Note: all htm files (2008 ~ 2015) are saved in the ~/R folder and need to create a folder, ~/R/qa\_section to save the Meta data.

> # save Navistar htm files

> Navistarurl2015 <- "~/R/Navistar2015.htm"

> Navistarurl2014 <- "~/R/Navistar2014.htm"

> Navistarurl2013 <- "~/R/Navistar2013.htm"

> Navistarurl2012 <- "~/R/Navistar2012.htm"

> Navistarurl2011 <- "~/R/Navistar2011.htm"

> Navistarurl2010 <- "~/R/Navistar2010.htm"

> Navistarurl2009 <- "~/R/Navistar2009.htm"

> Navistarurl2008 <- "~/R/Navistar2008.htm"

|  |
| --- |
| > # save the cursors for start qa section and end qa section  > qa\_sentence\_start <- "FREQUENTLY ASKED QUESTIONS REGARDING ATTENDANCE AND VOTING"  > qa\_sentence\_end <- "in an amendment to the Form 8-K as soon as they become available." |
| > # process 2015 file  > # read the data from the file  > dataNavistar2015 = readLines(Navistarurl2015)  > # find the start and end section  > grep(qa\_sentence\_start,dataNavistar2015, ignore.case = TRUE)  [1] 1471 5882  > grep(qa\_sentence\_end,dataNavistar2015, ignore.case = TRUE)  [1] 7628  > # save the qa section  > qa\_sectionNavistar2015 <- dataNavistar2015[5882:7628]  > # save the data into a new text file  > cat(qa\_sectionNavistar2015, file="~/R/qa\_section/Navistar2015.txt", sep="n", append = TRUE) |

> # process 2014 file

> # read the data from the file

> dataNavistar2014 = readLines(Navistarurl2014)

> # find the start and end section

> grep(qa\_sentence\_start,dataNavistar2014, ignore.case = TRUE)

[1] 1226 2796

> grep(qa\_sentence\_end,dataNavistar2014, ignore.case = TRUE)

[1] 3714

> # save the qa section

> qa\_sectionNavistar2014 <- dataNavistar2014[2796:3714]

> # save the data into a new text file

> cat(qa\_sectionNavistar2014, file="~/R/qa\_section/Navistar2014.txt", sep="n", append = TRUE)

> # process 2013 file

> # read the data from the file

> dataNavistar2013 = readLines(Navistarurl2013)

> # find the start and end section

> grep(qa\_sentence\_start,dataNavistar2013, ignore.case = TRUE)

[1] 1281 2795

> grep(qa\_sentence\_end,dataNavistar2013, ignore.case = TRUE)

[1] 3635

> # save the qa section

> qa\_sectionNavistar2013 <- dataNavistar2013[2795:3635]

> # save the data into a new text file

> cat(qa\_sectionNavistar2013, file="~/R/qa\_section/Navistar2013.txt", sep="n", append = TRUE)

>

> # process 2012 file

> # read the data from the file

> dataNavistar2012 = readLines(Navistarurl2012)

> # find the start and end section

> grep(qa\_sentence\_start,dataNavistar2012, ignore.case = TRUE)

[1] 330 751

> grep(qa\_sentence\_end,dataNavistar2012, ignore.case = TRUE)

[1] 978

> # save the qa section

> qa\_sectionNavistar2012 <- dataNavistar2012[751:978]

> # save the data into a new text file

> cat(qa\_sectionNavistar2012, file="~/R/qa\_section/Navistar2012.txt", sep="n", append = TRUE)

>

> # process 2011 file

> # read the data from the file

> dataNavistar2011 = readLines(Navistarurl2011)

> # find the start and end section

> grep(qa\_sentence\_start,dataNavistar2011, ignore.case = TRUE)

[1] 340 592

> grep(qa\_sentence\_end,dataNavistar2011, ignore.case = TRUE)

[1] 914

> # save the qa section

> qa\_sectionNavistar2011 <- dataNavistar2011[592:914]

> # save the data into a new text file

> cat(qa\_sectionNavistar2011, file="~/R/qa\_section/Navistar2011.txt", sep="n", append = TRUE)

>

> # process 2010 file

> # read the data from the file

> dataNavistar2010 = readLines(Navistarurl2010)

> # find the start and end section

> grep(qa\_sentence\_start,dataNavistar2010, ignore.case = TRUE)

[1] 329 583

> grep(qa\_sentence\_end,dataNavistar2010, ignore.case = TRUE)

[1] 937

> # save the qa section

> qa\_sectionNavistar2010 <- dataNavistar2010[583:937]

> # save the data into a new text file

> cat(qa\_sectionNavistar2010, file="~/R/qa\_section/Navistar2010.txt", sep="n", append = TRUE)

>

> # process 2009 file

> # read the data from the file

> dataNavistar2009 = readLines(Navistarurl2009)

> # find the start and end section

> grep(qa\_sentence\_start,dataNavistar2009, ignore.case = TRUE)

[1] 321 497

> grep("Stockholders may speak a second time only after all other stockholders who wish to speak have had their turn",dataNavistar2009, ignore.case = TRUE)

[1] 739

> # save the qa section

> qa\_sectionNavistar2009 <- dataNavistar2009[497:739]

> # save the data into a new text file

> cat(qa\_sectionNavistar2009, file="~/R/qa\_section/Navistar2009.txt", sep="n", append = TRUE)

>

> # process 2008 file

> # read the data from the file

> dataNavistar2008 = readLines(Navistarurl2008)

> # find the start and end section

> grep(qa\_sentence\_start,dataNavistar2008, ignore.case = TRUE)

[1] 284

> grep("Stockholders may speak a second time only after all other stockholders who wish to speak have had their turn",dataNavistar2008, ignore.case = TRUE)

[1] 610

> # save the qa section

> qa\_sectionNavistar2008 <- dataNavistar2008[284:610]

> # save the data into a new text file

> cat(qa\_sectionNavistar2008, file="~/R/qa\_section/Navistar2008.txt", sep="n", append = TRUE)

>

> # save the directory for corpus

> cname <- "~/R/qa\_section"

>

> library(tm)

> library(NLP)

> # save the corpus

> qa\_sections <- Corpus(DirSource(cname))

> # show details for the corpus

> inspect(qa\_sections[1])

<<VCorpus>>

Metadata: corpus specific: 0, document level (indexed): 0

Content: documents: 1

[[1]]

<<PlainTextDocument>>

Metadata: 7

Content: chars: 70786

> dir(cname)

[1] "Navistar2008.txt" "Navistar2009.txt" "Navistar2010.txt" "Navistar2011.txt" "Navistar2012.txt" "Navistar2013.txt" "Navistar2014.txt" "Navistar2015.txt"

> summary(qa\_sections)

Length Class Mode

Navistar2008.txt 2 PlainTextDocument list

Navistar2009.txt 2 PlainTextDocument list

Navistar2010.txt 2 PlainTextDocument list

Navistar2011.txt 2 PlainTextDocument list

Navistar2012.txt 2 PlainTextDocument list

Navistar2013.txt 2 PlainTextDocument list

Navistar2014.txt 2 PlainTextDocument list

Navistar2015.txt 2 PlainTextDocument list

> library(magrittr)

> # view the file

> viewDocs <- function(d,n) {d %>% extract2(n) %>% as.character() %>% writeLines()}

> viewDocs(qa\_sections, 1)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| > # convert to lower case  > qa\_sections <- tm\_map(qa\_sections, content\_transformer(tolower))  >  > # remove numbers  > qa\_sections <- tm\_map(qa\_sections, removeNumbers)  >  > # remove stop words  > qa\_sections <- tm\_map(qa\_sections, removeWords, stopwords("english"))   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | > # Replacing certin expressions with spaces:  > toSpace <- content\_transformer(function(x, pattern) gsub(pattern, " ",x))  > qa\_sections <- tm\_map(qa\_sections, toSpace, "/|<|>|”|=|@|\\|:|;|-|\"")  > # Remove Punctuations  > qa\_sections <- tm\_map(qa\_sections, removePunctuation)  >  > # Strip white spaces:  > qa\_sections <- tm\_map(qa\_sections, stripWhitespace)   |  |  |  |  | | --- | --- | --- | --- | | > # Remove known often words:  > qa\_sections <- tm\_map(qa\_sections, removeWords, c("b", "q", "a", "i", "e", "font", "style","n","trim","size","font", "can", "also", "e", "mail", "via", "td", "align","border", "familytimes", "roman", "p", "tr", "nbsp", "with", "table", "cellspacing", "valign", "cellpadding", "width", "top", "left", "sizepx", "telephone", "if", "may", "help", "us", "will", "please", "unless", "visit", "thnbsp","toppx","bottompx", "nnn", "address", "nonbsp", "new", "bottom", "em"))  > # Specific Transformation:  > toString <- content\_transformer(function(x, from, to) gsub(from, to, x))  > qa\_sections <- tm\_map(qa\_sections, toString, "broker bank", "bb")  > # Stemming:  > install.packages("SnowballC")  Installing package into ‘C:/Users/JJ/Documents/R/win-library/3.2’  (as ‘lib’ is unspecified)  trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.2/SnowballC\_0.5.1.zip'  Content type 'application/zip' length 3076512 bytes (2.9 MB)  downloaded 2.9 MB  package ‘SnowballC’ successfully unpacked and MD5 sums checked  The downloaded binary packages are in  C:\Users\JJ\AppData\Local\Temp\RtmpEt5XBN\downloaded\_packages  > library(SnowballC)  >  > qa\_sections <-tm\_map(qa\_sections, stemDocument)  > # create a term document matrix  > qa\_sections <-tm\_map(qa\_sections, PlainTextDocument)  > qa\_dtm <- DocumentTermMatrix(qa\_sections)  > qa\_dtm  <<DocumentTermMatrix (documents: 8, terms: 855)>>  Non-/sparse entries: 3389/3451  Sparsity : 50%  Maximal term length: 22  Weighting : term frequency (tf)   |  |  |  | | --- | --- | --- | | > # inpect the term document matrix  > inspect(qa\_dtm[1:8, 100:105])  <<DocumentTermMatrix (documents: 8, terms: 6)>>  Non-/sparse entries: 12/36  Sparsity : 75%  Maximal term length: 7  Weighting : term frequency (tf)  Terms  Docs benvot best bgcolor bind block bni  character(0) 0 0 6 0 0 0  character(0) 0 0 0 0 0 0  character(0) 0 0 0 8 0 2  character(0) 0 0 0 4 0 2  character(0) 2 0 0 4 0 0  character(0) 0 2 0 8 0 0  character(0) 0 0 0 4 0 0  character(0) 0 0 0 4 8 0  > freq <- colSums((as.matrix(qa\_dtm)))  > length(freq)  [1] 855  > ord <- order(freq)  > freq[head(ord)]  abstainnfrom abstentionnwil accountnstat addressneach allnshar allnstockhold  2 2 2 2 2 2  > freq[tail(ord)]  pad vote margin familyari div sizept  794 1328 1468 1670 1860 2146  > head(table(freq), 15)  freq  2 4 6 8 10 12 14 16 18 20 22 24 26 28 30  275 72 43 37 20 35 21 77 13 7 11 8 5 9 14  > tail(table(freq), 15)  freq  374 424 426 510 586 658 742 746 788 794 1328 1468 1670 1860 2146  2 1 1 1 2 1 1 1 1 3 1 1 1 1 1  > # convert to matrix and save it to CSV  > m<-as.matrix(qa\_dtm)  > dim(m)  [1] 8 855  > write.csv(m,file= "~/R/qa\_Navistar.csv")  > # remove sparse terms  > dtms <- removeSparseTerms(qa\_dtm, 0.1)  > dim(dtms)  [1] 8 243  > inspect(dtms)  <<DocumentTermMatrix (documents: 8, terms: 243)>>  Non-/sparse entries: 1944/0  Sparsity : 0%  Maximal term length: 12  Weighting : term frequency (tf)  > freq <-colSums(as.matrix(dtms))  > # Below we specify that we want terms / words  > # which were used 50 or more times (in all documents / paragraphs).  > findFreqTerms(dtms, lowfreq=50)  [1] "accord" "account" "act" "addit" "admiss" "affirm" "alway" "annual" "approv" "ask" "attend" "author" "avail" "bank"  [15] "board" "break" "broker" "brokerag" "call" "card" "combin" "common" "compani" "compens" "copi" "corpor" "date" "direct"  [29] "director" "doe" "elect" "entitl" "firm" "follow" "held" "hold" "holder" "illinoi" "includ" "independ" "individu" "instruct"  [43] "internet" "john" "later" "major" "matter" "meet" "must" "name" "navistar" "nomin" "nomine" "non" "notic" "number"  [57] "obtain" "one" "order" "outstand" "ownership" "page" "pay" "perform" "person" "phone" "plan" "present" "propos" "provid"  [71] "proxi" "public" "purpos" "question" "ratif" "receiv" "recommend" "record" "regist" "report" "repres" "request" "requir" "result"  [85] "routin" "rule" "sec" "secretari" "share" "solicit" "specifi" "statement" "stock" "stockhold" "street" "submit" "ticket" "time"  [99] "vote" "wish"  > # save this frequencies to CVS file  > freq50 <- findFreqTerms(dtms, lowfreq=50)  > write.csv(freq50,file= "~/R/frequencies50.csv")  > # Below we specify that we want terms / words  > # which were used 100 or more times (in all documents / paragraphs).  > findFreqTerms(dtms, lowfreq=100)  [1] "account" "annual" "attend" "avail" "board" "broker" "card" "combin" "common" "compani" "corpor" "date" "director" "elect"  [15] "firm" "follow" "held" "hold" "holder" "instruct" "internet" "matter" "meet" "must" "name" "navistar" "nomine" "notic"  [29] "one" "person" "present" "propos" "provid" "proxi" "receiv" "record" "regist" "repres" "requir" "result" "share" "solicit"  [43] "statement" "stock" "stockhold" "street" "ticket" "vote"  > # save this frequencies to CVS file  > freq100 <- findFreqTerms(dtms, lowfreq=100)  > write.csv(freq100,file= "~/R/frequencies100.csv")  > # Finding words which 'associate' together. Here, we are specifying  > # the Term Document Matrix to use, the term we want to find associates  > # for, and the lowest acceptable correlation limit with that term. This  > # returns a vector of terms which are associated with 'navistar' at  > # 0.50 or more (correlation) -- and reports each association in  > # decending order.  > findAssocs(dtms, "navistar", corlimit = .5)  $navistar  manner phone corpor necessari plan card intern perform hold  0.71 0.63 0.62 0.62 0.59 0.56 0.56 0.53 0.50  > # Plotting Frequent Words  > freq <- sort(colSums(as.matrix(dtms)), decreasing = TRUE)  > head(freq, 14)  vote proxi stockhold share annual meet propos card name record compani receiv matter instruct  1328 788 742 658 586 586 510 310 288 280 266 262 236 230  > wf <- data.frame(word=names(freq), freq=freq)  > head(wf)  word freq  vote vote 1328  proxi proxi 788  stockhold stockhold 742  share share 658  annual annual 586  meet meet 586   |  |  | | --- | --- | | > install.packages("ggplot2")  Installing package into ‘C:/Users/JJ/Documents/R/win-library/3.2’  (as ‘lib’ is unspecified)  trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.2/ggplot2\_2.1.0.zip'  Content type 'application/zip' length 2002463 bytes (1.9 MB)  downloaded 1.9 MB  package ‘ggplot2’ successfully unpacked and MD5 sums checked  The downloaded binary packages are in  C:\Users\JJ\AppData\Local\Temp\RtmpEt5XBN\downloaded\_packages  > library(ggplot2)  Attaching package: ‘ggplot2’  The following object is masked from ‘package:NLP’:  annotate  Warning message:  package ‘ggplot2’ was built under R version 3.2.5  > subset(wf, freq>200) %>% ggplot(aes(word, freq))  > subset(wf, freq>200)  word freq  vote vote 1328  proxi proxi 788  stockhold stockhold 742  share share 658  annual annual 586  meet meet 586  propos propos 510  card card 310  name name 288  record record 280  compani compani 266  receiv receiv 262  matter matter 236  instruct instruct 230  director director 212  navistar navistar 206  stock stock 206  >  > install.packages("dplyr")  Installing package into ‘C:/Users/JJ/Documents/R/win-library/3.2’  (as ‘lib’ is unspecified)  trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.2/dplyr\_0.4.3.zip'  Content type 'application/zip' length 2548225 bytes (2.4 MB)  downloaded 2.4 MB  package ‘dplyr’ successfully unpacked and MD5 sums checked  The downloaded binary packages are in  C:\Users\JJ\AppData\Local\Temp\RtmpEt5XBN\downloaded\_packages  > library(dplyr)  Attaching package: ‘dplyr’  The following objects are masked from ‘package:stats’:  filter, lag  The following objects are masked from ‘package:base’:  intersect, setdiff, setequal, union  Warning message:  package ‘dplyr’ was built under R version 3.2.5  > subset(wf, freq>200) %>% ggplot(aes(word, freq)) + geom\_bar(stat="identity") + theme(axis.text.x=element\_text(angle=45, hjust=1))   |  | | --- | | > # Generating a Word Cloud  > install.packages("wordcloud")  Installing package into ‘C:/Users/JJ/Documents/R/win-library/3.2’  (as ‘lib’ is unspecified)  trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.2/wordcloud\_2.5.zip'  Content type 'application/zip' length 465800 bytes (454 KB)  downloaded 454 KB  package ‘wordcloud’ successfully unpacked and MD5 sums checked  The downloaded binary packages are in  C:\Users\JJ\AppData\Local\Temp\RtmpEt5XBN\downloaded\_packages  > library(wordcloud)  Loading required package: RColorBrewer  Warning message:  package ‘wordcloud’ was built under R version 3.2.5  > set.seed(123)  > wordcloud(names(freq), freq, min.freq=50) | | | | |  | | |  | | --- | |  | | | |