
scanfiles <- FALSE
##scanfiles <- TRUE
library(tidyverse)
library(tidytext)
library(htmltools)
library(textclean)
library(pdftools)
library(XML)
library(tm)
library(data.table)
library(patchwork)
library(tools)
library(conflicted)
library(knitr)

opts_chunk$set(warning = FALSE)
conflicts_prefer(tools::file_ext(),
                 dplyr::filter,
                 readr::problems)

##path <- '/home/rmcd/doc/confer/fer/fer2019/submission/data/files/'
path <- './'
##setwd(path)
opts_chunk$set(root.dir = path,
               error = TRUE)
```

```
subdirs <- c('finra', 'pr', 'secother', 'secprop', 'secrules')
searchterms <- c('beta test', 'phase-in', 'pilot', 'pilot program',
                 'pilot study', 'pilot system')
```

```
allfiles <- list()
for (i in subdirs)  allfiles[[i]] <- list.files(i, full.names =  TRUE)
filelist <- unlist(allfiles, use.names = FALSE )
df <- data.frame(filelist) %>%
    mutate('beta test' = NA, 'phase-in' = NA, 'pilot' = NA,
           'pilot program' = NA, 'pilot study' =  NA, 'pilot system' = NA,
           dir = str_split_i(filelist, '/', 1),
           Year = str_sub(str_split_i(filelist, '/', 2), 1, 4)) %>%
    select(dir, Year, everything())

## problematic
unreadable_files <- c('secprop/2010-34-63556.pdf',
```

```
                      'finra/2012-34-68386.pdf',
                      'finra/2008-34-57252.pdf')

## These are files describing 4 of the 5 experiments we discuss in the paper
larryfiles <- paperfiles <- c('finra/2000-34-43616-nd9965n.htm',
                              'secother/2004-34-50104.htm',
                              'secother/2014-34-72460.pdf',
                              'secprop/2018-34-82873.pdf')
```

```
## This function takes a dataframe `df`containing a filelist column as
## input, and searches the documents for `searchterm` key words.
##
## Note that `pdf_text` frequently throws the error "PDF error:
## Expected the default config, but wasn't able to find it, or it
## isn't a Dictionary". Based on spot checking, this did not seem to
## affect the file being converted. Looks like it shouldn't matter for
## simple word counts. As examples, look at

## filelist[753:759]
## [1] "finra/2007-34-57073.pdf" "finra/2007-34-57074.pdf" "finra/2007-34-57076.pdf"
## [4] "finra/2007-34-57077.pdf" "finra/2007-34-57079.pdf" "finra/2007-34-57080.pdf"
## [7] "finra/2007-34-57081.pdf"


search_file <- function(df, searchterms) {
    for (i in 1:nrow(df)) {
        ##  i <- 20
        ## for (i in 1:500) {
        ##filepath <- "secprop/2022-34-95388.pdf"
        ##filepath <- "secprop/2000-34-42354.htm"
        ##filepath <- "pr/2022-2022-89"
        ## filepath <- "secprop/1999-ic-23815.txt"
        if (i %% 500 == 0) print(i)
        ##print(filepath)
        doc.text <- ''
        filepath <- df$filelist[i]
        if(file_ext(filepath) == "pdf") {
            doc.text <- pdf_text(filepath)
            doc.text = gsub('[\r]', '', doc.text)
            doc.text = gsub('\\n', ' ', doc.text)
        } else if (file_ext(filepath) == "txt") {
            doc.text = read_file(filepath)
```

```
        } else {
            doc.text = read_file(filepath)
            html = htmlTreeParse(filepath,useInternal = TRUE)
            doc.text = unlist(xpathApply(html, '//p', xmlValue))
            doc.text = paste(doc.text, unlist(xpathApply(html, '//h', xmlValue)))
            doc.text = gsub('[\r]', '', doc.text)
            doc.text = gsub('\\n', ' ', doc.text)
            doc.text <- gsub('--',  ' ',  doc.text)
        }
        doc.text <- paste(doc.text, collapse = ' ')
        ##issue with non ASCII character in file pr/2001-2001-133.txt
        Encoding(doc.text) <- "UTF-8"
        doc.text <- iconv(doc.text, "UTF-8", "ascii",sub='')
        ##doc.text = replace_non_ascii(doc.text, replacement = "", remove.nonconverted = TRU
        ##Remove punctuation and excess white space
        doc.text <- removePunctuation(doc.text, preserve_intra_word_dashes = TRUE,
                                      ucp = TRUE)
        ## This remove numbers within expressions, e.g.'b-34' becomes
        ## 'b-'. Maybe not desirable?
        doc.text <- removeNumbers(doc.text)
        doc.text <- stripWhitespace(doc.text)
        doc.text <- tolower(doc.text)
        for (j in searchterms) {
            ##print(j)
            df[i, j] <- str_count(doc.text, j)
        }
    }
    return(df)
}
```

In creating the dataframe, expect to get the error "PDF error: Expected the default config, but wasn't able to find it, or it isn't a Dictionary"

Not sure why is happens but based on spot checks it seems innocuous.

```
df2 <- search_file(df, searchterms)
write_csv(df2, 'search_results.csv')
```

Reproduce appendix results

```
x <- fread('search_results.csv') %>%
    rename('Webgroup' = 'dir')

tbls <- x %>%
    mutate(pilotgt0 = (pilot > 0),
           pilotgt2 = (pilot > 2))
```

```
tbls <- x %>%
    mutate(pilotgt0 = (pilot > 0),
           pilotgt2 = (pilot > 2))

tbl1 <- x %>%
    pivot_longer(`beta test`:`pilot system`, names_to = 'Phrase', values_to = 'Counts' ) %>%
    group_by(Phrase, Webgroup) %>%
    summarize(pilotgt0 = sum(Counts > 0)) %>%
    pivot_wider(values_from = pilotgt0, names_from = Phrase ) %>%
    kable(caption = "Breakdown of the use of "pilot" and variants for 5 classes of SEC docume
Proposals, Other, Press releases, and FINRA. The counts are for files where "pilot"
appears at least once.")
```

```
`summarise()` has grouped output by 'Phrase'. You can override using the
`.groups` argument.
```

```
tbl1
```

Table 1: Breakdown of the use of "pilot" and variants for 5 classes of SEC documents: Rules, Proposals, Other, Press releases, and FINRA. The counts are for files where "pilot" appears at least once.

Webgroup	beta test	phase-in	pilot	pilot program	pilot study	pilot system
finra	1	22	296	205	2	1
pr	2	23	86	45	0	0
secother	1	14	43	22	0	0
secprop	0	61	63	39	1	2
secrules	0	83	60	31	3	6

```
tbl2 <- x %>%
    pivot_longer(`beta test`:`pilot system`, names_to = 'Phrase', values_to = 'Counts' ) %>%
    group_by(Phrase, Webgroup) %>%
```

```
    summarize(pilotgt2 = sum(Counts > 2)) %>%
    pivot_wider(values_from = pilotgt2, names_from = Phrase ) %>%
    kable(caption = "Breakdown of the use of "pilot" and variants for 5 classes of SEC docume
appears more than twice.")
```

```
`summarise()` has grouped output by 'Phrase'. You can override using the
`.groups` argument.
```

```
tbl2
```

Table 2: Breakdown of the use of "pilot" and variants for 5 classes of SEC documents: Rules, Proposals, Other, Press releases, and FINRA. The counts are for files where "pilot" appears more than twice.

Webgroup	beta test	phase-in	pilot	pilot program	pilot study	pilot system
finra	0	4	231	114	0	0
pr	0	6	30	8	0	0
secother	0	10	27	9	0	0
secprop	0	31	29	14	0	0
secrules	0	49	30	13	2	0

```
## Want to do documents by year, tick by year, bigtick by year

p1 <- x %>% mutate(Year = as.Date(paste0(Year, '-1-1'))) %>%
    group_by(Year) %>%
    summarize(count =n()) %>%
    ggplot(aes(x = Year, y = count)) +
    geom_col(fill = 'darkgrey', color = 'black') +
    ggtitle('SEC Documents by Year')

tmp <- x %>% mutate(Year = as.Date(paste0(Year, '-1-1'))) %>%
    select(Webgroup, Year, filelist, pilot) %>%
    filter(pilot > 0) %>%
    mutate(`pilot > 2` = (pilot > 2)) %>%
    group_by(Year) %>%
    ##    mutate(count = sum(`pilot > 2`)) %>%
    mutate(count = n()) %>%
    ungroup()

p2 <- ggplot(tmp, aes(x = Year, fill = `pilot > 2`)) +
```

```
    geom_bar(color = 'black') +
    scale_fill_manual(values = c('white', 'darkgrey')) +
    ggtitle('Documents containing "pilot"')


p1/p2
```

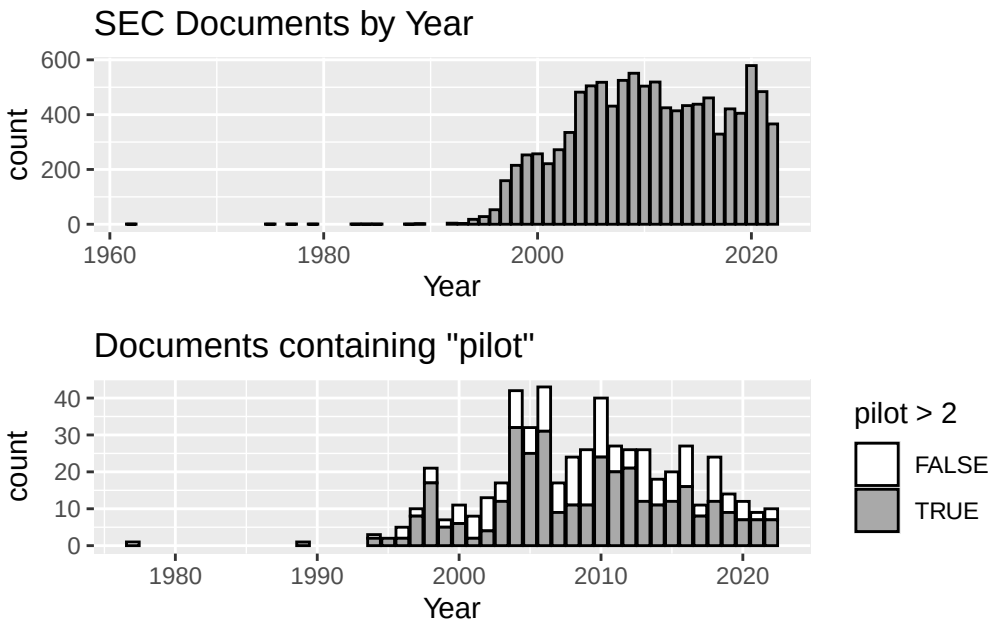

Figure 1