Clustering, Decision Trees, a Random Forest, and the Federalist Papers

Ronen Reouveni

8/11/2020

Introduction

Pivotal moments in history are often captured in the literature of the time. The federalist papers were written in order to try and convince the newly formed USA to accept the constitution. There were 85 papers written and all of those papers were written by either Hamilton, Madison, Jay, or Hamilton and Madison together. Interestingly, they all used the same pseudonym to release their writings in order to hide their identities. However, throughout their lifetimes and thereafter, a number of the papers were attributed to their actual author. Before Hamilton's death, he even gave out a list of the ones he claimed to have written. Herein lies the issue at hand, some of the papers have authorship claimed by multiple authors.

There have been many instances in history of multiple authors claiming ownership of a work. Although 100% certainty in these situations may be impossible, there are many linguistic ideas that can shed light. Literary style is noticeable to readers who know what to look for. Therefore, scholars have attempted to identify the true authors of various works by comparing the literary style. An author, by virtue of being who they are, will write with a certain style. These differences between how people write can be seen in simple things such as liking a specific word. For example, one author may like using a word and therefore that word is found more in their works. These words may or may not be words with deep meaning. One author could write the word "the" far more often than another writer. This simple difference can be used to try and identity an author.

As technology and the tools to understanding the world grow, so do the techniques available to scholars. Refining the idea of who wrote which federalist papers is of great historical importance to the USA and the world. The federalist papers were not simple everyday publications, they were in essence philosophical. They laid the foundation for what the US political and governmental system should look like. Understanding who wrote each work is important to understanding the original intentions for the USA. Since the true authorship may never be known with certainty, it is important to keep asking the question in order to refine results. Hopefully, the mystery of the disputed federalist papers will eventually be solved.

Analysis and Models

```
# install.packages('cluster')
# install.packages('factoextra')
# install.packages('NbClust')
# install.packages('ggplot2')
# install.packages("wordspace")
# install.packages('tidyverse')
# install.packages('dendextend')
```

```r
#install.packages('randomcoloR')
#install.packages('rpart')
#install.packages('randomForest')

#references used
#https://www.datanovia.com/en/lessons/determining-the-optimal-number-of-clusters-3-must-know-methods/
#https://www.r-graph-gallery.com/340-custom-your-dendrogram-with-dendextend.html
library(cluster)
library(factoextra)
```

```
## Loading required package: ggplot2

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

```r
library(NbClust)
library(ggplot2)
library(wordspace)
```

```
## Loading required package: Matrix
```

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------- tidyverse 1.3.0 --

## v tibble  3.0.3     v dplyr   1.0.1
## v tidyr   1.0.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
## v purrr   0.3.4

## -- Conflicts ------------------------------------------------------ tidyverse_conflicts() --
## x tidyr::expand() masks Matrix::expand()
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x tidyr::pack()   masks Matrix::pack()
## x tidyr::unpack() masks Matrix::unpack()
```

```r
library(dendextend)
```

```
##
## ---------------------
## Welcome to dendextend version 1.13.4
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## Or contact: <tal.galili@gmail.com>
##
##  To suppress this message use:  suppressPackageStartupMessages(library(dendextend))
## ---------------------
```

```
##
## Attaching package: 'dendextend'

## The following object is masked from 'package:stats':
##
##     cutree
```

```r
library(randomcoloR)
library(rpart)
```

```
##
## Attaching package: 'rpart'

## The following object is masked from 'package:dendextend':
##
##     prune
```

```r
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:dplyr':
##
##     combine

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

```r
library(randomForestExplainer)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

About the Data

The data consists of the term frequency inverse document frequency of 72 select words in 85 Federalist Papers. The majority of the Federalist Papers have a definite author, but there are 11 which are disputed.

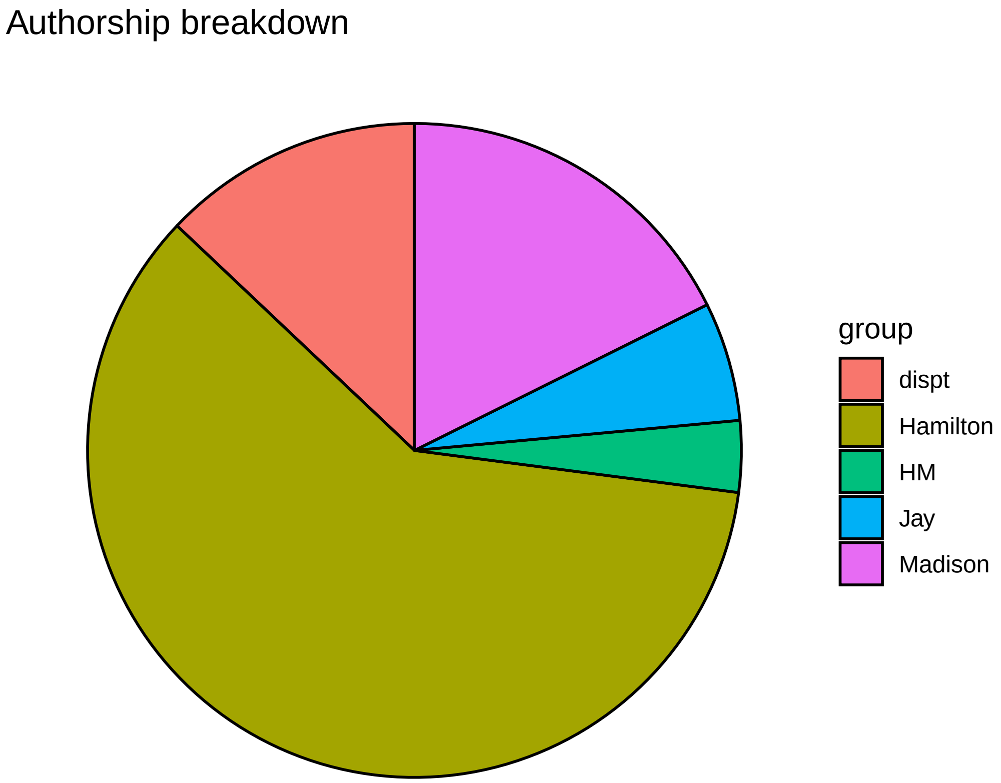
```r
#read in the pre cleaned data set
papers <- read.csv("/Users/ronenreouveni/Desktop/fedPapers85.csv")
```

The data consists of 51 Hamilton papers, 15 Madison papers, 3 written by both Hamilton and Madison, and 5 written by Jay. The 11 disputed papers were written by 1 of these 4 authors (including the possibility of joint authorship by Hamilton and Madison)

```r
#What is the breakdown of authorship
table(papers[,1])
```

```
##
##    dispt Hamilton       HM      Jay  Madison
##       11       51        3        5       15
```

```r
#build frame
authorFrame <- data.frame(
  group=levels(papers[,1]),
  value=table(papers[,1])
)

#viz
ggplot(authorFrame, aes(x="", y=value.Freq, fill=group)) +
  geom_bar(stat="identity", color="black") +
  coord_polar("y", start=0) +
  ggtitle("Authorship breakdown") +
  theme_void()
```



There needs to be a method in deciding which words to select for analysis. Not all words in the data set will be influential in distinguishing authors. Furthermore, including non relevant words will only add unwanted noise. Only columns (words) with a higher variance between writers will help distinguish authorship. To calculate the variance in word usage between authors there are intermediate steps. First, calculate the mean tf-idf of each word for each author. To accomplish this, create a new data frame for each author, (leaving out disputed papers) and then calculate the column means. Save a vector of the column means for each author into a new data frame. From there take the variance of each column and order them to see which words have the highest variance.

```r
#create a dataframe for each author
papersHam <- papers[papers$author == "Hamilton",]
papersH_and_M <- papers[papers$author == "HM",]
papersJay <- papers[papers$author == "Jay",]
papersMadison <- papers[papers$author == "Madison",]


#this function finds the average tfidf of a word for a specific author
createWordMean <- function(x) {
  y <- ncol(x)
  x <- colMeans(x[,3:y])
  newVec_1 <- c()
  for (i in 1:length(x)) {
    newVec_1[i] <- x[[i]]
  }
  print(newVec_1)
}
```

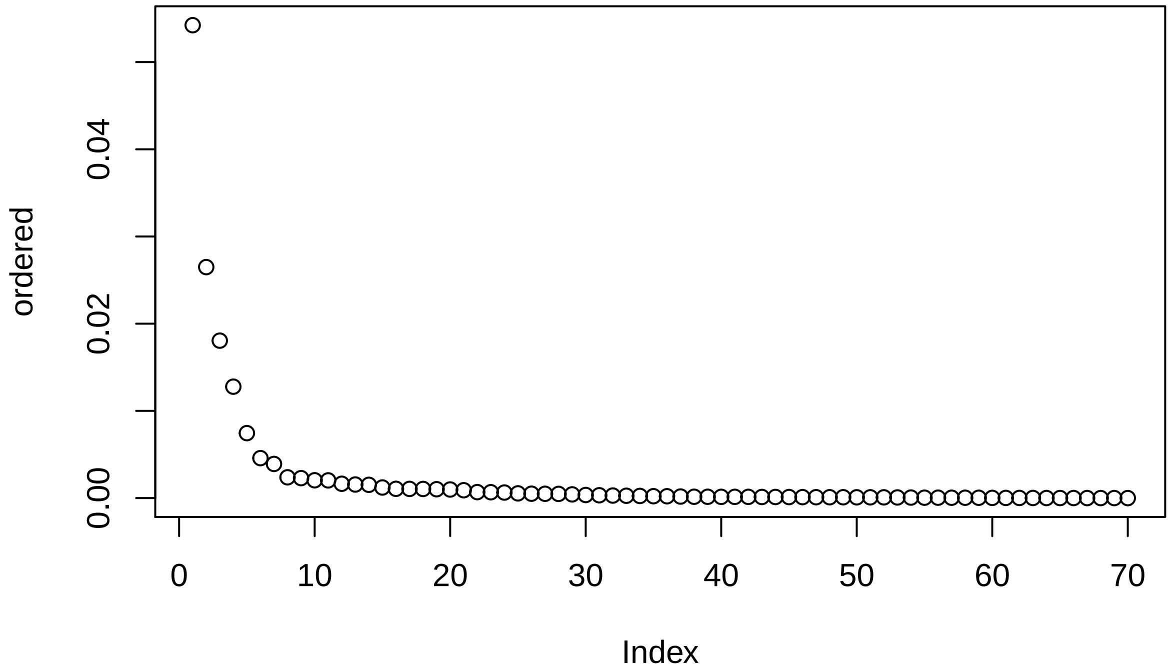
```r
#fun the function on each subsetted dataframe
hamiltonVector <- createWordMean(papersHam)
ham_n_mad_Vector <- createWordMean(papersH_and_M)
jay_Vector <- createWordMean(papersJay)
mad_Vector <- createWordMean(papersMadison)
columns <- colnames(papersHam)
```

```r
#put the results into a dataframe
newFrame <- papersHam
newFrame <- data.frame(rbind(hamiltonVector,ham_n_mad_Vector,jay_Vector,mad_Vector))



#name the columns of the new df their corresponding words
colnames(newFrame) <- columns[3:length(columns)]


#the dataframe is the average tfidf of a word for each author
newFrame[,1:3]
```

```
##                          a        all        also
## hamiltonVector   0.3156078 0.05376471 0.004784314
## ham_n_mad_Vector 0.2133333 0.04266667 0.006000000
## jay_Vector       0.1598000 0.03600000 0.019800000
## mad_Vector       0.2698000 0.05533333 0.011066667
```

After calculating the variance of words used by authors select the top 40% of words to use for analysis.

```r
#find the variance of each column (word)
wordVariance <- sapply(newFrame, var)


#order and plot the results
ordered <- sort(wordVariance, decreasing = TRUE)
plot(ordered)
```



```
#set the good words as the top 40% of words
goodWords <- data.frame(ordered[1:(length(ordered)*2/5)])

#number of words chosen
length(rownames(goodWords))
```

```
## [1] 28
```

```
#words chosen
rownames(goodWords)
```

```
##  [1] "the"    "and"    "of"     "be"     "to"     "a"      "that"   "would"  "will"
## [10] "was"    "or"     "in."    "it"     "is"     "his"    "their"  "not"    "as"
## [19] "by"     "had"    "which"  "our"    "been"   "on"     "an"     "may"    "upon"
## [28] "more"
```

Create a data frame only selecting the words from the ‘goodWords’ vector. These are the top 40% of words with the highest variance between authors. Another important aspect of the data frame is that the row numbers are replaced with the file name.

```
#create a new data frame of only the "good words"
smallFrame <- papers[,c("author", "filename",rownames(goodWords))]



#rename the rows with their corresponding filename
rownames(smallFrame) <- smallFrame$filename
rownames(papers) <- papers$filename
```

```
write.csv(smallFrame,"/Users/ronenreouveni/Desktop/fedPapers_varianceframe.csv")
write.csv(papers,"/Users/ronenreouveni/Desktop/fedPapers_FullPapersframe.csv")
```

This next section reloads the data for the preprocessing required for decision trees. Both Jay and Hamilton/Madison papers are removed. This is so there is a version of the data that only contains Hamilton and Madison papers. These are the two writers of interest in predicting which who could have written the disputed papers.

```
set.seed(1234)
treeData_small <- read.csv("/Users/ronenreouveni/Desktop/fedPapers_varianceframe.csv")
treeData_small <- treeData_small[,-1]


treeData_small <- treeData_small[-c(which(treeData_small$author==’Jay’), which(treeData_small$author==’
treeData_small <- droplevels(treeData_small)
treeData_small <- treeData_small[,-2]
treeData_small <- treeData_small[,c(2:ncol(treeData_small),1)]

treeData_small_disputed <- treeData_small[1:11,]
treeData_small_full_noD <- treeData_small[-c(1:11),]
```

This consutrction of indexes ensures that the proportion of Hamilton to Madison papers in the training set are equal to that of the testing set. This randomly selects 65% of the Hamilton papers and 65% of the Madison papers for training the decision tree.

```
indexes <- sample(1:55, .65*length(1:55))
indexes <- c(indexes, sample(56:nrow(treeData_small_full_noD), .65*length(56:nrow(treeData_small_full_n
```

Models

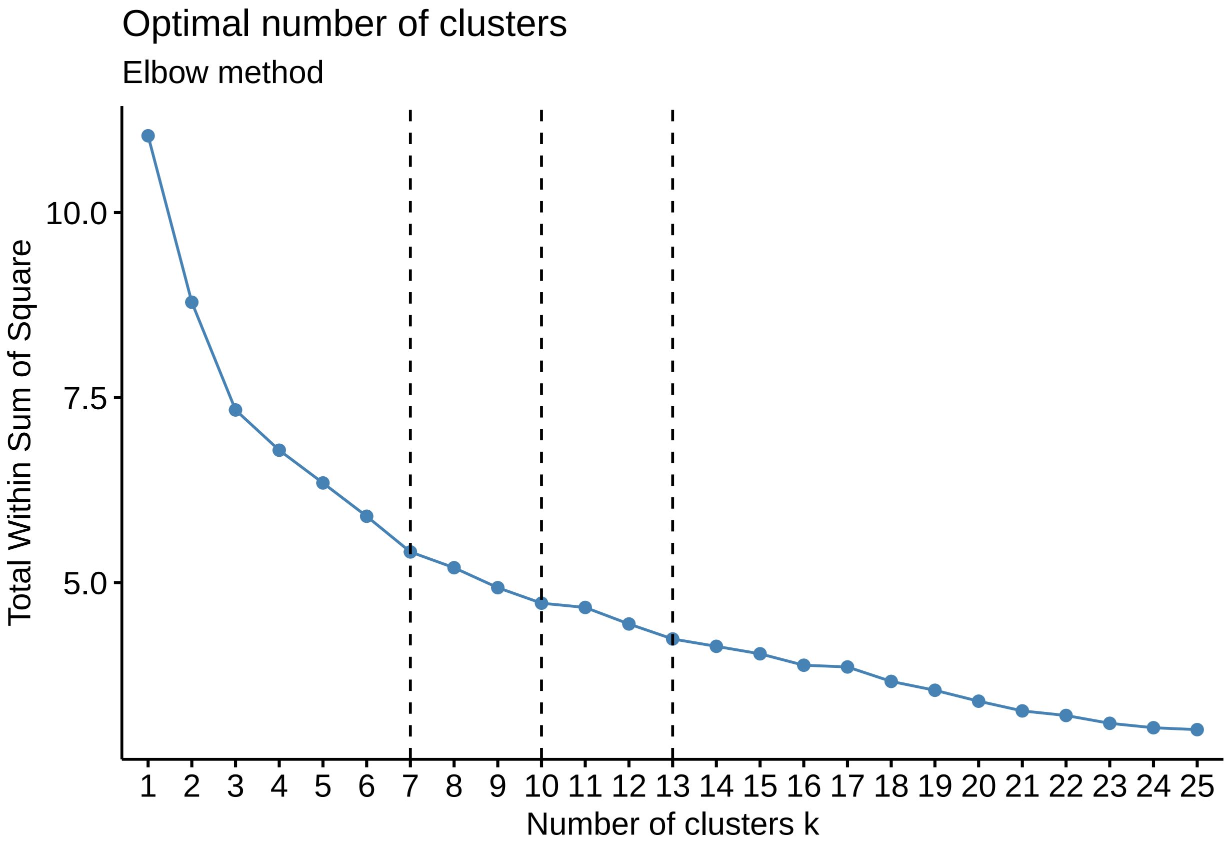
Two clustering algorithms will be used, K-means and Hierarchical. The purpose is to see if these can correctly cluster writers together. If so, then the clusters that the disputed works fall into will be able to shed light on who may have written them.

Decision trees will also be used for classification. This will allow for comparing results between decision trees and clustering. Furthermore, a random forest will also be implemented and compared to the other models.

K-means Clustering

K-means clustering is used with varying amount of clusters. Which clusters to use in the rest of the analysis is based on the ‘Elbow Method’. This visually shows good candidate amounts for clusters. Vertical lines are drawn at those points.

```
#Use "elbow method" to find optimal cluster amount
fviz_nbclust(smallFrame[,3:ncol(smallFrame)], kmeans, method = "wss", k.max = 25) +
  geom_vline(xintercept = 7, linetype = 2)+
  geom_vline(xintercept = 10, linetype = 2)+
  geom_vline(xintercept = 13, linetype = 2)+
  labs(subtitle = "Elbow method")
```

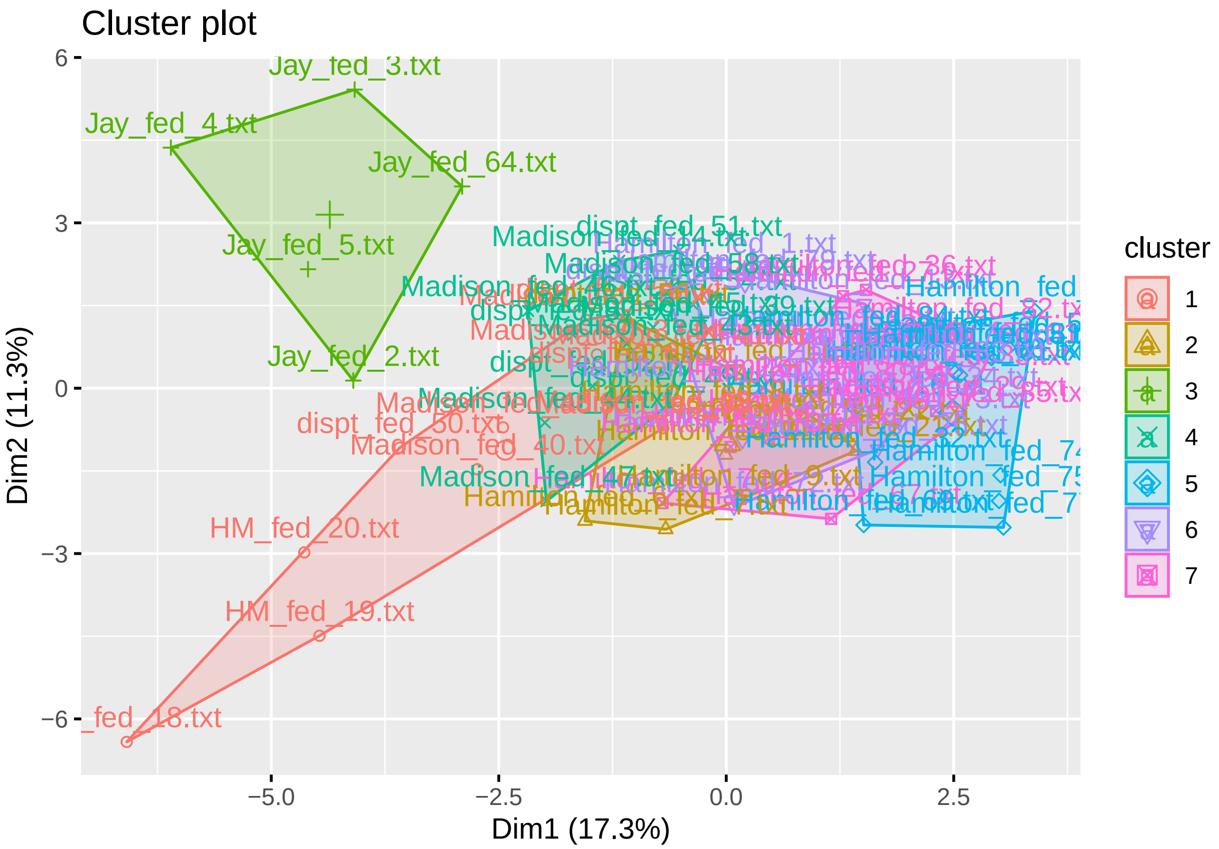


K-means is used with 7,10, and 13 clusters. Interestingly, when only using 7 clusters Jay is clustered perfectly together. The rest of the clusters overlap. However, when clusters are increased to 10 and then 13, HM (Hamilton and Madison) are clustered perfectly. These can be seen in the visualizations. Although the fviz_cluster is mostly not understandable, it is still interesting to see which observations were clustered without any overlap. The silhouette visualization shows the clusters. The distance shows how well an observation fits in a cluster. Closer to 0 means the observation is closer to the edge of a cluster.

```
#kmeans 7 clusters
k_model7 <- kmeans(smallFrame[,3:ncol(smallFrame)],7, nstart = 20)
k_model7$tot.withinss
```

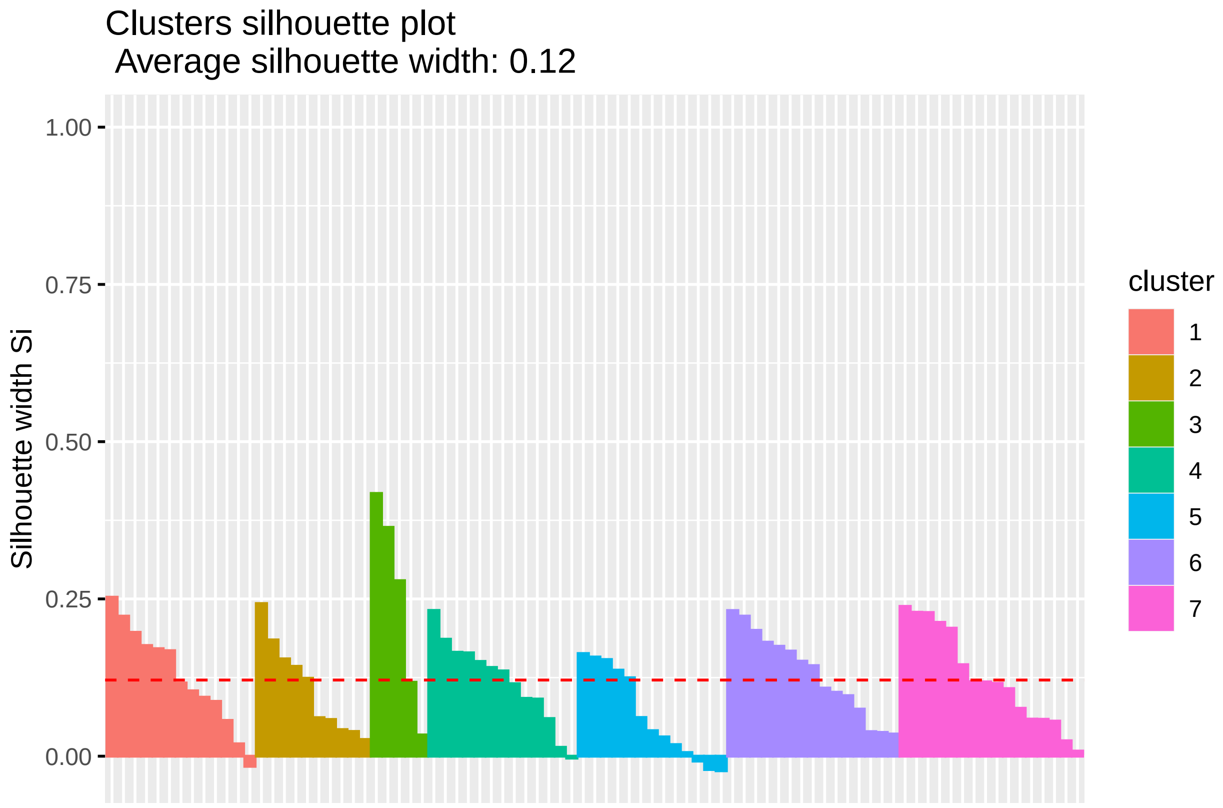
```
## [1] 5.414581
```

```
fviz_cluster(k_model7, smallFrame[,-c(1,2)])
```



```
sil7 <- silhouette(k_model7$cluster, dist(smallFrame[,3:ncol(smallFrame)]))
fviz_silhouette(sil7)
```

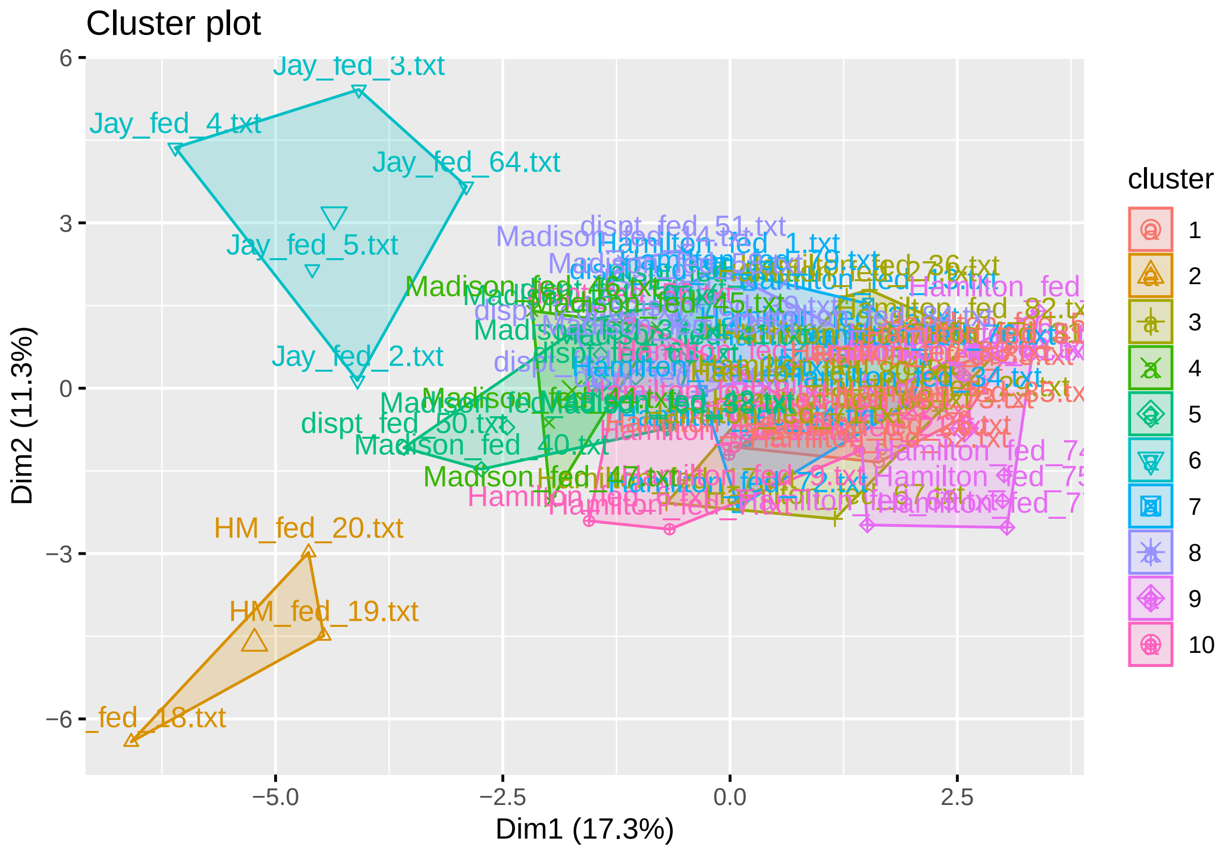
```
##   cluster size ave.sil.width
## 1       1   13          0.13
## 2       2   10          0.11
## 3       3    5          0.24
## 4       4   13          0.12
## 5       5   13          0.06
## 6       6   15          0.13
## 7       7   16          0.13
```



```
#kmeans 10 clusters
k_model10 <- kmeans(smallFrame[,3:ncol(smallFrame)],10, nstart = 20)
k_model10$tot.withinss
```

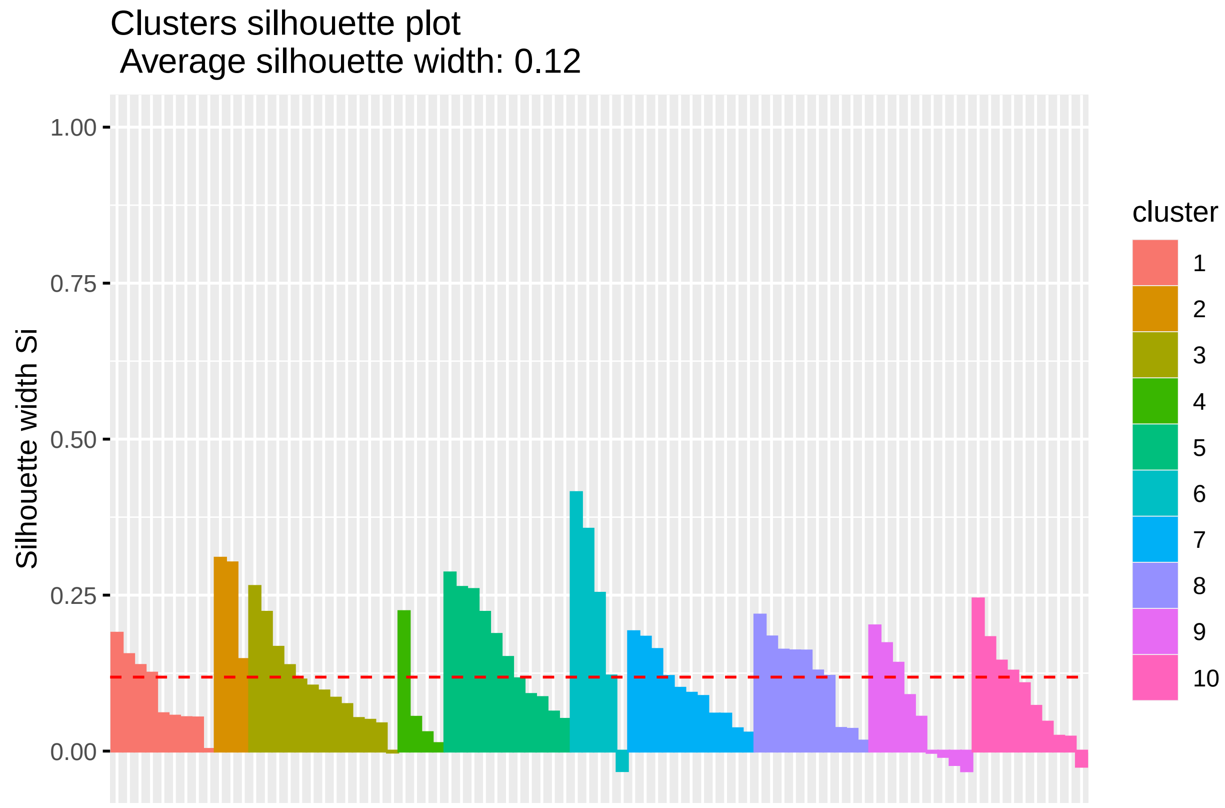
```
## [1] 4.696933
```

```
fviz_cluster(k_model10, smallFrame[,-c(1,2)])
```



```r
sil10 <- silhouette(k_model10$cluster, dist(smallFrame[,3:ncol(smallFrame)]))
fviz_silhouette(sil10)
```

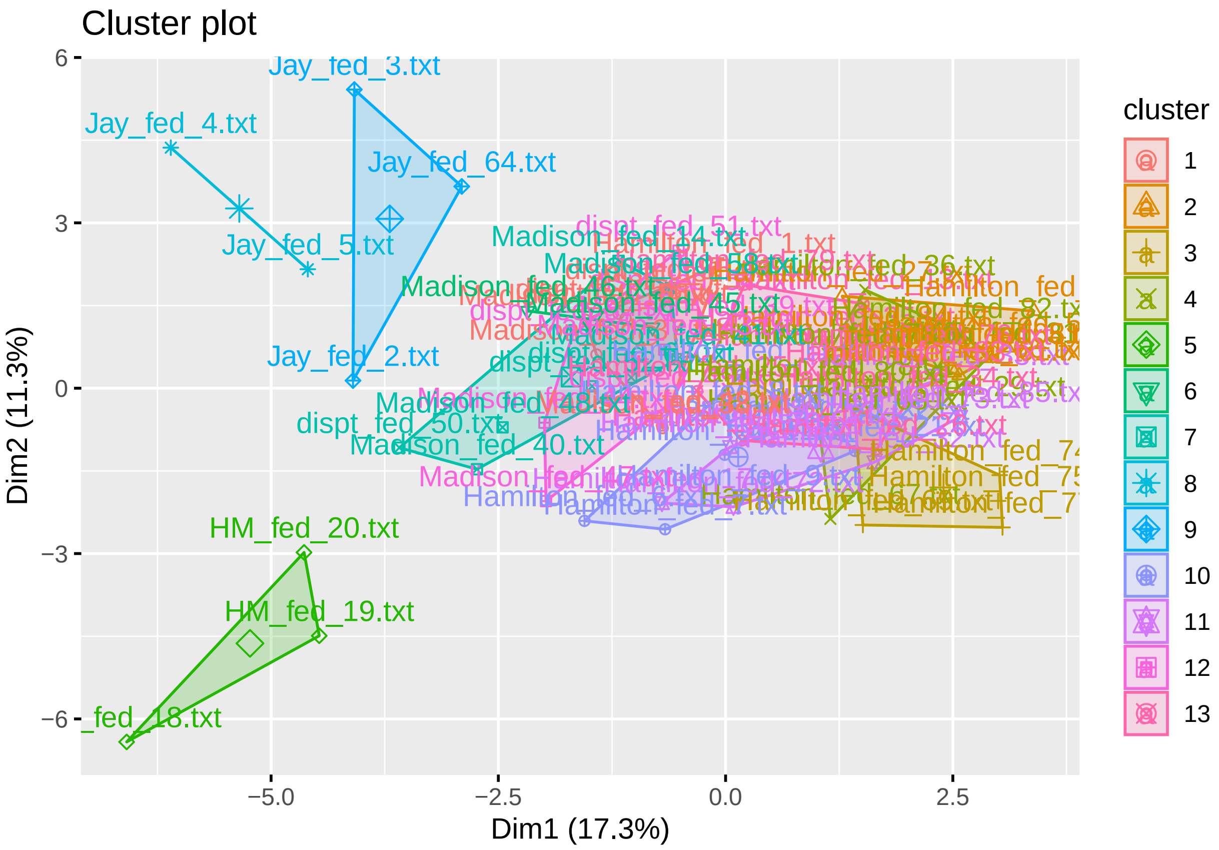
```
##    cluster size ave.sil.width
## 1        1    9          0.09
## 2        2    3          0.25
## 3        3   13          0.11
## 4        4    4          0.08
## 5        5   11          0.16
## 6        6    5          0.22
## 7        7   11          0.10
## 8        8   10          0.12
## 9        9    9          0.07
## 10      10   10          0.09
```



```
#kmeans 13 clusters
k_model13 <- kmeans(smallFrame[,3:ncol(smallFrame)],13, nstart = 20)
k_model13$tot.withinss
```

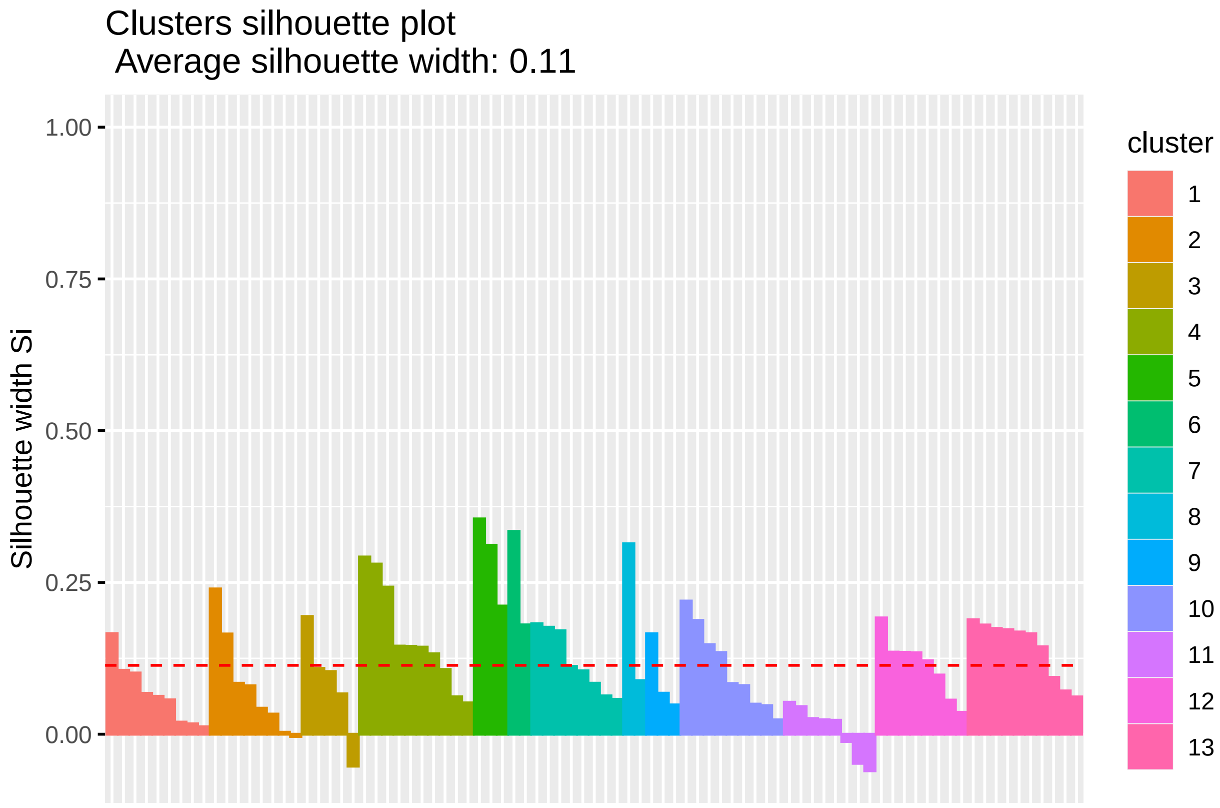
```
## [1] 4.176643
```

```
fviz_cluster(k_model13, smallFrame[,-c(1,2)])
```

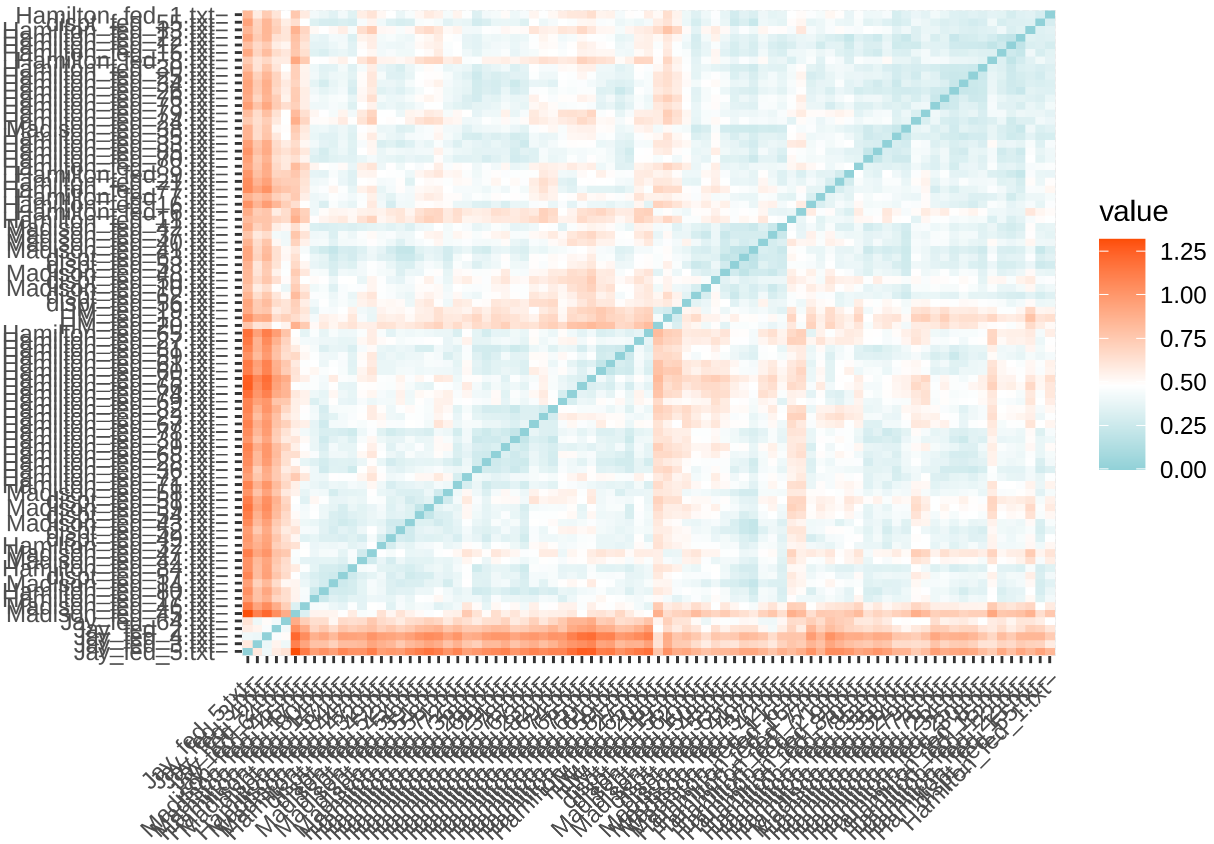


```
sil13 <- silhouette(k_model13$cluster, dist(smallFrame[,3:ncol(smallFrame)]))
fviz_silhouette(sil13)
```

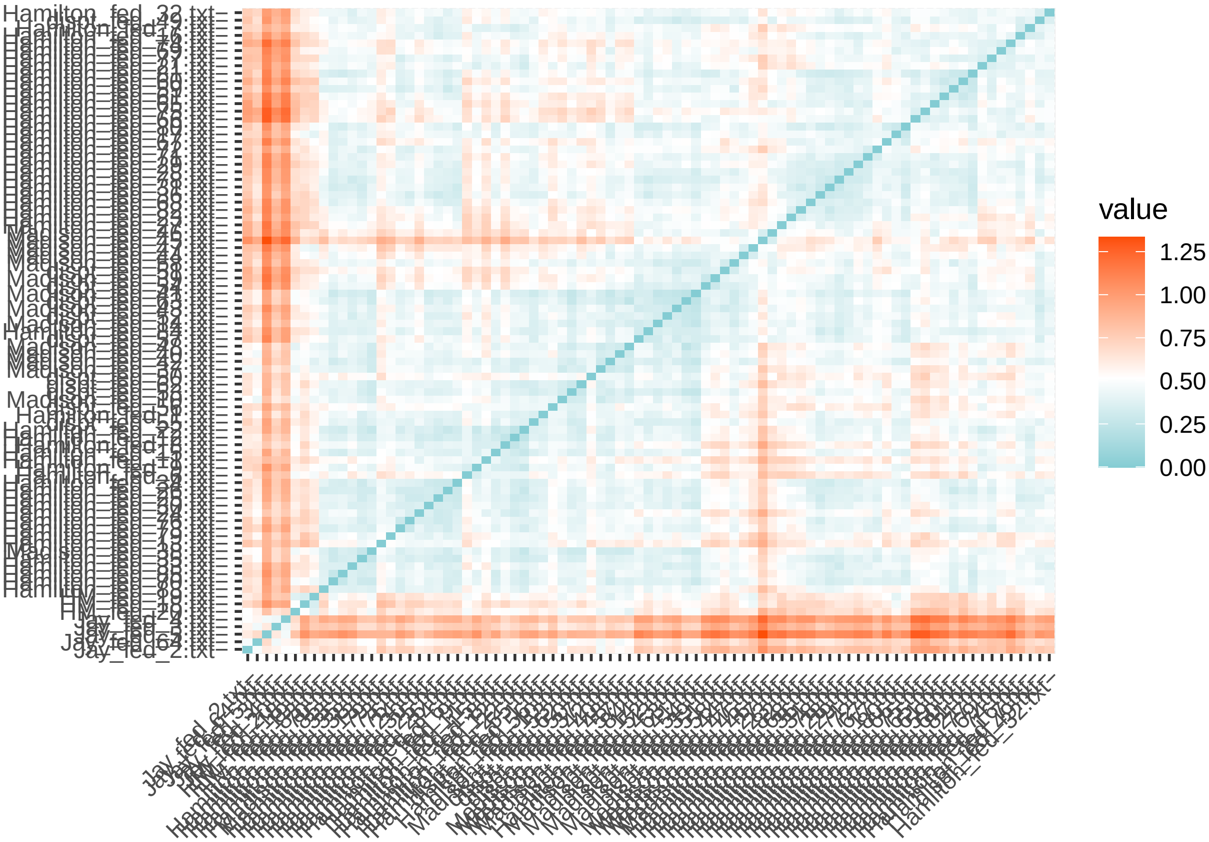
```
##    cluster size ave.sil.width
## 1        1    9          0.07
## 2        2    8          0.08
## 3        3    5          0.08
## 4        4   10          0.16
## 5        5    3          0.29
## 6        6    2          0.26
## 7        7    8          0.12
## 8        8    2          0.20
## 9        9    3          0.09
## 10      10    9          0.11
## 11      11    8          0.01
## 12      12    8          0.11
## 13      13   10          0.14
```



```
#distnace visualization on the trimmed dataframe
distance <- get_dist(smallFrame[,3:ncol(smallFrame)])
fviz_dist(distance, gradient = list(low = "#00AFBB", mid = "white", high = "#FC4E07"))
```



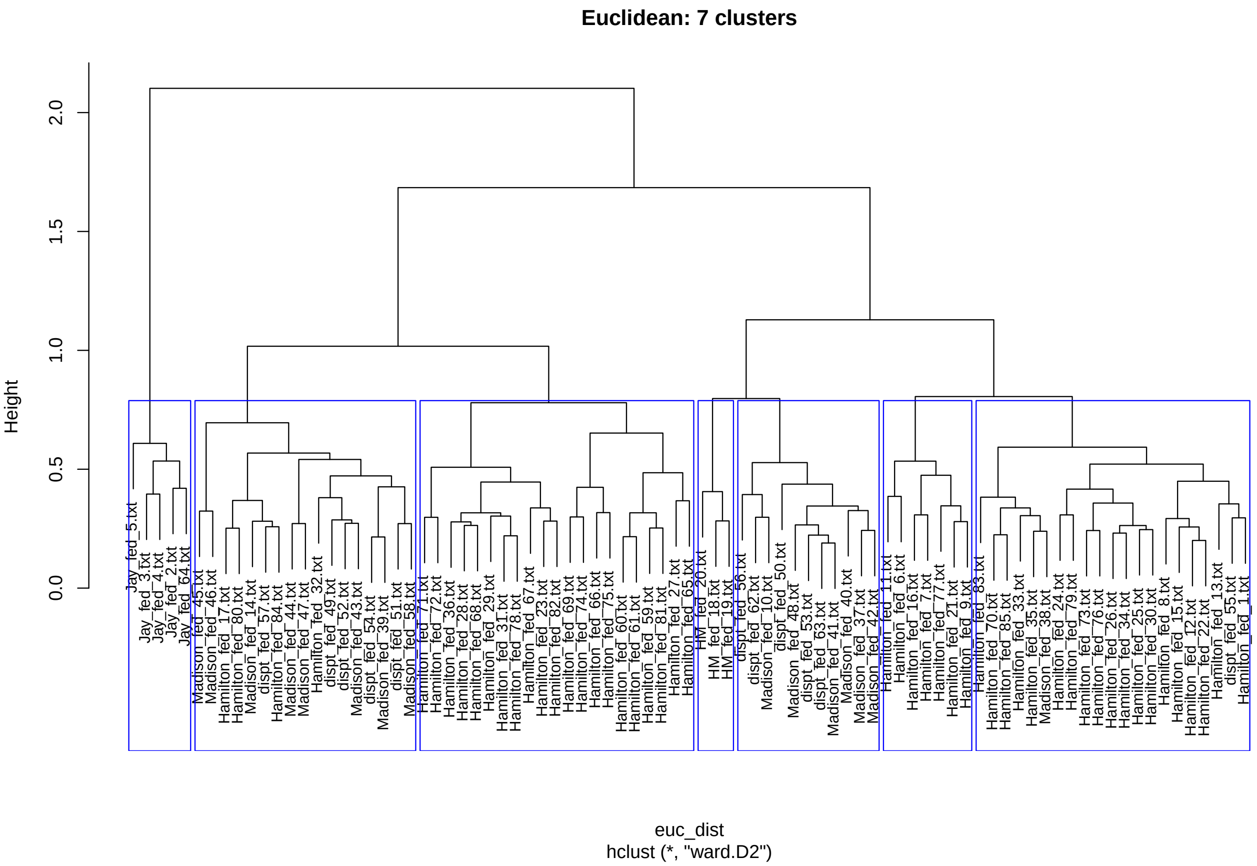
```
#distnace visualization on the full dataframe
distance2 <- get_dist(papers[,3:ncol(papers)])
fviz_dist(distance2, gradient = list(low = "#00AFBB", mid = "white", high = "#FC4E07"))
```



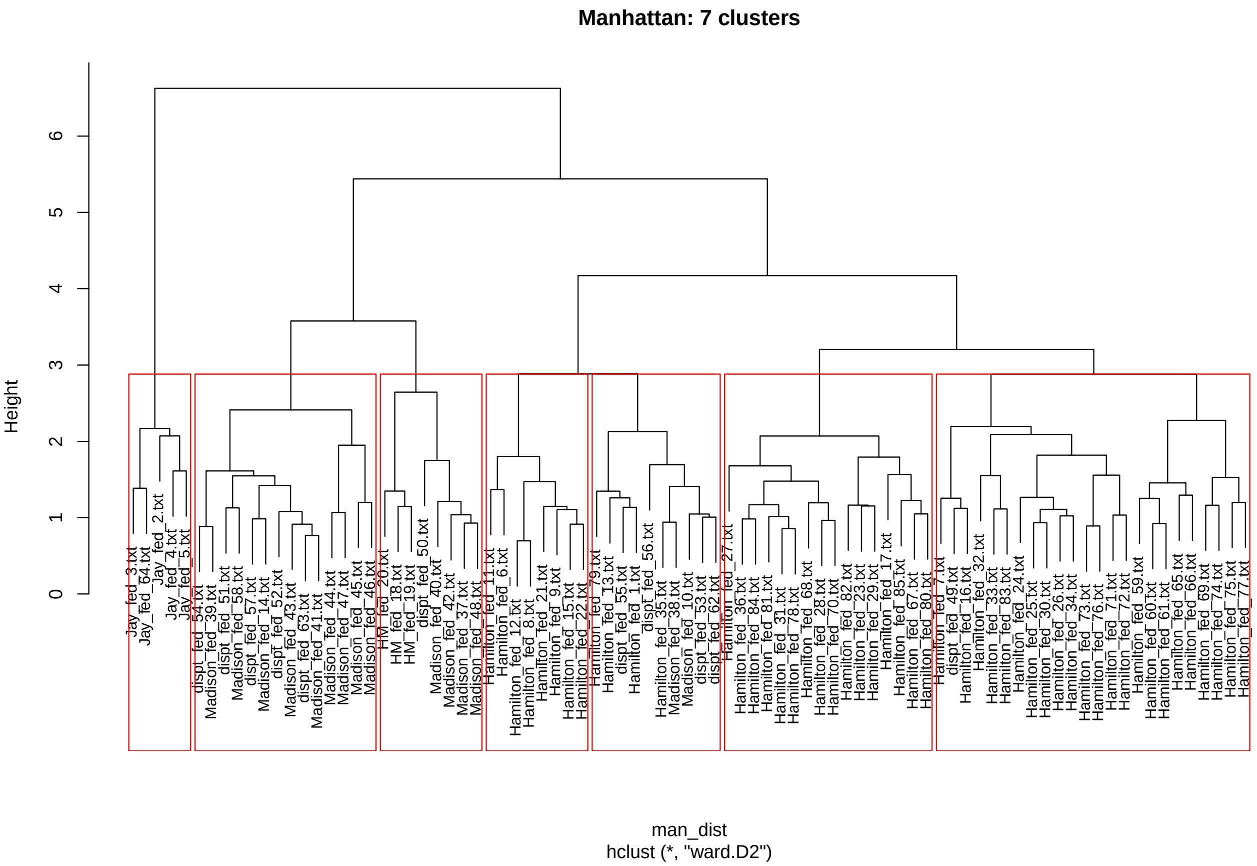
HAC Clustering

Hierarchical clustering is used with varying clusters and distance measures. It also uses the full data with all the words and the smaller data frame with less words based on variance. The differing results are comparable using dendograms. Subsequently, all the clusters are put into a data frame where the name of the cluster is replaced by the author who most appears in that cluster. This allows for extremely easy comparisons between the outcomes from cluster size, distance measure, and data. On the smaller more refined data frame, the 'goodWords' data frame, euclidean, manhattan, and cosine are all used with 7 clusters. Cosine distance is repeated with the 'goodWords' data for 10 and 13 clusters. Cosine, euclidean, and manhattan distance are also used with 13 clusters on the full data.

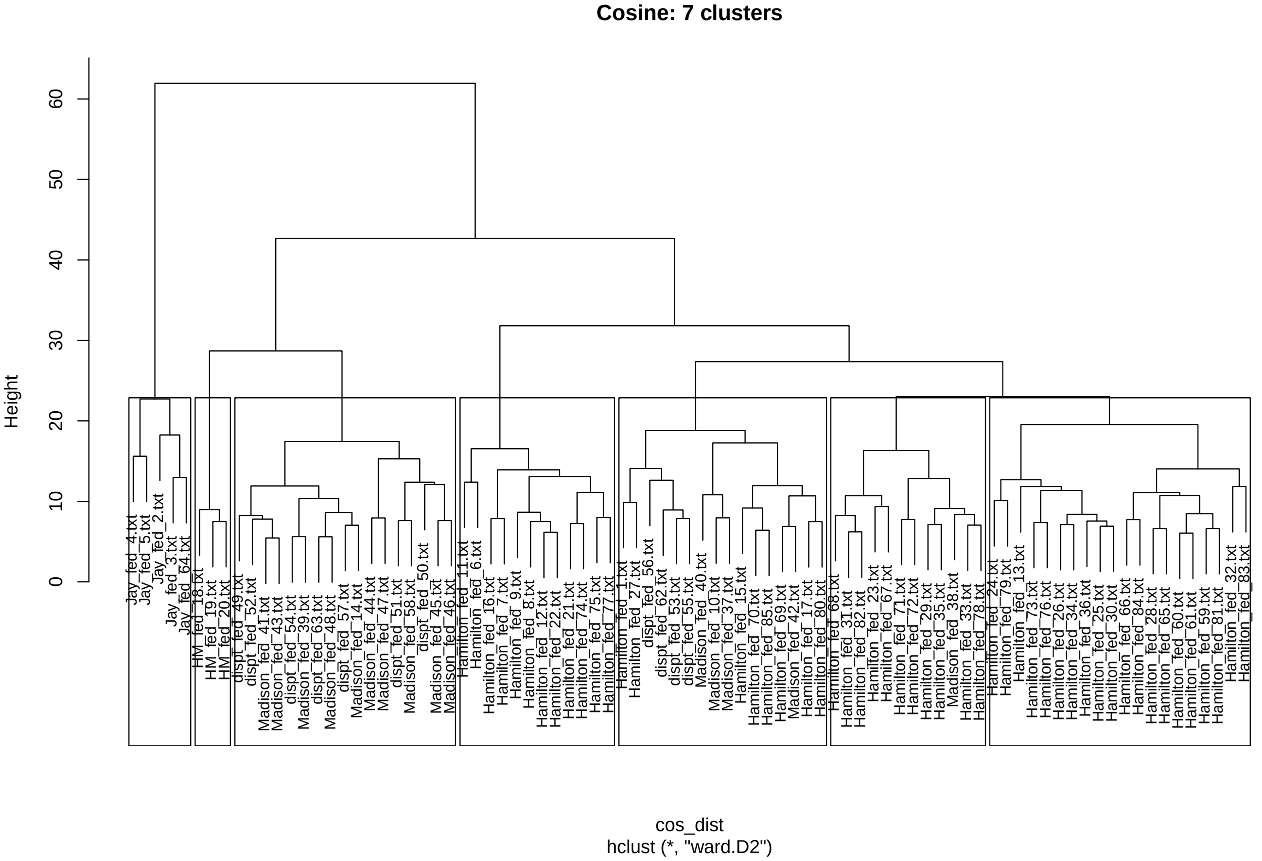
```r
#Create clusters

#eculidean distance
euc_dist <- as.dist(dist.matrix(as.matrix(smallFrame[,3:ncol(smallFrame)]), method = "euclidean"))
fit1 <- hclust(euc_dist, method="ward.D2")
plot(fit1, main = "Euclidean: 7 clusters", cex = .8)
euclid_small <- cutree(fit1, k=7)
rect.hclust(fit1, k=7, border="blue")
```



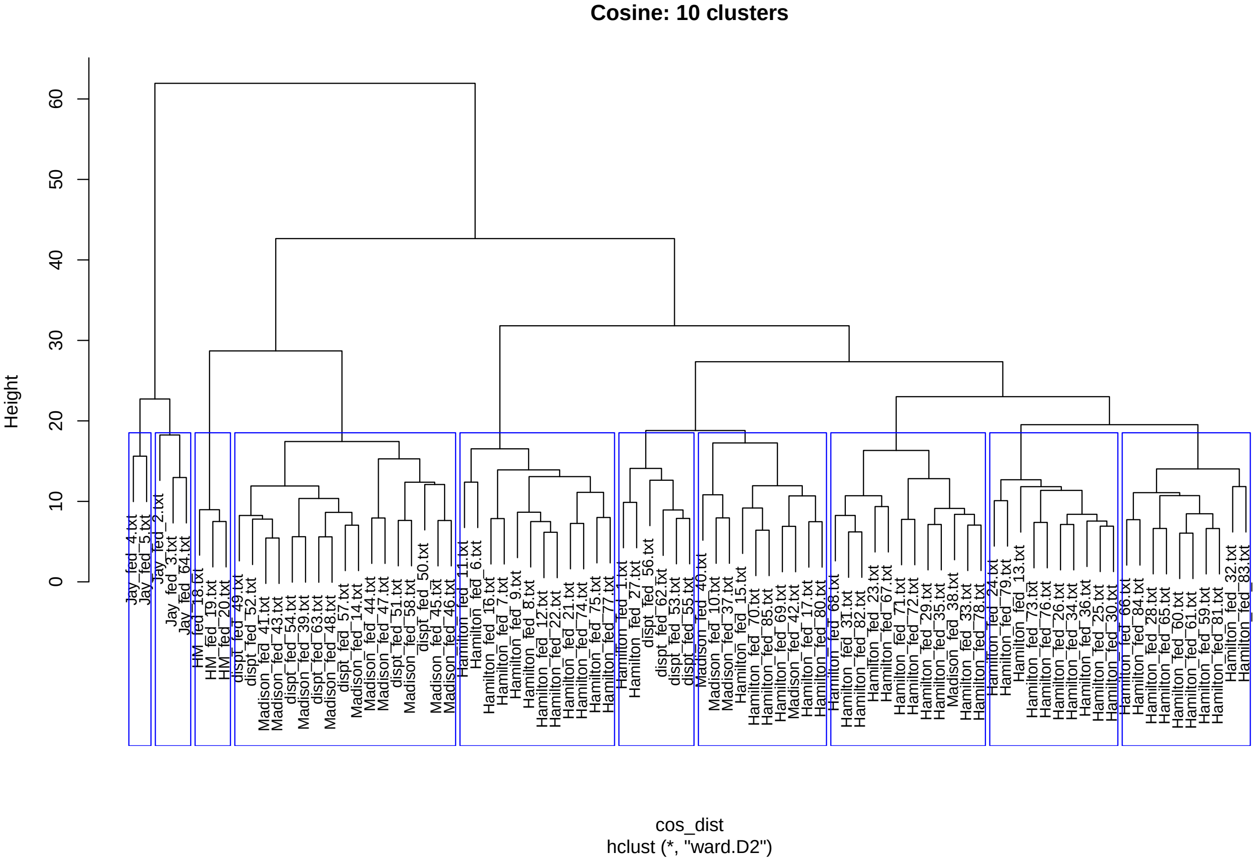
```
#manhattan distance
man_dist <- as.dist(dist.matrix(as.matrix(smallFrame[,3:ncol(smallFrame)]), method = "manhattan"))
fit2 <- hclust(man_dist, method="ward.D2")
plot(fit2, main = "Manhattan: 7 clusters", cex = .8)
manhat_small <- cutree(fit2, k=7)
rect.hclust(fit2, k=7, border="red")
```



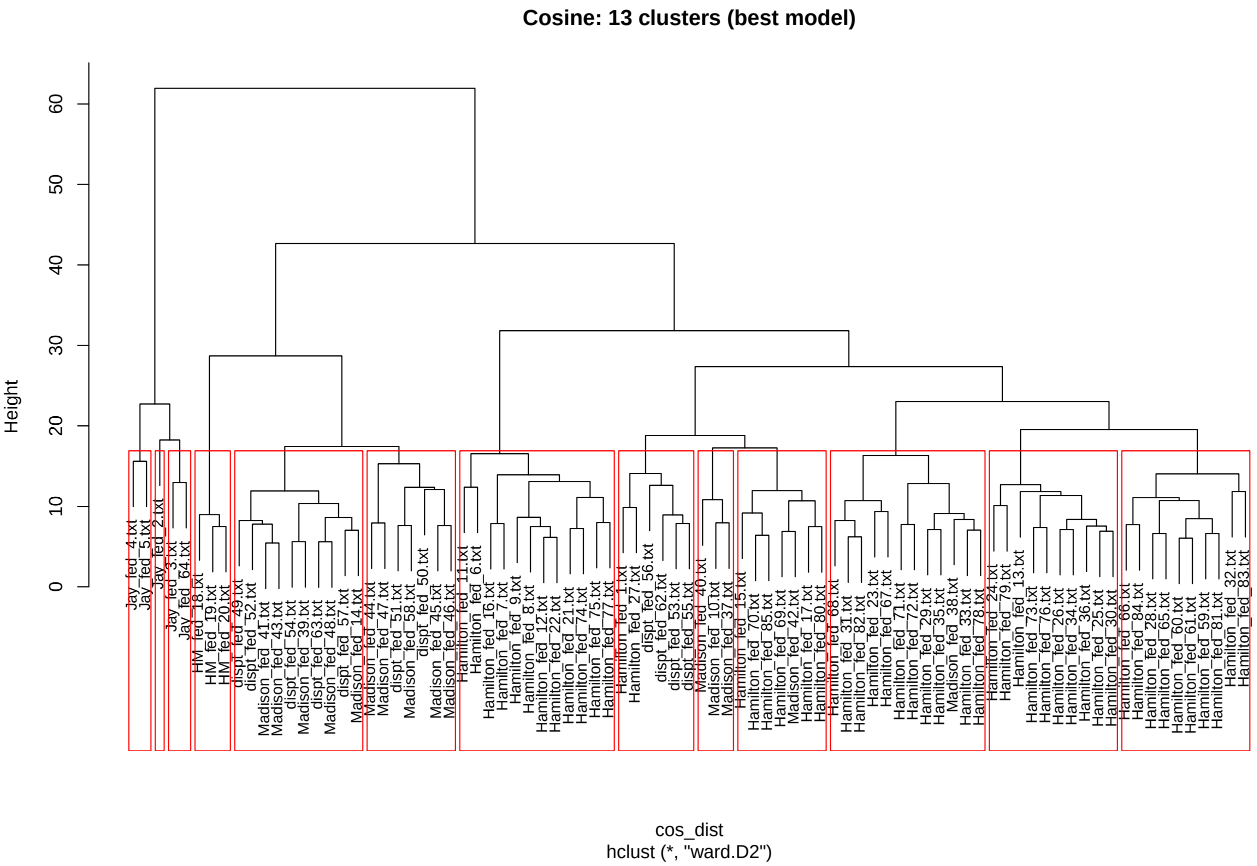
```
#cosine distance 7 clusters
cos_dist <- as.dist(dist.matrix(as.matrix(smallFrame[,3:ncol(smallFrame)]), method = "cosine"))
fit3 <- hclust(cos_dist, method="ward.D2")
plot(fit3, main = "Cosine: 7 clusters", cex = .8)
cosine_small7 <- cutree(fit3, k=7)
rect.hclust(fit3, k=7, border="black")
```



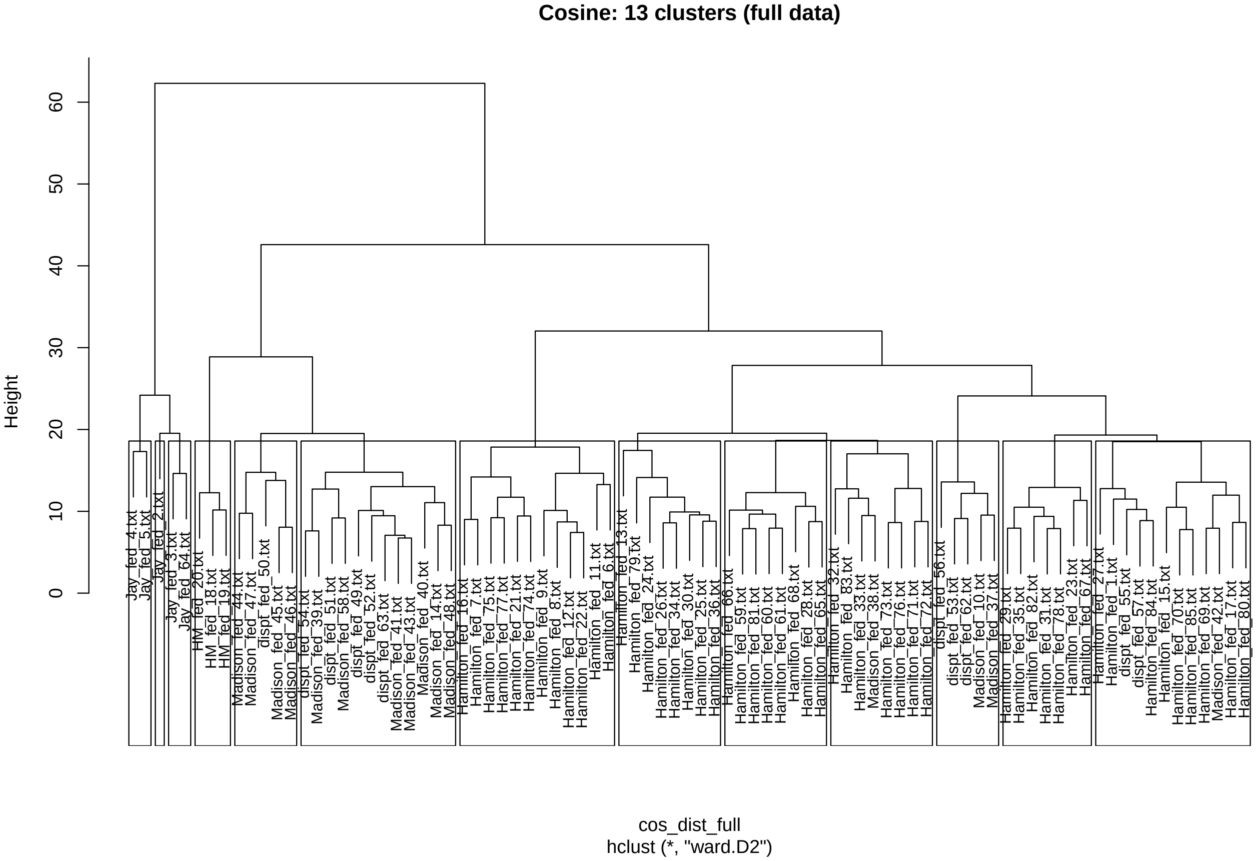
```
#cosine distance 10 clusters
plot(fit3, main = "Cosine: 10 clusters", cex = .8)
cosine_small10 <- cutree(fit3, k=10)
rect.hclust(fit3, k=10, border="blue")
```



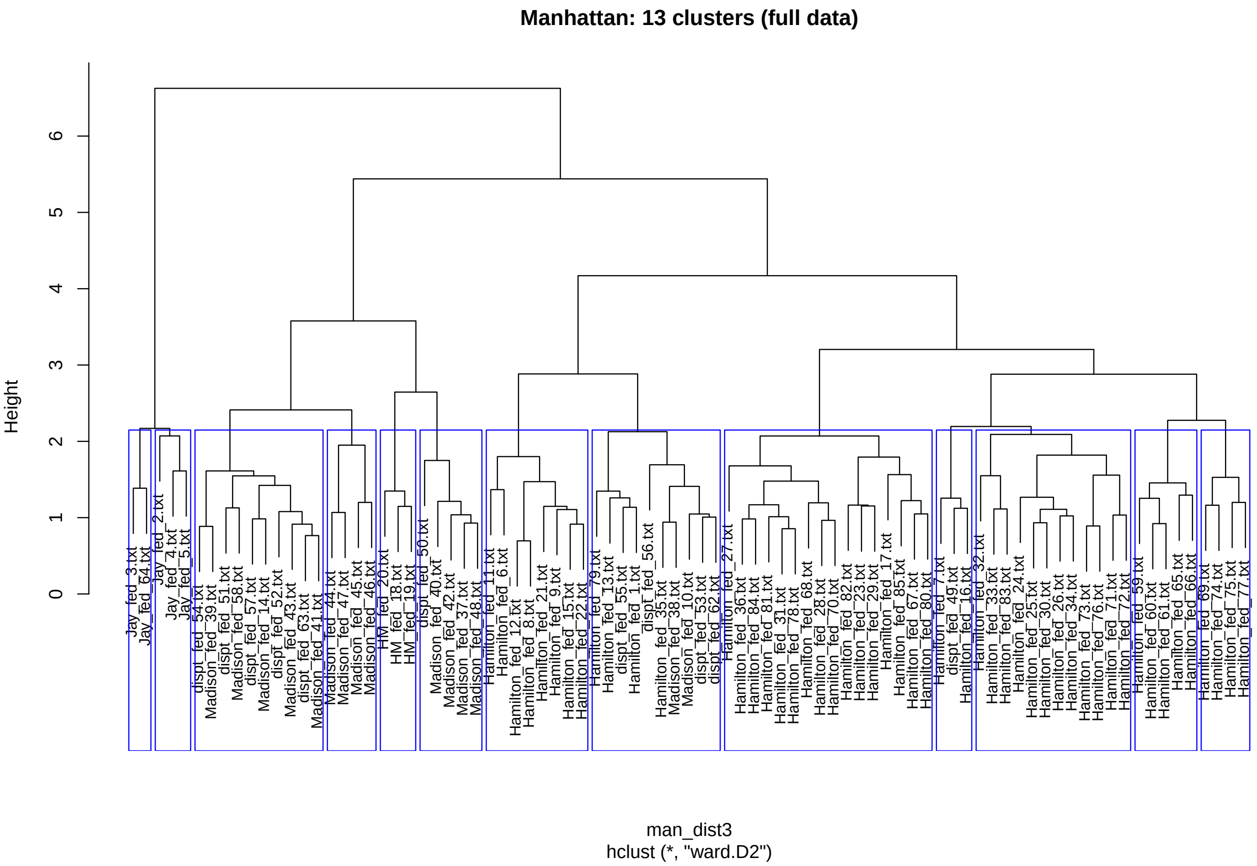
```
#cosine distance 13 clusters
plot(fit3, main = "Cosine: 13 clusters (best model)", cex = .8)
cosine_small13 <- cutree(fit3, k=13)
rect.hclust(fit3, k=13, border="red")
```



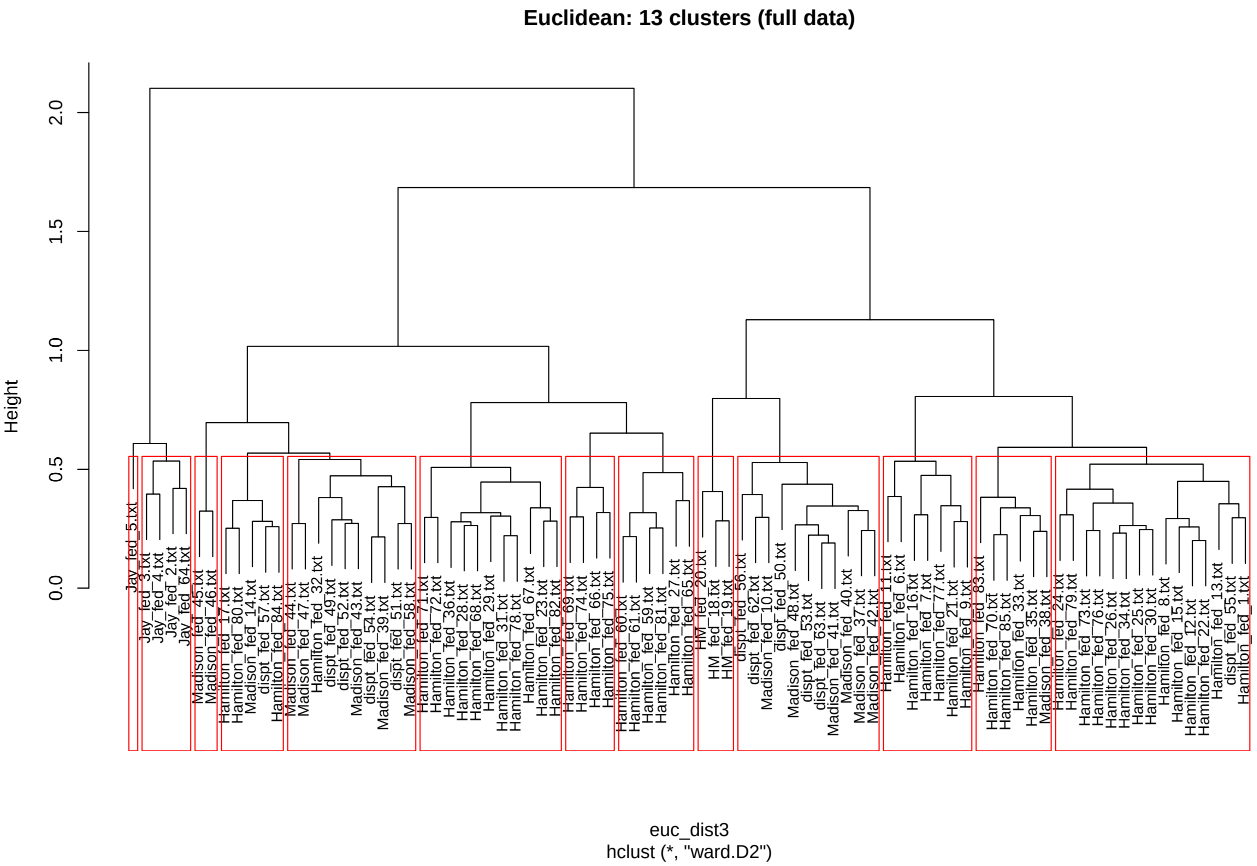
```
#cosine distance with full data 13 clusters
cos_dist_full <- as.dist(dist.matrix(as.matrix(papers[,3:ncol(papers)]), method = "cosine"))
fit6 <- hclust(cos_dist_full, method="ward.D2")
plot(fit6, main = "Cosine: 13 clusters (full data)", cex = .8)
cosine_full13 <- cutree(fit6, k=13)
rect.hclust(fit6, k=13, border="black")
```



```
#manhattan distance with full data 13 clusters
man_dist3 <- as.dist(dist.matrix(as.matrix(smallFrame[,3:ncol(smallFrame)]), method = "manhattan"))
fit7 <- hclust(man_dist3, method="ward.D2")
plot(fit7, main = "Manhattan: 13 clusters (full data)", cex = .8)
man_small13 <- cutree(fit7, k=13)
rect.hclust(fit7, k=13, border="blue")
```



```
#euclidean distance with full data 13 clusters
euc_dist3 <- as.dist(dist.matrix(as.matrix(smallFrame[,3:ncol(smallFrame)]), method = "euclidean"))
fit8 <- hclust(euc_dist3, method="ward.D2")
plot(fit8, main = "Euclidean: 13 clusters (full data)", cex = .8)
euc_small13 <- cutree(fit8, k=13)
rect.hclust(fit8, k=13, border="red")
```



Here the results are all inputted into a data frame. The names of the clusters are replaced with the author who appears most in that cluster.

```
#load in clusters into a dataframe as columns
disputedResults <- data.frame(as.factor(euclid_small)
                              ,as.factor(manhat_small)
                              ,as.factor(cosine_small7)
                              ,as.factor(cosine_small10)
                              ,as.factor(cosine_small13)
                              ,as.factor(cosine_full13)
                              ,as.factor(man_small13)
                              ,as.factor(euc_small13)
                              )



#name the columns by distance and cluster size
colnames(disputedResults) <- c("euclid_small","manhat_small",
                               "cosine_small7","cosine_small10"
                               ,"cosine_small13","cosine_full13",
                               "man_small13","euc_small13")





#name cluster by name of the author who appears most in that cluster
# disputedResults[disputedResults$euclid_small=="1",] #madison
# disputedResults[disputedResults$euclid_small=="2",] #madison
```

```r
# disputedResults[disputedResults$euclid_small=="3",] #hamilton
# disputedResults[disputedResults$euclid_small=="4",] #hamilton
# disputedResults[disputedResults$euclid_small=="5",] #hamilton
# disputedResults[disputedResults$euclid_small=="6",] #ham_mad
# disputedResults[disputedResults$euclid_small=="7",] #jay
levels(disputedResults$euclid_small) <- c("madison", "madison",
                                          "hamilton","hamilton","hamilton"
                                          ,"ham_mad","jay")


#name cluster by name of the author who appears most in that cluster
# disputedResults[disputedResults$manhat_small=="1",] #hamilton
# disputedResults[disputedResults$manhat_small=="2",] #madison
# disputedResults[disputedResults$manhat_small=="3",] #madison
# disputedResults[disputedResults$manhat_small=="4",] #hamilton
# disputedResults[disputedResults$manhat_small=="5",] #hamilton
# disputedResults[disputedResults$manhat_small=="6",] #hamilton
# disputedResults[disputedResults$manhat_small=="7",] #jay
levels(disputedResults$manhat_small) <- c("hamilton", "madison", "madison"
                                          ,"hamilton","hamilton","hamilton"
                                          ,"jay")


#name cluster by name of the author who appears most in that cluster
# disputedResults[disputedResults$cosine_small7=="1",] #madison
# disputedResults[disputedResults$cosine_small7=="2",] #hamilton
# disputedResults[disputedResults$cosine_small7=="3",] #hamilton
# disputedResults[disputedResults$cosine_small7=="4",] #hamilton
# disputedResults[disputedResults$cosine_small7=="5",] #hamilton
# disputedResults[disputedResults$cosine_small7=="6",] #ham_mad
# disputedResults[disputedResults$cosine_small7=="7",] #jay
levels(disputedResults$cosine_small7) <- c("madison", "hamilton",
                                           "hamilton","hamilton",
                                           "hamilton","ham_mad","jay")


#name cluster by name of the author who appears most in that cluster
# disputedResults[disputedResults$cosine_small10=="1",] #madison
# disputedResults[disputedResults$cosine_small10=="2",] #hamilton
# disputedResults[disputedResults$cosine_small10=="3",] #hamilton
# disputedResults[disputedResults$cosine_small10=="4",] #hamilton
# disputedResults[disputedResults$cosine_small10=="5",] #hamilton
# disputedResults[disputedResults$cosine_small10=="6",] #hamilton
# disputedResults[disputedResults$cosine_small10=="7",] #hamilton
# disputedResults[disputedResults$cosine_small10=="8",] #ham_mad
# disputedResults[disputedResults$cosine_small10=="9",] #jay
# disputedResults[disputedResults$cosine_small10=="10",] #jay
levels(disputedResults$cosine_small10) <- c("madison", "hamilton",
                                            "hamilton","hamilton",
                                            "hamilton","hamilton"
                                            ,"hamilton","ham_mad",
                                            "jay","jay")
```

```
#name cluster by name of the author who appears most in that cluster
# disputedResults[disputedResults$cosine_small13=="1",] #madison
# disputedResults[disputedResults$cosine_small13=="2",] #madison
# disputedResults[disputedResults$cosine_small13=="3",] #hamilton
# disputedResults[disputedResults$cosine_small13=="4",] #hamilton
# disputedResults[disputedResults$cosine_small13=="5",] #hamilton
# disputedResults[disputedResults$cosine_small13=="6",] #hamilton
# disputedResults[disputedResults$cosine_small13=="7",] #hamilton
# disputedResults[disputedResults$cosine_small13=="8",] #hamilton
# disputedResults[disputedResults$cosine_small13=="9",] #ham_mad
# disputedResults[disputedResults$cosine_small13=="10",] #jay
# disputedResults[disputedResults$cosine_small13=="11",] #jay
# disputedResults[disputedResults$cosine_small13=="12",] #jay
# disputedResults[disputedResults$cosine_small13=="13",] #madison
levels(disputedResults$cosine_small13) <- c("madison","madison",
                                            "hamilton", "hamilton",
                                            "hamilton","hamilton",
                                            "hamilton","hamilton",
                                            "ham_mad","jay","jay",
                                            "jay","madison")



#name cluster by name of the author who appears most in that cluster
# disputedResults[disputedResults$cosine_full13=="1",] #madison
# disputedResults[disputedResults$cosine_full13=="2",] #madison
# disputedResults[disputedResults$cosine_full13=="3",] #madison
# disputedResults[disputedResults$cosine_full13=="4",] #hamilton
# disputedResults[disputedResults$cosine_full13=="5",] #hamilton
# disputedResults[disputedResults$cosine_full13=="6",] #hamilton
# disputedResults[disputedResults$cosine_full13=="7",] #hamilton
# disputedResults[disputedResults$cosine_full13=="8",] #hamilton
# disputedResults[disputedResults$cosine_full13=="9",] #hamilton
# disputedResults[disputedResults$cosine_full13=="10",] #ham_mad
# disputedResults[disputedResults$cosine_full13=="11",] #jay
# disputedResults[disputedResults$cosine_full13=="12",] #jay
# disputedResults[disputedResults$cosine_full13=="13",] #jay
levels(disputedResults$cosine_full13) <- c("madison","madison", "madison",
                                           "hamilton","hamilton","hamilton"
                                           ,"hamilton","hamilton","hamilton"
                                           ,"ham_mad","jay","jay","jay")



#name cluster by name of the author who appears most in that cluster
# disputedResults[disputedResults$man_small13=="1",] #hamilton
# disputedResults[disputedResults$man_small13=="2",] #madison
# disputedResults[disputedResults$man_small13=="3",] #madison
# disputedResults[disputedResults$man_small13=="4",] #hamilton
# disputedResults[disputedResults$man_small13=="5",] #hamilton
# disputedResults[disputedResults$man_small13=="6",] #hamilton
# disputedResults[disputedResults$man_small13=="7",] #hamilton
# disputedResults[disputedResults$man_small13=="8",] #hamilton
# disputedResults[disputedResults$man_small13=="9",] #hamilton
# disputedResults[disputedResults$man_small13=="10",] #ham_mad
# disputedResults[disputedResults$man_small13=="11",] #jay
# disputedResults[disputedResults$man_small13=="12",] #jay
# disputedResults[disputedResults$man_small13=="13",] #madison
levels(disputedResults$man_small13) <- c("hamilton","madison", "madison",
                                         "hamilton","hamilton","hamilton"
                                         ,"hamilton","hamilton","hamilton"
                                         ,"ham_mad","jay","jay","madison")


#name cluster by name of the author who appears most in that cluster
# disputedResults[disputedResults$euc_small13=="1",] #madison
# disputedResults[disputedResults$euc_small13=="2",] #madison
# disputedResults[disputedResults$euc_small13=="3",] #hamilton
# disputedResults[disputedResults$euc_small13=="4",] #hamilton
# disputedResults[disputedResults$euc_small13=="5",] #hamilton
# disputedResults[disputedResults$euc_small13=="6",] #hamilton
# disputedResults[disputedResults$euc_small13=="7",] #hamilton
# disputedResults[disputedResults$euc_small13=="8",] #hamilton
# disputedResults[disputedResults$euc_small13=="9",] #hamilton
# disputedResults[disputedResults$euc_small13=="10",] #ham_mad
# disputedResults[disputedResults$euc_small13=="11",] #jay
# disputedResults[disputedResults$euc_small13=="12",] #jay
# disputedResults[disputedResults$euc_small13=="13",] #madison
levels(disputedResults$euc_small13) <- c("madison","madison", "hamilton",
                                         "hamilton","hamilton","hamilton"
                                         ,"hamilton","hamilton","hamilton"
                                         ,"ham_mad","jay","jay","madison")
```

Rename the columns so it is understandable which distance measure, cluster size, and data was used.

```
#make the column names understandable
colnames(disputedResults) <- c("7_euclid_small"
                               ,"7_man_small"
                               ,"7_cosine_small"
                               ,"10_cosine_small"
                               ,"13_cosine_small"
                               ,"13_cosine_full"
                               ,"13_man_full"
                               ,"13_euclid_full")
```

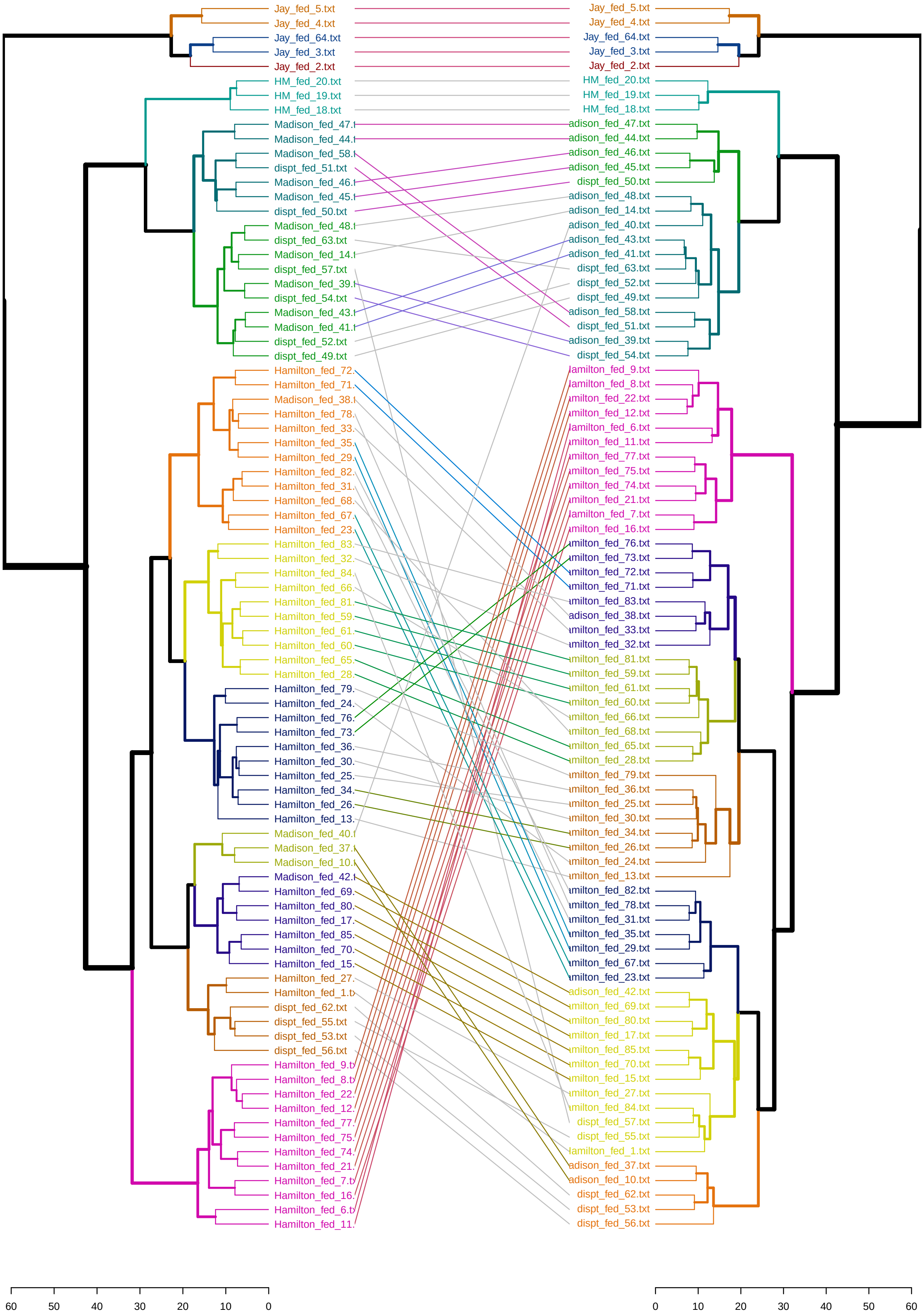
Examining the first 11 rows of the data frame show how the disputed papers were clustered.

```
disputedResults[1:11,]
```

```
##                  7_euclid_small 7_man_small 7_cosine_small 10_cosine_small
## dispt_fed_49.txt        madison    hamilton        madison         madison
## dispt_fed_50.txt        madison     madison        madison         madison
## dispt_fed_51.txt        madison     madison        madison         madison
## dispt_fed_52.txt        madison     madison        madison         madison
## dispt_fed_53.txt        madison    hamilton       hamilton        hamilton
## dispt_fed_54.txt        madison     madison        madison         madison
## dispt_fed_55.txt       hamilton    hamilton       hamilton        hamilton
## dispt_fed_56.txt        madison    hamilton       hamilton        hamilton
## dispt_fed_57.txt        madison     madison        madison         madison
## dispt_fed_62.txt        madison    hamilton       hamilton        hamilton
## dispt_fed_63.txt        madison     madison        madison         madison
##                  13_cosine_small 13_cosine_full 13_man_full 13_euclid_full
## dispt_fed_49.txt         madison        madison    hamilton        madison
## dispt_fed_50.txt         madison        madison     madison        madison
## dispt_fed_51.txt         madison        madison     madison        madison
## dispt_fed_52.txt         madison        madison     madison        madison
## dispt_fed_53.txt        hamilton        madison    hamilton        madison
## dispt_fed_54.txt         madison        madison     madison        madison
## dispt_fed_55.txt        hamilton       hamilton    hamilton       hamilton
## dispt_fed_56.txt        hamilton        madison    hamilton        madison
## dispt_fed_57.txt         madison       hamilton     madison       hamilton
## dispt_fed_62.txt        hamilton        madison    hamilton        madison
## dispt_fed_63.txt         madison        madison     madison        madison
```

This visualization makes a direct comparison of the dendogram for two HAC results. Both dendograms use cosine data but the dendogram on the left uses the smaller data frame compared to the one on the right with the full data. While some observations are clustered the same, a lot are not.

```r
# Make 2 dendrograms, using 2 different clustering methods
fit_3 <-  as.dendrogram(fit3)
fit_6 <- as.dendrogram(fit6)
colors <- randomColor(13, luminosity="dark")

# Custom these kendo, and place them in a list
compareCosineData <- dendlist(
  fit_3 %>%
    set("labels_col", value = colors, k=13) %>%
    set("branches_lty", 1) %>%
    set("branches_k_color", value = colors, k = 13),
  fit_6 %>%
    set("labels_col", value = colors, k=13) %>%
    set("branches_lty", 1) %>%
    set("branches_k_color", value = colors, k = 13)
)

# Plot them together
tanglegram(compareCosineData, sort = TRUE,
           common_subtrees_color_lines = TRUE, highlight_distinct_edges  = TRUE
           , highlight_branches_lwd=TRUE,
           margin_inner=7,
           lwd=1
)
```


Jay_fed_5.txt
Jay_fed_4.txt
Jay_fed_64.txt
Jay_fed_3.txt
Jay_fed_2.txt
HM_fed_20.txt
HM_fed_19.txt
HM_fed_18.txt
Madison_fed_47.t
Madison_fed_44.t
Madison_fed_58.t
dispt_fed_51.txt
Madison_fed_46.t
Madison_fed_45.t
dispt_fed_50.txt
Madison_fed_48.t
dispt_fed_63.txt
Madison_fed_14.t
dispt_fed_57.txt
Madison_fed_39.t
dispt_fed_54.txt
Madison_fed_43.t
Madison_fed_41.t
dispt_fed_52.txt
dispt_fed_49.txt
Hamilton_fed_72.
Hamilton_fed_71.
Madison_fed_38.t
Hamilton_fed_78.
Hamilton_fed_33.
Hamilton_fed_35.
Hamilton_fed_29.
Hamilton_fed_82.
Hamilton_fed_31.
Hamilton_fed_68.
Hamilton_fed_67.
Hamilton_fed_23.
Hamilton_fed_83.
Hamilton_fed_32.
Hamilton_fed_84.
Hamilton_fed_66.
Hamilton_fed_81.
Hamilton_fed_59.
Hamilton_fed_61.
Hamilton_fed_60.
Hamilton_fed_65.
Hamilton_fed_28.
Hamilton_fed_79.
Hamilton_fed_24.
Hamilton_fed_76.
Hamilton_fed_73.
Hamilton_fed_36.
Hamilton_fed_30.
Hamilton_fed_25.
Hamilton_fed_34.
Hamilton_fed_26.
Hamilton_fed_13.
Madison_fed_40.t
Madison_fed_37.t
Madison_fed_10.t
Madison_fed_42.t
Hamilton_fed_69.
Hamilton_fed_80.
Hamilton_fed_17.
Hamilton_fed_85.
Hamilton_fed_70.
Hamilton_fed_15.
Hamilton_fed_27.
Hamilton_fed_1.t
dispt_fed_62.txt
dispt_fed_55.txt
dispt_fed_53.txt
dispt_fed_56.txt
Hamilton_fed_9.t
Hamilton_fed_8.t
Hamilton_fed_22.
Hamilton_fed_12.
Hamilton_fed_77.
Hamilton_fed_75.
Hamilton_fed_74.
Hamilton_fed_21.
Hamilton_fed_7.t
Hamilton_fed_16.
Hamilton_fed_6.t
Hamilton_fed_11.
Jay_fed_5.txt
Jay_fed_4.txt
Jay_fed_64.txt
Jay_fed_3.txt
Jay_fed_2.txt
HM_fed_20.txt
HM_fed_19.txt
HM_fed_18.txt
adison_fed_47.txt
adison_fed_44.txt
adison_fed_46.txt
adison_fed_45.txt
dispt_fed_50.txt
adison_fed_48.txt
adison_fed_14.txt
adison_fed_40.txt
adison_fed_43.txt
adison_fed_41.txt
dispt_fed_63.txt
dispt_fed_52.txt
dispt_fed_49.txt
adison_fed_58.txt
dispt_fed_51.txt
adison_fed_39.txt
dispt_fed_54.txt
amilton_fed_9.txt
amilton_fed_8.txt
milton_fed_22.txt
milton_fed_12.txt
amilton_fed_6.txt
milton_fed_11.txt
milton_fed_77.txt
milton_fed_75.txt
milton_fed_74.txt
milton_fed_21.txt
amilton_fed_7.txt
milton_fed_16.txt
milton_fed_76.txt
milton_fed_73.txt
milton_fed_72.txt
milton_fed_71.txt
milton_fed_83.txt
adison_fed_38.txt
milton_fed_33.txt
milton_fed_32.txt
milton_fed_81.txt
milton_fed_59.txt
milton_fed_61.txt
milton_fed_60.txt
milton_fed_66.txt
milton_fed_68.txt
milton_fed_65.txt
milton_fed_28.txt
milton_fed_79.txt
milton_fed_36.txt
milton_fed_25.txt
milton_fed_30.txt
milton_fed_34.txt
milton_fed_26.txt
milton_fed_24.txt
milton_fed_13.txt
milton_fed_82.txt
milton_fed_78.txt
milton_fed_31.txt
milton_fed_35.txt
milton_fed_29.txt
milton_fed_67.txt
milton_fed_23.txt
adison_fed_42.txt
milton_fed_69.txt
milton_fed_80.txt
milton_fed_17.txt
milton_fed_85.txt
milton_fed_70.txt
milton_fed_15.txt
milton_fed_27.txt
milton_fed_84.txt
dispt_fed_57.txt
dispt_fed_55.txt
amilton_fed_1.txt
adison_fed_37.txt
adison_fed_10.txt
dispt_fed_62.txt
dispt_fed_53.txt
dispt_fed_56.txt
60
50
40
30
20
10
0
0
10
20
30
40
50
60

Decision Trees

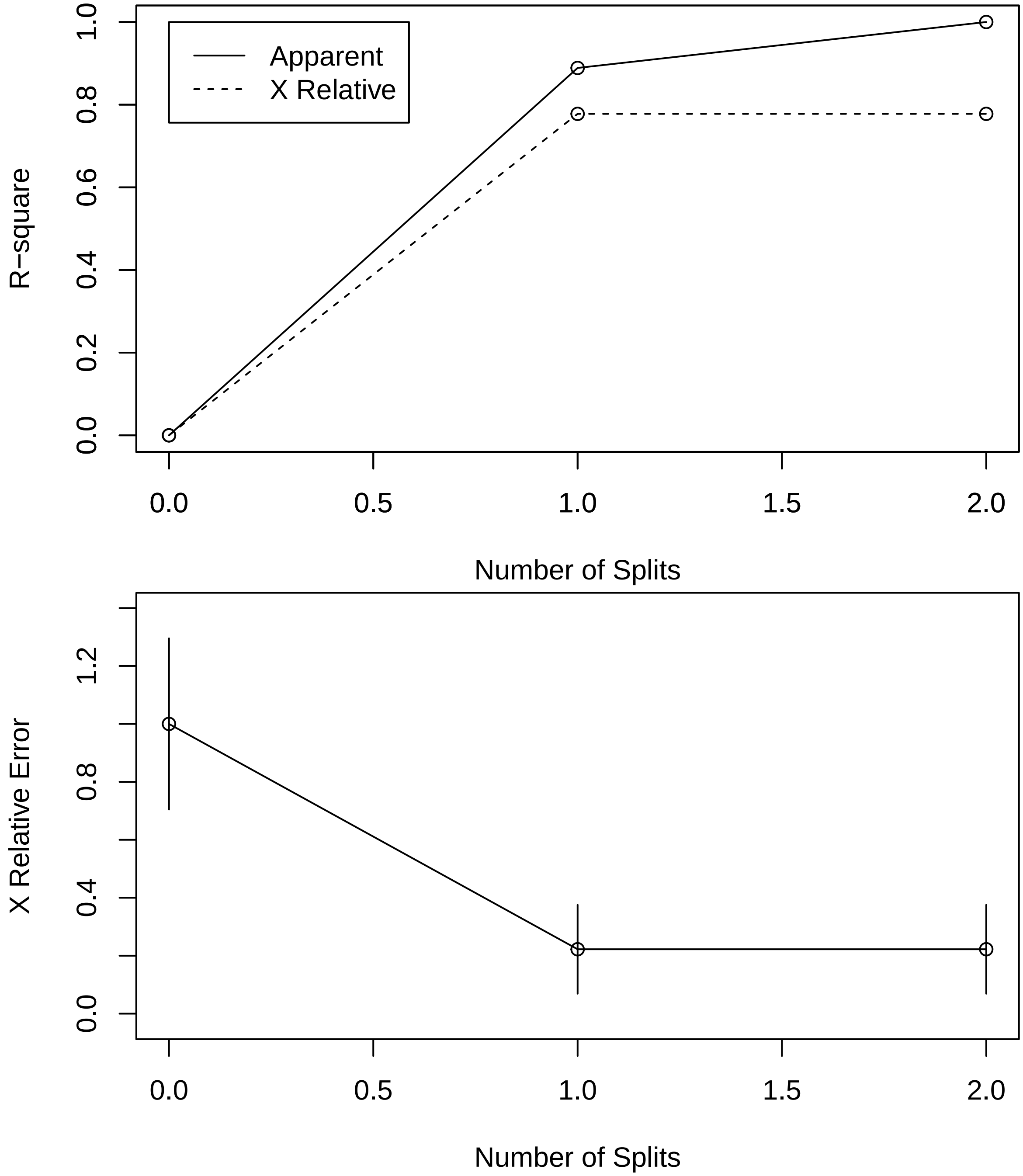
A decision tree is similar to asking a series of questions about an observation. Once the end of the tree is reached there is a classification result. Unseen observations by the model are passed in and the result is a prediction. The splits or questions asked are decided by node impurity.

First, a fully unpruned tree is built and tested. Then, a pruned tree is built. The pruning on the second tree is done automatically with rpart's internal optimization based on cross validation.

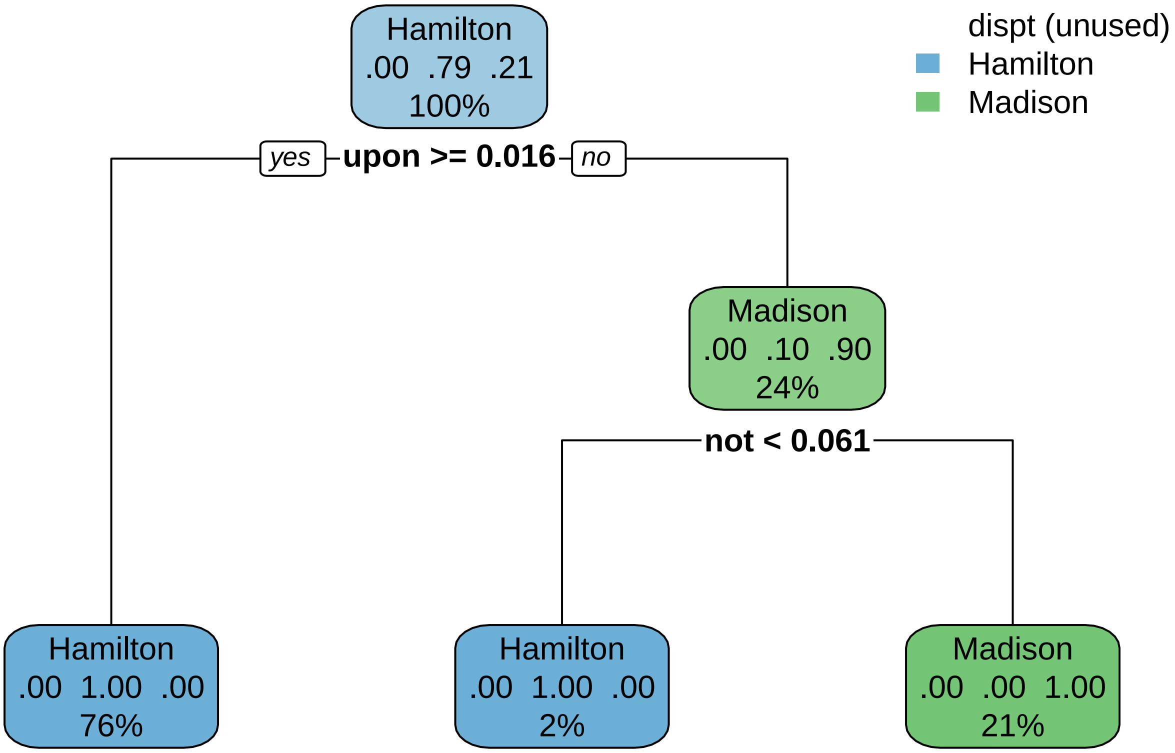
```
#fully unpruned
tree_model1 <- rpart(author ~ . , data = treeData_small_full_noD[indexes,]
                     , method = 'class'
                     , control = rpart.control(minbucket = 1, minsplit=1, cp=-1)
                     , model = T
                     )

rsq.rpart(tree_model1)
```

```
##
## Classification tree:
## rpart(formula = author ~ ., data = treeData_small_full_noD[indexes,
##     ], method = "class", model = T, control = rpart.control(minbucket = 1,
##     minsplit = 1, cp = -1))
##
## Variables actually used in tree construction:
## [1] not  upon
##
## Root node error: 9/42 = 0.21429
##
## n= 42
##
##         CP nsplit rel error  xerror    xstd
## 1  0.88889      0   1.00000 1.00000 0.29547
## 2  0.11111      1   0.11111 0.22222 0.15335
## 3 -1.00000      2   0.00000 0.22222 0.15335
```

```
## Warning in rsq.rpart(tree_model1): may not be applicable for this method
```



```
rpart.plot(tree_model1)
```

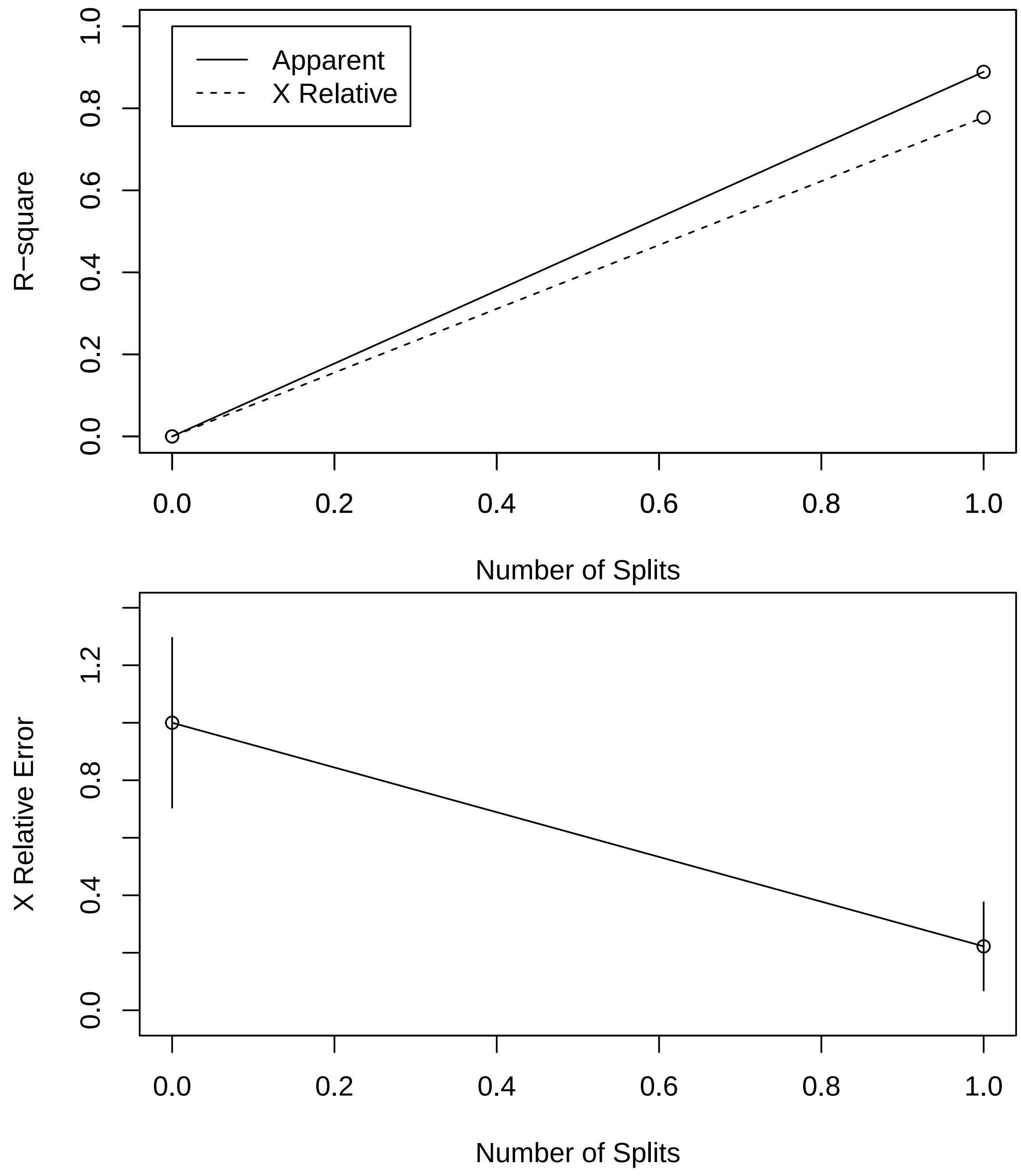


```
preds1 <- predict(tree_model1, treeData_small_full_noD[-indexes,], type = 'class')
table(treeData_small_full_noD$author[-indexes],preds1)
```

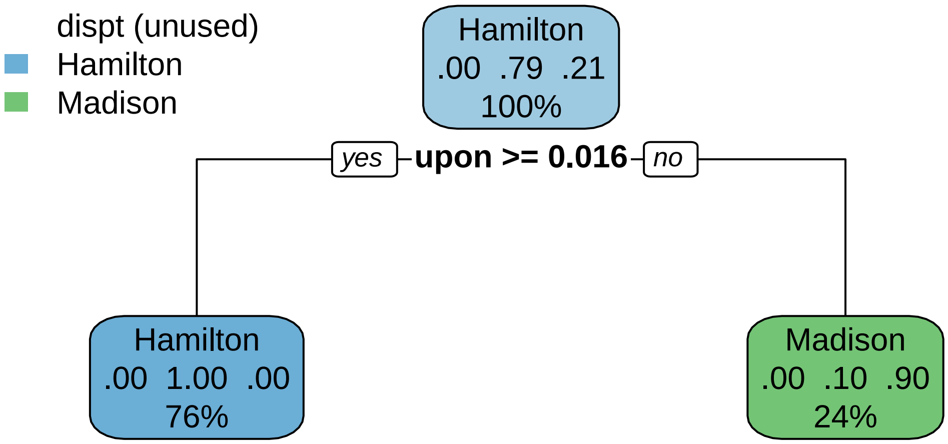
```
##           preds1
##             dispt Hamilton Madison
##   dispt         0        0       0
##   Hamilton      0       18       0
##   Madison       0        1       5
```

```
#fully pruned
tree_model3 <- rpart(author ~ . , data = treeData_small_full_noD[indexes,], method = 'class', model = T

rsq.rpart(tree_model3)
```

```
##
## Classification tree:
## rpart(formula = author ~ ., data = treeData_small_full_noD[indexes,
##     ], method = "class", model = TRUE)
##
## Variables actually used in tree construction:
## [1] upon
##
## Root node error: 9/42 = 0.21429
##
## n= 42
##
##         CP nsplit rel error  xerror    xstd
## 1 0.88889      0   1.00000 1.00000 0.29547
## 2 0.01000      1   0.11111 0.22222 0.15335
## Warning in rsq.rpart(tree_model3): may not be applicable for this method
```



```r
rpart.plot(tree_model3)
```



```
preds3 <- predict(tree_model3, treeData_small_full_noD[-indexes,], type = 'class')
table(treeData_small_full_noD$author[-indexes],preds3)
```

```
##           preds3
##            dispt Hamilton Madison
##   dispt        0        0       0
##   Hamilton     0       18       0
##   Madison      0        1       5
```

```
#Tree predictions


tree_model1 <- rpart(author ~ . , data = treeData_small_full_noD
                     , method = 'class'
                     , control = rpart.control(minbucket = 1, minsplit=1, cp=-1)
                     , model = T
                     )


tree_model3 <- rpart(author ~ . , data = treeData_small_full_noD, method = 'class', model = TRUE)



predict(tree_model1, treeData_small_disputed, type = 'class')
```

```
##       1       2       3       4       5       6       7       8       9      10
## Madison Madison Madison Madison Madison Madison Madison Madison Madison Madison
##      11
## Madison
## Levels: dispt Hamilton Madison
```

```
predict(tree_model3, treeData_small_disputed, type = 'class')
```

```
##       1       2       3       4       5       6       7       8       9      10
## Madison Madison Madison Madison Madison Madison Madison Madison Madison Madison
##      11
## Madison
## Levels: dispt Hamilton Madison
```

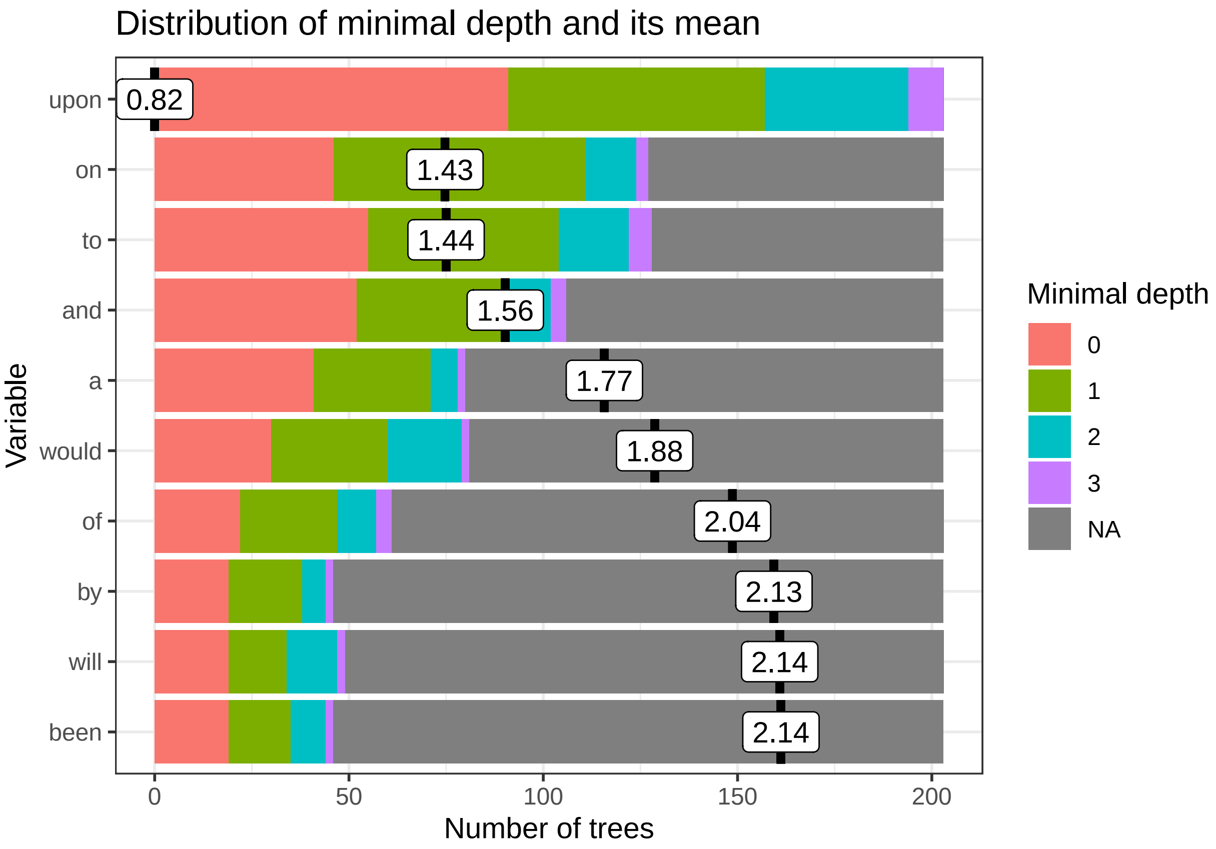
Random Forest

A random forest is an ensemble method that uses many randomly selected trees. It is classification by committee of trees.

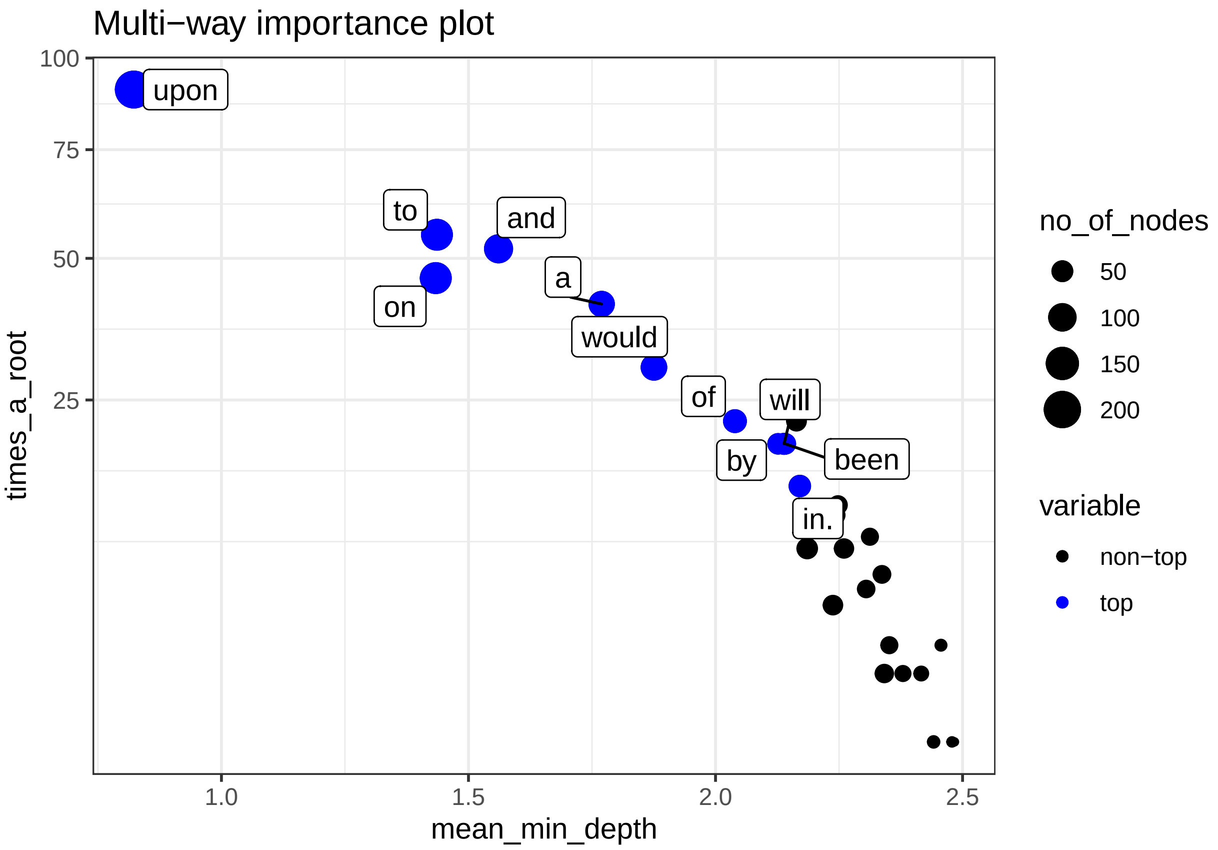
```
treeData_small_full_noD <- droplevels(treeData_small_full_noD)
rf1 <- randomForest(treeData_small_full_noD[indexes,-ncol(treeData_small_full_noD)]
                    ,treeData_small_full_noD[indexes,ncol(treeData_small_full_noD)]
                    ,localImp = TRUE)

preds4 <- predict(rf1, treeData_small_full_noD[-indexes,-ncol(treeData_small_full_noD)], type = ’class’
table(treeData_small_full_noD$author[-indexes],preds4)
```

```
##           preds4
##            Hamilton Madison
##   Hamilton       18       0
##   Madison         1       5
```

```
plot_min_depth_distribution(min_depth_distribution(rf1))
```



```
plot_multi_way_importance(measure_importance(rf1), size_measure = "no_of_nodes")
```



```
rf1 <- randomForest(treeData_small_full_noD[,-ncol(treeData_small_full_noD)]
                    ,treeData_small_full_noD[,ncol(treeData_small_full_noD)]
                    ,localImp = TRUE)
predict(rf1, treeData_small_disputed[,-ncol(treeData_small_disputed)], type = ’class’)
```

```
##        1        2        3        4        5        6        7        8
##  Madison  Madison  Madison  Madison  Madison  Madison Hamilton  Madison
##        9       10       11
##  Madison  Madison  Madison
## Levels: Hamilton Madison
```

Results

Preliminary analysis using the models highlighted a very important feature. Even with 7 clusters, k-means was able to correctly distinguish papers written by Jay. One of the clusters was exclusively Jay. Furthermore, other clusters were primarily Hamilton or Madison. Despite this, there were many instances of Hamilton, Madison, and their joint works being clustered together. These issues are apparent in the visualizations for k-means. Jay, and the joint Hamilton and Madison papers seemed to be easily distinguishable. However, past that k-means tended to have very overlapping results with its clustering. When deciding how many clusters to try and use the elbow method was used. By looking at the path the visualization laid out, various elbows were selected. These were 7, 10, and 13 for the number of clusters.

The nature of the data is suited better for hierarchical clustering. The majority of the analysis was driven by hierarchical clustering. Unlike k-means, euclidean and cosine distance with 7 clusters were able to distinguish between Jay and Hamilton-Madison clusters from the rest. Each paper in those categories was clustered together with only 7 clusters. This was not true for manhattan distance. In the case of manhattan distance and 7 clusters Hamilton-Madison papers were clustered with Madison papers. When increasing to 10 clusters, more specific splits are also made and improvement is seen. For example, with all distance measures, there are clusters with almost half Madison and half Hamilton papers. Increasing to 13 clusters with all distance measures split those clusters. These outcomes still need to be compared to those of the identical model parameters being used on the full data.

To more easily make comparisons, the different clustering results are put into a data frame. Each row of the data frame is a different paper. Each column is the clustering result by a different model. These include 7 clusters euclidean, 7 clusters manhattan, 7 clusters cosine, 10 clusters cosine, and 13 clusters cosine. All of the above mentioned are done using the smaller data frame based on variance. It is a smaller data set only containing 40% of the words from the original data. This data frame also includes 13 clusters cosine, 13 clusters manhattan, and 13 clusters euclidean all with the full data. The clusters in each column are renamed as the name of the author who appears most in that cluster.

All of the 8 hierarchical clustering models agreed that Madison wrote papers 50,51,52,54, and 63. Furthermore, all 8 models agree that Hamilton wrote paper 55. This leaves 5 more disputed papers that were not consistently clustered across the 8 models. Paper 49 sheds light on varying distance measures. Both euclidean and cosine distance, for all k amounts and data used, clustered paper 49 in a Madison majority cluster. Both manhattan distance models, one with the smaller data frame and one without both placed paper 49 in a Hamilton majority cluster. Paper 53 differed across distance measure and data used. Euclidean distance measures, irrespective of data used, always clustered paper 53 with Madison papers. The most interesting and accurate model uses cosine distance with 13 clusters on the smaller data frame. For paper 53, cosine distance with 13 clusters on the smaller data frame clusters this observation with Hamilton, while the same model on the full data frame clusters the observation with Madison papers.

When focusing in on the best model, 13 clusters on the smaller data frame, it becomes apparent how powerful the results are. Only two papers, both Hamilton, are clustered with Madison papers. Other than those observations, all groups were successfully clustered. This success is very encouraging when assessing the results of the disputed papers.

All trees and random forests are trained and tested with a 65% split of the data. However, for making the predictions on the disputed papers the models are fully retrained on all of the training data. Both fully pruned and unpruned trees predicted that Madison wrote all 11 disputed papers. Although this may seem surprising, it actually coincides with the 13 cosine clustered HAC predictions. In that setting, 9/11 papers are written by Madison, and both decision trees predicted 11/11 papers are written by Madison.

Comparing the pruned and unpruned trees is done using internal cross validation with rpart. Xerror is the error with cross validation. One can easily see that while the error goes down all the way to zero in the unpruned tree, the xerror stops going down at .22222. This is where the tree should be pruned back to. It is exactly where rpart prunes the tree. It recognizes that using cross validation there is no more gain after a depth of 1 node, not including the root. When focusing in on the pruned tree, it correctly classified all papers on the testing data other than one. This one error predicted a Madison paper when the paper was written by Hamilton. This is important to keep in mind when further comparing the clustering to tree results. It would be surprising if all papers were actually written by Madison.

The random forest seems to have very promising results. It predicted that all papers were written by Madison except for the 7th disputed paper. This almost perfectly matches with the clustering approach. Both the random forest and the best clustering model agree on all disputed papers except for one. This is convincing evidence that very different models are capable of coming to similar conclusions.

One of the most important comparisons between the decision trees, random forests, and clustering is that the trees and forests give specific reasons as to why things are classified how they are. For the decision trees, the frequency of the word “upon” is the deciding factor between Madison and Hamilton. For the random forest, the visualization titled, “Distribution of minimal depth and its mean” from the random forest explainer package tells a similar story. It shows how the word “upon” was almost always chosen as the root. The next most important words were “on” and “and”. Furthermore, the visualization titled “Multi-way importance plot” shows that upon is the most important word followed by “on”, “and”, and “to”. These insights are possible with decision trees and random forests, this is a huge benefit in truly understanding the data set and problem.

Conclusion

Clustering is simply grouping like observations. This idea can be used to try and understand who may have written the disputed Federalist Papers. Technology and new ways to model the world are constantly evolving. These improvements may help shed light on historical mysteries. The Federalist Papers are some of the most fundamental works in framing what the USA was supposed to be. Solving a historical mystery with modern technology can be a great success and moment of reflection. However, what makes a good mystery is an inability to solve it.

To build on this, two very different approaches were used. Clustering, decision trees, and a random forest were all used to try and predict who wrote the disputed papers. Although results were not identical, for the best clustering and random forest model, the predictions were the same for 10/11 papers.

Scholars believe they have an idea as to who wrote some of the disputed papers, but not all. This was reflected in what was accomplished using clustering. A number of the disputed Federalist Papers seemed to have very definite results. However, some did not have clear results. Even if clustering is only successful at reinforcing understood ideas, this can be considered a success. Although a well built model can cluster disputed papers there is by nature no way to check the results. This is in part what makes this mystery so great.

Despite this, there were significant results. There was clear evidence that certain authors can be characterized by the words they chose. The fact that the methods in this report were able to accomplish this means that the results for the disputed papers should be considered viable. Pattern recognition in data is a task that is constantly improving and growing. Most problems, with creativity, can be framed in a way to leverage pattern recognition. The mystery of the Federalist Papers is no exception. Although the true authorships may never be known, there is clear evidence that at least some of the disputed papers can be correctly attributed with confidence.