IST707\_HW4

Jared Mosley

7/23/2020

# Mystery in History: Cluster Analysis & The Federalist Papers

## Introduction

The Federalist Papers consists of 85 essays penned by Alexander Hamilton, James Madison, and John Jay. They were authored under the pseudonym “Publius” in varius journals and news publications in 1787 and 1788. They were published in a move to encourage the ratification of the United States Constitution. Interestingly enough, in 1804 only after Hamilton died did the authors true names emerge.

Once the names were released, there were several papers whose authorship has been disputed to this day. While Hamilton claimed to have authored the majority of the papers, recent advances in science have shown that that some of the papers do not match the style and tone of the claimed authors.

This report will perform an analysis on the author’s styles in attempt to cluster the disputed articles among papers with the most similar styles. In the end, we hope to match the styles in an attempt to identify the true author.

<https://en.wikipedia.org/wiki/The_Federalist_Papers>

# Data Preparation

## About the Data

Tell me about the dataset

## Data Structure

What does it contain?

### Read in Libraries

library(SnowballC)  
library(tm)  
library(wordcloud)  
library(cluster)  
library(proxy)  
library(factoextra)

### Load in the Corpus

Files provided from <https://github.com/boltonvandy/IST707repo/tree/master/FedPapersCorpus>.

Load in the corpus and check to see it was performed properly.

# Read in corpus from disk  
FPC <- Corpus(DirSource("C:/Users/jared/iCloudDrive/Documents/Coding/R/Homework/FedPapersCorpus"))  
  
# Check to ensure files read in properly  
summary(FPC)

## 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(FPC[[1]])

## author : character(0)  
## datetimestamp: 2020-07-26 18:27:37  
## description : character(0)  
## heading : character(0)  
## id : dispt\_fed\_49.txt  
## language : en  
## origin : character(0)

meta(FPC[[1]],5)

## [1] "dispt\_fed\_49.txt"

### Initial Cleaning

Begin exploring the data and using the DocumentTermMatrix, vectorize it. Eliminate overly common and extremely rare words. Then, examine the counts and frequencies of terms.

# Remove Punctuatuion, numbers, spaces  
getTransformations()

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

# Set length and ignore words with less than 1% use and appearing in more than 50% of the documents.  
nFPC <- length(FPC)  
minTermFreq <- nFPC\*.001  
maxTermFreq <- nFPC\*1  
  
# Load in stop words  
STOPS <- stopwords('english')  
  
# Run DTM with stopwords, set lengths, removing punctuation and numbers, setting to lowercase, using stemming, removing separators, and setting frequency boundaries.  
Papers\_DTM <- DocumentTermMatrix(FPC,  
 control = list(  
 stopwords = TRUE,  
 stopwords = STOPS,  
 wordLengths = c(3,15),  
 removePunctuation = TRUE,  
 removeNumbers = TRUE,  
 tolower = TRUE,  
 stemming = TRUE,  
 remove\_separators = TRUE,  
 bounds = list(global=c(minTermFreq,maxTermFreq))  
 ))  
DTM <- as.matrix(Papers\_DTM) # Set as matrix  
  
DTM[1:11,1:10] # Inspect

## Terms  
## Docs abandon abat abb abet abhorr abil abject abl ablest abolish  
## dispt\_fed\_49.txt 0 0 0 0 0 0 0 2 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 0 0 1 0 0  
## dispt\_fed\_52.txt 0 0 0 0 0 1 0 1 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 1 0 0 0 0 0  
## dispt\_fed\_62.txt 0 0 0 0 0 0 0 1 0 0  
## dispt\_fed\_63.txt 0 0 0 0 0 0 0 4 0 0

# Count column sumns as word frequencies  
WordFreq <- colSums(DTM)  
length(WordFreq) # Check total counts

## [1] 4900

orderedWordFreq <- order(WordFreq) # Order frequencies  
WordFreq[head(orderedWordFreq)] # Inspect lowest 5

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

WordFreq[tail(orderedWordFreq)] # Inspect top 5

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

RowSumPerDoc <- rowSums(DTM) # Count matching terms per document  
RowSumPerDoc # Inspect

## dispt\_fed\_49.txt dispt\_fed\_50.txt dispt\_fed\_51.txt dispt\_fed\_52.txt   
## 758 530 923 853   
## dispt\_fed\_53.txt dispt\_fed\_54.txt dispt\_fed\_55.txt dispt\_fed\_56.txt   
## 1035 882 968 765   
## dispt\_fed\_57.txt dispt\_fed\_62.txt dispt\_fed\_63.txt Hamilton\_fed\_1.txt   
## 1023 1124 1432 767   
## Hamilton\_fed\_11.txt Hamilton\_fed\_12.txt Hamilton\_fed\_13.txt Hamilton\_fed\_15.txt   
## 1164 1044 479 1411   
## Hamilton\_fed\_16.txt Hamilton\_fed\_17.txt Hamilton\_fed\_21.txt Hamilton\_fed\_22.txt   
## 918 767 937 1692   
## Hamilton\_fed\_23.txt Hamilton\_fed\_24.txt Hamilton\_fed\_25.txt Hamilton\_fed\_26.txt   
## 828 925 927 1093   
## Hamilton\_fed\_27.txt Hamilton\_fed\_28.txt Hamilton\_fed\_29.txt Hamilton\_fed\_30.txt   
## 690 755 1010 948   
## Hamilton\_fed\_31.txt Hamilton\_fed\_32.txt Hamilton\_fed\_33.txt Hamilton\_fed\_34.txt   
## 797 686 773 1020   
## Hamilton\_fed\_35.txt Hamilton\_fed\_36.txt Hamilton\_fed\_59.txt Hamilton\_fed\_6.txt   
## 1052 1272 860 984   
## Hamilton\_fed\_60.txt Hamilton\_fed\_61.txt Hamilton\_fed\_65.txt Hamilton\_fed\_66.txt   
## 1006 681 912 997   
## Hamilton\_fed\_67.txt Hamilton\_fed\_68.txt Hamilton\_fed\_69.txt Hamilton\_fed\_7.txt   
## 781 683 1359 1073   
## Hamilton\_fed\_70.txt Hamilton\_fed\_71.txt Hamilton\_fed\_72.txt Hamilton\_fed\_73.txt   
## 1436 766 925 1061   
## Hamilton\_fed\_74.txt Hamilton\_fed\_75.txt Hamilton\_fed\_76.txt Hamilton\_fed\_77.txt   
## 478 905 883 887   
## Hamilton\_fed\_78.txt Hamilton\_fed\_79.txt Hamilton\_fed\_8.txt Hamilton\_fed\_80.txt   
## 1376 478 998 1132   
## Hamilton\_fed\_81.txt Hamilton\_fed\_82.txt Hamilton\_fed\_83.txt Hamilton\_fed\_84.txt   
## 1798 749 2620 1907   
## Hamilton\_fed\_85.txt Hamilton\_fed\_9.txt HM\_fed\_18.txt HM\_fed\_19.txt   
## 1264 931 1029 1023   
## HM\_fed\_20.txt Jay\_fed\_2.txt Jay\_fed\_3.txt Jay\_fed\_4.txt   
## 776 804 736 780   
## Jay\_fed\_5.txt Jay\_fed\_64.txt Madison\_fed\_10.txt Madison\_fed\_14.txt   
## 657 1072 1437 1016   
## Madison\_fed\_37.txt Madison\_fed\_38.txt Madison\_fed\_39.txt Madison\_fed\_40.txt   
## 1268 1529 1169 1340   
## Madison\_fed\_41.txt Madison\_fed\_42.txt Madison\_fed\_43.txt Madison\_fed\_44.txt   
## 1701 1330 1601 1382   
## Madison\_fed\_45.txt Madison\_fed\_46.txt Madison\_fed\_47.txt Madison\_fed\_48.txt   
## 1018 1233 1306 846   
## Madison\_fed\_58.txt   
## 978

# Analysis & Models

### Distance Metrics

We will be computing the distances based off of various methods such as Euclidean, Manhattan, Cosine, and Cosine with a normalized distribution to run our models.

The Euclidean distance measures “as the crow flies,” or rather the shortest distance between two points. The Manhattan distance measures in “blocks” or rather the absolute values between two points. The Cosine measurement works best in “high-dimensional” space.. It works by finding the angular distance between points. Lastly, the Cosine Normalized is exactly what it sounds like, a normalized approach to cosine simlarity, but for our data size it should not provide much variability, though we will still examine it.

# Normalize the DTM  
NormDTM <- t(apply(DTM,1,function(i) round (i/sum(i),3)))  
# Examine for accuracy  
NormDTM[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

EuclideanDist <- dist(DTM, method='euclidean')  
#print(EuclideanDist)  
  
ManhattanDist <- dist(DTM, method='manhattan')  
#print(ManhattanDist)  
  
CosineDist <- dist(DTM, method='cosine')  
#print(CosineDist)  
  
CosineDistNorm <- dist(NormDTM, method='cosine')  
#print(CosineDistNorm)

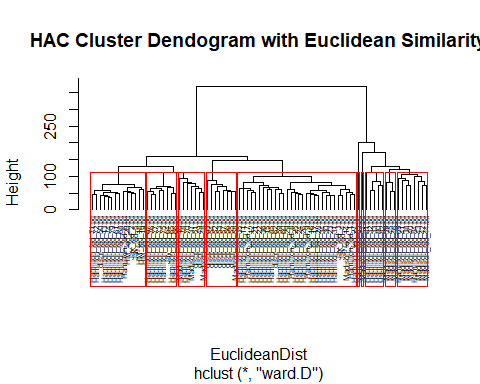
## Clustering Methods

Two types of clustering methods will be performed. First we will evaluate the Heirarchical Algorithm Clustering Method, and later we will evaluate the K-Means Algorithm Clustering Method.

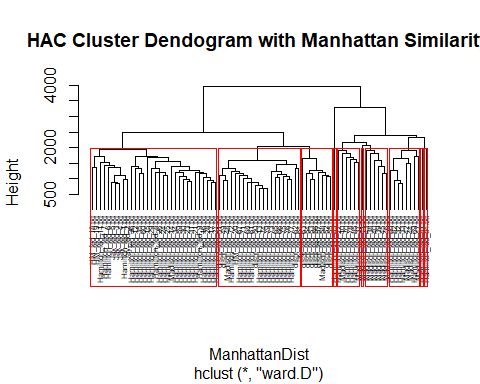
### Heirarchical Algorithm Clustering

The Heirarchical Algorithm Clustering Method measures the distance by starting with initial points as clusters and measuring the distances between other points, merging the clusters until only one remains. We will measure the distance in several ways.

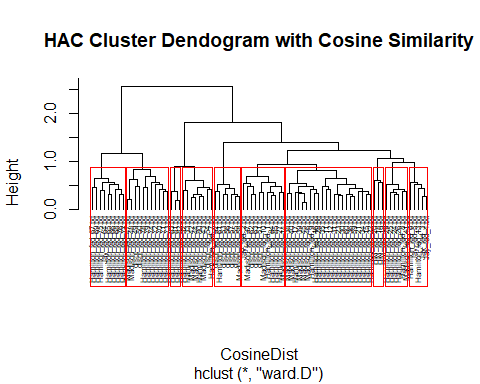
# HAC: Heirarchical Algorithm Clustering Method  
# Euclidean  
group\_E <- hclust(EuclideanDist, method="ward.D")  
plot(group\_E, cex=0.5, font=22, hang=-1, main= "HAC Cluster Dendogram with Euclidean Similarity")  
rect.hclust(group\_E, k=10)



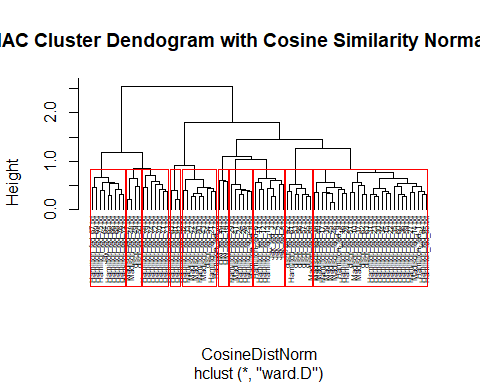
# Manhattan  
group\_M <- hclust(ManhattanDist, method="ward.D")  
plot(group\_M, cex=0.5, font=22, hang=-1, main= "HAC Cluster Dendogram with Manhattan Similarity")  
rect.hclust(group\_M, k=10)



# Cosine  
group\_C <- hclust(CosineDist, method="ward.D")  
plot(group\_C, cex=0.5, font=22, hang=-1, main= "HAC Cluster Dendogram with Cosine Similarity")  
rect.hclust(group\_C, k=10)



# Cosine Normalized  
group\_N <- hclust(CosineDistNorm, method="ward.D")  
plot(group\_N, cex=0.5, font=22, hang=-1, main= "HAC Cluster Dendogram with Cosine Similarity Normalized")  
rect.hclust(group\_N, k=10)



The dendrographs created isolate best in my opinion at 10 clusters. Anything less and you cannot accurately place the locations of the disputed papers.

The Euclidean similarity performed adequately, creating one large cluster, and some clusters being far too small. It did however place many of the disputed papers with Madison, though only matching to one of his papers, and generally created clusters that contained little crossover between authors, though they did contain inaccurate authorship. It did suggest a large portion (Papers 63, 54, 52, 53, 57, 56, and 55) of the disputed papers were Madsion’s, with a couple given to Hamilton (Papers 50, 62), and some unclear.

The Manhattan similarity performed better, with two large clusters that are primarily Hamilton, though with Jay, Madison and the disputed papers sprinkled in. It appears to cluster the majority of the papers with Madison (Papers 63, 54, 53, 56, 57, 52, and 55), and the remaining with Hamilton (Papers 51, 49, 50, and 62). The clustering appears more precise for the disputed papers.

The Cosine similarity performed rather indecisively, with decent cluster sizes, though with much crossover in authorship. Additionally the normalized Cosine similarity performed rather poorly, with one large mixed cluster and many other clusters that are not perfectly matching.

It appears that the Manhattan actually performed the best when evaluating the Heirarchical Algorithm Clustering Method.

### K-Means Clustering

The K-Means Clustering works by finding the mean of a group of central points, called the centroid, and begins by partitioning the data objects in relation to the distance from that mean.

# K-Means Algorithm Clustering Method  
k1 <- kmeans(DTM, centers=10, nstart=100, iter.max=50)  
str(k1)

## List of 9  
## $ cluster : Named int [1:85] 7 7 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 [1:10, 1:4900] 0 0 0 0.429 0 ...  
## ..- attr(\*, "dimnames")=List of 2  
## .. ..$ : chr [1:10] "1" "2" "3" "4" ...  
## .. ..$ : chr [1:4900] "abandon" "abat" "abb" "abet" ...  
## $ totss : num 257061  
## $ withinss : num [1:10] 28066 3298 9404 16985 0 ...  
## $ tot.withinss: num 155365  
## $ betweenss : num 101695  
## $ size : int [1:10] 13 2 4 7 1 1 40 1 13 3  
## $ iter : int 3  
## $ ifault : int 0  
## - attr(\*, "class")= chr "kmeans"

summary(k1)

## Length Class Mode   
## cluster 85 -none- numeric  
## centers 49000 -none- numeric  
## totss 1 -none- numeric  
## withinss 10 -none- numeric  
## tot.withinss 1 -none- numeric  
## betweenss 1 -none- numeric  
## size 10 -none- numeric  
## iter 1 -none- numeric  
## ifault 1 -none- numeric

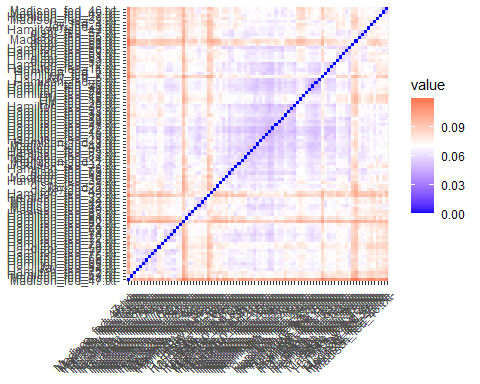
k2 <- kmeans(DTM, centers=10, nstart=50, iter.max=50)  
str(k2)

## List of 9  
## $ cluster : Named int [1:85] 10 10 8 3 3 3 3 3 3 3 ...  
## ..- attr(\*, "names")= chr [1:85] "dispt\_fed\_49.txt" "dispt\_fed\_50.txt" "dispt\_fed\_51.txt" "dispt\_fed\_52.txt" ...  
## $ centers : num [1:10, 1:4900] 0.167 0 0 0 0 ...  
## ..- attr(\*, "dimnames")=List of 2  
## .. ..$ : chr [1:10] "1" "2" "3" "4" ...  
## .. ..$ : chr [1:4900] "abandon" "abat" "abb" "abet" ...  
## $ totss : num 257061  
## $ withinss : num [1:10] 16487 6752 22040 0 0 ...  
## $ tot.withinss: num 155414  
## $ betweenss : num 101647  
## $ size : int [1:10] 6 3 10 1 1 2 3 31 13 15  
## $ iter : int 4  
## $ ifault : int 0  
## - attr(\*, "class")= chr "kmeans"

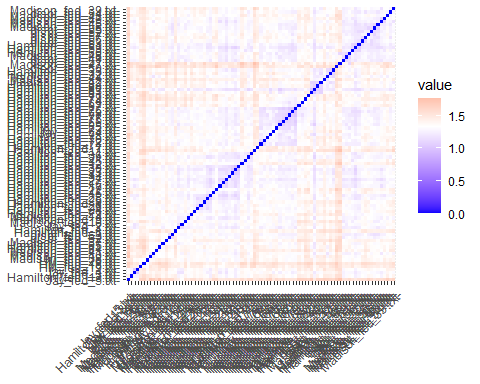
summary(k2)

## Length Class Mode   
## cluster 85 -none- numeric  
## centers 49000 -none- numeric  
## totss 1 -none- numeric  
## withinss 10 -none- numeric  
## tot.withinss 1 -none- numeric  
## betweenss 1 -none- numeric  
## size 10 -none- numeric  
## iter 1 -none- numeric  
## ifault 1 -none- numeric

# Visualize the k means   
#Euclidean  
dist1 <- get\_dist(NormDTM, method= "euclidean")  
fviz\_dist(dist1, gradient=list(low="blue", mid="white", high="Red"))



#Manhattan  
dist2 <- get\_dist(NormDTM, method= "manhattan")  
fviz\_dist(dist2, gradient=list(low="blue", mid="white", high="red"))



When visualizing the K-Means results, it immediately stands out that the Euclidean method was much more decisive in determining similarity. The Blue values cluster in the middle, which align with primarily Hamilton’s papers. It picks out Jay and Madison in the Red, but isn’t intuitively selecting them accurately. The Manhattan method has much less dicisiveness, and majority of the values return closer to the middle. It does cluster Hanmilton a little more accurately around the middle, and Madison towards the end, though the disputed papers are not intuitive.

## Initial Results

Initial results are a bit unclear, but it appears the Heirarchical approach is more intuitive and perhaps more accurate. The Heirarchical approach clusters the majority of the disputed papers to Madison, with only a couple identified as being Hamilton’s. This goes against Hamilton’s claims that he authored the majority of the papers.

# Analysis Using “Top 10” Words

In an attempt to further clarify the results, we will perform the same analysis as above, though narrowing our scope of words from 85 to only the top 10, as decided by use rates between the authors.

### Word Clouds for Common Word Usage Visualization

Create and visualize a wordcloud for Hamilton, Madison, and the disputed authorship papers, excluding those by John Jay. We can see that the word usage by the Hamilton set of papers lacks as high of a frequency as the word usage in both the Madison and Disputed sets of papers.

# Word cloud viz for Disputed Papers  
DisputedPapersWC <- wordcloud(colnames(DTM), DTM[11,])

## Warning in wordcloud(colnames(DTM), DTM[11, ]): elect could not be fit on page.  
## It will not be plotted.

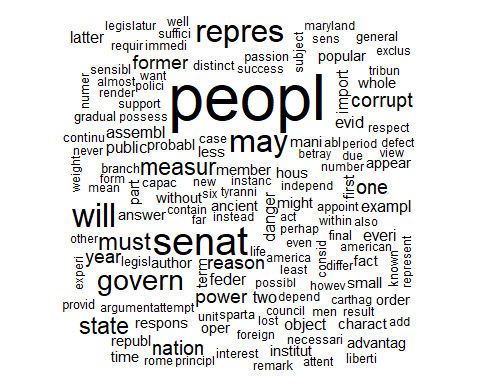
## Warning in wordcloud(colnames(DTM), DTM[11, ]): annual could not be fit on page.  
## It will not be plotted.

## Warning in wordcloud(colnames(DTM), DTM[11, ]): constitut could not be fit on  
## page. It will not be plotted.

## Warning in wordcloud(colnames(DTM), DTM[11, ]): can could not be fit on page. It  
## will not be plotted.

## Warning in wordcloud(colnames(DTM), DTM[11, ]): bodi could not be fit on page.  
## It will not be plotted.

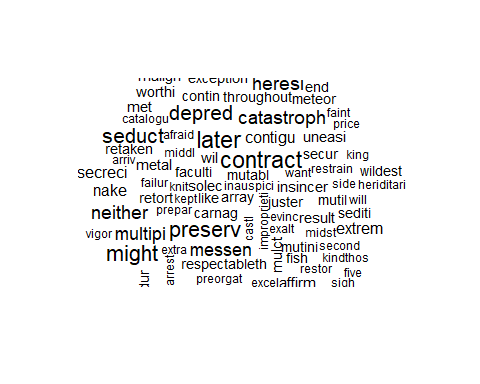
## Warning in wordcloud(colnames(DTM), DTM[11, ]): particular could not be fit on  
## page. It will not be plotted.



# Word cloud viz for Hamilton Papers  
HamiltonPapersWC <- wordcloud(colnames(DTM), DTM[12:62,])



# Word cloud viz for Madison Papers  
MadisonPapersWC <- wordcloud(colnames(DTM), DTM[63:77,])



### Setup the Data

Upon seeing the wordclouds, it might be easier and more intuitive to identify the top words based on use rate comparison between authors. We will choose a set of 10 words that have the highest usage, and minimal stemming. These 10 words will serve as a guide to performing a second set of cluster analysis in hopes of further identifying the unknown authors.

# Identify Top words for Hamilton  
HamiltonWords <- head(sort(colSums(DTM[12:62,]), decreasing=T), n=100)  
  
# Identify Top words for Madison  
MadisonWords <- head(sort(colSums(DTM[63:77,]), decreasing=T), n=100)  
# Put in a DF for comparison  
worddf <- data.frame(HamiltonWords, MadisonWords)  
# Compare use rate averages  
worddf$HamiltonDif <- (worddf$HamiltonWords/sum(worddf$HamiltonWords))/(worddf$MadisonWords/sum(worddf$MadisonWords))  
# Reclustering based off of "top" 10 words  
wordstop10 <- c("interest", "time", "shall", "general", "part", "must",   
 "great", "particula", "upon", "might")  
  
DTM10 <- DocumentTermMatrix(FPC,  
 control = list(  
 dictionary = wordstop10,  
 removePunctuation = TRUE,  
 removeNumbers = TRUE,  
 tolower = TRUE,  
 stemming = TRUE,  
 remove\_separators = TRUE,  
 bounds = list(global=c(minTermFreq,maxTermFreq))  
 ))  
DTM10 <- as.matrix(DTM10)  
# Normalize the DTM  
NormDTM10 <- t(apply(DTM10,1,function(i) round (i/sum(i),3)))

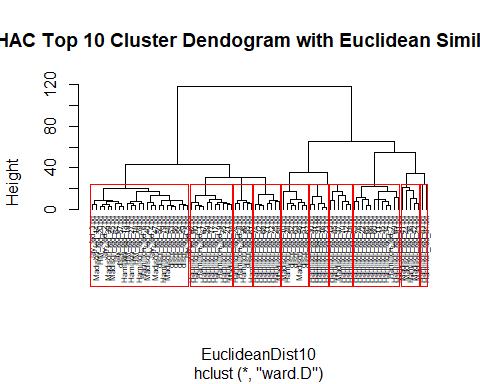
### Distance Metrics

# Create Distance Metrics  
  
EuclideanDist10 <- dist(DTM10, method='euclidean')  
#print(EuclideanDist10)  
  
ManhattanDist10 <- dist(DTM10, method='manhattan')  
#print(ManhattanDist10)  
  
CosineDist10 <- dist(DTM10, method='cosine')  
#print(CosineDist10)  
  
CosineDistNorm10 <- dist(NormDTM10, method='cosine')  
#print(CosineDistNorm10)

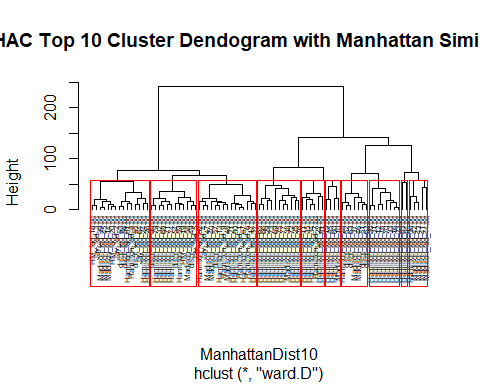
## Clustering Methods

### Heirarchical Algorithm Clustering Method

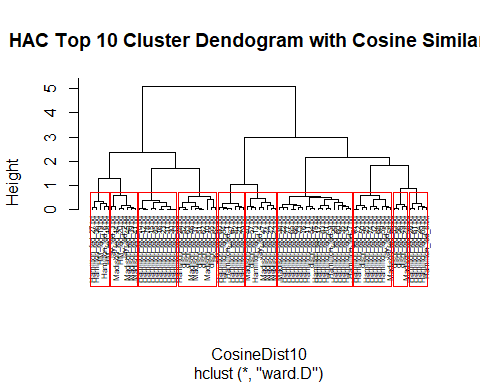
# HAC: Heirarchical Algorithm Clustering Method  
# Euclidean  
group\_E10 <- hclust(EuclideanDist10, method="ward.D")  
plot(group\_E10, cex=0.5, font=22, hang=-1, main= "HAC Top 10 Cluster Dendogram with Euclidean Similarity")  
rect.hclust(group\_E10, k=10)



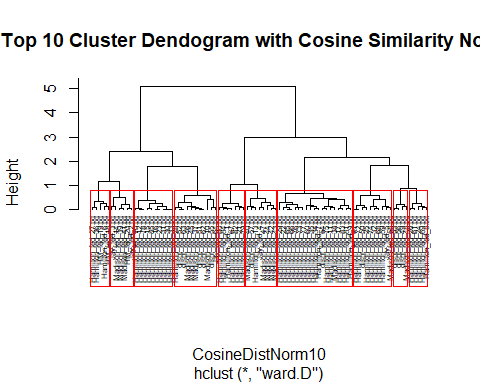
# Manhattan  
group\_M10 <- hclust(ManhattanDist10, method="ward.D")  
plot(group\_M10, cex=0.5, font=22, hang=-1, main= "HAC Top 10 Cluster Dendogram with Manhattan Similarity")  
rect.hclust(group\_M10, k=10)



# Cosine  
group\_C10 <- hclust(CosineDist10, method="ward.D")  
plot(group\_C10, cex=0.5, font=22, hang=-1, main= "HAC Top 10 Cluster Dendogram with Cosine Similarity")  
rect.hclust(group\_C10, k=10)



# Cosine Normalized  
group\_N10 <- hclust(CosineDistNorm10, method="ward.D")  
plot(group\_N10, cex=0.5, font=22, hang=-1, main= "HAC Top 10 Cluster Dendogram with Cosine Similarity Normalized")  
rect.hclust(group\_N10, k=10)



When evaluating the 10 chosen words under the Euclidean clustering method, we see strong accurate clustering on the last nine clusters, with ther first cluster containins a wide spread of authorship. This result is quite poor, and therefore will not be explored.

The Manhattan clustering method performed much better, though still left much to be desired. The clusters are mostly accurate unless including the disputed papers, in which the authorship is too broad in that cluster. The cosine and cosine simlarity performed very similarly, and were surprisingly decisive. The cosine similarity determined that papers 52, 63, 51, 62, 55, 53, 56, 57, and 49 to be authored by Madison, similar to our results above. Papers 54 and 50 are determined to be authored by Hamilton.

### K-Means Clustering

# K-Means Algorithm Clustering Method  
k1\_10 <- kmeans(NormDTM10, centers=3, nstart=100, iter.max=50)  
str(k1\_10)

## List of 9  
## $ cluster : Named int [1:85] 1 1 3 3 1 1 3 1 3 3 ...  
## ..- attr(\*, "names")= chr [1:85] "dispt\_fed\_49.txt" "dispt\_fed\_50.txt" "dispt\_fed\_51.txt" "dispt\_fed\_52.txt" ...  
## $ centers : num [1:3, 1:9] 0.0793 0.1166 0.1271 0.0942 0.0814 ...  
## ..- attr(\*, "dimnames")=List of 2  
## .. ..$ : chr [1:3] "1" "2" "3"  
## .. ..$ : chr [1:9] "general" "great" "interest" "might" ...  
## $ totss : num 6.85  
## $ withinss : num [1:3] 1.75 1.55 1.53  
## $ tot.withinss: num 4.84  
## $ betweenss : num 2.01  
## $ size : int [1:3] 29 31 25  
## $ iter : int 3  
## $ ifault : int 0  
## - attr(\*, "class")= chr "kmeans"

summary(k1\_10)

## Length Class Mode   
## cluster 85 -none- numeric  
## centers 27 -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

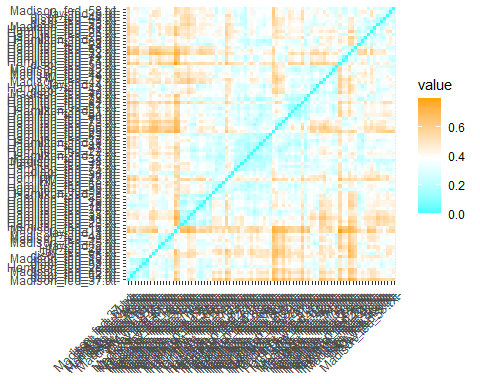
k2\_10 <- kmeans(NormDTM10, centers=3, nstart=50, iter.max=50)  
str(k2\_10)

## List of 9  
## $ cluster : Named int [1:85] 2 2 3 3 2 2 3 2 3 3 ...  
## ..- attr(\*, "names")= chr [1:85] "dispt\_fed\_49.txt" "dispt\_fed\_50.txt" "dispt\_fed\_51.txt" "dispt\_fed\_52.txt" ...  
## $ centers : num [1:3, 1:9] 0.1166 0.0793 0.1271 0.0814 0.0942 ...  
## ..- attr(\*, "dimnames")=List of 2  
## .. ..$ : chr [1:3] "1" "2" "3"  
## .. ..$ : chr [1:9] "general" "great" "interest" "might" ...  
## $ totss : num 6.85  
## $ withinss : num [1:3] 1.55 1.75 1.53  
## $ tot.withinss: num 4.84  
## $ betweenss : num 2.01  
## $ size : int [1:3] 31 29 25  
## $ iter : int 3  
## $ ifault : int 0  
## - attr(\*, "class")= chr "kmeans"

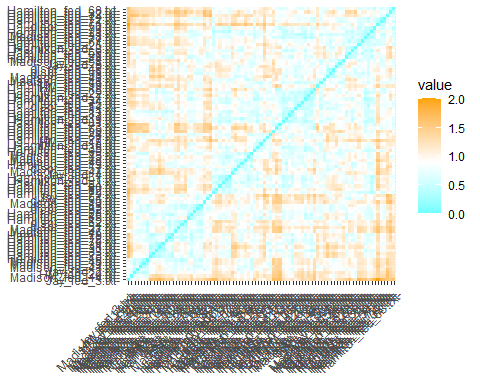
summary(k2\_10)

## Length Class Mode   
## cluster 85 -none- numeric  
## centers 27 -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

# Visualize the k means   
#Euclidean  
dist1\_10 <- get\_dist(NormDTM10, method= "euclidean")  
fviz\_dist(dist1\_10, gradient=list(low="cyan", mid="white", high="Orange"))



#Manhattan  
dist2\_10 <- get\_dist(NormDTM10, method= "manhattan")  
fviz\_dist(dist2\_10, gradient=list(low="cyan", mid="white", high="Orange"))



Visualizing both the K- Means Euclidean and Manhattan method show very checkered dispersion, with Hamilton near the middle and Madison on the fringes. As above, it is not intuitively obvious where the disputed papers lie.

## Results

When evaluating the top 10 words chosen, we did find perhaps our best clustering yet for the Cosine Similarity in the Heirarchical Algorithmic Clustering method, though found that we had less “usable” methods overall. However, we did find similar clustering as above with the Cosine similarity, and have enough data to dispute authorship.

# Conclusion

The dispute of the authorship of the Federalist papers has been a mystery to many avid history scholars, and may continue to confound the best minds out there. When evaluating the authorship of the 11 disputed papers via Clustering, we can deduct the following:

There is strong evidence to suggest papers 52, 53, 55, 56, 57, and 63 were authored by Madison. There is less strong evidence that suggests Madison also authored paper 54. There is strong evidence that suggests that Hamilton authored paper 50, and less strong evidence that Hamilton also authored paper 62. Papers 49 and 51 remain disputed in this analysis.

Contrary to Hamilton’s claims, our analysis suggests that Madison in fact authored the majority of the disputed papers, though Hamilton did author at least one.

* Madison: 63, 52, 53, 55, 56, 57
* Probably Madison: 54
* Hamilton: 50
* Probably Hamilton: 62
* Disputed: 51, 49