# Disputed Federalist Papers: Clustering Models

Jason Tompkins

5/1/2021

## Introduction

The Library of Congress is the largest library in the world, and is tasked with preserving historic and culturally important books, documents, and media. As a source for knowledge and research it is vital to the Library of Congress that they have all the information available pertaining to the documents housed within its archives.

There are eleven documents among the Federalist Papers whose authorship is in dispute. Anyone who had first-hand knowledge of who wrote each of the eleven disputed Federalist Papers has passed away as they were written in 1787 and 1788. The texts themselves are the only evidence for there own authorship. For many years experts could only speculate on who wrote them based on writing style and drawing basic comparisons.The use of clustering data mining methods will be used to measure the similarities in work usage between the disputed papers and the ones with identified authors.

According to the Library of Congress, The Federalist Papers were a series of eighty-five essays urging the citizens of New York to ratify the new United States Constitution. Written by Alexander Hamilton, James Madison, and John Jay, the essays originally appeared anonymously in New York newspapers in 1787 and 1788 under the pen name "Publius." The Federalist Papers are considered one of the most important sources for interpreting and understanding the original intent of the Constitution.

## Loading and Preparing the Data

The data consists of 85 observations each representing a separate essay from the collection of essays known as the "Federalist Papers". There are 72 variables for each essay. The variables are summarized below.

Author: The author indicates the authorship of each paper. There are three acknowledged authors of the Federalist papers; Alexander Hamilton, John Jay, and James Madison. Three were written as a collaboration between Hamilton and Madison. The author of eleven of the essays are disputed.

Filename: The name of each text file that includes an indication of the authorship, the abbreviation "fed', and the sequential number of the essay (e.g. Hamilton_fed_1.txt, dispt_fed_50.txt)

a, all, also… some: Columns three through seventy-two list words that appear in the essays. The text files are assigned a factor for each word that indicates how prevalent that word is in the text.

```r
# Read in CSV
fedPapersRAW <- read.csv("~/rData/fedPapers85_fromClass.csv", sep=",")

# View the data
#View(fedPapersRAW)

# Check for missing values
sum(is.na(fedPapersRAW))

## [1] 0
```

## K-Means Clustering

K-Means clustering is method that places each observation in a multidimensional space and assigns each observation into a distinct cluster based on it's relation to the other observations.

The number of clusters is not determined by the algorithm and therefore must be entered as an input to the model. After a series of trials with different numbers. A model with 12 clusters created enough visual stratification to make some conclusions about authorship.

```r
# Create a subset that excludes the authors names
fedPapers_KM <- fedPapersRAW[,2:72]

# Reduce the dimensionality.. focus on signal and not noise
#fedPapers_KM <- select(fedPapers_KM, filename, upon, all, may, also, even, f
rom, shall, only)

# Make the names of each file an index for the rownames
rownames(fedPapers_KM) <- fedPapers_KM[,1]
fedPapers_KM[,1] <- NULL

#View(fedPapers_KM)

# Determine "optimal" number of clusters

# Set seed for fixed random seed
set.seed(20)

# run k-means
Clusters <- kmeans(fedPapers_KM, 12)
fedPapers_KM$Clusters <- as.factor(Clusters$cluster)

str(Clusters)

## List of 9
##  $ cluster     : Named int [1:85] 11 3 11 3 6 11 6 5 11 6 ...
##   ..- attr(*, "names")= chr [1:85] "dispt_fed_49.txt" "dispt_fed_50.txt" "
dispt_fed_51.txt" "dispt_fed_52.txt" ...
##  $ centers     : num [1:12, 1:70] 0.361 0.197 0.282 0.267 0.358 ...
```

```
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:12] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:70] "a" "all" "also" "an" ...
##  $ totss       : num 12.6
##  $ withinss    : num [1:12] 0.479 0.162 0.348 0.651 0.615 ...
##  $ tot.withinss: num 5.71
##  $ betweenss   : num 6.86
##  $ size        : int [1:12] 7 3 7 10 9 7 4 5 8 10 ...
##  $ iter        : int 3
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

```
Clusters$centers
```

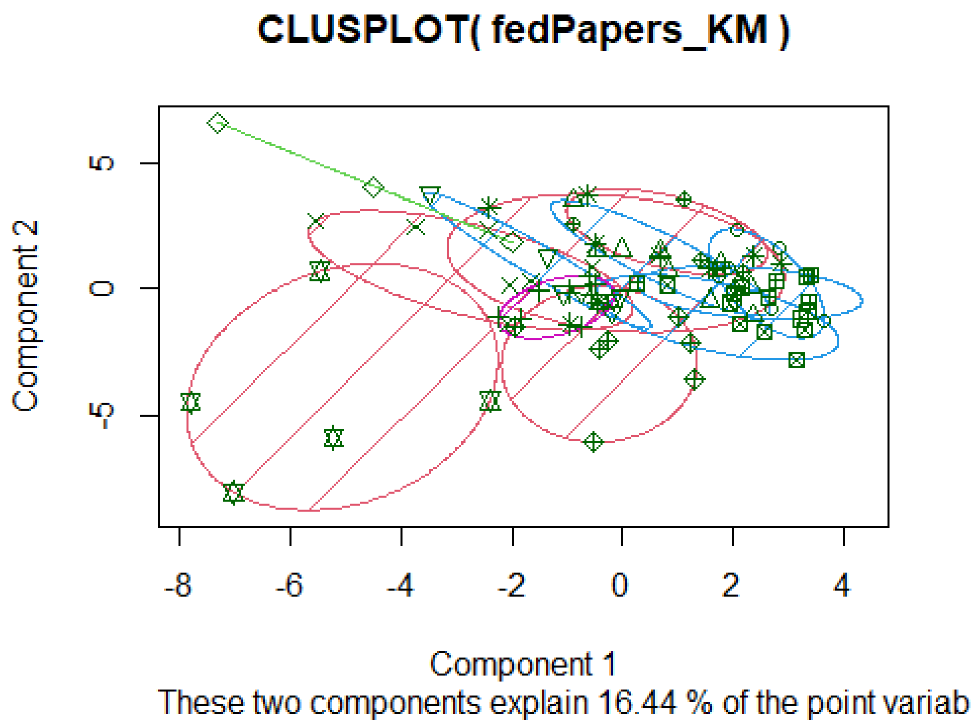
```
##            a        all        also          an       and         any
are
## 1  0.3607143 0.04157143 0.011714286 0.08614286 0.2598571 0.04285714 0.0547
1429
## 2  0.1973333 0.04200000 0.015666667 0.04766667 0.5113333 0.02666667 0.0883
3333
## 3  0.2818571 0.05428571 0.010714286 0.05671429 0.3430000 0.03971429 0.0567
1429
## 4  0.2672000 0.05950000 0.003400000 0.08550000 0.3413000 0.04850000 0.0846
0000
## 5  0.3583333 0.05833333 0.004444444 0.07100000 0.3911111 0.02733333 0.0878
8889
## 6  0.3234286 0.05057143 0.007857143 0.05014286 0.3585714 0.05757143 0.0947
1429
## 7  0.2505000 0.08425000 0.004000000 0.06550000 0.4370000 0.02550000 0.0627
5000
## 8  0.1598000 0.03600000 0.019800000 0.02520000 0.7152000 0.03760000 0.0852
0000
## 9  0.3372500 0.05162500 0.004750000 0.09612500 0.3341250 0.06725000 0.0742
5000
## 10 0.3218000 0.04620000 0.002800000 0.07980000 0.3271000 0.04910000 0.0590
0000
## 11 0.2465000 0.06225000 0.008375000 0.05200000 0.3620000 0.03487500 0.0955
0000
## 12 0.2835714 0.04714286 0.010000000 0.06485714 0.4760000 0.02100000 0.0830
0000
##            as         at        be       been        but          by
can
## 1  0.12700000 0.04442857 0.3422857 0.06157143 0.01642857 0.09485714 0.0274
2857
## 2  0.07766667 0.03233333 0.1063333 0.04266667 0.02200000 0.19066667 0.0180
0000
## 3  0.13714286 0.04971429 0.3375714 0.09257143 0.02671429 0.16485714 0.0414
2857
## 4  0.13960000 0.04160000 0.3477000 0.04460000 0.03300000 0.09280000 0.0650
0000
```

```
## 5  0.11022222 0.06411111 0.2387778 0.07100000 0.03522222 0.10933333 0.0295
5556
## 6  0.10085714 0.04814286 0.3592857 0.03971429 0.02600000 0.11614286 0.0614
2857
## 7  0.08700000 0.03650000 0.2255000 0.07525000 0.02025000 0.08800000 0.0412
5000
## 8  0.15680000 0.03600000 0.2754000 0.02680000 0.04920000 0.13620000 0.0330
0000
## 9  0.11187500 0.03750000 0.2786250 0.08287500 0.04812500 0.11550000 0.0291
2500
## 10 0.11730000 0.06000000 0.3491000 0.05620000 0.02880000 0.12460000 0.0253
0000
## 11 0.16600000 0.02775000 0.3493750 0.03850000 0.03350000 0.14787500 0.0231
2500
## 12 0.12185714 0.03271429 0.2264286 0.07628571 0.04028571 0.18671429 0.0201
4286
##             do         down        even      every       for.       from
## 1  0.003571429 0.0010000000 0.002857143 0.01457143 0.11485714 0.07685714
## 2  0.004000000 0.0033333333 0.009333333 0.01166667 0.04366667 0.07366667
## 3  0.006000000 0.0005714286 0.018000000 0.03214286 0.11185714 0.07285714
## 4  0.006600000 0.0042000000 0.013500000 0.01870000 0.09000000 0.09310000
## 5  0.009111111 0.0005555556 0.012666667 0.02611111 0.07188889 0.09377778
## 6  0.008571429 0.0018571429 0.009285714 0.04157143 0.10928571 0.08685714
## 7  0.005250000 0.0030000000 0.017000000 0.01025000 0.08325000 0.04650000
## 8  0.008200000 0.0000000000 0.007600000 0.00600000 0.09600000 0.09100000
## 9  0.008625000 0.0000000000 0.006250000 0.02950000 0.08175000 0.07250000
## 10 0.005700000 0.0022000000 0.022600000 0.01970000 0.10240000 0.07650000
## 11 0.005250000 0.0018750000 0.006750000 0.03887500 0.08825000 0.06762500
## 12 0.002142857 0.0000000000 0.006428571 0.02028571 0.10614286 0.08600000
##            had        has       have         her         his        if.
## 1  0.020000000 0.03457143 0.09500000 0.000000000 0.030428571 0.02542857
## 2  0.077666667 0.04533333 0.06100000 0.011333333 0.074333333 0.01000000
## 3  0.029857143 0.04500000 0.11871429 0.012857143 0.006285714 0.02885714
## 4  0.011300000 0.03300000 0.08680000 0.002700000 0.027700000 0.03010000
## 5  0.016666667 0.06388889 0.10988889 0.012777778 0.015222222 0.02166667
## 6  0.005285714 0.04700000 0.06171429 0.027000000 0.021428571 0.03171429
## 7  0.013750000 0.05975000 0.10525000 0.000000000 0.050000000 0.02425000
## 8  0.016400000 0.02880000 0.08680000 0.014800000 0.009000000 0.05260000
## 9  0.016625000 0.04937500 0.11525000 0.001625000 0.014750000 0.02850000
## 10 0.028500000 0.04900000 0.09580000 0.004800000 0.062100000 0.03470000
## 11 0.015375000 0.02525000 0.08050000 0.004250000 0.015875000 0.01812500
## 12 0.034142857 0.05414286 0.10085714 0.009142857 0.039714286 0.01642857
##          in.       into        is        it        its        may       mo
re
## 1  0.4078571 0.01828571 0.1468571 0.1972857 0.03885714 0.06100000 0.042142
86
## 2  0.2490000 0.02233333 0.1316667 0.1110000 0.06533333 0.02533333 0.051666
67
## 3  0.2745714 0.02128571 0.1322857 0.1634286 0.05500000 0.07871429 0.043428
57
```

```
## 4  0.3262000 0.01670000 0.1688000 0.1648000 0.06650000 0.07130000 0.035600
00
## 5  0.3166667 0.02922222 0.1306667 0.1178889 0.04677778 0.05288889 0.042555
56
## 6  0.2805714 0.03157143 0.1801429 0.1232857 0.02628571 0.08471429 0.073285
71
## 7  0.2837500 0.02050000 0.1632500 0.1270000 0.04075000 0.04925000 0.038500
00
## 8  0.2714000 0.04460000 0.0936000 0.2048000 0.03340000 0.05680000 0.086800
00
## 9  0.4012500 0.01612500 0.2168750 0.2016250 0.05300000 0.07975000 0.035125
00
## 10 0.3256000 0.01840000 0.1258000 0.1520000 0.06290000 0.04530000 0.034800
00
## 11 0.2963750 0.02587500 0.1786250 0.1501250 0.04400000 0.05675000 0.051250
00
## 12 0.2922857 0.03257143 0.1808571 0.1467143 0.03614286 0.05600000 0.034857
14
##           must          my         no        not         now        of
## 1  0.023000000 0.004714286 0.02142857 0.09028571 0.005714286 1.0714286
## 2  0.006333333 0.002333333 0.04233333 0.04333333 0.006666667 0.9133333
## 3  0.039142857 0.003000000 0.03471429 0.10128571 0.005142857 0.8365714
## 4  0.040900000 0.001800000 0.03220000 0.07940000 0.003700000 0.9058000
## 5  0.031444444 0.002777778 0.02455556 0.08788889 0.008666667 1.0322222
## 6  0.033285714 0.009000000 0.03742857 0.11142857 0.003714286 0.8734286
## 7  0.016500000 0.002250000 0.02675000 0.06975000 0.005000000 1.0182500
## 8  0.021200000 0.001800000 0.01500000 0.10800000 0.006600000 0.6390000
## 9  0.039250000 0.008625000 0.05000000 0.10400000 0.013750000 0.9067500
## 10 0.039600000 0.000700000 0.02950000 0.09340000 0.004200000 0.9229000
## 11 0.037500000 0.000875000 0.03912500 0.09887500 0.004250000 0.9122500
## 12 0.035428571 0.001285714 0.03371429 0.09271429 0.005285714 0.8125714
##            on        one       only         or        our      shall
## 1  0.03600000 0.03600000 0.02128571 0.10128571 0.00400000 0.01771429
## 2  0.10566667 0.05633333 0.01233333 0.05900000 0.00000000 0.01666667
## 3  0.11942857 0.03514286 0.03071429 0.08328571 0.01971429 0.01271429
## 4  0.02990000 0.03580000 0.02410000 0.10490000 0.01650000 0.01840000
## 5  0.05344444 0.03200000 0.01944444 0.10055556 0.04933333 0.01233333
## 6  0.06657143 0.05057143 0.01457143 0.07885714 0.02557143 0.02857143
## 7  0.03200000 0.03250000 0.02125000 0.09075000 0.00950000 0.02825000
## 8  0.07460000 0.08140000 0.04340000 0.16080000 0.06600000 0.01740000
## 9  0.06950000 0.03375000 0.02525000 0.08075000 0.01087500 0.02725000
## 10 0.05430000 0.03650000 0.01740000 0.11270000 0.03180000 0.01640000
## 11 0.13000000 0.04237500 0.02425000 0.09712500 0.01150000 0.01700000
## 12 0.08514286 0.04128571 0.02200000 0.07585714 0.01942857 0.01685714
##         should         so       some       such       than      that
the
## 1  0.028714286 0.03657143 0.02042857 0.02628571 0.05471429 0.2465714 1.452
571
## 2  0.005666667 0.01833333 0.01466667 0.02833333 0.03133333 0.1316667 1.409
000
```

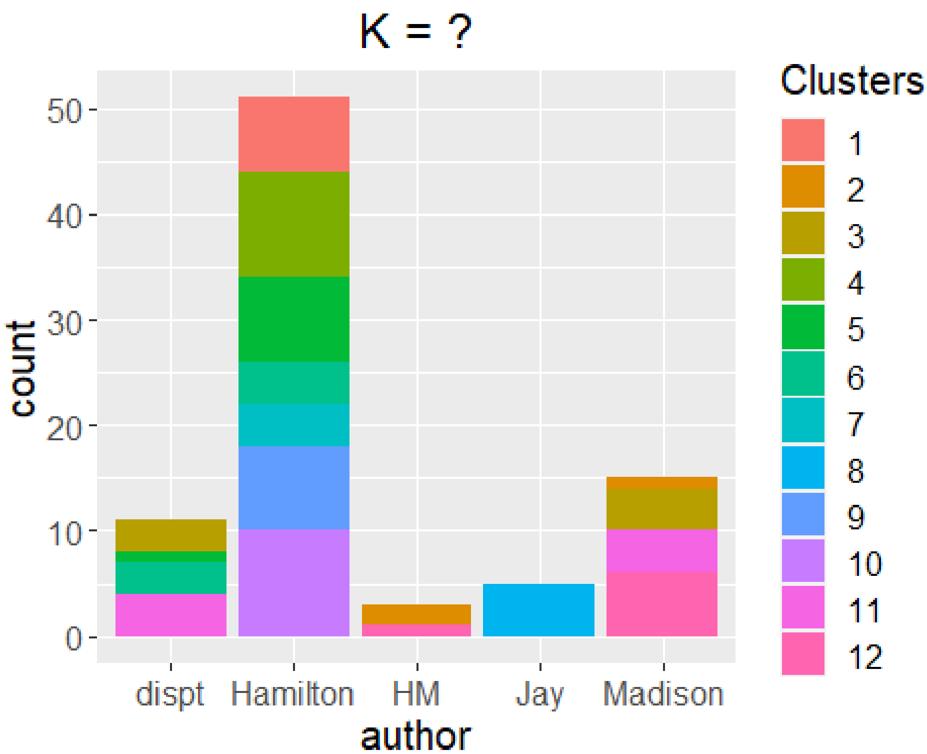
```
## 3  0.016285714 0.02771429 0.03342857 0.02385714 0.02942857 0.2080000 1.326
000
## 4  0.041400000 0.02980000 0.00980000 0.03400000 0.03750000 0.2340000 1.415
600
## 5  0.023888889 0.02955556 0.01622222 0.02133333 0.03444444 0.1691111 1.116
556
## 6  0.017428571 0.03514286 0.01914286 0.02685714 0.08642857 0.2320000 1.142
286
## 7  0.018750000 0.03300000 0.01775000 0.01850000 0.04350000 0.1765000 1.399
250
## 8  0.041400000 0.04460000 0.02140000 0.05120000 0.06280000 0.2434000 0.854
400
## 9  0.022625000 0.02362500 0.01250000 0.03262500 0.03637500 0.2451250 1.309
375
## 10 0.031000000 0.02360000 0.02520000 0.03100000 0.03890000 0.2179000 1.206
300
## 11 0.025625000 0.02387500 0.02000000 0.02550000 0.04200000 0.1873750 1.534
500
## 12 0.028142857 0.03557143 0.02885714 0.03185714 0.03714286 0.1997143 1.186
429
##         their        then       there      things       this         to
## 1  0.05671429 0.004571429 0.032857143 0.001000000 0.11000000 0.5740000
## 2  0.07400000 0.010000000 0.007666667 0.000000000 0.08466667 0.3680000
## 3  0.08971429 0.003714286 0.007000000 0.001285714 0.08028571 0.4445714
## 4  0.09310000 0.006100000 0.044600000 0.003500000 0.08200000 0.6809000
## 5  0.08011111 0.003555556 0.033777778 0.006666667 0.09522222 0.4807778
## 6  0.09400000 0.006857143 0.029714286 0.001714286 0.07371429 0.5712857
## 7  0.04550000 0.003750000 0.026500000 0.007750000 0.07900000 0.5897500
## 8  0.14160000 0.008000000 0.014000000 0.001400000 0.05320000 0.4834000
## 9  0.05887500 0.007000000 0.044500000 0.001375000 0.10000000 0.5496250
## 10 0.07490000 0.006400000 0.031300000 0.002800000 0.10660000 0.6119000
## 11 0.09450000 0.007500000 0.009375000 0.003250000 0.07825000 0.4536250
## 12 0.12100000 0.007571429 0.008857143 0.000000000 0.07757143 0.4645714
##             up       upon        was       were        what        when
## 1  0.001142857 0.054285714 0.01600000 0.01200000 0.019857143 0.022857143
## 2  0.009000000 0.002333333 0.10000000 0.05433333 0.005666667 0.010000000
## 3  0.001428571 0.001857143 0.03671429 0.02128571 0.010000000 0.010000000
## 4  0.000900000 0.045000000 0.02500000 0.01330000 0.015700000 0.007700000
## 5  0.003333333 0.032888889 0.01777778 0.02277778 0.009777778 0.005111111
## 6  0.004142857 0.024714286 0.00800000 0.01071429 0.009571429 0.015285714
## 7  0.003750000 0.048750000 0.03025000 0.01575000 0.010750000 0.005000000
## 8  0.000000000 0.001800000 0.02480000 0.02880000 0.018400000 0.021000000
## 9  0.002875000 0.045375000 0.01700000 0.01962500 0.017250000 0.008125000
## 10 0.012600000 0.055600000 0.02760000 0.02310000 0.013200000 0.017600000
## 11 0.000750000 0.000000000 0.02112500 0.01600000 0.011000000 0.009000000
## 12 0.001857143 0.006000000 0.03357143 0.02671429 0.008857143 0.010000000
##        which        who       will       with      would        your
## 1  0.1291429 0.03571429 0.09842857 0.07914286 0.16185714 0.000000000
## 2  0.1363333 0.04866667 0.01000000 0.10600000 0.02966667 0.000000000
## 3  0.1735714 0.02142857 0.12600000 0.06871429 0.06600000 0.004857143
```

```
## 4  0.1630000 0.03660000 0.11720000 0.08300000 0.07470000 0.000000000
## 5  0.1724444 0.02100000 0.08744444 0.10333333 0.10888889 0.001111111
## 6  0.1720000 0.04228571 0.14000000 0.07900000 0.07842857 0.010571429
## 7  0.1987500 0.02400000 0.06725000 0.08700000 0.08450000 0.000000000
## 8  0.0986000 0.05160000 0.12600000 0.09500000 0.12520000 0.006400000
## 9  0.1840000 0.03337500 0.09125000 0.05825000 0.10500000 0.002750000
## 10 0.1368000 0.03130000 0.07710000 0.07030000 0.18530000 0.000000000
## 11 0.1520000 0.02275000 0.12800000 0.07550000 0.09575000 0.000000000
## 12 0.1644286 0.03600000 0.06014286 0.07300000 0.03757143 0.000000000

# Plot results
clusplot(fedPapers_KM, fedPapers_KM$Clusters, color = T, shade = T, labels =
0, lines = 0)
```




The clusplot shows a visual representation of the K-Means model. It is difficult to see how the documents are clustered from this visualization, but it does show that the model was successful in assigning the 85 observations to clusters.

```
# Create a separate data frame that includes the author names
fedPapers_KM2 <- fedPapersRAW
fedPapers_KM2$Clusters <- as.factor(Clusters$cluster)

ggplot(data = fedPapers_KM2, aes(x=author, fill=Clusters))+
  geom_bar(stat = "count")+
  labs(title = "K = ?")+
  theme(plot.title = element_text(hjust=0.5), text=element_text(size = 15))
```




The stacked bar chart can give us a better visual of the contents of each cluster. The disputed papers seem to be assigned three papers to cluster 3, one paper to cluster 5, three papers to cluster 6, and four papers to cluster 11.

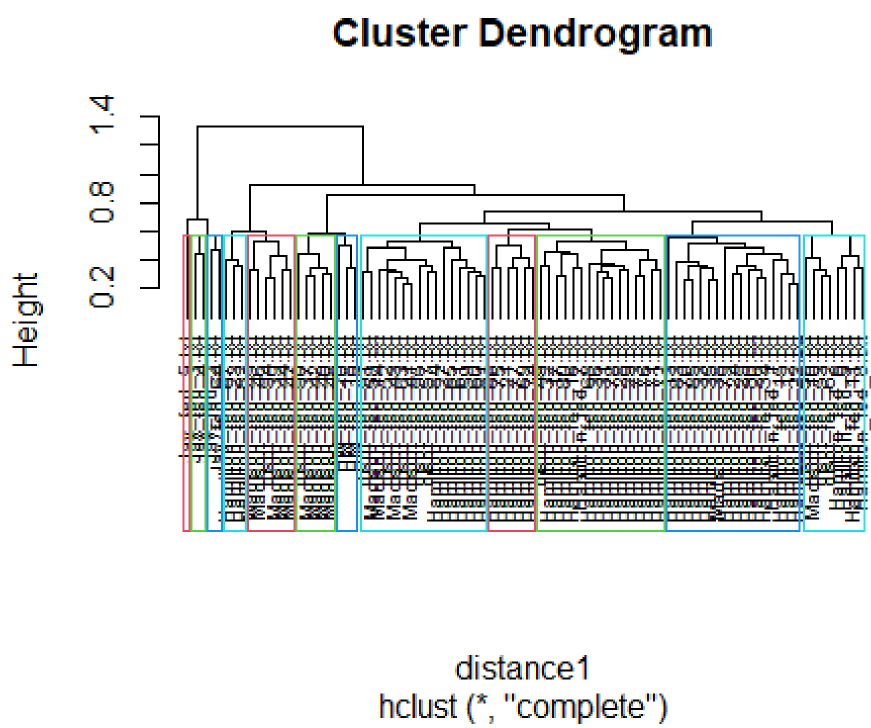
Based on the contents of clusters 3, 5, 6 & 11 we can draw conclusions about those papers. Cluster 3 and 11 seem to be distinctly Madison. Cluster 5 and 6 seem to be distinctly Hamilton. Based on these results it is likely that Hamilton wrote approximately seven of the disputed papers, and Hamilton wrote approximately four or the disputed papers.

## Hierarchical Clustering Algorithms (HAC)

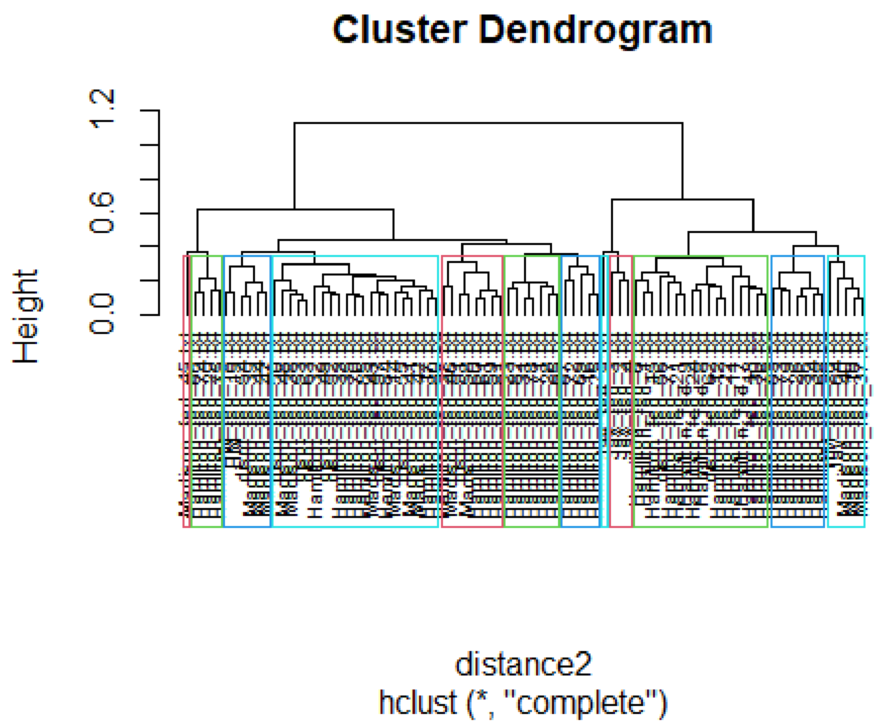
The Hierarchical Clustering Algorithms or HAC create nested clusters. This method will produce a series of diagrams that assigns each paper to a cluster and identifies its nearest neighbors. This data will be used to make conclusions about the authorship of each disputed paper.

```
# Create a subset that excludes the authors names
fedPapers_HAC <- fedPapersRAW[, c(2:72)]

# Make the names of each file an index for the rownames
rownames(fedPapers_HAC) <- fedPapers_HAC[,1]
fedPapers_HAC[,1] <- NULL

# Calculate the distance using various methods
distance1 <- dist(fedPapers_HAC, method = "euclidean")
distance2 <- dist(fedPapers_HAC, method = "maximum")
distance3 <- dist(fedPapers_HAC, method = "manhattan")
```

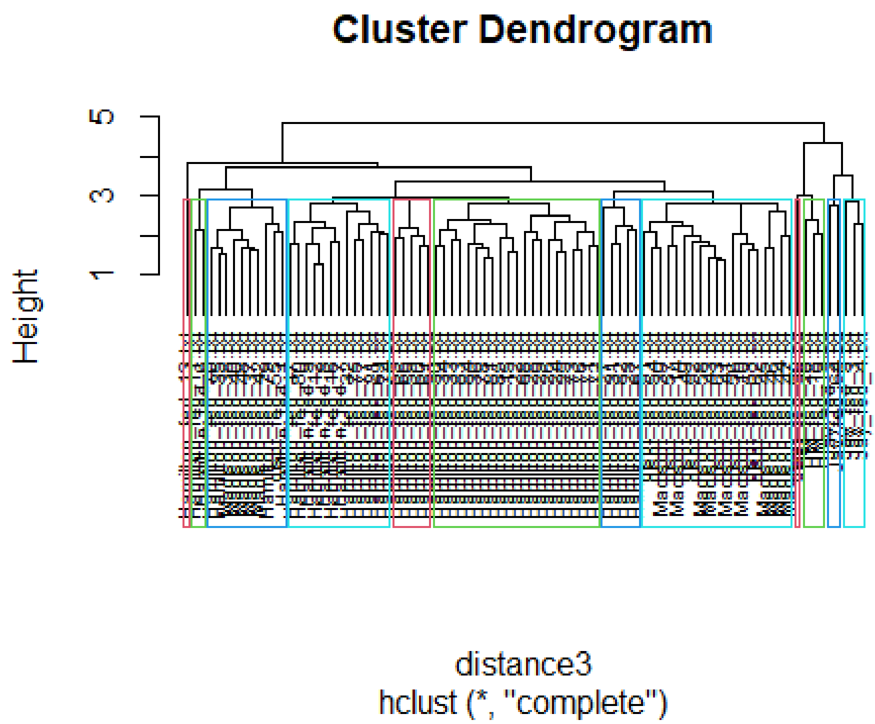
```
distance4 <- dist(fedPapers_HAC, method = "canberra")
distance5 <- dist(fedPapers_HAC, method = "binary")
distance6 <- dist(fedPapers_HAC, method = "minkowski", p=0.5)
distance7 <- dist(fedPapers_HAC, method = "minkowski", p=4)

# Display the results of HAC
# Euclidean distance
HAC1 <- hclust(distance1, method = "complete")
plot(HAC1, cex=0.6, hang=-1)
rect.hclust(HAC1, k=12, border = 2:5)
```



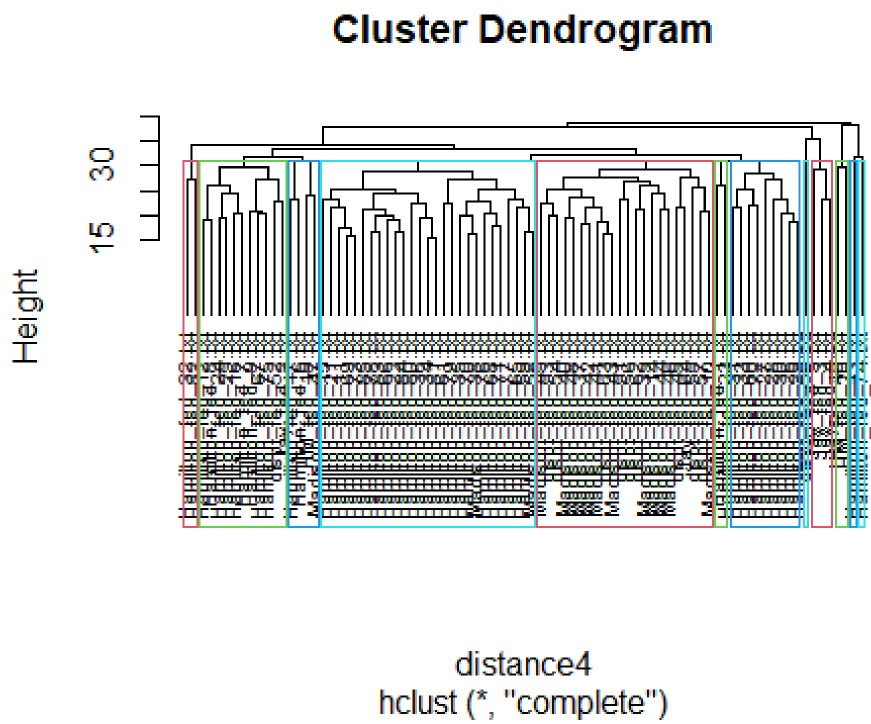

```
# Maximum Distance
HAC2 <- hclust(distance2, method = "complete")
plot(HAC2, cex=0.6, hang=-1)
rect.hclust(HAC2, k=12, border = 2:5)
```



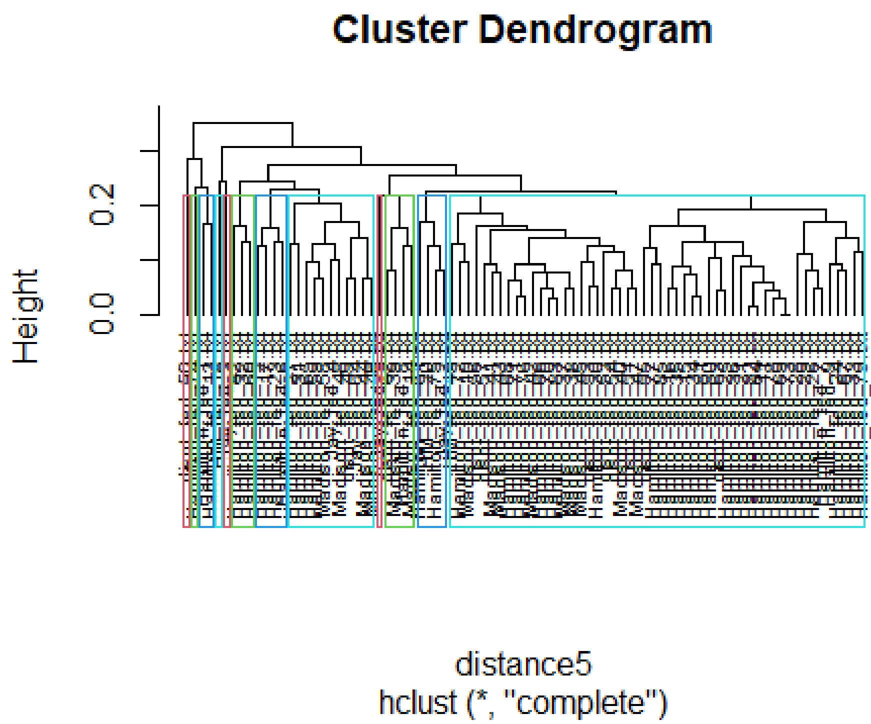

```r
# Manhattan distance
HAC3 <- hclust(distance3, method = "complete")
plot(HAC3, cex=0.6, hang=-1)
rect.hclust(HAC3, k=12, border = 2:5)
```



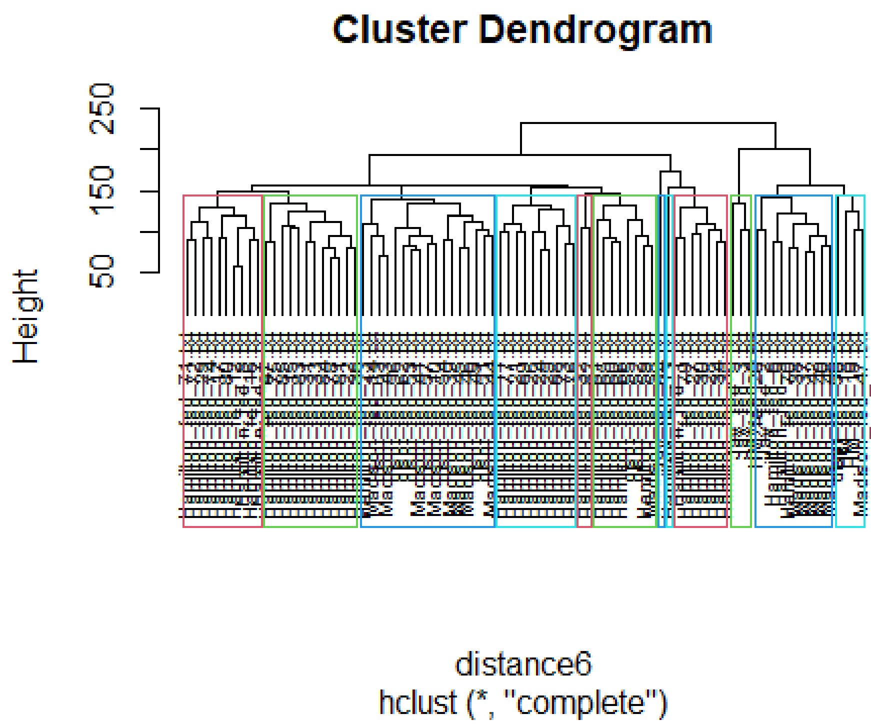

```r
# Canberra distance
HAC4 <- hclust(distance4, method = "complete")
plot(HAC4, cex=0.6, hang=-1)
rect.hclust(HAC4, k=12, border = 2:5)
```



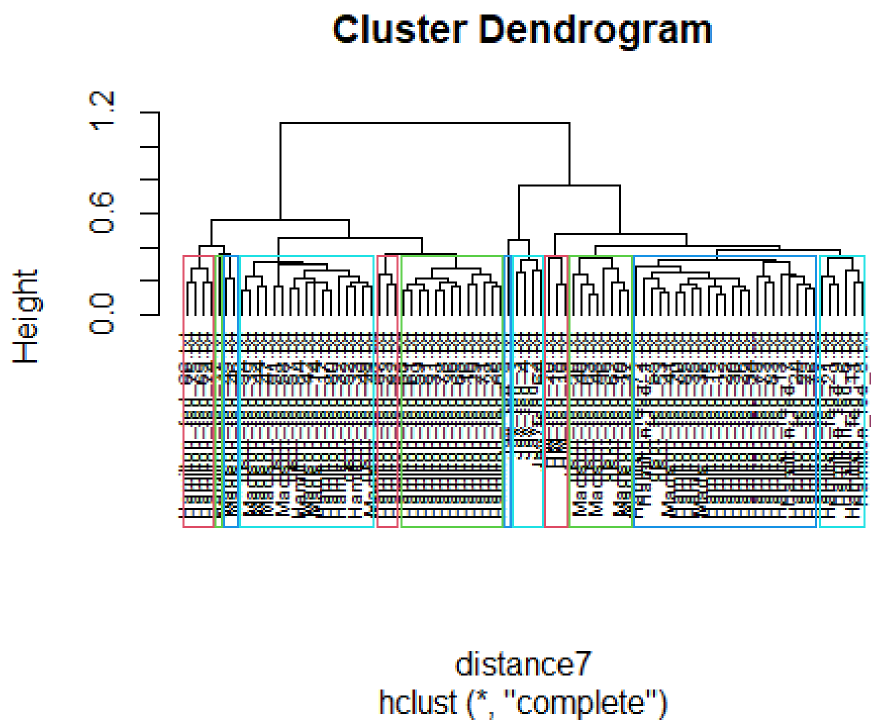

```r
# Binary distance
HAC5 <- hclust(distance5, method = "complete")
plot(HAC5, cex=0.6, hang=-1)
rect.hclust(HAC5, k=12, border = 2:5)
```




```
# Minkowksi distance p=0.5
HAC6 <- hclust(distance6, method = "complete")
plot(HAC6, cex=0.6, hang=-1)
rect.hclust(HAC6, k=12, border = 2:5)
```




```
# Minkowksi distance p=4
HAC7 <- hclust(distance7, method = "complete")
plot(HAC7, cex=0.6, hang=-1)
rect.hclust(HAC7, k=12, border = 2:5)
```




## Conclusions

The K-Means model seemed to have more definitive results, because each observation is placed in a distinct cluster. The Hierarchical Clustering Algorithms (HAC) were much more textured. The different distance measuring methods didn't always agree. Both clustering methods indicated that the majority of the disputed papers were written by James Madison, though they did not have the same number of Madison vs. Hamilton identifications. Most of the models correctly cluster John Jay's papers together, and John Jay was not considered to be a possible author of any of the disputed papers.

The highly detailed dendrograms which resulted from the HAC models (summarized in the tables below) contain features that illustrate the similarity between the disputed papers and the papers whose authors are known. Some of the disputed papers were assigned to mixed clusters of Madison and Hamilton papers. In those cases the disputed paper's authorship was identified based on a close branch to a known author.

Papers 51 and 54 were unanimously chosen as Madison essays by all of the models. Papers 53, 56, 62, and 63 were identified as Madison papers six out of seven times. Papers 49, 50 and 57 were identified slightly more times as Madison over Hamilton. Paper 52 had as many Madison identifications as Hamilton. It is important to note that some of the models identified a disputed paper as authored by John Jay or as a collaboration between Madison and Hamilton. As stated before John Jay was not considered a candidate for authorship, and Hamilton-Madison did not occur enough times in any observation to be considered as a serious possibility. Only one paper was identifed as Hamilton by the majority of models, paper 55.

| **dispt_fed_49.txt** | Neighbor1 | Neighbor2 | Cluster Contents | Authorship |
| --- | --- | --- | --- | --- |
| Distance1: Euclidean | Hamilton_74 | Hamilton_32 | Hamilton | Hamilton |
| Distance2: Maximum | Hamilton_32 | Dispt_52 | Madison/Hamilton | Hamilton |
| Distance3: Manhattan | Madison_14 | Dispt_52 | Madison | Madison |
| Distance4: Canberra | Madison_46 | Jay_64 | Madison/Jay | Jay? |
| Distance5: Binary | Madison_48 | Jay_64 | Madison/Hamilton/Jay | Jay? |
| Distance6: Minkowski, p=0.5 | Madison_46 | Dispt_51 | Madison/Hamilton | Madison |
| Distance7: Minkowski, p=4 | Hamilton_33 | Madison_43 | Madison/Hamilton | Madison |

| **dispt_fed_50.txt** | Neighbor1 | Neighbor2 | Cluster Contents | Authorship |
| --- | --- | --- | --- | --- |
| Distance1: Euclidean | Madison_47 | Madison_37 | Madison | Madison |
| Distance2: Maximum | Madison_47 | Madison_48 | Madison/Hamilton | Madison |
| Distance3: Manhattan | Madison_47 | HM_20 | Dispt | HM? |
| Distance4: Canberra | Hamilton_29 | Jay_5 | Dispt | Jay? |
| Distance5: Binary | NA | Hamilton_74 | Dispt | Hamilton |
| Distance6: Minkowski, p=0.5 | Madison_48 | HM_18 | Madison/HM | HM? |
| Distance7: Minkowski, p=4 | HM_19 | Madison_40 | Madison | Madison |

| **dispt_fed_51.txt** | Neighbor1 | Neighbor2 | Cluster Contents | Authorship |
| --- | --- | --- | --- | --- |
| Distance1: Euclidean | HM_19 | Madison_58 | Madison/Hamilton | Madison |
| Distance2: Maximum | Madison_46 | Madison_58 | Madison/Hamilton | Madison |
| Distance3: Manhattan | Hamilton_67 | Dispt_54 | Madison/Jay | Madison |
| Distance4: Canberra | Madison_43 | Dispt_56 | Madison/Jay | Madison |
| Distance5: Binary | Madison_40 | Madison_47 | Madison/Hamilton | Madison |
| Distance6: Minkowski, p=0.5 | Dispt_49 | Madison_44 | Madison/Hamilton | Madison |
| Distance7: Minkowski, p=4 | Madison_47 | Madison_58 | Madison/Hamilton | Madison |

| **dispt_fed_52.txt** | Neighbor1 | Neighbor2 | Cluster Contents | Authorship |
| --- | --- | --- | --- | --- |
| Distance1: Euclidean | Madison_14 | Madison_43 | Madison/Hamilton | Madison |
| Distance2: Maximum | Dispt_49 | Hamilton_33 | Madison/Hamilton | Hamilton |
| Distance3: Manhattan | Dispt_49 | Madison_58 | Madison | Madison |
| Distance4: Canberra | Hamilton_75 | Jay_2 | Hamilton/Jay | Jay? |
| Distance5: Binary | Hamilton_30 | Dispt_54 | Madison/Hamilton | Madison |
| Distance6: Minkowski, p=0.5 | Dispt_55 | Hamilton_59 | Madison/Hamilton | Hamilton |
| Distance7: Minkowski, p=4 | Hamilton_32 | Hamilton_33 | Madison/Hamilton | Hamilton |

| **dispt_fed_53.txt** | Neighbor1 | Neighbor2 | Cluster Contents | Authorship |
|---|---|---|---|---|
| Distance1: Euclidean | Madison_10 | Dispt_62 | Madison/Hamilton | Madison |
| Distance2: Maximum | Madison_38 | Dispt_63 | Madison/Hamilton | Madison |
| Distance3: Manhattan | Madison_10 | Dispt_62 | Madison | Madison |
| Distance4: Canberra | Madison_45 | Dispt_53 | Madison/Jay | Madison |
| Distance5: Binary | Madison_45 | Hamilton_30 | Madison/Hamilton | Hamilton |
| Distance6: Minkowski, p=0.5 | Dispt_62 | Madison_41 | Madison/Hamilton | Madison |
| Distance7: Minkowski, p=4 | Dispt_55 | Madison_41 | Madison/Hamilton | Madison |

| **dispt_fed_54.txt** | Neighbor1 | Neighbor2 | Cluster Contents | Authorship |
|---|---|---|---|---|
| Distance1: Euclidean | Madison_46 | Madison_39 | Madison | Madison |
| Distance2: Maximum | HM_19 | Madison_39 | HM/Madison | Madison |
| Distance3: Manhattan | Dispt_51 | Madison_39 | Madison | Madison |
| Distance4: Canberra | Dispt_54 | Madison_40 | Madison/Jay | Madison |
| Distance5: Binary | Dispt_52 | Madison_40 | Madison/Hamilton | Madison |
| Distance6: Minkowski, p=0.5 | Madison_10 | Madison_39 | Madison/Hamilton | Madison |
| Distance7: Minkowski, p=4 | Madison_46 | Madison_39 | Madison/Hamilton | Madison |

| **dispt_fed_55.txt** | Neighbor1 | Neighbor2 | Cluster Contents | Authorship |
|---|---|---|---|---|
| Distance1: Euclidean | Madison_41 | Dispt_57 | Madison/Hamilton | Hamilton |
| Distance2: Maximum | Madison_14 | Madison_41 | Madison/Hamilton | Madison |
| Distance3: Manhattan | Hamilton_79 | Hamilton_1 | Madison/Hamilton | Hamilton |
| Distance4: Canberra | Jay_64 | Dispt_57 | Madison/Jay | Madison |
| Distance5: Binary | Madison_47 | Hamilton_77 | Madison/Hamilton | Hamilton |
| Distance6: Minkowski, p=0.5 | Hamilton_66 | Dispt_52 | Madison/Hamilton | Hamilton |
| Distance7: Minkowski, p=4 | Hamilton_1 | Dispt_55 | Madison/Hamilton | Madison |

| **dispt_fed_56.txt** | Neighbor1 | Neighbor2 | Cluster Contents | Authorship |
|---|---|---|---|---|
| Distance1: Euclidean | Hamilton_22 | Madison_10 | Madison/Hamilton | Madison |
| Distance2: Maximum | Hamilton_15 | Hamilton77 | Hamilton | Hamilton |
| Distance3: Manhattan | Madison_41 | Madison_10 | Madison | Madison |
| Distance4: Canberra | Dispt_51 | Dispt_62 | Madison/Jay | Madison |
| Distance5: Binary | Madison_10 | Dispt_57 | Madison/Hamilton | Madison |
| Distance6: Minkowski, p=0.5 | Madison_43 | Dispt_62 | Madison/Hamilton | Madison |
| Distance7: Minkowski, p=4 | Madison_48 | Dispt_62 | Madison | Madison |

| **dispt_fed_57.txt** | Neighbor1 | Neighbor2 | Cluster Contents | Authorship |
|---|---|---|---|---|
| Distance1: Euclidean | Dispt_55 | Hamilton_84 | Madison/Hamilton | Hamilton |
| Distance2: Maximum | Madison_43 | Hamilton_84 | Madison/Hamilton | Hamilton |
| Distance3: Manhattan | Madison_39 | Madison_14 | Madison | Madison |
| Distance4: Canberra | Dispt_55 | Madison_10 | Madison/Jay | Madison |
| Distance5: Binary | Dispt_56 | Madison_41 | Madison/Hamilton | Madison |
| Distance6: Minkowski, p=0.5 | Madison_41 | Madison_10 | Madison/Hamilton | Madison |
| Distance7: Minkowski, p=4 | Madison_58 | Hamilton_84 | Madison/Hamilton | Hamilton |

| **dispt_fed_62.txt** | Neighbor1 | Neighbor2 | Cluster Contents | Authorship |
|---|---|---|---|---|
| Distance1: Euclidean | Dispt_53 | Hamilton_6 | Madison/Hamilton | Madison |
| Distance2: Maximum | Hamilton_8 | Hamilton_12 | Hamilton | Hamilton |
| Distance3: Manhattan | Dispt_53 | Madison_45 | Madison | Madison |
| Distance4: Canberra | Dispt_56 | Madison_39 | Madison/Jay | Madison |
| Distance5: Binary | Jay_5 | Madison_39 | Madison/Hamilton | Madison |
| Distance6: Minkowski, p=0.5 | Dispt_56 | Dispt_53 | Madison/Hamilton | Madison |
| Distance7: Minkowski, p=4 | Dispt_56 | Madison_10 | Madison | Madison |

| **dispt_fed_63.txt** | Neighbor1 | Neighbor2 | Cluster Contents | Authorship |
|---|---|---|---|---|
| Distance1: Euclidean | Madison_43 | Madison_41 | Madison/Hamilton | Madison |
| Distance2: Maximum | Dispt_53 | Hamilton_32 | Madison/Hamilton | Madison |
| Distance3: Manhattan | Madison_43 | Madison_41 | Madison | Madison |
| Distance4: Canberra | Madison_41 | Madison_43 | Madison/Jay | Madison |
| Distance5: Binary | Hamilton_80 | Hamilton_25 | Madison/Hamilton | Hamilton |
| Distance6: Minkowski, p=0.5 | Madison_14 | Madison_43 | Madison/Hamilton | Madison |
| Distance7: Minkowski, p=4 | Madison_40 | Madison_48 | Madison | Madison |