

Sharat_Sripada_HW5

```
# install.packages('tm')
# install.packages('tmap')
# install.packages('quanteda')
# install.packages('philentropy')
# install.packages('factoextra')
# install.packages('rpart')
# install.packages('tibble')
# install.packages('rattle')
# install.packages('randomForest')

library(tm)

## Loading required package: NLP

library(tmap)
library(quanteda)

## Package version: 2.1.1

## Parallel computing: 2 of 4 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 objects are masked from 'package:NLP':
##
##   meta, meta<-

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

library(RColorBrewer)
library(wordcloud)
library(philentropy)
library(factoextra)

## Loading required package: ggplot2

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

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

## Welcome! Want to learn more? See two factoextra-related books at
https://goo.gl/ve3WBa

library(rpart)
library(tibble)
library(rattle)

## Loading required package: bitops

## Rattle: A free graphical interface for data science with R.
## Version 5.4.0 Copyright (c) 2006-2020 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.

library(rpart.plot)
library(randomForest)

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:rattle':
##
##      importance

## The following object is masked from 'package:ggplot2':
##
##      margin
```

Introduction

See section related 'Classification using Decision Trees' for work related to HW5. The remainder of the homework uses the same data-structures, visualizations etc as HW-4

We begin our analysis by ingesting a corpus of documents and running through the following pipelines:

- loading the documents using the R Corpus function
- build a document term matrix (DTM)
- visualize wordclouds
- dive into the core concepts of clustering
- classify disputed documents from results of clustering

```
#Load the data/corpus
```

```
FedPapersCorpus <-
```

```
Corpus(DirSource("/Users/venkatasharatsripada/Downloads/IST707repo-  
master/FedPapersCorpus"))
```

```
numFedPapers <- length(FedPapersCorpus)
```

```
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]])

```

```

#Ignore extremely rare words - <2% of documents
(minTermFreq <- 0.02 * numFedPapers)

```

```
## [1] 1.7

#Also, ignore common words - >75%-95% of documents
(maxTermFreq <- 0.95 * numFedPapers)

## [1] 80.75

#
Papers_DTM <- DocumentTermMatrix(FedPapersCorpus,
                                  control=list(
                                    stopwords=TRUE,
                                    wordLengths=c(3,15),
                                    removePunctuation=T,
                                    removeNumbers=T,
                                    tolower=T,
                                    stemming=T,
                                    remove_separators=T,
                                    bounds=list(global=c(minTermFreq,
maxTermFreq)))
                                  ))

DTM <- as.matrix(Papers_DTM)
(DTM[1:11,1:10])

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

col_WordFreq <- colSums(as.matrix(Papers_DTM))
(head(col_WordFreq))
```

```

## abandon    abat    abb    abet    abil    abl
##          9      2      5      2     15     74

#Length of all words
(length(col_WordFreq))

## [1] 3370

(row_WordFreq <- rowSums(as.matrix(Papers_DTM)))

##    dispt_fed_49.txt    dispt_fed_50.txt    dispt_fed_51.txt
dispt_fed_52.txt
##              677              480              783
743
##    dispt_fed_53.txt    dispt_fed_54.txt    dispt_fed_55.txt
dispt_fed_56.txt
##              903              766              865
649
##    dispt_fed_57.txt    dispt_fed_62.txt    dispt_fed_63.txt
Hamilton_fed_1.txt
##              889              983              1244
659
## Hamilton_fed_11.txt Hamilton_fed_12.txt Hamilton_fed_13.txt
Hamilton_fed_15.txt
##              1020              901              400
1256
## Hamilton_fed_16.txt Hamilton_fed_17.txt Hamilton_fed_21.txt
Hamilton_fed_22.txt
##              814              663              823
1494
## Hamilton_fed_23.txt Hamilton_fed_24.txt Hamilton_fed_25.txt
Hamilton_fed_26.txt
##              717              826              825
983
## Hamilton_fed_27.txt Hamilton_fed_28.txt Hamilton_fed_29.txt
Hamilton_fed_30.txt
##              573              639              876
819
## Hamilton_fed_31.txt Hamilton_fed_32.txt Hamilton_fed_33.txt
Hamilton_fed_34.txt
##              673              589              640
883
## Hamilton_fed_35.txt Hamilton_fed_36.txt Hamilton_fed_59.txt
Hamilton_fed_6.txt
##              942              1095              720
868
## Hamilton_fed_60.txt Hamilton_fed_61.txt Hamilton_fed_65.txt
Hamilton_fed_66.txt
##              892              591              816
899
## Hamilton_fed_67.txt Hamilton_fed_68.txt Hamilton_fed_69.txt

```

```

Hamilton_fed_7.txt
##          688          604          1174
952
## Hamilton_fed_70.txt Hamilton_fed_71.txt Hamilton_fed_72.txt
Hamilton_fed_73.txt
##          1295          677          842
941
## Hamilton_fed_74.txt Hamilton_fed_75.txt Hamilton_fed_76.txt
Hamilton_fed_77.txt
##          422          822          796
798
## Hamilton_fed_78.txt Hamilton_fed_79.txt Hamilton_fed_8.txt
Hamilton_fed_80.txt
##          1245          421          892
974
## Hamilton_fed_81.txt Hamilton_fed_82.txt Hamilton_fed_83.txt
Hamilton_fed_84.txt
##          1581          642          2374
1656
## Hamilton_fed_85.txt Hamilton_fed_9.txt HM_fed_18.txt
HM_fed_19.txt
##          1114          808          926
907
##      HM_fed_20.txt      Jay_fed_2.txt      Jay_fed_3.txt
Jay_fed_4.txt
##          692          709          622
663
##      Jay_fed_5.txt      Jay_fed_64.txt Madison_fed_10.txt
Madison_fed_14.txt
##          605          966          1316
882
## Madison_fed_37.txt Madison_fed_38.txt Madison_fed_39.txt
Madison_fed_40.txt
##          1122          1348          981
1132
## Madison_fed_41.txt Madison_fed_42.txt Madison_fed_43.txt
Madison_fed_44.txt
##          1479          1140          1344
1178
## Madison_fed_45.txt Madison_fed_46.txt Madison_fed_47.txt
Madison_fed_48.txt
##          810          980          1167
738
## Madison_fed_58.txt
##          847

```

Normalization

#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)
```

```
#compare the original and normalized version
```

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

```
##              Terms
## Docs          edit effect effectu efficaci effici effort eight eighth
## dispt_fed_49.txt    0      1      1        0      0      0      0      0
## dispt_fed_50.txt    0      3      0        0      0      0      0      0
## dispt_fed_51.txt    0      0      0        0      0      0      0      0
## dispt_fed_52.txt    0      1      1        0      0      0      0      0
## dispt_fed_53.txt    0      2      1        0      0      0      0      0
## dispt_fed_54.txt    0      3      0        2      0      0      0      0
## dispt_fed_55.txt    0      0      0        0      0      0      1      0
## dispt_fed_56.txt    0      2      0        0      0      0      3      0
## dispt_fed_57.txt    0      0      2        0      0      0      0      0
## dispt_fed_62.txt    0      4      0        0      0      0      0      0
## dispt_fed_63.txt    0      2      2        0      0      0      0      0
```

```
##              Terms
## Docs          either elaps elect
## dispt_fed_49.txt    1      0      1
## dispt_fed_50.txt    3      0      2
## dispt_fed_51.txt    0      0      1
## dispt_fed_52.txt    0      0     21
## dispt_fed_53.txt    2      1     20
## dispt_fed_54.txt    0      0      1
## dispt_fed_55.txt    2      0      3
## dispt_fed_56.txt    2      0      3
## dispt_fed_57.txt    0      0     10
## dispt_fed_62.txt    0      0      2
## dispt_fed_63.txt    0      0     14
```

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

```
##              Terms
## Docs          edit effect effectu efficaci effici effort eight eighth
## dispt_fed_49.txt    0 0.001 0.001 0.000      0      0 0.000      0
## dispt_fed_50.txt    0 0.006 0.000 0.000      0      0 0.000      0
## dispt_fed_51.txt    0 0.000 0.000 0.000      0      0 0.000      0
## dispt_fed_52.txt    0 0.001 0.001 0.000      0      0 0.000      0
## dispt_fed_53.txt    0 0.002 0.001 0.000      0      0 0.000      0
## dispt_fed_54.txt    0 0.004 0.000 0.003      0      0 0.000      0
## dispt_fed_55.txt    0 0.000 0.000 0.000      0      0 0.001      0
## dispt_fed_56.txt    0 0.003 0.000 0.000      0      0 0.005      0
## dispt_fed_57.txt    0 0.000 0.002 0.000      0      0 0.000      0
## dispt_fed_62.txt    0 0.004 0.000 0.000      0      0 0.000      0
## dispt_fed_63.txt    0 0.002 0.002 0.000      0      0 0.000      0
```

```
##              Terms
## Docs          either elaps elect
## dispt_fed_49.txt 0.001 0.000 0.001
```



```
## dispt_fed_50.txt 0.006 0.000 0.004
## dispt_fed_51.txt 0.000 0.000 0.001
## dispt_fed_52.txt 0.000 0.000 0.028
## dispt_fed_53.txt 0.002 0.001 0.022
## dispt_fed_54.txt 0.000 0.000 0.001
## dispt_fed_55.txt 0.002 0.000 0.003
## dispt_fed_56.txt 0.003 0.000 0.005
## dispt_fed_57.txt 0.000 0.000 0.011
## dispt_fed_62.txt 0.000 0.000 0.002
## dispt_fed_63.txt 0.000 0.000 0.011
```

*#verify for word 'embarrass' in document 'dispt_fed_62.txt' if the
#normalization math is correct*

(row_WordFreq)

```
## dispt_fed_49.txt dispt_fed_50.txt dispt_fed_51.txt
dispt_fed_52.txt
## 677 480 783
743
## dispt_fed_53.txt dispt_fed_54.txt dispt_fed_55.txt
dispt_fed_56.txt
## 903 766 865
649
## dispt_fed_57.txt dispt_fed_62.txt dispt_fed_63.txt
Hamilton_fed_1.txt
## 889 983 1244
659
## Hamilton_fed_11.txt Hamilton_fed_12.txt Hamilton_fed_13.txt
Hamilton_fed_15.txt
## 1020 901 400
1256
## Hamilton_fed_16.txt Hamilton_fed_17.txt Hamilton_fed_21.txt
Hamilton_fed_22.txt
## 814 663 823
1494
## Hamilton_fed_23.txt Hamilton_fed_24.txt Hamilton_fed_25.txt
Hamilton_fed_26.txt
## 717 826 825
983
## Hamilton_fed_27.txt Hamilton_fed_28.txt Hamilton_fed_29.txt
Hamilton_fed_30.txt
## 573 639 876
819
## Hamilton_fed_31.txt Hamilton_fed_32.txt Hamilton_fed_33.txt
Hamilton_fed_34.txt
## 673 589 640
883
## Hamilton_fed_35.txt Hamilton_fed_36.txt Hamilton_fed_59.txt
Hamilton_fed_6.txt
```

##	942	1095	720
868			
##	Hamilton_fed_60.txt	Hamilton_fed_61.txt	Hamilton_fed_65.txt
	Hamilton_fed_66.txt		
##	892	591	816
899			
##	Hamilton_fed_67.txt	Hamilton_fed_68.txt	Hamilton_fed_69.txt
	Hamilton_fed_7.txt		
##	688	604	1174
952			
##	Hamilton_fed_70.txt	Hamilton_fed_71.txt	Hamilton_fed_72.txt
	Hamilton_fed_73.txt		
##	1295	677	842
941			
##	Hamilton_fed_74.txt	Hamilton_fed_75.txt	Hamilton_fed_76.txt
	Hamilton_fed_77.txt		
##	422	822	796
798			
##	Hamilton_fed_78.txt	Hamilton_fed_79.txt	Hamilton_fed_8.txt
	Hamilton_fed_80.txt		
##	1245	421	892
974			
##	Hamilton_fed_81.txt	Hamilton_fed_82.txt	Hamilton_fed_83.txt
	Hamilton_fed_84.txt		
##	1581	642	2374
1656			
##	Hamilton_fed_85.txt	Hamilton_fed_9.txt	HM_fed_18.txt
	HM_fed_19.txt		
##	1114	808	926
907			
##	HM_fed_20.txt	Jay_fed_2.txt	Jay_fed_3.txt
	Jay_fed_4.txt		
##	692	709	622
663			
##	Jay_fed_5.txt	Jay_fed_64.txt	Madison_fed_10.txt
	Madison_fed_14.txt		
##	605	966	1316
882			
##	Madison_fed_37.txt	Madison_fed_38.txt	Madison_fed_39.txt
	Madison_fed_40.txt		
##	1122	1348	981
1132			
##	Madison_fed_41.txt	Madison_fed_42.txt	Madison_fed_43.txt
	Madison_fed_44.txt		
##	1479	1140	1344
1178			
##	Madison_fed_45.txt	Madison_fed_46.txt	Madison_fed_47.txt
	Madison_fed_48.txt		
##	810	980	1167
738			

```
## Madison_fed_58.txt
## 847
```

*#dispt_fed_62 has 798 words in total
 #there are 2x words of 'embarrass' so, 2/798 = 0.0025 ~0.003 (3 places after decimal)*

Data-structures

```
Papers_dtm_matrix <- as.matrix(Papers_DTM)
str(Papers_dtm_matrix)
```

```
## num [1:85, 1:3370] 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 [1:3370] "abandon" "abat" "abb" "abet" ...
```

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

```
##               Terms
## Docs          abat abb abet abil abl ablest abolish abolit abort
## dispt_fed_49.txt    0  0   0   0  2    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  1    0      0      0      0
## dispt_fed_52.txt    0  0   0   1  1    0      0      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   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   0   0  4    0      0      0      0
```

Convert to a data-frame

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

```
## 'data.frame':    85 obs. of  3370 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 ...
## $ abil         : num  0 0 0 1 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 ...
## $ abridg       : num  0 0 0 1 0 0 0 0 0 0 ...
## $ abroad       : num  0 0 0 0 0 0 0 0 0 0 ...
```

##	\$ absolut	:	num	0	2	2	1	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	...
##	\$ 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	...
##	\$ 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	...
##	\$ 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	...
##	\$ 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	...
##	\$ achaeen	:	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	...
##	\$ 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	...
##	\$ 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	...
##	\$ adjourn	:	num	0	0	0	0	0	0	0	0	0	0	...
##	\$ adjud	:	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 ...
## $ 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 ...
## $ 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 ...
## $ adventur   : num 0 0 0 0 0 0 0 0 0 0 ...
## $ advers     : num 2 0 0 0 0 0 0 0 0 0 ...
## $ adversari  : num 0 0 0 0 0 0 0 0 0 0 ...
## $ advert     : num 0 0 0 0 0 0 0 0 0 0 ...
## $ advertis   : num 0 0 0 0 0 0 0 0 0 0 ...
## $ advic      : num 0 0 0 0 0 0 0 0 0 1 ...
## $ advis      : num 0 0 0 0 0 0 0 0 0 0 ...
## $ advoc      : num 0 0 0 0 0 1 0 1 0 0 ...
## $ affair     : num 0 0 1 0 9 0 1 5 0 4 ...
## $ affect     : num 0 0 0 1 0 0 0 0 1 1 ...
## $ affin      : num 0 0 0 0 0 0 0 0 0 0 ...
## $ affirm     : num 0 0 0 0 2 0 0 0 0 1 ...
## $ afford     : num 0 0 0 0 1 0 0 0 0 0 ...
## $ affront    : num 0 0 0 0 0 0 0 0 0 0 ...
## $ afraid     : num 0 0 0 0 0 0 1 0 0 0 ...
## $ afterward  : num 0 0 0 0 0 0 0 0 0 0 ...
## $ age        : num 0 0 0 1 0 0 0 0 0 2 ...
## $ agenc      : num 0 0 1 0 0 0 0 0 0 1 ...
## $ agent      : num 1 1 0 0 0 0 0 0 0 0 ...
## $ aggrand    : num 0 0 0 0 0 0 0 0 0 0 ...
## $ aggrandiz  : num 1 0 0 0 0 0 0 0 1 0 ...
## $ aggreg     : num 0 0 0 0 0 2 0 0 0 0 ...
## $ aggress    : num 0 0 0 0 0 0 0 0 0 0 ...
## $ aggressor  : num 0 0 0 0 0 0 0 0 0 0 ...
## $ agit       : num 0 0 0 0 0 0 0 0 0 0 ...
## [list output truncated]
```

Example word cloud

Breaking the word clouds based on the document list: - 1:11 -> disputed papers - 12:62 -> Hamilton papers - 63:70 -> Ignoring HM_fed, Jay_fed papers - 71:85 -> Madison papers

```
disputedpaperswc <- wordcloud(colnames(Papers_dtm_matrix),
Papers_dtm_matrix[11,], scale=c(3.5,0.25))

## Warning in wordcloud(colnames(Papers_dtm_matrix), Papers_dtm_matrix[11, :
senat
## could not be fit on page. It will not be plotted.
```



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

##	senat	repres	bodi	can	elect	measur
corrupt						
##	24	18	15	14	14	11
9						
##	nation	constitut	former	reason	year	assembl
exempl						
##	9	8	8	8	8	7
7						
##	two	annual	danger	everi	evid	feder
import						
##	7	6	6	6	6	6
6						
##	latter	object	particular	public	advantag	ancient
answer						
##	6	6	6	6	5	5
5						
##	appear	charact	fact	first	hous	institut
less						
##	5	5	5	5	5	5
5						
##	mani	member	might	oper	order	popular

```

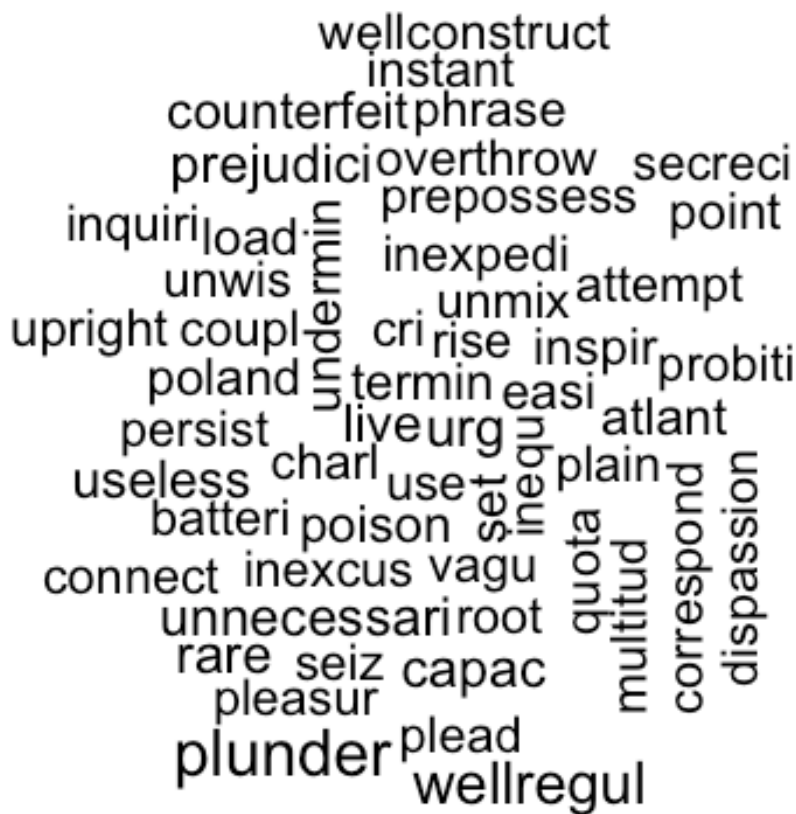
probal
##      5      5      5      5      5      5
5
##      republ      respons      small      term      time      whole
without
##      5      5      5      5      5      5
5
##      abl
##      4

```

```

HamiltonPapersWC <- wordcloud(colnames(Papers_dtm_matrix),
Papers_dtm_matrix[12:62, ], scale=c(3.5,1.25))

```



```

MadisonPapersWC <- wordcloud(colnames(Papers_dtm_matrix),
Papers_dtm_matrix[71:85, ], scale=c(2.5,1.25))

## Warning in wordcloud(colnames(Papers_dtm_matrix), Papers_dtm_matrix[71:85,
:
## mankind could not be fit on page. It will not be plotted.

## Warning in wordcloud(colnames(Papers_dtm_matrix), Papers_dtm_matrix[71:85,
:
## imposs could not be fit on page. It will not be plotted.

```

```
## Warning in wordcloud(colnames(Papers_dtm_matrix), Papers_dtm_matrix[71:85,  
:   
## escap could not be fit on page. It will not be plotted.  
## Warning in wordcloud(colnames(Papers_dtm_matrix), Papers_dtm_matrix[71:85,  
:   
## impos could not be fit on page. It will not be plotted.  
## Warning in wordcloud(colnames(Papers_dtm_matrix), Papers_dtm_matrix[71:85,  
:   
## particip could not be fit on page. It will not be plotted.  
## Warning in wordcloud(colnames(Papers_dtm_matrix), Papers_dtm_matrix[71:85,  
:   
## afford could not be fit on page. It will not be plotted.  
## Warning in wordcloud(colnames(Papers_dtm_matrix), Papers_dtm_matrix[71:85,  
:   
## chargeabl could not be fit on page. It will not be plotted.  
## Warning in wordcloud(colnames(Papers_dtm_matrix), Papers_dtm_matrix[71:85,  
:   
## moral could not be fit on page. It will not be plotted.  
## Warning in wordcloud(colnames(Papers_dtm_matrix), Papers_dtm_matrix[71:85,  
:   
## pardon could not be fit on page. It will not be plotted.  
## Warning in wordcloud(colnames(Papers_dtm_matrix), Papers_dtm_matrix[71:85,  
:   
## continu could not be fit on page. It will not be plotted.  
## Warning in wordcloud(colnames(Papers_dtm_matrix), Papers_dtm_matrix[71:85,  
:   
## suffic could not be fit on page. It will not be plotted.
```




Analysis

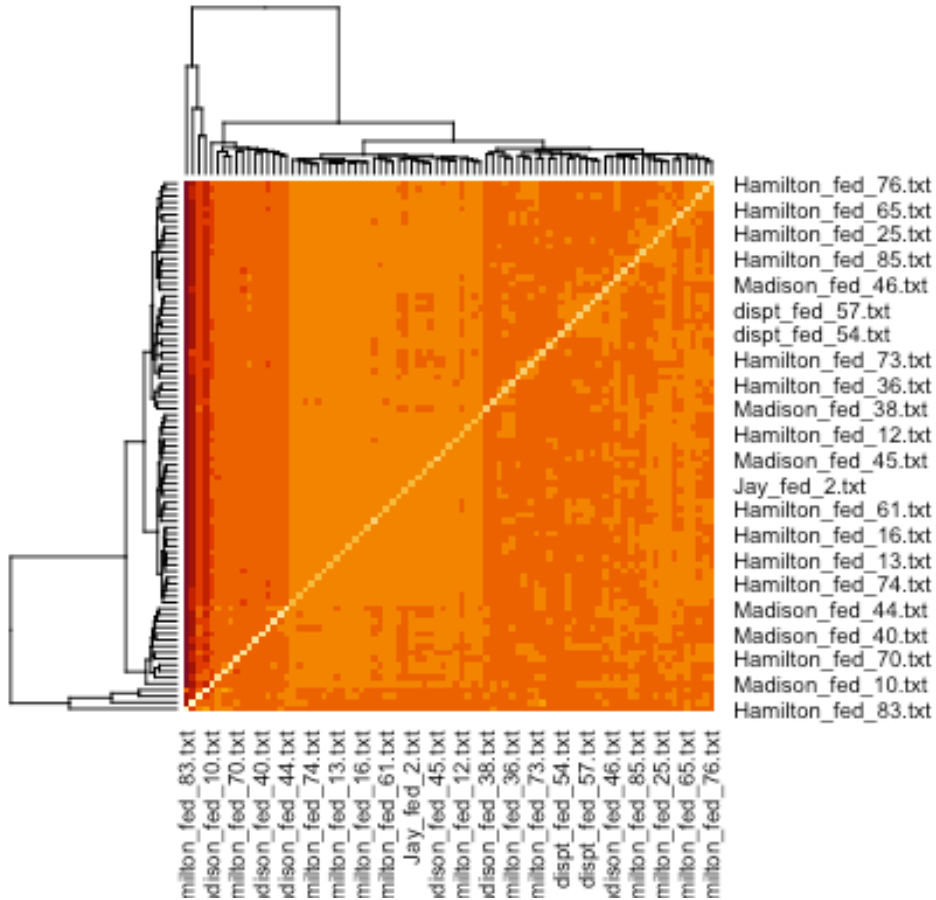
Distance metrics

```
m <- Papers_dtm_matrix
m_norm <- Papers_Matrix_Norm

distMatrix_E <- distance(m, method='euclidean', use.row.names = TRUE)

## Metric: 'euclidean'; comparing: 85 vectors.

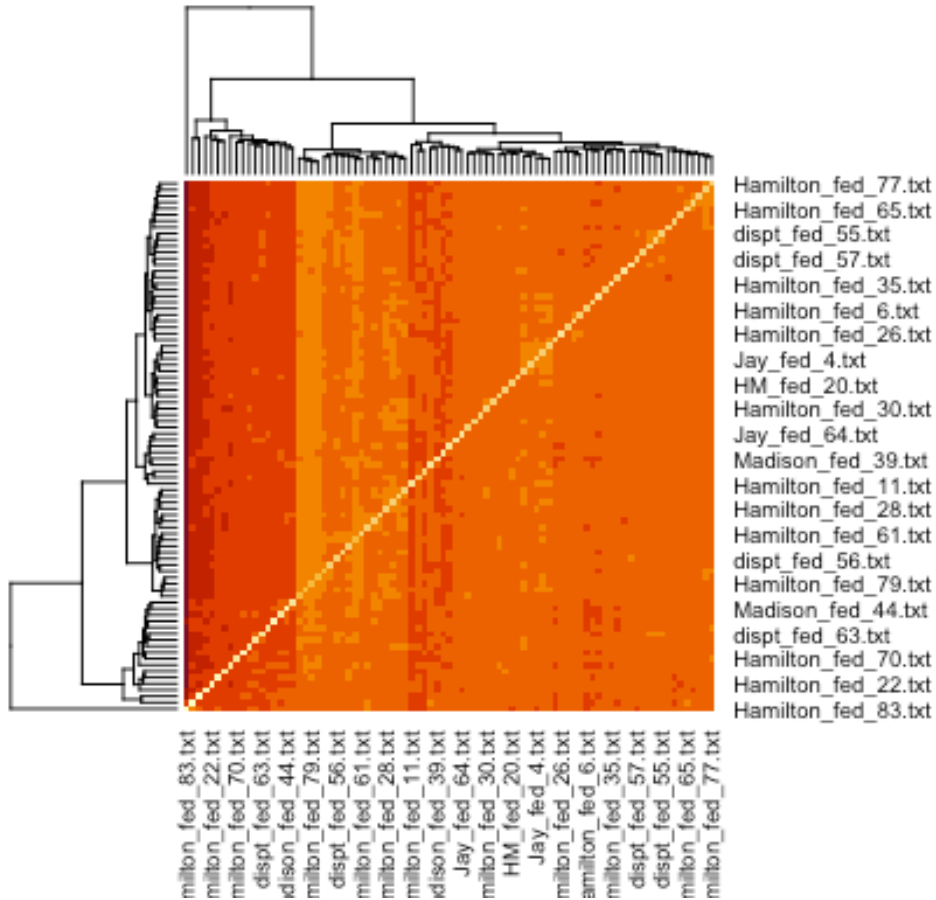
# print(distMatrix_E)
heatmap(distMatrix_E)
```



```
distMatrix_M <- distance(m, method='manhattan', use.row.names = TRUE)

## Metric: 'manhattan'; comparing: 85 vectors.

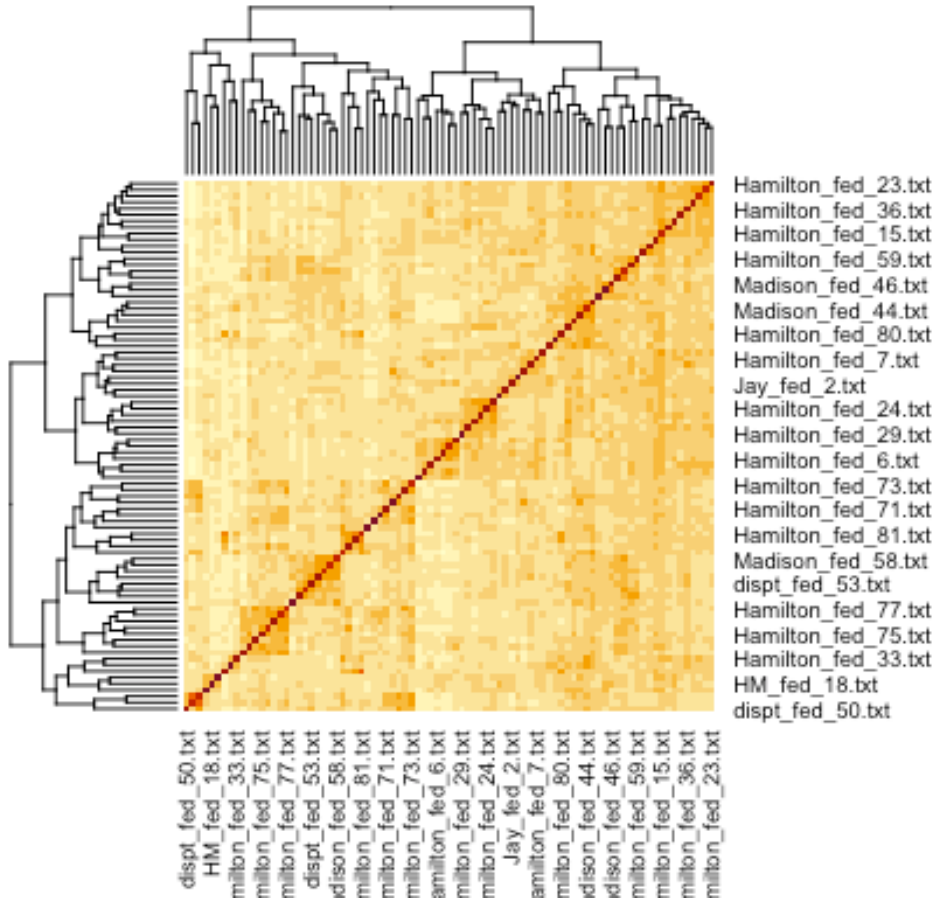
# print(distMatrix_M)
heatmap(distMatrix_M)
```



```
distMatrix_C <- distance(m, method = 'cosine', use.row.names = TRUE)

## Metric: 'cosine'; comparing: 85 vectors.

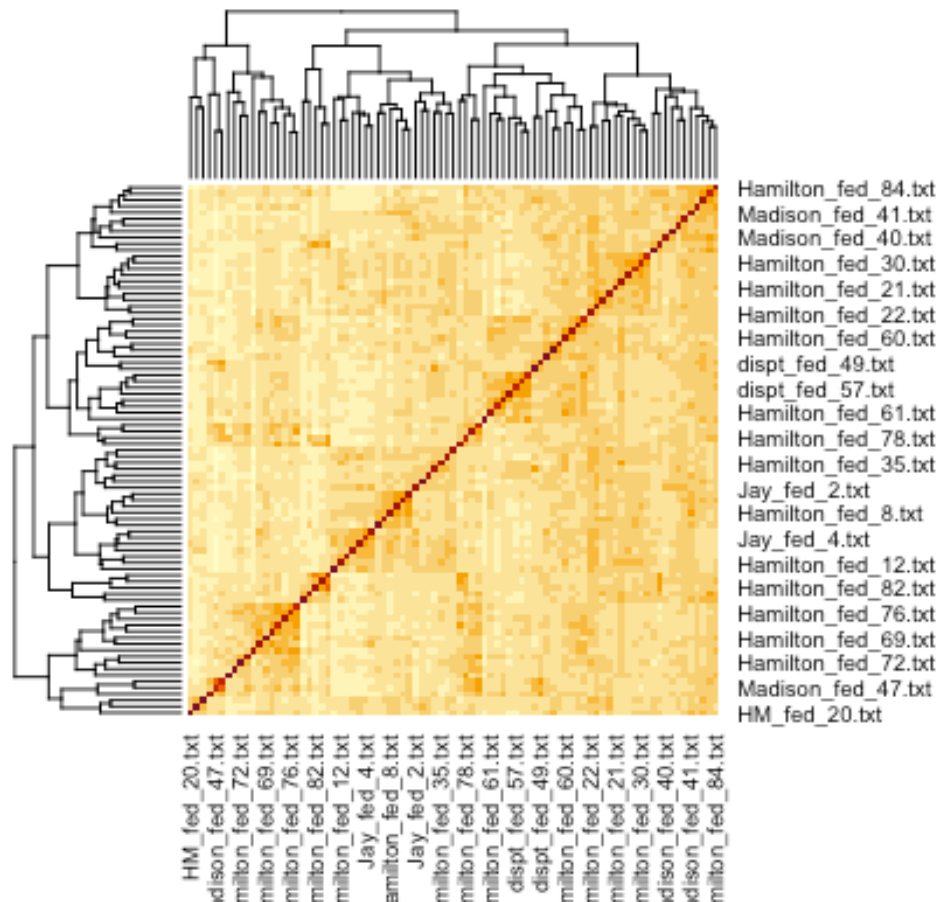
# print(distMatrix_C)
heatmap(distMatrix_C)
```



```
distMatrix_C_norm <- distance(m_norm, method='cosine', use.row.names = TRUE)

## Metric: 'cosine'; comparing: 85 vectors.

# print(distMatrix_C_norm)
heatmap(distMatrix_C_norm)
```



The `dist()` function has issues with 'cosine' methods. Instead, used `distance()` function and obtain cosine similarity visualization. Heat-maps prove cosine similarity measurements are likely more suitable for document analysis.

Data

We will explore the following two methods to cluster the data and determine an author to the disputed papers:

- K-means algorithm
- HAC algorithm

Given that the number of authors here are namely Hamilton and Madison, we will start with choosing number of clusters = 2.

First, is the k-means algorithm:

```
k <- 2
set.seed(5)
km.res <- kmeans(Papers_dtm_matrix, k, nstart=100, iter.max=50)
str(km.res)
```

```
## List of 9
## $ cluster      : Named int [1:85] 1 1 1 1 1 1 1 1 1 1 ...
## ..- attr(*, "names")= chr [1:85] "dispt_fed_49.txt" "dispt_fed_50.txt"
"dispt_fed_51.txt" "dispt_fed_52.txt" ...
## $ centers      : num [1:2, 1:3370] 0.1084 0 0.0241 0 0.0602 ...
## ..- attr(*, "dimnames")=List of 2
## .. ..$ : chr [1:2] "1" "2"
## .. ..$ : chr [1:3370] "abandon" "abat" "abb" "abet" ...
## $ totss       : num 202176
## $ withinss    : num [1:2] 174195 6448
## $ tot.withinss: num 180642
## $ betweenss   : num 21533
## $ size        : int [1:2] 83 2
## $ iter        : int 1
## $ ifault      : int 0
## - attr(*, "class")= chr "kmeans"
```

#plot a visualization

```
fviz_cluster(km.res, Papers_dtm_matrix)
```



```
k <- 7
km.res <- kmeans(Papers_Matrix_Norm, k, nstart=50, iter.max=50)
str(km.res)
```

```
## List of 9
## $ cluster      : Named int [1:85] 1 1 1 5 5 5 5 5 7 ...
## ..- attr(*, "names")= chr [1:85] "dispt_fed_49.txt" "dispt_fed_50.txt"
"dispt_fed_51.txt" "dispt_fed_52.txt" ...
## $ centers      : num [1:7, 1:3370] 7.69e-05 0.00 7.14e-05 0.00 0.00 ...
## ..- attr(*, "dimnames")=List of 2
## .. ..$ : chr [1:7] "1" "2" "3" "4" ...
## .. ..$ : chr [1:3370] "abandon" "abat" "abb" "abet" ...
## $ totss       : num 0.226
## $ withinss    : num [1:7] 0.03396 0.00231 0.02952 0.00754 0.02239 ...
## $ tot.withinss: num 0.174
## $ betweenss   : num 0.0514
## $ size        : int [1:7] 13 2 14 4 10 5 37
## $ iter        : int 4
## $ ifault      : int 0
## - attr(*, "class")= chr "kmeans"
```

#plot a visualization

```
fviz_cluster(km.res, Papers_Matrix_Norm)
```



Now, we explore the HAC algorithms

#Euclidean distance measure

```
dist.eul <- as.dist(distMatrix_E)
```

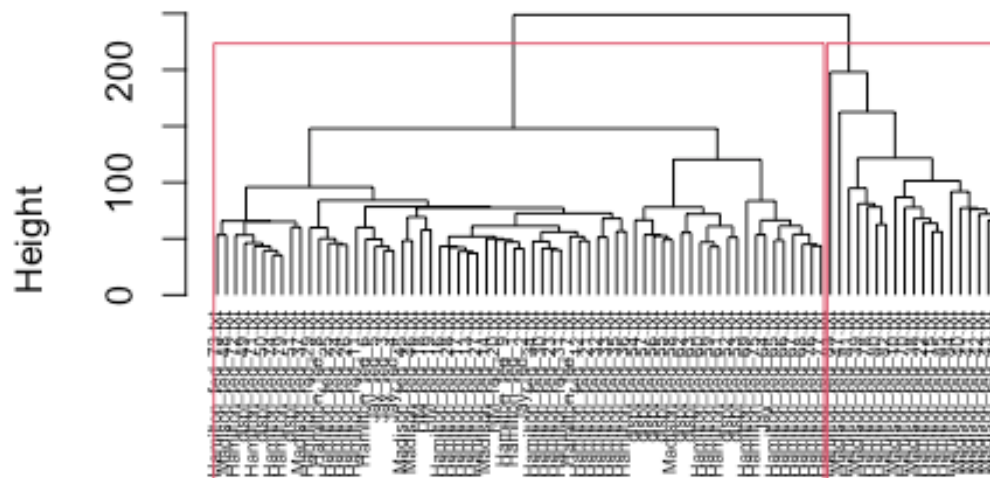
```
groups_E <- hclust(dist.eul, method='ward.D')
```

```
#Visualizations
```

```
plot(groups_E, cex=0.5, font=22, hang=-1, main="HAC cluster dendrogram with  
Euclidean Similarity")
```

```
rect.hclust(groups_E, k=2)
```

HAC cluster dendrogram with Euclidean Similarity



dist.eul
hclust (*, "ward.D")

```
#Cosine distance measure
```

```
dist.cos <- as.dist(distMatrix_C)
```

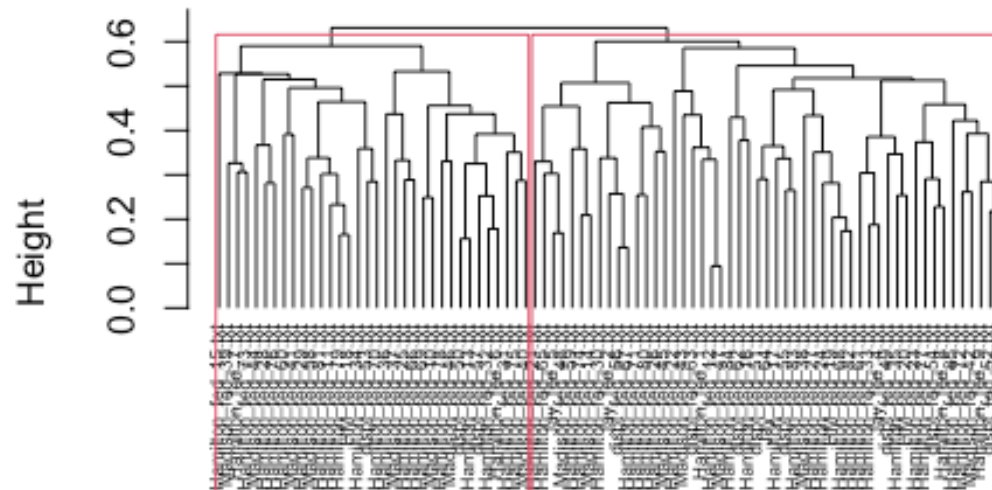
```
groups_C <- hclust(dist.cos, method='ward.D')
```

```
#Visualizations
```

```
plot(groups_C, cex=0.5, font=22, hang=-1, main="HAC cluster dendrogram with  
Cosine Similarity")
```

```
rect.hclust(groups_C, k=2)
```


HAC cluster dendrogram with Cosine Similarity

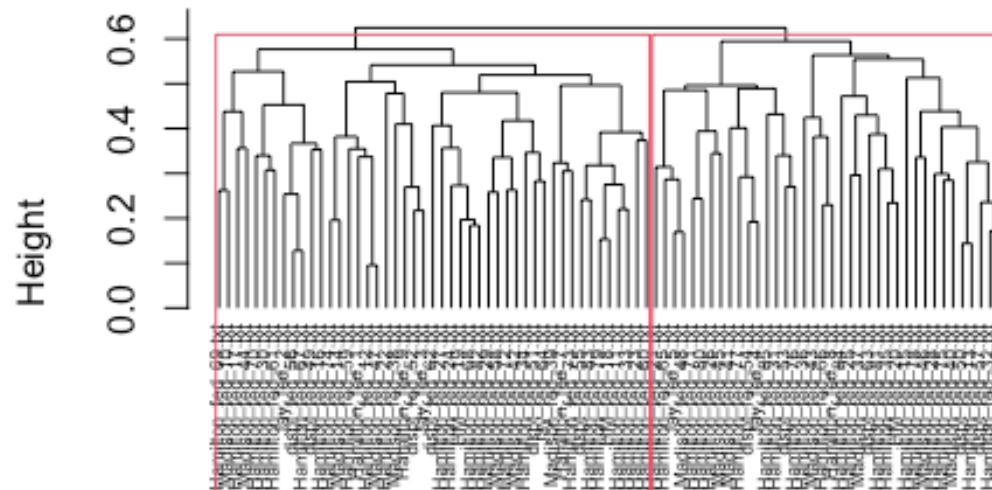


```
dist.cos
hclust (*, "ward.D")
```

```
#Cosine distance measure (Normalized)
dist.cosnorm <- as.dist(distMatrix_C_norm)
groups_C_norm <- hclust(dist.cosnorm, method='ward.D')

#Visualizations
plot(groups_C_norm, cex=0.5, font=22, hang=-1, main="HAC cluster dendrogram
with Cosine Similarity (Normalized)")
rect.hclust(groups_C_norm, k=2)
```

AC cluster dendrogram with Cosine Similarity (Normalized)



dist.cosnorm
hclust (*, "ward.D")

Analysis and Results

K-Means

Here are some results/observations with experiments around different cluster sizes:

- cluster-size=2

SSEs

Within cluster sum of squares by cluster is high:

[1] 174194.7 6447.5

This is an indication of high deviation between data-points and the centroid which we would ideally like to be lower. To explore k-means further, we could consider using the k-medoids/expectation-max or PAM algorithms.

Data

Most of the data-points were grouped into cluster-1 and this did not help to clearly determine the author for the disputed papers.

- cluster-size=7

SSE

SSEs look a lot better with increased cluster-size

Within cluster sum of squares by cluster:

```
[1] 0.00754175 0.03396400 0.06862076 0.00231200 0.02952307 0.00990520  
0.02239410
```

Data

Disputed papers were placed in clusters - 2, 7, 3:

- Number of disputed papers in cluster-2 = 3
- Number of disputed papers in cluster-7 = 7
- Number of disputed papers in cluster-3 = 1

Cluster-7 that has the highest papers does not have sufficient majority of Hamilton/Madison papers to make a decision.

Overall, k-means does not seem like a good algorithm for document analysis use-cases.

HAC algorithm

In comparison, seems like plotting and analyzing dendograms, seems a plausible means to realize the exercise. To a very large extent we can classify the disputed documents to the corresponding authors.

Conclusions

With Hierarchical Agglomerative Clustering (HAC) techniques (and dendograms to analyze the results) we conclude by analyzing one disputed document `dispt_fed_49.txt` across:

- Euclidian

In plot 'HAC cluster dendogram with Euclidean Similarity', see document 'dispt_fed_49.txt' present in the first-cluster on the left and is associated by nodes/leafs that belong to Hamilton so, we can conclude it was written by author Hamilton with moderate confidence.

- Cosine

In plot 'HAC cluster dendogram with Cosine Similarity', see document 'dispt_fed_49.txt' belonging to a cluster towards the end. Again, the nodes/leafs around it are documents by author Hamilton.

- Cosine-Normalized

Likewise, in plot 'HAC cluster dendogram with Cosine Similarity (Normalized)' the surrounding nodes/leafs are related to author Hamilton.

In similar lines, we could extend the study to all disputed documents and hence classify them between the two authors.

Classification using Decision Trees

This week we will explore classification using Decision Trees and use it as a means to classify the disputed papers.

First we will prepare a train, test data-sets and use 60% data for training.

```
# Make a copy of Papers_DF
Papers_DF1 <- Papers_DF %>% tibble::rownames_to_column()
names(Papers_DF1)[1] <- "Author"
Papers_DF1[1:11,1] <- "dispt"
Papers_DF1[12:62,1] <- "hamil"
Papers_DF1[63:85,1] <- "madis"
# head(Papers_DF1)

numDisputed <- 11
numTotalPapers <- nrow(Papers_DF1)

# Decide the train ratio
trainRatio <- .60

set.seed(11)
sample <- sample.int(n = numTotalPapers - numDisputed,
size=floor(trainRatio*numTotalPapers), replace=FALSE)
newsample <- sample + numDisputed

# Construct the train/test data-sets such that:
# train - Hamilton/Madison papers
# test - will have disputed and few Hamilton/Madison papers
train <- Papers_DF1[newsample,]
test <- Papers_DF1[-newsample,]

length(newsample)/nrow(Papers_DF1)

## [1] 0.6
```

Classification

rpart, a recursive partitioning and regression test package in R will be used to make predictions through classification. Further, rpart uses the CART algorithm, a non-greedy algorithm and is known to be efficient.

DT-Model1

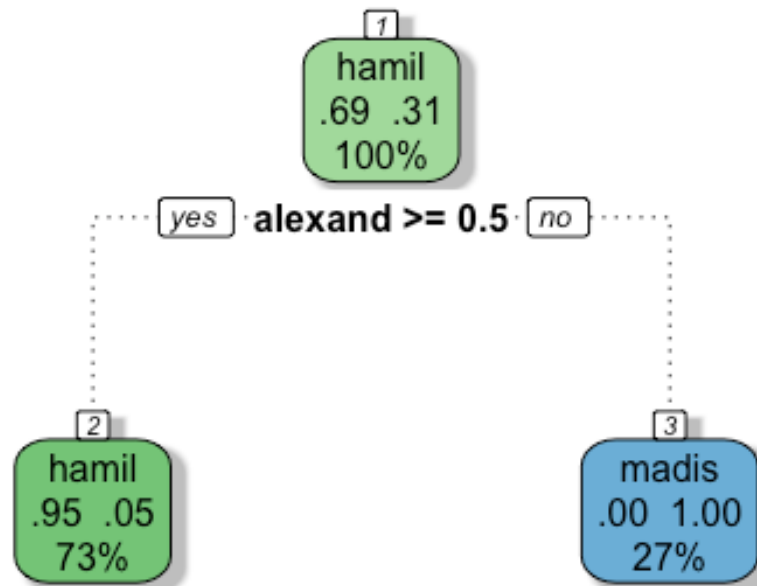
```
train_tree1 <- rpart(Author ~ ., data = train, method='class',
control=rpart.control(cp=0))
summary(train_tree1)
```

```

## Call:
## rpart(formula = Author ~ ., data = train, method = "class", control =
rpart.control(cp = 0))
##    n= 51
##
##      CP nsplit rel error xerror      xstd
## 1 0.875      0      1.000  1.000 0.2071042
## 2 0.000      1      0.125  0.375 0.1438059
##
## Variable importance
## alexand hamilton      jame  madison      upon although
##      22      22      15      15      15      11
##
## Node number 1: 51 observations,      complexity param=0.875
##   predicted class=hamil expected loss=0.3137255 P(node) =1
##   class counts:      35      16
##   probabilities: 0.686 0.314
##   left son=2 (37 obs) right son=3 (14 obs)
##   Primary splits:
##     alexand < 0.5 to the right, improve=18.177, (0 missing)
##     hamilton < 0.5 to the right, improve=18.177, (0 missing)
##     jame < 0.5 to the left, improve=18.177, (0 missing)
##     madison < 0.5 to the left, improve=18.177, (0 missing)
##     upon < 1.5 to the right, improve=18.177, (0 missing)
##   Surrogate splits:
##     hamilton < 0.5 to the right, agree=1.000, adj=1.000, (0 split)
##     jame < 0.5 to the left, agree=0.922, adj=0.714, (0 split)
##     madison < 0.5 to the left, agree=0.922, adj=0.714, (0 split)
##     upon < 0.5 to the right, agree=0.922, adj=0.714, (0 split)
##     although < 0.5 to the left, agree=0.863, adj=0.500, (0 split)
##
## Node number 2: 37 observations
##   predicted class=hamil expected loss=0.05405405 P(node) =0.7254902
##   class counts:      35      2
##   probabilities: 0.946 0.054
##
## Node number 3: 14 observations
##   predicted class=madis expected loss=0 P(node) =0.2745098
##   class counts:      0      14
##   probabilities: 0.000 1.000

# Plot rpart using fancyRpartPlot
fancyRpartPlot(train_tree1)

```



Rattle 2020-Aug-16 13:53:52 venkatasharatsripada

```

# Get confusion matrix
predicted1 <- predict(train_tree1, test, type='class')
table(Authorship=predicted1, true=test$Author)

##           true
## Authorship dispt hamil madis
##      hamil     11    16     1
##      madis      0     0     6
  
```

The confusion matrix shows the disputed papers may have belonged to Hamilton with sufficient confidence since a high percentage of predictions are accurate. Following couple of observations: - It wrongly predicted, 2 papers as Madison where the Author was Hamilton - It accurately predicted all Madison papers

DT-Model2

Notice with Model-1 that the tree was not allowed to grow and limited to a root and two immediate leaf nodes. In the second model, we will attempt to attempt to override the defaults and grow the tree namely using the following: - minsplit - minimum number of observations that must exist in a node - maxdepth - maximum depth the tree can grow

Hopefully, this will also lead to better accuracy.

```

train_tree2 <- rpart(Author ~ ., data = train, method='class',
control=rpart.control(cp=0, minsplit = 2, maxdepth=5))
summary(train_tree2)

## Call:
## rpart(formula = Author ~ ., data = train, method = "class", control =
rpart.control(cp = 0,
##      minsplit = 2, maxdepth = 5))
##      n= 51
##
##      CP nsplit rel error xerror      xstd
## 1 0.875      0    1.000 1.0000 0.2071042
## 2 0.125      1    0.125 0.2500 0.1199980
## 3 0.000      2    0.000 0.3125 0.1327269
##
## Variable importance
## alexand hamilton      jame      madison      upon      although
catastroph
##      17      17      16      16      16      9
4
##      templ      abb
##      4      2
##
## Node number 1: 51 observations,      complexity param=0.875
## predicted class=hamil expected loss=0.3137255 P(node) =1
## class counts:      35      16
## probabilities: 0.686 0.314
## left son=2 (37 obs) right son=3 (14 obs)
## Primary splits:
## alexand < 0.5 to the right, improve=18.177, (0 missing)
## hamilton < 0.5 to the right, improve=18.177, (0 missing)
## jame < 0.5 to the left, improve=18.177, (0 missing)
## madison < 0.5 to the left, improve=18.177, (0 missing)
## upon < 1.5 to the right, improve=18.177, (0 missing)
## Surrogate splits:
## hamilton < 0.5 to the right, agree=1.000, adj=1.000, (0 split)
## jame < 0.5 to the left, agree=0.922, adj=0.714, (0 split)
## madison < 0.5 to the left, agree=0.922, adj=0.714, (0 split)
## upon < 0.5 to the right, agree=0.922, adj=0.714, (0 split)
## although < 0.5 to the left, agree=0.863, adj=0.500, (0 split)
##
## Node number 2: 37 observations,      complexity param=0.125
## predicted class=hamil expected loss=0.05405405 P(node) =0.7254902
## class counts:      35      2
## probabilities: 0.946 0.054
## left son=4 (35 obs) right son=5 (2 obs)
## Primary splits:
## catastroph < 0.5 to the left, improve=3.783784, (0 missing)
## jame < 0.5 to the left, improve=3.783784, (0 missing)
## madison < 0.5 to the left, improve=3.783784, (0 missing)

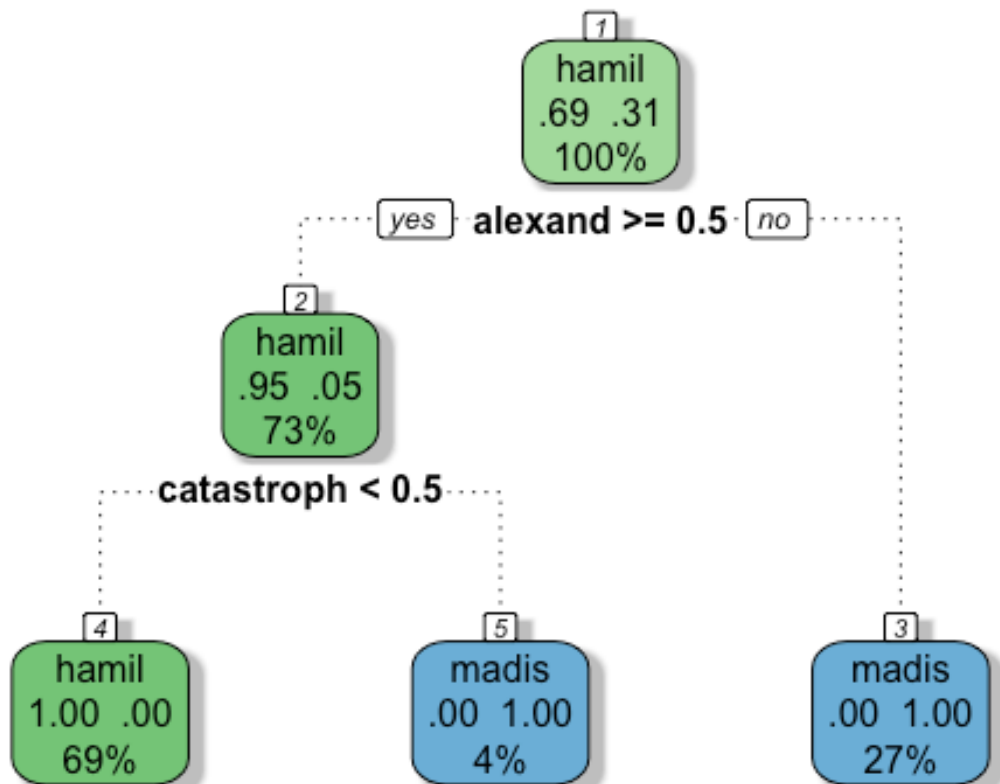
```

```

##      templ      < 1   to the left,  improve=3.783784, (0 missing)
##      upon       < 1.5 to the right, improve=3.783784, (0 missing)
##  Surrogate splits:
##      jame      < 0.5 to the left,  agree=1.000, adj=1.0, (0 split)
##      madison   < 0.5 to the left,  agree=1.000, adj=1.0, (0 split)
##      templ     < 1   to the left,  agree=1.000, adj=1.0, (0 split)
##      upon      < 1.5 to the right, agree=1.000, adj=1.0, (0 split)
##      abb       < 0.5 to the left,  agree=0.973, adj=0.5, (0 split)
##
## Node number 3: 14 observations
##   predicted class=madis  expected loss=0  P(node) =0.2745098
##   class counts:      0      14
##   probabilities: 0.000 1.000
##
## Node number 4: 35 observations
##   predicted class=hamil  expected loss=0  P(node) =0.6862745
##   class counts:      35      0
##   probabilities: 1.000 0.000
##
## Node number 5: 2 observations
##   predicted class=madis  expected loss=0  P(node) =0.03921569
##   class counts:      0      2
##   probabilities: 0.000 1.000

# Plot rpart using fancyRpartPlot
fancyRpartPlot(train_tree2)

```

Rattle 2020-Aug-16 13:53:57 venkatasharatsripada

```

# Get confusion matrix
predicted2 <- predict(train_tree2, test, type='class')
table(Authorship=predicted2, true=test$Author)

##           true
## Authorship dispt hamil madis
##      hamil    11    16     1
##      madis     0     0     6

```

The accuracy improved a tad bit with this model which further gives us confidence about the disputed papers belonging to Hamilton.

DT-Model3

Finally, let's explore this via Random Forests. Random Forests improve predictive accuracy by generating a large number of bootstrapped trees (based on random sample of variables), classifying a case using each tree in this new 'forest', and deciding a final predicted outcome by combining the results across all of the trees. This is also commonly termed an Ensemble Method.

```

train$Author <- as.factor(train$Author)
train_tree3 <- randomForest(x=train[,2:ncol(train)], y=train$Author, data =
train, ntree=100, mtry=2, importance=TRUE)

```

```
predicted3 <- predict(train_tree3, test)
table(Authorship=predicted3, true=test$Author)

##           true
## Authorship dispt hamil madis
##      hamil    11    16     7
##      madis     0     0     0
```

Conclusion

The prediction using Decision Trees points to the disputed papers belonging to Hamilton. As pointed out in each of the sections above, we have sufficient confidence, given the accuracy of prediction in the confusion-matrix. Also, the ensemble method via Random Forests provides similar results.

An important characteristic of Decision Trees on such data-sets/problems is that, while building the trees and the underlying computations can be complex, the algorithm is fairly quick when making subsequent decisions.