HW 4 Hints: Fed Papers

Load Libraries

```
## Read in the documents and convert them to
## a format that we can evaluate.
library(wordcloud)
```

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

```
library(tm)
```

```
## Loading required package: NLP
```

```
library(slam)
library(quanteda)
```

```
## Package version: 1.4.3

## Parallel computing: 2 of 8 threads used.

## See https://quanteda.io for tutorials and examples.

##
## Attaching package: 'quanteda'

## The following objects are masked from 'package:tm':
##
##     as.DocumentTermMatrix, stopwords

## The following object is masked from 'package:utils':
##
##     View
```

```
library(SnowballC)
library(arules)
```

```
## Loading required package: Matrix

##
## Attaching package: 'arules'

## The following object is masked from 'package:quanteda':
##
##     affinity

## The following object is masked from 'package:tm':
##
##     inspect

## The following objects are masked from 'package:base':
##
##     abbreviate, write
```

```
library(proxy)
```

```
##
## Attaching package: 'proxy'
## The following object is masked from 'package:Matrix':
##
##     as.matrix

## The following objects are masked from 'package:stats':
##
##     as.dist, dist

## The following object is masked from 'package:base':
##
##     as.matrix
```

```r
library(cluster)
library(stringi)
library(Matrix)
library(tidytext)
library(plyr)
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'

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

```r
library(factoextra)
```

```
## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ
```

```r
library(mclust)
```

```
## Package 'mclust' version 5.4.3
## Type 'citation("mclust")' for citing this R package in publications.
```

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following objects are masked from 'package:arules':
##
##     intersect, recode, setdiff, setequal, union

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Load Data (as Corpus).

In this example, we will load the data in corpus form. We will need to do much of the data cleaning, text processing, ourselves.

```
###Load Fed Papers Corpus
FedPapersCorpus <- Corpus(DirSource("FedPapersCorpus"))
(numberFedPapers<-length(FedPapersCorpus))
```

```
## [1] 85
```

```
##The following will show you that you read in all the documents
(summary(FedPapersCorpus))
```

```
##                     Length Class             Mode
## dispt_fed_49.txt    2      PlainTextDocument list
## dispt_fed_50.txt    2      PlainTextDocument list
## dispt_fed_51.txt    2      PlainTextDocument list
## dispt_fed_52.txt    2      PlainTextDocument list
## dispt_fed_53.txt    2      PlainTextDocument list
## dispt_fed_54.txt    2      PlainTextDocument list
## dispt_fed_55.txt    2      PlainTextDocument list
## dispt_fed_56.txt    2      PlainTextDocument list
## dispt_fed_57.txt    2      PlainTextDocument list
## dispt_fed_62.txt    2      PlainTextDocument list
## dispt_fed_63.txt    2      PlainTextDocument list
## Hamilton_fed_1.txt  2      PlainTextDocument list
## Hamilton_fed_11.txt 2      PlainTextDocument list
## Hamilton_fed_12.txt 2      PlainTextDocument list
## Hamilton_fed_13.txt 2      PlainTextDocument list
## Hamilton_fed_15.txt 2      PlainTextDocument list
## Hamilton_fed_16.txt 2      PlainTextDocument list
## Hamilton_fed_17.txt 2      PlainTextDocument list
## Hamilton_fed_21.txt 2      PlainTextDocument list
## Hamilton_fed_22.txt 2      PlainTextDocument list
## Hamilton_fed_23.txt 2      PlainTextDocument list
## Hamilton_fed_24.txt 2      PlainTextDocument list
## Hamilton_fed_25.txt 2      PlainTextDocument list
## Hamilton_fed_26.txt 2      PlainTextDocument list
## Hamilton_fed_27.txt 2      PlainTextDocument list
## Hamilton_fed_28.txt 2      PlainTextDocument list
## Hamilton_fed_29.txt 2      PlainTextDocument list
## Hamilton_fed_30.txt 2      PlainTextDocument list
## Hamilton_fed_31.txt 2      PlainTextDocument list
## Hamilton_fed_32.txt 2      PlainTextDocument list
## Hamilton_fed_33.txt 2      PlainTextDocument list
## Hamilton_fed_34.txt 2      PlainTextDocument list
## Hamilton_fed_35.txt 2      PlainTextDocument list
## Hamilton_fed_36.txt 2      PlainTextDocument list
## Hamilton_fed_59.txt 2      PlainTextDocument list
## Hamilton_fed_6.txt  2      PlainTextDocument list
## Hamilton_fed_60.txt 2      PlainTextDocument list
## Hamilton_fed_61.txt 2      PlainTextDocument list
## Hamilton_fed_65.txt 2      PlainTextDocument list
## Hamilton_fed_66.txt 2      PlainTextDocument list
## Hamilton_fed_67.txt 2      PlainTextDocument list
## Hamilton_fed_68.txt 2      PlainTextDocument list
## Hamilton_fed_69.txt 2      PlainTextDocument list
## Hamilton_fed_7.txt  2      PlainTextDocument list
## Hamilton_fed_70.txt 2      PlainTextDocument list
## Hamilton_fed_71.txt 2      PlainTextDocument list
## Hamilton_fed_72.txt 2      PlainTextDocument list
## Hamilton_fed_73.txt 2      PlainTextDocument list
## Hamilton_fed_74.txt 2      PlainTextDocument list
## Hamilton_fed_75.txt 2      PlainTextDocument list
## Hamilton_fed_76.txt 2      PlainTextDocument list
## Hamilton_fed_77.txt 2      PlainTextDocument list
## Hamilton_fed_78.txt 2      PlainTextDocument list
## Hamilton_fed_79.txt 2      PlainTextDocument list
## Hamilton_fed_8.txt  2      PlainTextDocument list
## Hamilton_fed_80.txt 2      PlainTextDocument list
## Hamilton_fed_81.txt 2      PlainTextDocument list
## Hamilton_fed_82.txt 2      PlainTextDocument list
## Hamilton_fed_83.txt 2      PlainTextDocument list
## Hamilton_fed_84.txt 2      PlainTextDocument list
## Hamilton_fed_85.txt 2      PlainTextDocument list
## Hamilton_fed_9.txt  2      PlainTextDocument list
## HM_fed_18.txt       2      PlainTextDocument list
## HM_fed_19.txt       2      PlainTextDocument list
## HM_fed_20.txt       2      PlainTextDocument list
## Jay_fed_2.txt       2      PlainTextDocument list
## Jay_fed_3.txt       2      PlainTextDocument list
## Jay_fed_4.txt       2      PlainTextDocument list
## Jay_fed_5.txt       2      PlainTextDocument list
## Jay_fed_64.txt      2      PlainTextDocument list
## Madison_fed_10.txt  2      PlainTextDocument list
## Madison_fed_14.txt  2      PlainTextDocument list
## Madison_fed_37.txt  2      PlainTextDocument list
## Madison_fed_38.txt  2      PlainTextDocument list
## Madison_fed_39.txt  2      PlainTextDocument list
## Madison_fed_40.txt  2      PlainTextDocument list
## Madison_fed_41.txt  2      PlainTextDocument list
## Madison_fed_42.txt  2      PlainTextDocument list
## Madison_fed_43.txt  2      PlainTextDocument list
## Madison_fed_44.txt  2      PlainTextDocument list
## Madison_fed_45.txt  2      PlainTextDocument list
## Madison_fed_46.txt  2      PlainTextDocument list
## Madison_fed_47.txt  2      PlainTextDocument list
## Madison_fed_48.txt  2      PlainTextDocument list
## Madison_fed_58.txt  2      PlainTextDocument list
```

```
(meta(FedPapersCorpus[[1]]))
```

```
##   author       : character(0)
##   datetimestamp: 2019-10-01 16:59:20
##   description  : character(0)
##   heading      : character(0)
##   id           : dispt_fed_49.txt
##   language     : en
##   origin       : character(0)
```

```
(meta(FedPapersCorpus[[1]],5))
```

```
## [1] "dispt_fed_49.txt"
```

Data Cleaning

Here we investigate the data and vectorize it using DocumentTermMatrix. We will ignore very infrequent words and very frequent words during the vectorization process. Note: The DocumentTermMatrix method will perform much data cleaning for us.

```
##Data Preparation and Transformation on Fed Papers
###Remove punctuation,numbers, and space
(getTransformations())
```

```
## [1] "removeNumbers"     "removePunctuation" "removeWords"
## [4] "stemDocument"      "stripWhitespace"
```

```
(nFedPapersCorpus<-length(FedPapersCorpus))
```

```
## [1] 85
```

```
### ignore extremely rare words i.e. terms that appear in less then 1% of the documents
(minTermFreq <- nFedPapersCorpus * 0.0001)
```

```
## [1] 0.0085
```

```
###Ignore overly common words i.e. terms that appear in more than 50% of the documents
(maxTermFreq <- nFedPapersCorpus * 1)
```

```
## [1] 85
```

```
(MyStopwords <- c("will","one","two", "may","less", "well","might","withou","small", "single", "several
```

```
##  [1] "will"    "one"     "two"     "may"     "less"    "well"    "might"
##  [8] "withou"  "small"   "single"  "several" "but"     "very"    "can"
## [15] "must"    "also"    "any"     "and"     "are"     "however" "into"
## [22] "almost"  "can"     "for"     "add"
```

```
  #stopwords))
(STOPS <-stopwords('english'))
```

```
##   [1] "i"          "me"         "my"         "myself"     "we"
##   [6] "our"        "ours"       "ourselves"  "you"        "your"
##  [11] "yours"      "yourself"   "yourselves" "he"         "him"
##  [16] "his"        "himself"    "she"        "her"        "hers"
##  [21] "herself"    "it"         "its"        "itself"     "they"
##  [26] "them"       "their"      "theirs"     "themselves" "what"
##  [31] "which"      "who"        "whom"       "this"       "that"
##  [36] "these"      "those"      "am"         "is"         "are"
##  [41] "was"        "were"       "be"         "been"       "being"
##  [46] "have"       "has"        "had"        "having"     "do"
##  [51] "does"       "did"        "doing"      "would"      "should"
##  [56] "could"      "ought"      "i'm"        "you're"     "he's"
##  [61] "she's"      "it's"       "we're"      "they're"    "i've"
##  [66] "you've"     "we've"      "they've"    "i'd"        "you'd"
##  [71] "he'd"       "she'd"      "we'd"       "they'd"     "i'll"
##  [76] "you'll"     "he'll"      "she'll"     "we'll"      "they'll"
##  [81] "isn't"      "aren't"     "wasn't"     "weren't"    "hasn't"
##  [86] "haven't"    "hadn't"     "doesn't"    "don't"      "didn't"
##  [91] "won't"      "wouldn't"   "shan't"     "shouldn't"  "can't"
##  [96] "cannot"     "couldn't"   "mustn't"    "let's"      "that's"
## [101] "who's"      "what's"     "here's"     "there's"    "when's"
## [106] "where's"    "why's"      "how's"      "a"          "an"
## [111] "the"        "and"        "but"        "if"         "or"
## [116] "because"    "as"         "until"      "while"      "of"
## [121] "at"         "by"         "for"        "with"       "about"
## [126] "against"    "between"    "into"       "through"    "during"
## [131] "before"     "after"      "above"      "below"      "to"
## [136] "from"       "up"         "down"       "in"         "out"
## [141] "on"         "off"        "over"       "under"      "again"
## [146] "further"    "then"       "once"       "here"       "there"
## [151] "when"       "where"      "why"        "how"        "all"
## [156] "any"        "both"       "each"       "few"        "more"
## [161] "most"       "other"      "some"       "such"       "no"
## [166] "nor"        "not"        "only"       "own"        "same"
## [171] "so"         "than"       "too"        "very"       "will"
```

```
Papers_DTM <- DocumentTermMatrix(FedPapersCorpus,
                         control = list(
                           stopwords = TRUE,
                           wordLengths=c(3, 15),
                           removePunctuation = T,
                           removeNumbers = T,
                           tolower=T,
                           stemming = T,
                           remove_separators = T,
                           stopwords = MyStopwords,
                           #removeWords(STOPS), # use the "built-in" STOP words
                           bounds = list(global = c(minTermFreq, maxTermFreq))
                         ))

#inspect FedPapers Document Term Matrix (DTM)
DTM <- as.matrix(Papers_DTM)
(DTM[1:11,1:10])
```

```
##                     Terms
## Docs                 abandon abat abb abet abhorr abil abject abl ablest
##   dispt_fed_49.txt         0    0   0    0      0    0      0   2      0
##   dispt_fed_50.txt         0    0   0    0      0    0      0   0      0
##   dispt_fed_51.txt         0    0   0    0      0    0      0   1      0
##   dispt_fed_52.txt         0    0   0    0      0    1      0   1      0
##   dispt_fed_53.txt         0    1   0    0      0    0      0   0      0
##   dispt_fed_54.txt         0    0   0    0      0    0      0   0      0
##   dispt_fed_55.txt         0    0   0    0      0    0      0   0      0
##   dispt_fed_56.txt         0    0   0    0      0    0      0   0      0
##   dispt_fed_57.txt         0    0   0    0      1    0      0   0      0
##   dispt_fed_62.txt         0    0   0    0      0    0      0   1      0
##   dispt_fed_63.txt         0    0   0    0      0    0      0   4      0
##                     Terms
## Docs                 abolish
##   dispt_fed_49.txt         0
##   dispt_fed_50.txt         0
##   dispt_fed_51.txt       0
##   dispt_fed_52.txt       0
##   dispt_fed_53.txt       0
##   dispt_fed_54.txt       0
##   dispt_fed_55.txt       0
##   dispt_fed_56.txt       0
##   dispt_fed_57.txt       0
##   dispt_fed_62.txt       0
##   dispt_fed_63.txt       0
```

Inspect Initial Cleaning Results

Investigate the initial results of data cleaning. Depending on the results, we may decide to go back and “re-clean" the data, eg, add more stop words. Lets inspect the word frequencies.

```
## Look at word freuquncies
WordFreq <- colSums(as.matrix(Papers_DTM))
(head(WordFreq))
```

```
## abandon    abat     abb    abet  abhorr    abil
##       9       2       5       2       1      15
```

```
(length(WordFreq))
```

```
## [1] 4900
```

```
ord <- order(WordFreq)
(WordFreq[head(ord)])
```

```
##  abhorr  abject abraham   abreg  absenc  absolv
##       1       1       1       1       1       1
```

```
(WordFreq[tail(ord)])
```

```
## constitut       may     power    govern      will     state
##       686       811       937      1040      1263      1662
```

```
## Row Sums per Fed Papers
(Row_Sum_Per_doc <- rowSums((as.matrix(Papers_DTM))))
```

```
##    dispt_fed_49.txt    dispt_fed_50.txt    dispt_fed_51.txt
##                 758                 530                 923
##    dispt_fed_52.txt    dispt_fed_53.txt    dispt_fed_54.txt
##                 853                1035                 882
##    dispt_fed_55.txt    dispt_fed_56.txt    dispt_fed_57.txt
##                 968                 765                1023
##    dispt_fed_62.txt    dispt_fed_63.txt  Hamilton_fed_1.txt
##                1124                1432                 767
## Hamilton_fed_11.txt Hamilton_fed_12.txt Hamilton_fed_13.txt
##                1164                1044                 479
## Hamilton_fed_15.txt Hamilton_fed_16.txt Hamilton_fed_17.txt
##                1411                 918                 767
## Hamilton_fed_21.txt Hamilton_fed_22.txt Hamilton_fed_23.txt
##                 937                1692                 828
## Hamilton_fed_24.txt Hamilton_fed_25.txt Hamilton_fed_26.txt
##                 925                 927                1093
## Hamilton_fed_27.txt Hamilton_fed_28.txt Hamilton_fed_29.txt
##                 690                 755                1010
## Hamilton_fed_30.txt Hamilton_fed_31.txt Hamilton_fed_32.txt
##                 948                 797                 686
## Hamilton_fed_33.txt Hamilton_fed_34.txt Hamilton_fed_35.txt
##                 773                1020                1052
## Hamilton_fed_36.txt Hamilton_fed_59.txt  Hamilton_fed_6.txt
##                1272                 860                 984
## Hamilton_fed_60.txt Hamilton_fed_61.txt Hamilton_fed_65.txt
##                1006                 681                 912
## Hamilton_fed_66.txt Hamilton_fed_67.txt Hamilton_fed_68.txt
##                 997                 781                 683
## Hamilton_fed_69.txt  Hamilton_fed_7.txt Hamilton_fed_70.txt
##                1359                1073                1436
## Hamilton_fed_71.txt Hamilton_fed_72.txt Hamilton_fed_73.txt
##                 766                 925                1061
## Hamilton_fed_74.txt Hamilton_fed_75.txt Hamilton_fed_76.txt
##                 478                 905                 883
## Hamilton_fed_77.txt Hamilton_fed_78.txt Hamilton_fed_79.txt
##                 887                1376                 478
##  Hamilton_fed_8.txt Hamilton_fed_80.txt Hamilton_fed_81.txt
##                 998                1132                1798
## Hamilton_fed_82.txt Hamilton_fed_83.txt Hamilton_fed_84.txt
##                 749                2620                1907
## Hamilton_fed_85.txt  Hamilton_fed_9.txt       HM_fed_18.txt
##                1264                 931                1029
##       HM_fed_19.txt       HM_fed_20.txt       Jay_fed_2.txt
##                1023                 776                 804
##       Jay_fed_3.txt       Jay_fed_4.txt       Jay_fed_5.txt
##                 736                 780                 657
##      Jay_fed_64.txt  Madison_fed_10.txt  Madison_fed_14.txt
##                1072                1437                1016
##  Madison_fed_37.txt  Madison_fed_38.txt  Madison_fed_39.txt
##                1268                1529                1169
##  Madison_fed_40.txt  Madison_fed_41.txt  Madison_fed_42.txt
##                1340                1701                1330
##  Madison_fed_43.txt  Madison_fed_44.txt  Madison_fed_45.txt
##                1601                1382                1018
##  Madison_fed_46.txt  Madison_fed_47.txt  Madison_fed_48.txt
##                1233                1306                 846
##  Madison_fed_58.txt
##                 978
```

Normalization

In text processing, it is often beneficial to normalize the word vectors before applying standard analysis techniques.

```
## Create a normalized version of Papers_DTM
Papers_M <- as.matrix(Papers_DTM)
Papers_M_N1 <- apply(Papers_M, 1, function(i) round(i/sum(i),3))
Papers_Matrix_Norm <- t(Papers_M_N1)
## Have a look at the original and the norm to make sure
(Papers_M[c(1:11),c(1000:1010)])
```

```
##                    Terms
## Docs               crude cruel crush culpabl cultiv culumni cun cupid cure
##   dispt_fed_49.txt     0     0     0       0      0       0   0     0    0
##   dispt_fed_50.txt     0     0     0       0      0       0   0     0    0
##   dispt_fed_51.txt     0     0     0       0      0       0   0     0    0
##   dispt_fed_52.txt     0     0     0       0      0       0   0     0    0
##   dispt_fed_53.txt     0     0     0       0      0       0   0     0    0
##   dispt_fed_54.txt     0     0     0       0      0       0   0     0    0
##   dispt_fed_55.txt     0     0     0       0      0       0   0     0    0
##   dispt_fed_56.txt     0     0     0       0      0       0   0     0    0
##   dispt_fed_57.txt     0     0     0       0      0       0   0     0    0
##   dispt_fed_62.txt     0     0     0       0      1       0   0     0    0
##   dispt_fed_63.txt     0     0     1       0      0       0   0     0    0
##                    Terms
## Docs               curios curious
##   dispt_fed_49.txt      0       0
##   dispt_fed_50.txt      0       0
##   dispt_fed_51.txt      0       0
##   dispt_fed_52.txt      0       0
##   dispt_fed_53.txt      1       0
##   dispt_fed_54.txt      0       0
##   dispt_fed_55.txt      0       0
##   dispt_fed_56.txt      0       0
##   dispt_fed_57.txt      0       0
##   dispt_fed_62.txt      0       0
##   dispt_fed_63.txt      0       0
```

```
(Papers_Matrix_Norm[c(1:11),c(1000:1010)])
```

```
##                    Terms
## Docs               crude cruel crush culpabl cultiv culumni cun cupid cure
##   dispt_fed_49.txt     0     0 0.000       0  0.000       0   0     0    0
##   dispt_fed_50.txt     0     0 0.000       0  0.000       0   0     0    0
##   dispt_fed_51.txt     0     0 0.000       0  0.000       0   0     0    0
##   dispt_fed_52.txt     0     0 0.000       0  0.000       0   0     0    0
##   dispt_fed_53.txt     0     0 0.000       0  0.000       0   0     0    0
##   dispt_fed_54.txt     0     0 0.000       0  0.000       0   0     0    0
##   dispt_fed_55.txt     0     0 0.000       0  0.000       0   0     0    0
##   dispt_fed_56.txt     0     0 0.000       0  0.000       0   0     0    0
##   dispt_fed_57.txt     0     0 0.000       0  0.000       0   0     0    0
##   dispt_fed_62.txt     0     0 0.000       0  0.001       0   0     0    0
##   dispt_fed_63.txt     0     0 0.001       0  0.000       0   0     0    0
##                    Terms
## Docs               curios curious
##   dispt_fed_49.txt  0.000       0
##   dispt_fed_50.txt  0.000       0
##   dispt_fed_51.txt  0.000       0
##   dispt_fed_52.txt  0.000       0
##   dispt_fed_53.txt  0.001       0
##   dispt_fed_54.txt  0.000       0
##   dispt_fed_55.txt  0.000       0
##   dispt_fed_56.txt  0.000       0
##   dispt_fed_57.txt  0.000       0
##   dispt_fed_62.txt  0.000       0
##   dispt_fed_63.txt  0.000       0
## From the line of code
## (Row_Sum_Per_doc <- rowSums((as.matrix(FedPapersDTM))))
## above, we can see that dispt_fed_53.txt has a row sum of 1035
## So, we can confirm correctness. For word "curious" we should have
## 1/1035 = 0.001 rounded, which is what we have.
```

Data Structures

Depending on the subsequent analysis, we may need to restructure the data. Here is an example …

```
## Convert to matrix and view
Papers_dtm_matrix = as.matrix(Papers_DTM)
str(Papers_dtm_matrix)
```

```
##  num [1:85, 1:4900] 0 0 0 0 0 0 0 0 0 0 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ Docs : chr [1:85] "dispt_fed_49.txt" "dispt_fed_50.txt" "dispt_fed_51.txt" "dispt_fed_52.txt"
##   ..$ Terms: chr [1:4900] "abandon" "abat" "abb" "abet" ...
```

```
(Papers_dtm_matrix[c(1:11),c(2:10)])
```

```
##                    Terms
## Docs                 abat abb abet abhorr abil abject abl ablest abolish
##   dispt_fed_49.txt    0   0    0      0    0      0   2      0       0
##   dispt_fed_50.txt    0   0    0      0    0      0   0      0       0
##   dispt_fed_51.txt    0   0    0      0    0      0   1      0       0
##   dispt_fed_52.txt    0   0    0      0    1      0   1      0       0
##   dispt_fed_53.txt    1   0    0      0    0      0   0      0       0
##   dispt_fed_54.txt    0   0    0      0    0      0   0      0       0
##   dispt_fed_55.txt    0   0    0      0    0      0   0      0       0
##   dispt_fed_56.txt    0   0    0      0    0      0   0      0       0
##   dispt_fed_57.txt    0   0    0      1    0      0   0      0       0
##   dispt_fed_62.txt    0   0    0      0    0      0   1      0       0
##   dispt_fed_63.txt    0   0    0      0    0      0   4      0       0
```

Also convert to DF

```
Papers_DF <- as.data.frame(as.matrix(Papers_DTM))
str(Papers_DF)
```

```
## 'data.frame':    85 obs. of  4900 variables:
##  $ abandon        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ abat           : num  0 0 0 0 1 0 0 0 0 0 ...
##  $ abb            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ abet           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ abhorr         : num  0 0 0 0 0 0 0 0 1 0 ...
##  $ abil           : num  0 0 0 1 0 0 0 0 0 0 ...
##  $ abject         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ abl            : num  2 0 1 1 0 0 0 0 0 1 ...
##  $ ablest         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ abolish        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ abolit         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ abort          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ abound         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ abraham        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ abreg          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ abridg         : num  0 0 0 1 0 0 0 0 0 0 ...
##  $ abroad         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ abrog          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ absenc         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ absolut        : num  0 2 2 1 0 0 0 0 0 0 ...
##  $ absolv         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ absorb         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ abstain        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ abstract       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ abstrus        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ absurd         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ abund          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ abus           : num  1 1 2 1 1 0 0 0 0 0 ...
##  $ abyss          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ acced          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ acceler        : num  0 0 0 0 1 0 0 0 0 0 ...
##  $ accept         : num  0 0 0 0 0 0 0 0 0 1 ...
##  $ access         : num  0 0 0 2 0 0 0 0 0 0 ...
##  $ accid          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ accident       : num  0 0 0 1 0 0 0 0 0 0 ...
##  $ accommod       : num  0 0 0 0 1 0 0 0 0 0 ...
##  $ accomod        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ accompani      : num  0 0 0 0 0 0 0 1 0 0 ...
##  $ accomplic      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ accomplish     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ accord         : num  0 0 0 0 1 2 2 1 1 0 ...
##  $ account        : num  0 0 0 0 0 0 1 0 0 0 ...
##  $ accret         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ accru          : num  0 0 0 0 0 0 0 0 0 1 ...
##  $ accumul        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ accur          : num  1 0 0 0 1 0 0 0 0 1 ...
##  $ accuraci       : num  0 0 0 0 0 1 0 0 0 0 ...
##  $ accus          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ accustom       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ achaean        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ achaeus        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ achaia         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ achiev         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ acknowledg     : num  0 1 0 0 0 0 0 0 0 1 ...
##  $ acquaint       : num  1 0 0 0 2 0 0 2 0 1 ...
##  $ acquiesc       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ acquir         : num  1 0 0 0 5 0 0 2 0 0 ...
##  $ acquisit       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ acquit         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ acr            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ act            : num  0 0 0 1 2 1 0 1 0 1 ...
##  $ action         : num  0 0 1 0 0 0 0 0 0 1 ...
##  $ activ          : num  0 4 0 0 0 0 0 0 0 0 ...
##  $ actor          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ actual         : num  1 2 0 0 4 0 0 0 1 0 ...
##  $ actuat         : num  0 0 0 0 0 0 1 0 1 0 ...
##  $ acut           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ adag           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ adapt          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ add            : num  0 0 0 0 1 0 0 1 1 0 ...
##  $ addict         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ addit          : num  0 0 1 1 0 0 0 0 1 1 ...
##  $ address        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ adduc          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ adept          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ adequ          : num  1 1 0 0 0 0 0 0 0 0 ...
##  $ adher          : num  0 0 1 0 0 1 0 0 0 0 ...
##  $ adjac          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ adjoin         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ adjourn        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ adjud          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ adjudg         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ adjust         : num  0 0 0 0 0 1 0 0 0 0 ...
##  $ administ       : num  0 0 2 0 0 0 0 0 0 1 ...
##  $ administr      : num  1 2 1 0 0 0 0 0 1 0 ...
##  $ admir          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ admiralgener   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ admiralti      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ admiss         : num  0 0 0 0 0 1 0 0 1 1 ...
##  $ admit          : num  1 0 3 0 1 5 2 0 1 0 ...
##  $ admitt         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ admonish       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ admonit        : num  0 0 0 0 0 0 0 0 0 1 ...
##  $ adopt          : num  0 0 0 1 0 1 0 0 0 1 ...
##  $ adroit         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ adul           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ advanc         : num  0 0 0 0 1 0 0 1 1 2 ...
##  $ advantag       : num  4 1 0 2 2 4 0 1 0 7 ...
##  $ adventiti      : num  0 0 0 0 0 0 0 0 0 0 ...
##   [list output truncated]
```

```
(Papers_DF$abolit)
```

```
##  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [36] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 3 0 0 1 1 0 0 0 0 0 0 0
## [71] 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0
```

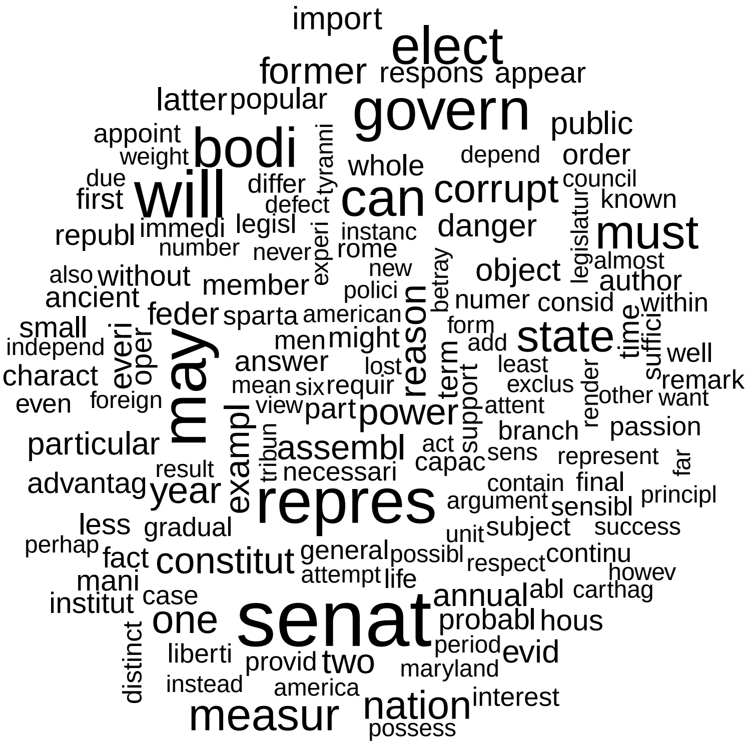
```
(nrow(Papers_DF))  ## Each row is Paper
```

```
## [1] 85
```

Example Word Cloud

Note: this word cloud package requires our data to be in DTM format

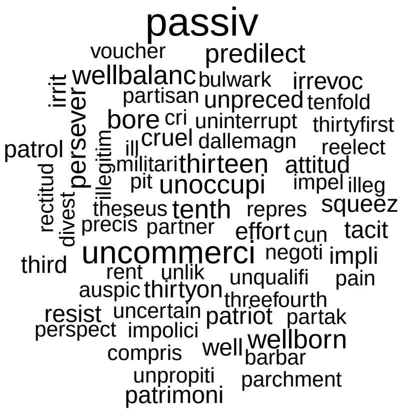
```
#Wordcloud Visualization Hamilton, Madison and Disputed Papers
DisputedPapersWC<- wordcloud(colnames(Papers_dtm_matrix), Papers_dtm_matrix[11, ])
```



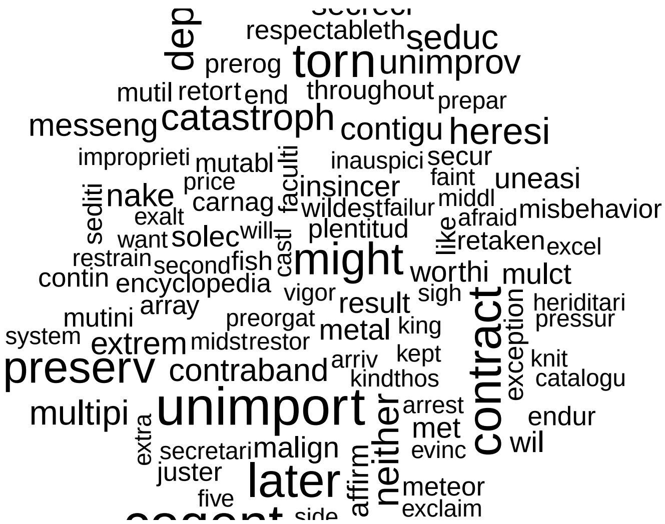
```
(head(sort(as.matrix(Papers_DTM)[11,], decreasing = TRUE), n=50))
```

```
##      peopl      senat       will        may     repres     govern
##         42         24         19         18         18         16
##       bodi        can      elect       must     measur      state
##         15         14         14         12         11         11
##    corrupt     nation        one  constitut     former      power
##          9          9          9          8          8          8
##     reason       year    assembl     exampl        two     annual
##          8          8          7          7          7          6
##     danger      everi       evid      feder     import     latter
##          6          6          6          6          6          6
##     object particular     public   advantag    ancient     answer
##          6          6          6          5          5          5
##     appear     author    charact       fact      first       hous
##          5          5          5          5          5          5
##   institut       less       mani     member      might       oper
##          5          5          5          5          5          5
##      order       part
##          5          5
```

```
HamiltonPapersWC <- wordcloud(colnames(Papers_dtm_matrix), Papers_dtm_matrix[12:62, ])
```



```r
MadisonPapersHW <- wordcloud(colnames(Papers_dtm_matrix), Papers_dtm_matrix[63:77, ])
```



Analysis

Once the data is vectorized, we can analyze! Lets apply some distance metrics and see how the data clusters!! Note: Cosine distance usually works well with high dimensional data.

Distance Metrics

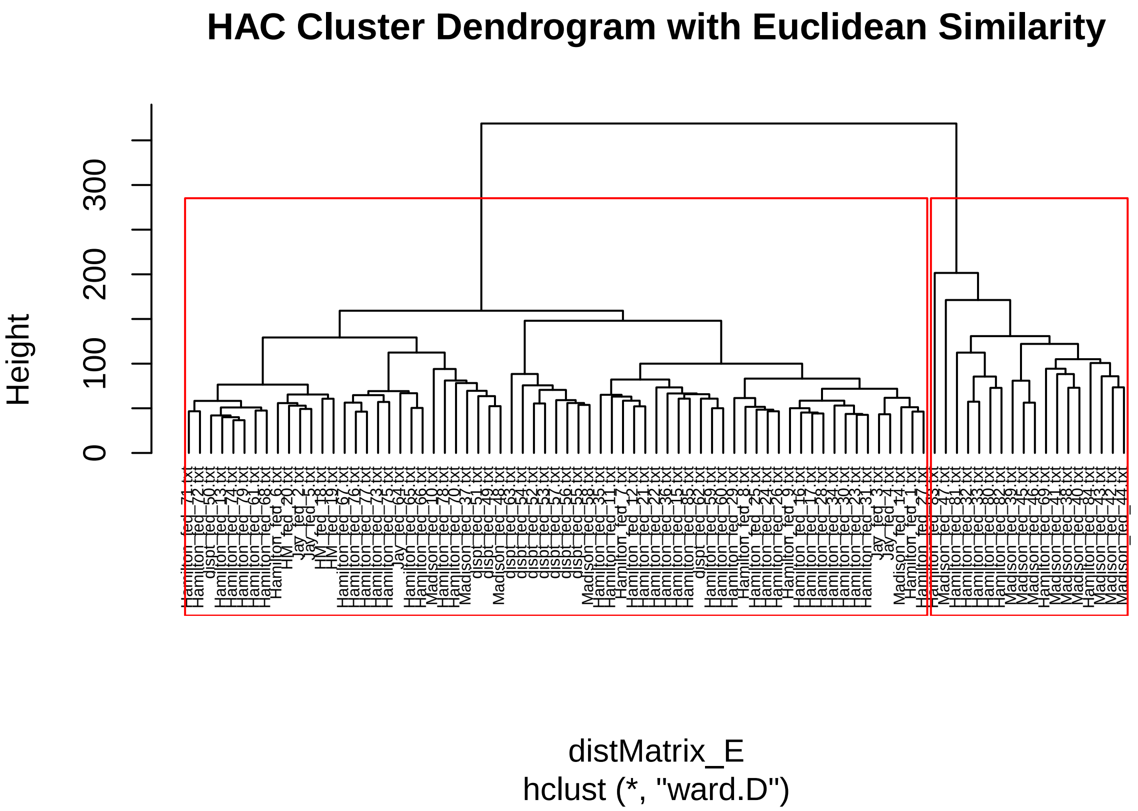
Below we compute a variety of distance matrices to determine which seems to work the best!

```r
###Distance Measure
m <- Papers_dtm_matrix
m_norm <- Papers_Matrix_Norm
#m <- [1:2, 1:3]
distMatrix_E <- dist(m, method="euclidean")
#print(distMatrix_E)
distMatrix_M <- dist(m, method="manhattan")
#print(distMatrix_M)
distMatrix_C <- dist(m, method="cosine")
#print(distMatrix_C)
distMatrix_C_norm <- dist(m_norm, method="cosine")
#print(distMatrix_C_norm)
##Cosine similarity works the best. Norm and not norm is about
## the same because the size of the Papers are not sig diff.
```

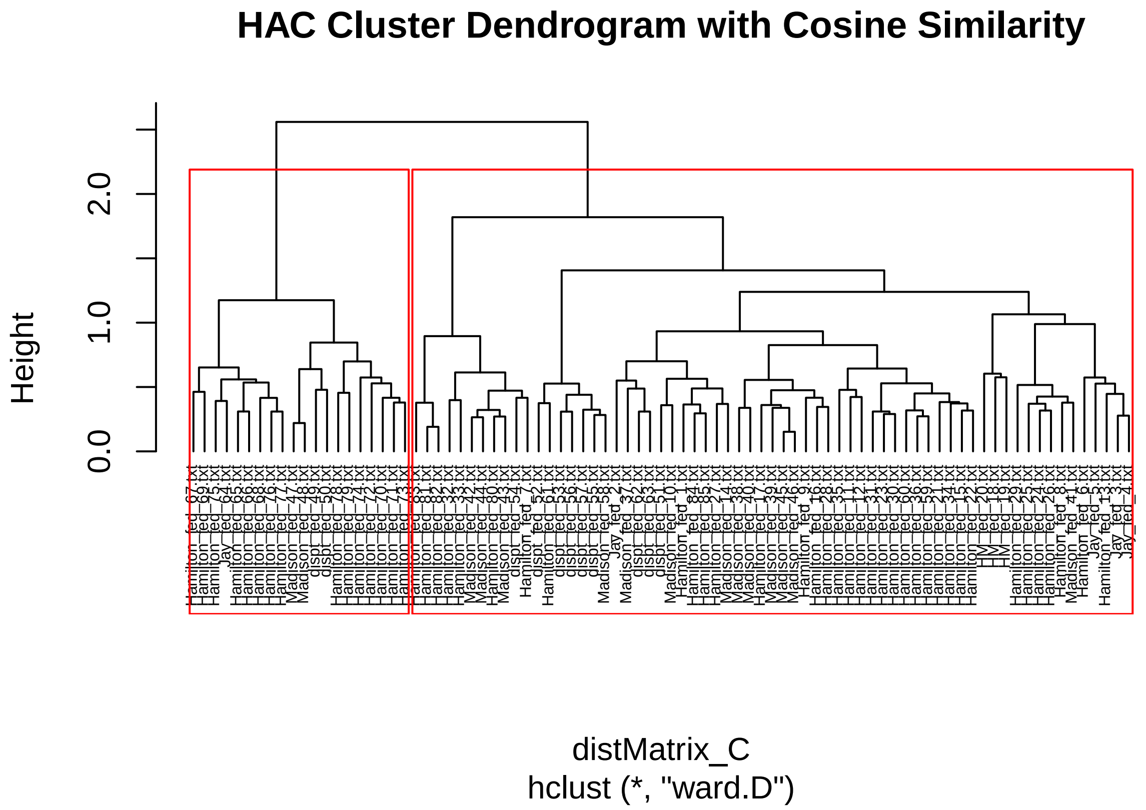
##Clustering

Below are some HAC results. Which does best??? Why?

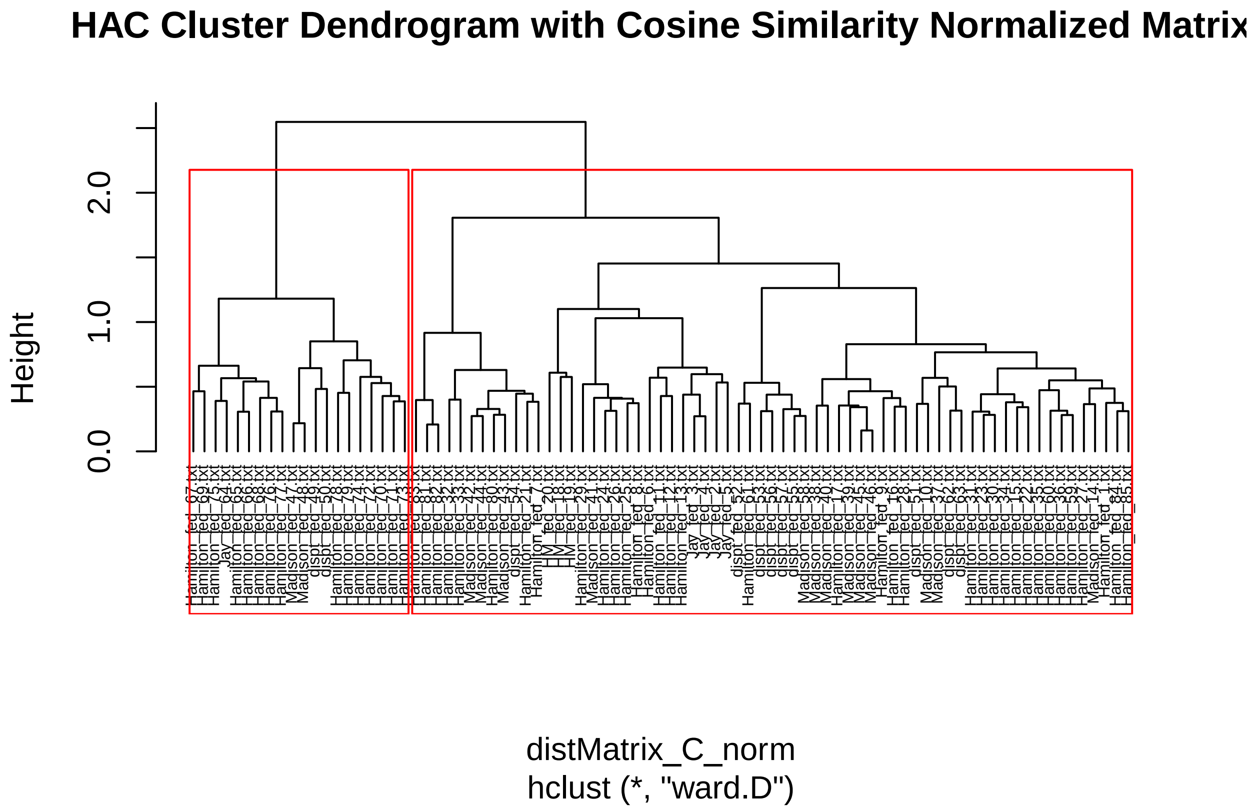
```
###Clustering Methods:
## HAC: Hierarchical Algorithm Clustering Method
## Euclidean
groups_E <- hclust(distMatrix_E,method="ward.D")
plot(groups_E, cex=0.5, font=22, hang=-1, main = "HAC Cluster Dendrogram with Euclidean Similarity")
rect.hclust(groups_E, k=2)
```



```
## Cosine Similarity
groups_C <- hclust(distMatrix_C,method="ward.D")
plot(groups_C, cex=0.5,font=22, hang=-1,main = "HAC Cluster Dendrogram with Cosine Similarity")
rect.hclust(groups_C, k=2)
```



```
## Cosine Similarity for Normalized Matrix
groups_C_n <- hclust(distMatrix_C_norm,method="ward.D")
plot(groups_C_n, cex=0.5, font=22, hang=-1,  main = "HAC Cluster Dendrogram with Cosine Similarity Norma
rect.hclust(groups_C_n, k=2)
```



Below are some k-means results … how would you assess these results? Are they intuitive? Why?

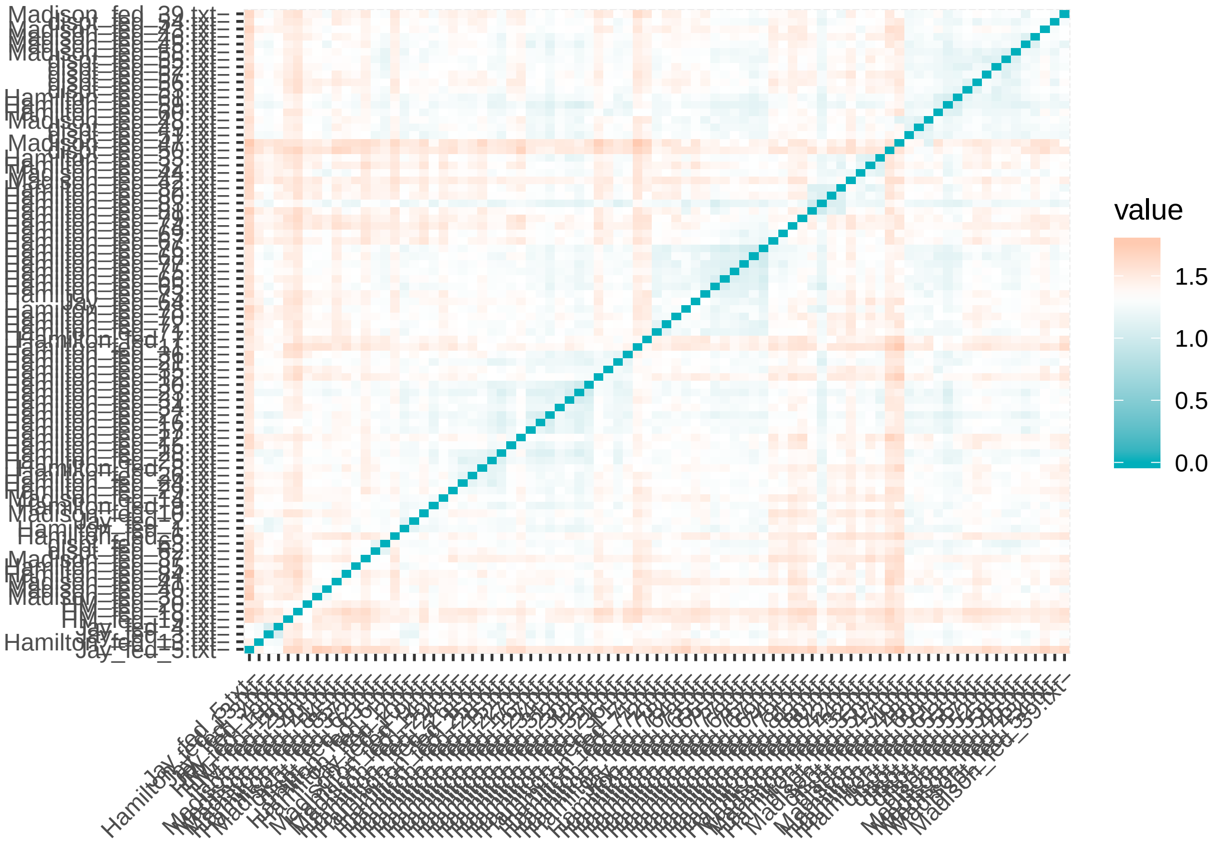
```
## k means clustering Methods
X <- m_norm
k2 <- kmeans(X, centers = 2, nstart = 100, iter.max = 50)
str(k2)
```

```
## List of 9
##  $ cluster     : Named int [1:85] 1 1 1 2 2 2 1 2 2 1 ...
##   ..- attr(*, "names")= chr [1:85] "dispt_fed_49.txt" "dispt_fed_50.txt" "dispt_fed_51.txt" "dispt_fe
##  $ centers     : num [1:2, 1:4900] 1.09e-04 6.67e-05 1.82e-05 3.33e-05 9.09e-05 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:2] "1" "2"
##   .. ..$ : chr [1:4900] "abandon" "abat" "abb" "abet" ...
##  $ totss       : num 0.216
##  $ withinss    : num [1:2] 0.1231 0.0794
##  $ tot.withinss: num 0.203
##  $ betweenss   : num 0.0137
##  $ size        : int [1:2] 55 30
##  $ iter        : int 1
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

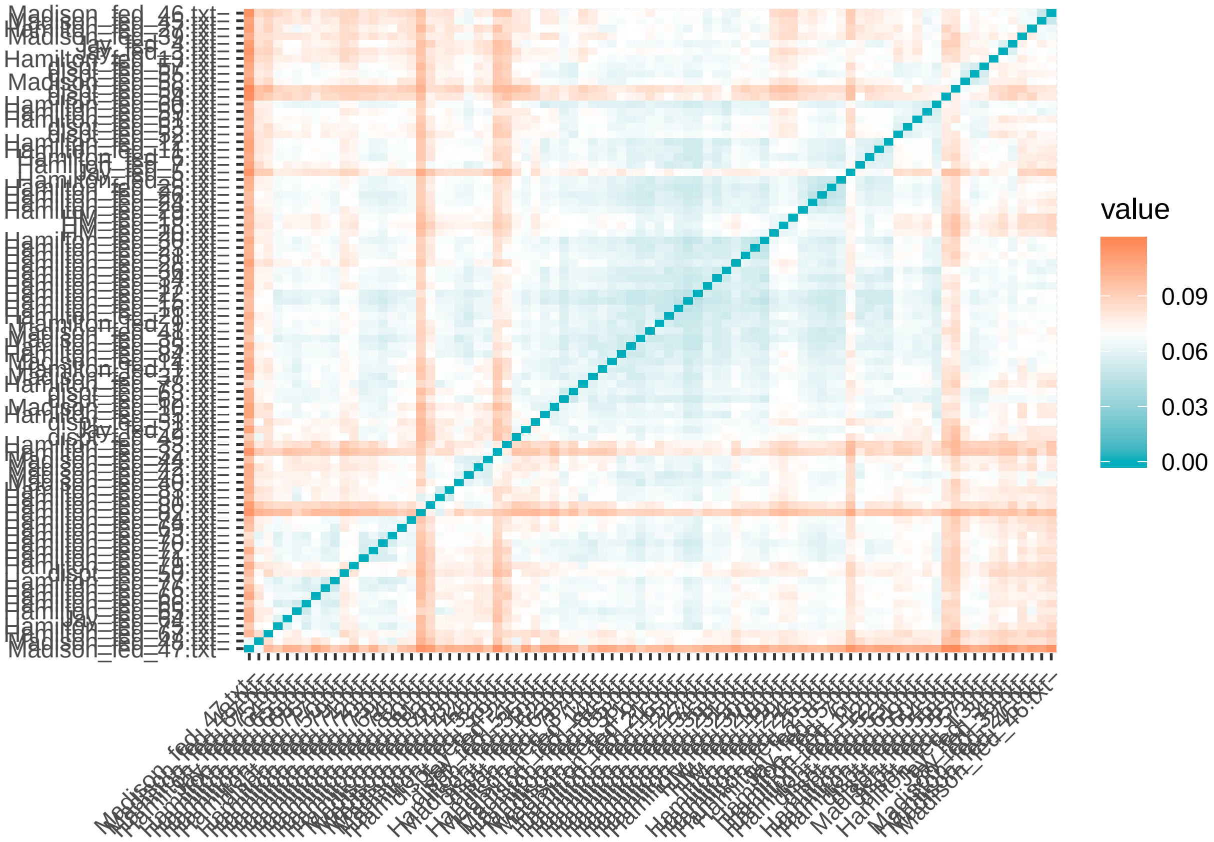
```
k3 <- kmeans(X, centers = 7, nstart = 50, iter.max= 50)
str(k3)
```

```
## List of 9
##  $ cluster     : Named int [1:85] 5 5 3 7 7 7 7 7 7 3 ...
##   ..- attr(*, "names")= chr [1:85] "dispt_fed_49.txt" "dispt_fed_50.txt" "dispt_fed_51.txt" "dispt_fe
##  $ centers     : num [1:7, 1:4900] 0.00 0.00 2.14e-04 0.00 5.26e-05 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:7] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:4900] "abandon" "abat" "abb" "abet" ...
##  $ totss       : num 0.216
##  $ withinss    : num [1:7] 0.01163 0.0068 0.02827 0.00201 0.04062 ...
##  $ tot.withinss: num 0.163
##  $ betweenss   : num 0.0531
##  $ size        : int [1:7] 6 4 14 2 19 32 8
##  $ iter        : int 4
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

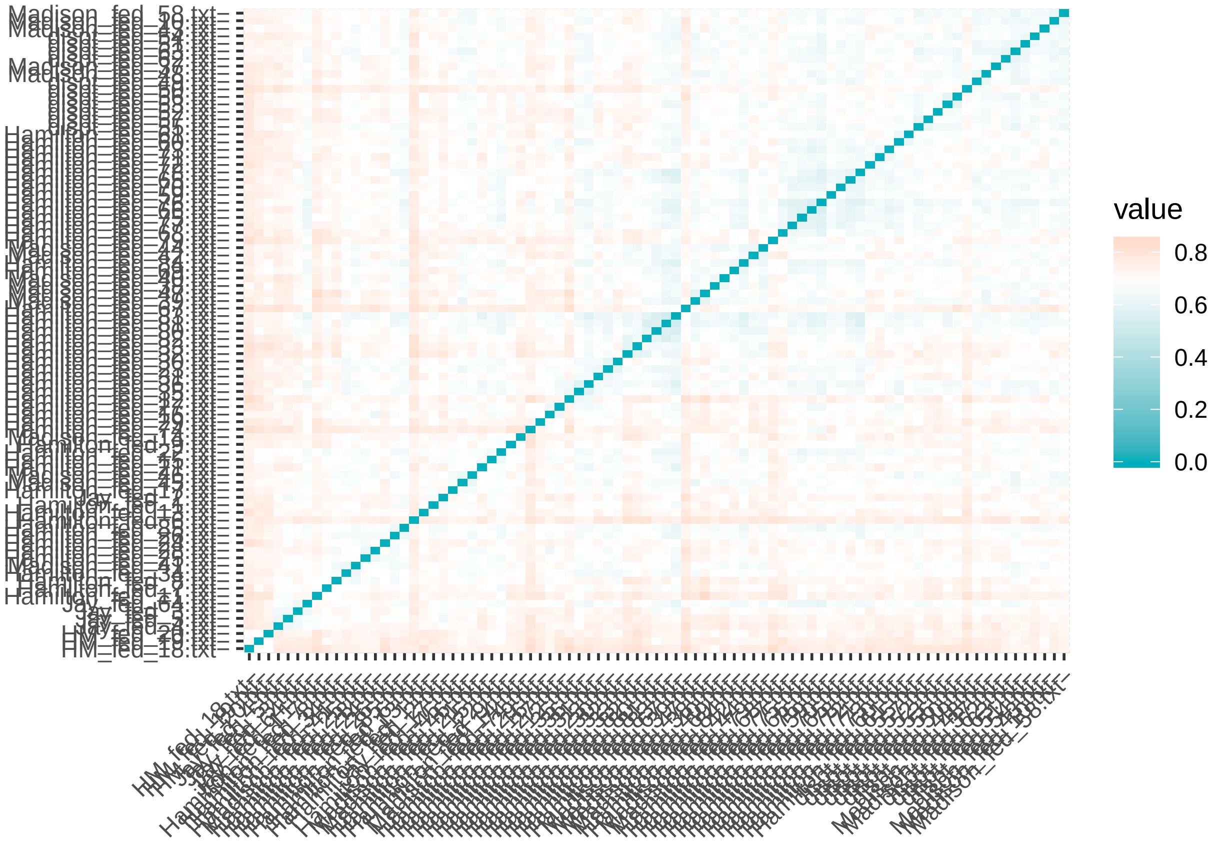
```
## k means visualization results!
distance1 <- get_dist(X,method = "manhattan")
fviz_dist(distance1, gradient = list(low = "#00AFBB", mid = "white", high = "#FC4E07"))
```



```
distance2 <- get_dist(X,method = "euclidean")
fviz_dist(distance2, gradient = list(low = "#00AFBB", mid = "white", high = "#FC4E07"))
```



```
distance3 <- get_dist(X,method = "spearman")
fviz_dist(distance3, gradient = list(low = "#00AFBB", mid = "white", high = "#FC4E07", title= "Distance
```



We can visualize the k-means results as follows. This viz package can be finicky – check the other tutorial for other options :)

```r
str(X)
```

```
##  num [1:85, 1:4900] 0 0 0 0 0 0 0 0 0 0 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ Docs : chr [1:85] "dispt_fed_49.txt" "dispt_fed_50.txt" "dispt_fed_51.txt" "dispt_fed_52.txt"
##   ..$ Terms: chr [1:4900] "abandon" "abat" "abb" "abet" ...
```

```r
## k means
kmeansFIT_1 <- kmeans(X, centers = 4)
#(kmeansFIT1)
summary(kmeansFIT_1)
```

```
##              Length Class  Mode   
## cluster         85  -none- numeric
## centers      19600  -none- numeric
## totss            1  -none- numeric
## withinss         4  -none- numeric
## tot.withinss     1  -none- numeric
## betweenss        1  -none- numeric
## size             4  -none- numeric
## iter             1  -none- numeric
## ifault           1  -none- numeric
```

```r
#(kmeansFIT_1$cluster)
#fviz_cluster(kmeansFIT_1, data = X)
```

```
kmeansFIT_2 <- kmeans(X, centers = 3)
#(kmeansFIT2)
summary(kmeansFIT_2)
```

```
##              Length Class  Mode   
## cluster         85  -none- numeric
## centers      14700  -none- numeric
## totss            1  -none- numeric
## withinss         3  -none- numeric
## tot.withinss     1  -none- numeric
## betweenss        1  -none- numeric
## size             3  -none- numeric
## iter             1  -none- numeric
## ifault           1  -none- numeric
```

```
#(kmeansFIT_2$cluster)
#fviz_cluster(kmeansFIT_2, data = X)
```