HW4

Jose Reyes

7.28.2020

#### Introduction

In the words of the famous Alexander Hamilton, “Men give me credit for some genius… the effort that I have made is what people are pleased to call the fruit of genius.” In the curious case of the Federalist Papers, it is uncertain if credit can be given for the authorship of 11 essays. This authorship is disputed between two famous figures in U.S. history, Alexander Hamilton and James Madison. Which founding father should man give credit for this genius?

To identify the rightful author(s) of these 11 essays, a data mining technique called classification is conducted on all 85 Federalist Papers. The algorithm will identify which essays with known authorships most closely match the words used in the essays with disputed authorships. As a result, it can be predicted based on the classification results whether Alexander Hamilton or James Madison authored these 11 essays.

James Madison once said “Knowledge will forever govern ignorance... [people] must arm themselves with the power which knowledge gives.” In that spirit, this data mining technique will provide that knowledge and hopefully solve one of the great mysteries of United States history.

#### Analysis and Models

#### About the Data

Before starting the classification process, a data frame is created with information on each of the Federalist paper including each documents’ words and their frequencies. Of the 85 documents, 51 are known to be written by Hamilton, 15 by Madison, 5 by Jay, and 3 are by both Hamilton and Madison. The authorship of 11 of the documents are disputed.

The .txt files of all 85 Federalist Papers are loaded as a corpus, cleansed, and formed into a Document Term Matrix (DTM). To allow for proper analysis, the proper parameters were set for the DTM:

-Include words only between the length of 3 and 15 characters

-Remove “stop words” as they are very common and do not provide much value to the meaning of the documents

-Remove words that appear in less than 1% and more than 50% of the documents

### Load Data

FedPapersCorpus <- Corpus(DirSource("fedpapers"))  
FedPapersCorpus

## <<SimpleCorpus>>  
## Metadata: corpus specific: 1, document level (indexed): 0  
## Content: documents: 85

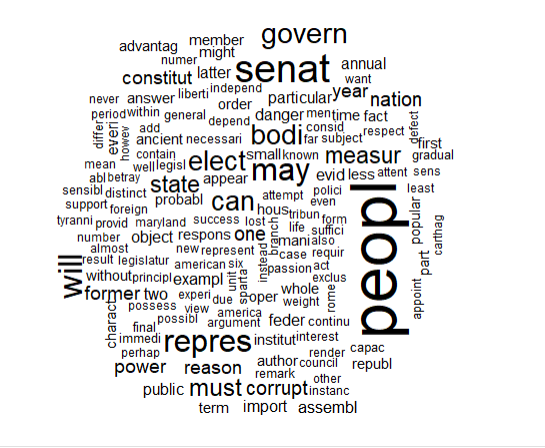
### Clean Data

#filter out "stop words" before processing of natural language data  
StopWords <- c("will", "one", "two", "may", "less", "well","might","withou","small","single",  
 "several","but","very","can","must","also","any","and","are","however","into",  
 "almost","can","for","add")  
  
STOPS <- stopwords('english')  
  
#Create Document Term Matrix  
#Remove Stop words  
#Remove words that appear less than 1% and more than 50% of the documents  
  
FedPapers <- DocumentTermMatrix(FedPapersCorpus, control = list(  
 stopwords = TRUE,  
 wordLengths=c(3,15),  
 removePunctuation = T,  
 removeNumbers = T,  
 tolower = T,  
 stemming = T,  
 remove\_separators = T,  
 stopwords = StopWords,  
 bound = list(global=c(.0085,85))  
 ))  
  
FedPapers <- as.matrix(FedPapers)

The DTM is now created which contains each Federalist Paper and the frequency of all words. The most common words across all 85 Papers can now be identified to take an initial look at the data.

### Explore Data

#Create World Cloud  
WC <- wordcloud(colnames(FedPapers),FedPapers[11,])



#View Most Common words  
WordFreq <- colSums(FedPapers)  
ord <- order(WordFreq)  
WordFreq[tail(ord)]

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

### As seen above, the most common words are state, will, govern, power, may, and ‘constitut’.

The data is then normalized to allow for better analysis and put into a data frame for classification.

### Normalization

#Normalization is beneficial for analysis  
N\_FedPapers <- apply(FedPapers, 1, function(i) round(i/sum(i),3))  
N\_FedPapers <- t(N\_FedPapers)

### Check data structure and convert to DF

N\_FedPapers <- as.data.frame(N\_FedPapers)

#### Classification

Now that the data has been prepared, classification can be performed to create clusters. By comparing the normalized frequencies of words used in all the documents, the algorithm will identify which documents are most closely related. It can thus be predicted based on the clusters who authored which documents.

The following clustering methods were performed:

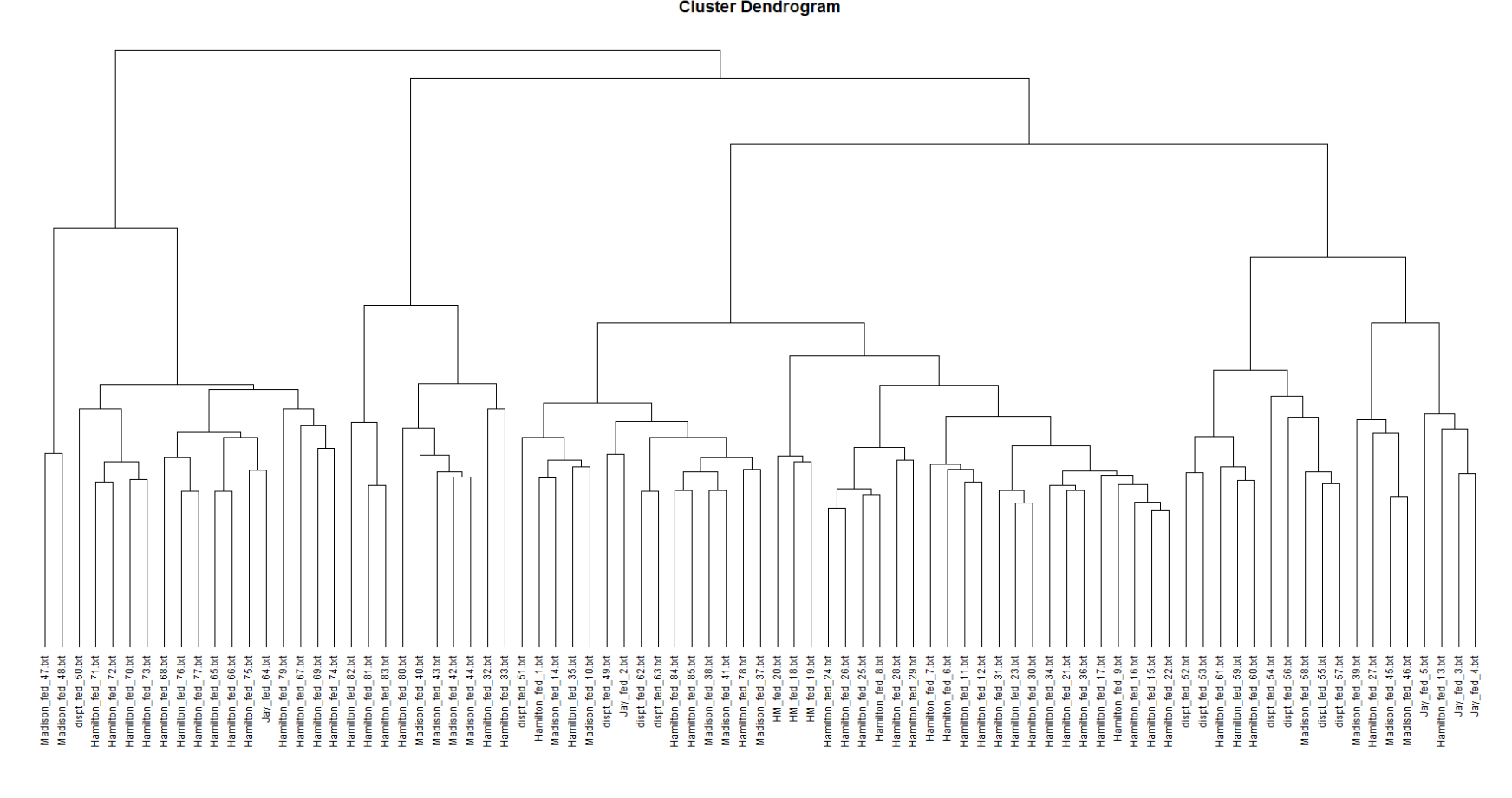
**Heirarchial clustering:** All data points start as their own cluster and are then merged to the nearest clusters. The clusters are continuously connected until a single cluster is created.

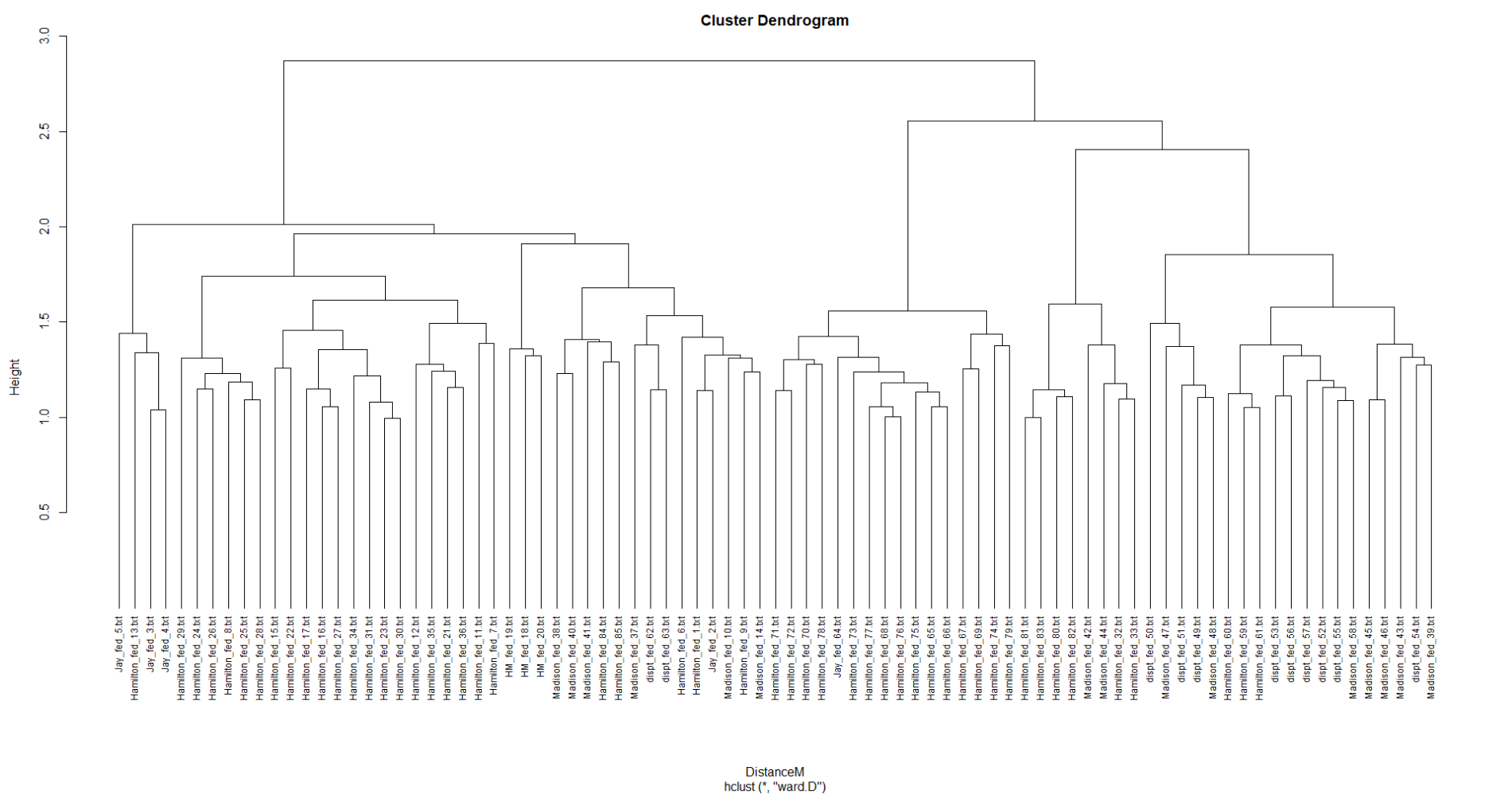
**K-means:** Data points are classified into clusters based on their proximity to the nearest ‘centroid’ or cluster centers. The number of centroids and their initial placement must first be defined prior to the process.

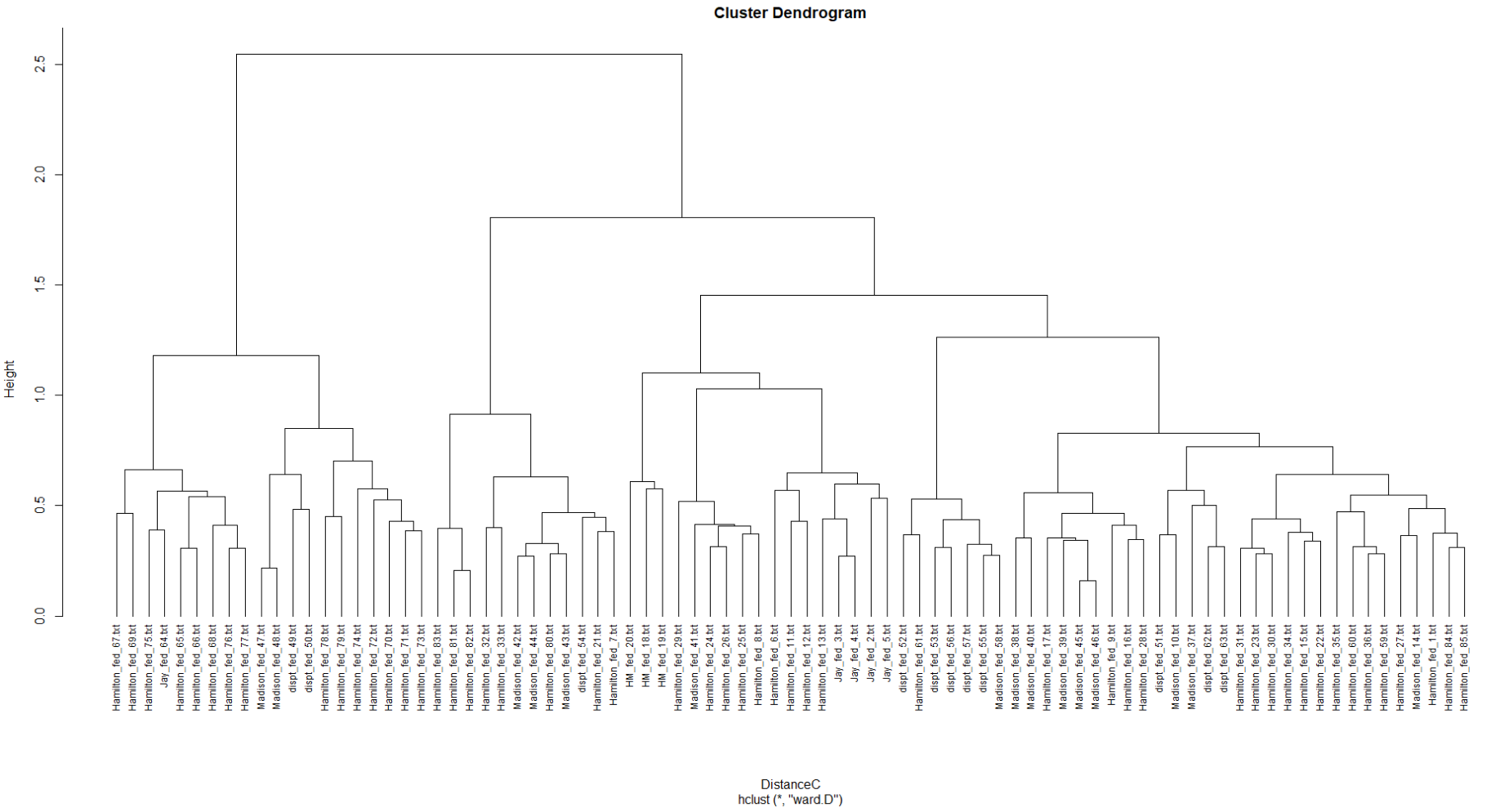
### Start HAC Analysis

The ‘distance’ between each document can be defined three ways: Euclidan, Manhattan, and cosine similarity. In order to identify the dendogram that provides the ‘clearest’ clustering results, all three methods were used to compare the results.

#Define different distance measures  
m=N\_FedPapers  
DistanceE <- dist(m, method="euclidean")  
DistanceM <- dist(m, method="manhattan")  
DistanceC <- dist(m, method="cosine")







All three methods provided different results. However, the dendogram using cosine similarity appears to have most correctly classified the articles written by Jay and both Hamilton and Madison while the others were more scattered. Across all three results, it appears that articles 52-57 seem very closely related.

### Start k-means Analysis

Because there are 4 different author types (Madison, Hamilton, Jay, Madison/Hamilton), 4 centers were chosen for the k-means analysis. The algorithm was also conducted with various other number of centers (ranging from 3-7), and none produced more consistent resu

Because k-means results can vary based on the setup and initial placement of centroids, three different k-means with different parameters were performed on the dataset.

#k-means clustering  
  
K1 <- kmeans(N\_FedPapers,centers = 4, nstart=100, iter.max = 50)  
K1Analysis <- N\_FedPapers  
K1Analysis$cluster <- K1$cluster

K2 <- kmeans(N\_FedPapers,centers = 4, nstart=125, iter.max = 50)  
K2Analysis <- N\_FedPapers  
K2Analysis$cluster <- K2$cluster

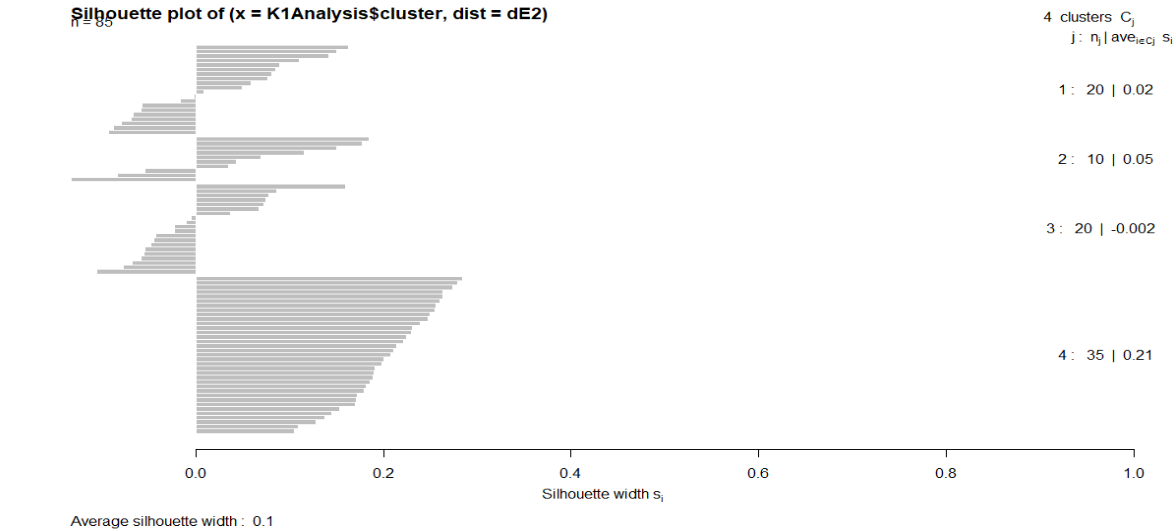
K3 <- kmeans(N\_FedPapers,centers = 4, nstart=150, iter.max = 50)  
K3Analysis <- N\_FedPapers  
K3Analysis$cluster <- K3$cluster

|  |  |  |  |
| --- | --- | --- | --- |
|  | K1 cluster | K2 cluster | K3 cluster |
| **dispt\_fed\_49.txt** | 4 | 1 | 3 |
| **dispt\_fed\_50.txt** | 4 | 1 | 3 |
| **dispt\_fed\_51.txt** | 2 | 3 | 1 |
| **dispt\_fed\_52.txt** | 2 | 3 | 1 |
| **dispt\_fed\_53.txt** | 2 | 3 | 1 |
| **dispt\_fed\_54.txt** | 2 | 3 | 1 |
| **dispt\_fed\_55.txt** | 2 | 3 | 1 |
| **dispt\_fed\_56.txt** | 2 | 3 | 1 |
| **dispt\_fed\_57.txt** | 2 | 3 | 1 |
| **dispt\_fed\_62.txt** | 2 | 3 | 1 |
| **dispt\_fed\_63.txt** | 2 | 3 | 1 |

There appears to be one singular theme that occurred across all three k-means analysis. Papers 49 and 50 are in one cluster while other 9 papers were in another cluster.

Papers 49 and 50 appear in a cluster with papers written mostly by Hamilton. However, the rest of the documents are in a separate cluster that appears to be a mix of Hamilton and Madison.

The following silhouette analysis provides a graphical representation of the first k-means results:



### Results

The following table provides a summary of results for each disputed Federalist Paper:

|  |  |  |
| --- | --- | --- |
| **Federalist Paper Number** | **HAC Prediction** | **K-Means Prediction** |
| 49 | Mostly Madison | Hamilton |
| 50 | Mostly Madison | Hamilton |
| 51 | Mostly Hamilton | Mostly Hamilton |
| 52 | Hamilton | Mostly Hamilton |
| 53 | Hamilton | Mostly Hamilton |
| 54 | Hamilton | Mostly Hamilton |
| 55 | Hamilton | Mostly Hamilton |
| 56 | Hamilton | Mostly Hamilton |
| 57 | Hamilton | Mostly Hamilton |
| 62 | Hamilton | Mostly Hamilton |
| 63 | Hamilton | Mostly Hamilton |

For Papers 49 and 50, the k-means algorithm classified them almost exclusively with other papers with Hamilton while HAC appeared mixed. Because of this, it can be predicted that they are *most likely*authored by Hamilton.

The other 9 papers are vague. The k-means classified these papers in a single cluster which is mixed with both Hamilton and Madison. However, the cluster contained slightly more Hamilton papers than Madison. The HAC provided similarly vague results, It is most likely that these 9 papers are a mix of Hamilton and Madison, but unfortunately the results did not provide a clear conclusion. Based on probability and the fact that the k-means cluster for these 9 papers contained slightly more Hamilton than Madison, there is a higher chance that these papers were authored by Hamilton.

#### Conclusion

To determine the author of 11 Federalist Papers, a data mining technique called classification was performed. This technique identified which documents are most similar to each other based on their normalized word frequency similarities.

Prior to classification, the .txt files of all 85 documents were loaded, cleansed, and loaded into a data frame to allow for classification analysis. Words that are common or rare were removed to allow for proper analysis. Two different classification algorithms, hierarchical and k-means classification, were performed on the dataset. Overall, both algorithms provided similarly vague results.

It can be concluded based on the classification results that Hamilton authored Federalist Papers 49 and 50. All 9 other papers may slightly more likely be authored by Hamilton, but the results did not provide clear results.

For future projects, it is recommend to try additional classification techniques or perhaps explore other methods such as neural networks.