HW4_Federalist-Papers

Ben Dieck

2023-08-04

Load Muckrakers

```
library(tidyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tibble)
library(factoextra)
```

```
## Loading required package: ggplot2
```

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

```
library(cluster)
```

Load Data

Creating a corpus of the dataset

```
fedpapers <- read.csv("C:/R/data/fedPapers85.csv")
fedpapers2<-fedpapers
str(fedpapers)
```

```
## 'data.frame':    85 obs. of  72 variables:
##  $ author  : chr  "dispt" "dispt" "dispt" "dispt" ...
##  $ filename: chr  "dispt_fed_49.txt" "dispt_fed_50.txt" "dispt_fed_51.txt" "dispt_fed_52.txt"
...
##  $ a       : num  0.28 0.177 0.339 0.27 0.303 0.245 0.349 0.414 0.248 0.442 ...
##  $ all     : num  0.052 0.063 0.09 0.024 0.054 0.059 0.036 0.083 0.04 0.062 ...
##  $ also    : num  0.009 0.013 0.008 0.016 0.027 0.007 0.007 0.009 0.007 0.006 ...
##  $ an      : num  0.096 0.038 0.03 0.024 0.034 0.067 0.029 0.018 0.04 0.075 ...
##  $ and     : num  0.358 0.393 0.301 0.262 0.404 0.282 0.335 0.478 0.356 0.423 ...
##  $ any     : num  0.026 0.063 0.008 0.056 0.04 0.052 0.058 0.046 0.034 0.037 ...
##  $ are     : num  0.131 0.051 0.068 0.064 0.128 0.111 0.087 0.11 0.154 0.093 ...
##  $ as      : num  0.122 0.139 0.203 0.111 0.148 0.252 0.073 0.074 0.161 0.1 ...
##  $ at      : num  0.017 0.114 0.023 0.056 0.013 0.015 0.116 0.037 0.047 0.031 ...
##  $ be      : num  0.411 0.393 0.474 0.365 0.344 0.297 0.378 0.331 0.289 0.379 ...
##  $ been    : num  0.026 0.165 0.015 0.127 0.047 0.03 0.044 0.046 0.027 0.025 ...
##  $ but     : num  0.009 0 0.038 0.032 0.061 0.037 0.007 0.055 0.027 0.037 ...
##  $ by      : num  0.14 0.139 0.173 0.167 0.209 0.186 0.102 0.092 0.168 0.174 ...
##  $ can     : num  0.035 0 0.023 0.056 0.088 0 0.058 0.037 0.047 0.056 ...
##  $ do      : num  0.026 0.013 0 0 0 0 0.015 0.028 0 0 ...
##  $ down    : num  0 0 0.008 0 0 0.007 0 0 0 0 ...
##  $ even    : num  0.009 0.025 0.015 0.024 0.02 0.007 0.007 0.018 0 0.006 ...
##  $ every   : num  0.044 0 0.023 0.04 0.027 0.007 0.087 0.064 0.081 0.05 ...
##  $ for.    : num  0.096 0.076 0.098 0.103 0.141 0.067 0.116 0.055 0.127 0.1 ...
##  $ from    : num  0.044 0.101 0.053 0.079 0.074 0.096 0.08 0.083 0.074 0.124 ...
##  $ had     : num  0.035 0.101 0.008 0.016 0 0.022 0.015 0.009 0.007 0 ...
##  $ has     : num  0.017 0.013 0.015 0.024 0.054 0.015 0.036 0.037 0.02 0.019 ...
##  $ have    : num  0.044 0.152 0.023 0.143 0.047 0.119 0.044 0.074 0.074 0.044 ...
##  $ her     : num  0 0 0 0 0 0 0.007 0 0.034 0.025 ...
##  $ his     : num  0.017 0 0 0.024 0.02 0.067 0 0.018 0.02 0.05 ...
##  $ if.     : num  0 0.025 0.023 0.04 0.034 0.03 0.029 0 0 0.025 ...
##  $ in.     : num  0.262 0.291 0.308 0.238 0.263 0.401 0.189 0.267 0.248 0.274 ...
##  $ into    : num  0.009 0.025 0.038 0.008 0.013 0.037 0 0.037 0.013 0.037 ...
##  $ is      : num  0.157 0.038 0.15 0.151 0.189 0.26 0.167 0.083 0.208 0.23 ...
##  $ it      : num  0.175 0.127 0.173 0.222 0.108 0.156 0.102 0.165 0.134 0.131 ...
##  $ its     : num  0.07 0.038 0.03 0.048 0.013 0.015 0 0.046 0.02 0.019 ...
##  $ may     : num  0.035 0.038 0.12 0.056 0.047 0.074 0.08 0.092 0.027 0.106 ...
##  $ more    : num  0.026 0 0.038 0.056 0.067 0.045 0.08 0.064 0.06 0.081 ...
##  $ must    : num  0.026 0.013 0.083 0.071 0.013 0.015 0.044 0.018 0.027 0.068 ...
##  $ my      : num  0 0 0 0 0 0 0.007 0 0 0 ...
##  $ no      : num  0.035 0 0.03 0.032 0.047 0.059 0.022 0.018 0.02 0.044 ...
##  $ not     : num  0.114 0.127 0.068 0.087 0.128 0.134 0.102 0.101 0.094 0.106 ...
##  $ now     : num  0 0 0 0 0 0 0.007 0 0.007 0.012 ...
##  $ of      : num  0.9 0.747 0.858 0.802 0.869 ...
##  $ on      : num  0.14 0.139 0.15 0.143 0.054 0.141 0.051 0.083 0.127 0.118 ...
##  $ one     : num  0.026 0.025 0.03 0.032 0.047 0.052 0.073 0.046 0.06 0.031 ...
##  $ only    : num  0.035 0 0.023 0.048 0.027 0.022 0.007 0.046 0.02 0.012 ...
##  $ or      : num  0.096 0.114 0.06 0.064 0.081 0.074 0.153 0.037 0.154 0.081 ...
##  $ our     : num  0.017 0 0 0.016 0.027 0.03 0.051 0 0.007 0.025 ...
##  $ shall   : num  0.017 0 0.008 0.016 0 0.015 0.007 0 0.02 0 ...
##  $ should  : num  0.017 0.013 0.068 0.032 0 0.03 0.007 0 0 0.012 ...
##  $ so      : num  0.035 0.013 0.038 0.04 0.027 0.007 0.051 0.018 0.04 0.05 ...
##  $ some    : num  0.009 0.063 0.03 0.024 0.067 0.045 0.007 0.028 0.027 0.025 ...
```

```
##  $ such    : num  0.026 0 0.045 0.008 0.027 0.015 0.015 0 0.013 0.031 ...
##  $ than    : num  0.009 0 0.023 0 0.047 0.03 0.109 0.055 0.067 0.044 ...
##  $ that    : num  0.184 0.152 0.188 0.238 0.162 0.208 0.233 0.165 0.208 0.218 ...
##  $ the     : num  1.42 1.25 1.49 1.33 1.19 ...
##  $ their   : num  0.114 0.165 0.053 0.071 0.027 0.089 0.109 0.083 0.154 0.081 ...
##  $ then    : num  0 0 0.015 0.008 0.007 0.007 0.015 0.009 0.007 0.012 ...
##  $ there   : num  0.009 0 0.015 0 0.007 0.007 0.036 0.028 0.02 0 ...
##  $ things  : num  0.009 0 0 0 0 0 0 0 0 0.012 ...
##  $ this    : num  0.044 0.051 0.075 0.103 0.094 0.126 0.08 0.11 0.067 0.093 ...
##  $ to      : num  0.507 0.355 0.361 0.532 0.485 0.445 0.56 0.34 0.49 0.498 ...
##  $ up      : num  0 0 0 0 0 0 0.007 0 0 0 ...
##  $ upon    : num  0 0.013 0 0 0 0 0 0 0 0 ...
##  $ was     : num  0.009 0.051 0.008 0.087 0.027 0.007 0.015 0.018 0.027 0 ...
##  $ were    : num  0.017 0 0.015 0.079 0.02 0.03 0.029 0.009 0.007 0 ...
##  $ what    : num  0 0 0.008 0.008 0.02 0.015 0.015 0.009 0.02 0.025 ...
##  $ when    : num  0.009 0 0 0.024 0.007 0.037 0.007 0 0.02 0.012 ...
##  $ which   : num  0.175 0.114 0.105 0.167 0.155 0.186 0.211 0.175 0.201 0.199 ...
##  $ who     : num  0.044 0.038 0.008 0 0.027 0.045 0.022 0.018 0.04 0.031 ...
##  $ will    : num  0.009 0.089 0.173 0.079 0.168 0.111 0.145 0.267 0.154 0.106 ...
##  $ with    : num  0.087 0.063 0.045 0.079 0.074 0.089 0.073 0.129 0.027 0.081 ...
##  $ would   : num  0.192 0.139 0.068 0.064 0.04 0.037 0.073 0.037 0.04 0.031 ...
##  $ your    : num  0 0 0 0 0 0 0 0 0 0 ...
```

Look for NA

```
sum(is.na(fedpapers))
```

```
## [1] 0
```

Take a peak

```
head(fedpapers)
```

```
##   author        filename     a   all  also    an   and   any   are    as    at
## 1  dispt dispt_fed_49.txt 0.280 0.052 0.009 0.096 0.358 0.026 0.131 0.122 0.017
## 2  dispt dispt_fed_50.txt 0.177 0.063 0.013 0.038 0.393 0.063 0.051 0.139 0.114
## 3  dispt dispt_fed_51.txt 0.339 0.090 0.008 0.030 0.301 0.008 0.068 0.203 0.023
## 4  dispt dispt_fed_52.txt 0.270 0.024 0.016 0.024 0.262 0.056 0.064 0.111 0.056
## 5  dispt dispt_fed_53.txt 0.303 0.054 0.027 0.034 0.404 0.040 0.128 0.148 0.013
## 6  dispt dispt_fed_54.txt 0.245 0.059 0.007 0.067 0.282 0.052 0.111 0.252 0.015
##      be  been   but    by   can    do  down  even every  for.  from   had   has
## 1 0.411 0.026 0.009 0.140 0.035 0.026 0.000 0.009 0.044 0.096 0.044 0.035 0.017
## 2 0.393 0.165 0.000 0.139 0.000 0.013 0.000 0.025 0.000 0.076 0.101 0.101 0.013
## 3 0.474 0.015 0.038 0.173 0.023 0.000 0.008 0.015 0.023 0.098 0.053 0.008 0.015
## 4 0.365 0.127 0.032 0.167 0.056 0.000 0.000 0.024 0.040 0.103 0.079 0.016 0.024
## 5 0.344 0.047 0.061 0.209 0.088 0.000 0.000 0.020 0.027 0.141 0.074 0.000 0.054
## 6 0.297 0.030 0.037 0.186 0.000 0.000 0.007 0.007 0.007 0.067 0.096 0.022 0.015
##    have her   his   if.   in.  into    is    it   its   may  more  must my
## 1 0.044   0 0.017 0.000 0.262 0.009 0.157 0.175 0.070 0.035 0.026 0.026  0
## 2 0.152   0 0.000 0.025 0.291 0.025 0.038 0.127 0.038 0.038 0.000 0.013  0
## 3 0.023   0 0.000 0.023 0.308 0.038 0.150 0.173 0.030 0.120 0.038 0.083  0
## 4 0.143   0 0.024 0.040 0.238 0.008 0.151 0.222 0.048 0.056 0.056 0.071  0
## 5 0.047   0 0.020 0.034 0.263 0.013 0.189 0.108 0.013 0.047 0.067 0.013  0
## 6 0.119   0 0.067 0.030 0.401 0.037 0.260 0.156 0.015 0.074 0.045 0.015  0
##      no   not now    of    on   one  only    or   our shall should    so  some
## 1 0.035 0.114   0 0.900 0.140 0.026 0.035 0.096 0.017 0.017  0.017 0.035 0.009
## 2 0.000 0.127   0 0.747 0.139 0.025 0.000 0.114 0.000 0.000  0.013 0.013 0.063
## 3 0.030 0.068   0 0.858 0.150 0.030 0.023 0.060 0.000 0.008  0.068 0.038 0.030
## 4 0.032 0.087   0 0.802 0.143 0.032 0.048 0.064 0.016 0.016  0.032 0.040 0.024
## 5 0.047 0.128   0 0.869 0.054 0.047 0.027 0.081 0.027 0.000  0.000 0.027 0.067
## 6 0.059 0.134   0 0.876 0.141 0.052 0.022 0.074 0.030 0.015  0.030 0.007 0.045
##    such  than  that   the their  then there things  this    to up  upon   was
## 1 0.026 0.009 0.184 1.425 0.114 0.000 0.009  0.009 0.044 0.507  0 0.000 0.009
## 2 0.000 0.000 0.152 1.254 0.165 0.000 0.000  0.000 0.051 0.355  0 0.013 0.051
## 3 0.045 0.023 0.188 1.490 0.053 0.015 0.015  0.000 0.075 0.361  0 0.000 0.008
## 4 0.008 0.000 0.238 1.326 0.071 0.008 0.000  0.000 0.103 0.532  0 0.000 0.087
## 5 0.027 0.047 0.162 1.193 0.027 0.007 0.007  0.000 0.094 0.485  0 0.000 0.027
## 6 0.015 0.030 0.208 1.469 0.089 0.007 0.007  0.000 0.126 0.445  0 0.000 0.007
##    were  what  when which   who  will  with would your
## 1 0.017 0.000 0.009 0.175 0.044 0.009 0.087 0.192    0
## 2 0.000 0.000 0.000 0.114 0.038 0.089 0.063 0.139    0
## 3 0.015 0.008 0.000 0.105 0.008 0.173 0.045 0.068    0
## 4 0.079 0.008 0.024 0.167 0.000 0.079 0.079 0.064    0
## 5 0.020 0.020 0.007 0.155 0.027 0.168 0.074 0.040    0
## 6 0.030 0.015 0.037 0.186 0.045 0.111 0.089 0.037    0
```

So in this case we have 85 federalist papers and 72 possible words in each one which means that this data is already cleaned. Thank god.

```
fedpapers2 <- fedpapers2[,2:72]
fedpapers2 <-select(fedpapers2,filename, upon,all,may,also,even,from,shall,only)
rownames(fedpapers2)<-fedpapers2[,1]
fedpapers2[,1]<-NULL
```

Not sure what I did here but lets see

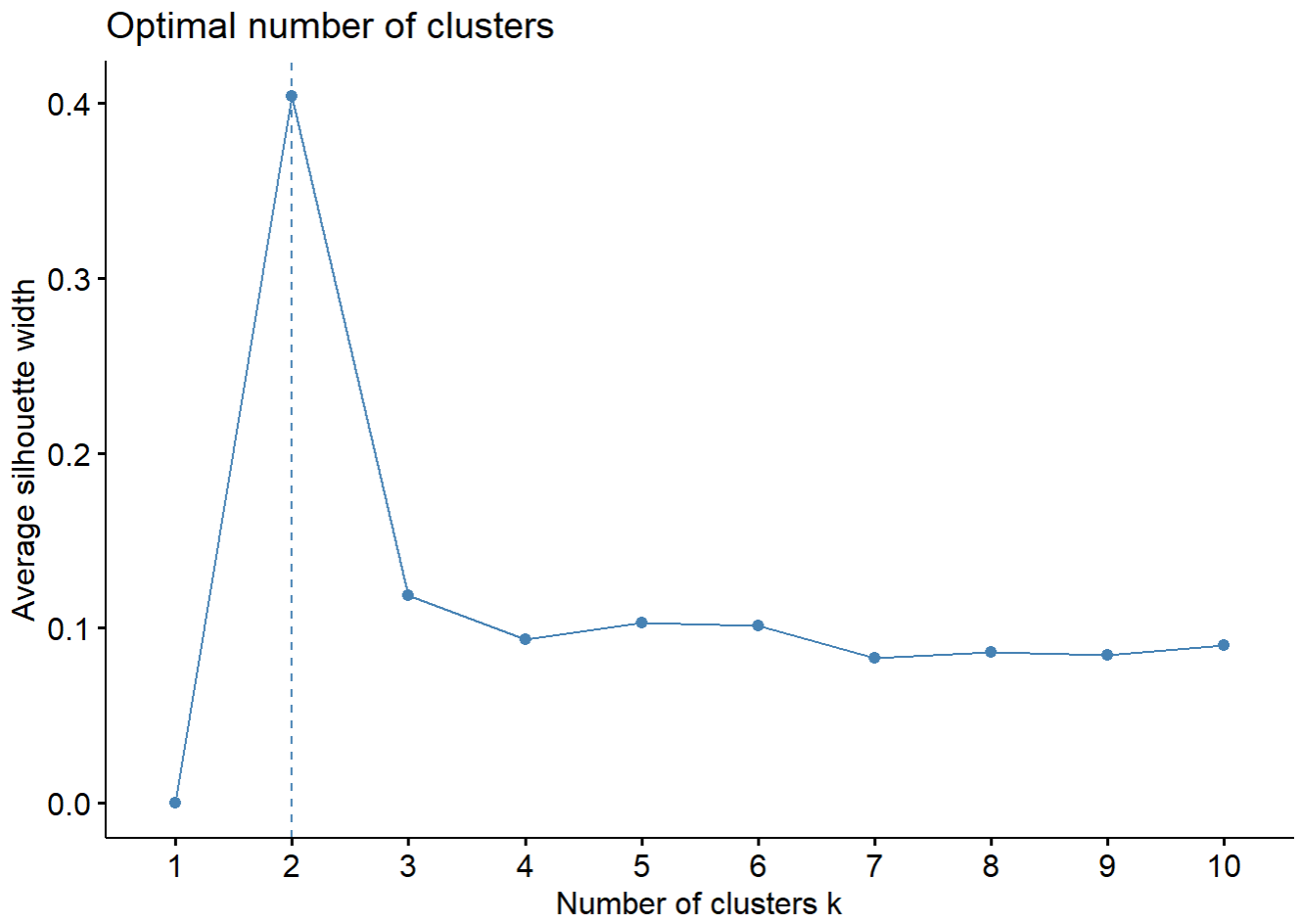
```
#had to load the tibble package
view(fedpapers2)
```

That made a really finite dataframe… The example says that this is signaling words of the pieces.

```
#library factoextra
#Determines the optimal number of clusters, thats cool
fviz_nbclust(fedpapers, FUN = hcut, method="silhouette")
```

```
## Warning in stats::dist(x): NAs introduced by coercion
```

```
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion
```



This already indicates that there are two clear demarcations of where the data should be separated. I guess one of the lasting questions is, how were those words chosen?

```
set.seed(20)
#Running k-means
clusters <- kmeans(fedpapers2,6)
str(clusters)
```

```
## List of 9
##  $ cluster     : Named int [1:85] 2 3 6 2 2 6 6 6 2 6 ...
##   ..- attr(*, "names")= chr [1:85] "dispt_fed_49.txt" "dispt_fed_50.txt" "dispt_fed_51.txt"
"dispt_fed_52.txt" ...
##  $ centers     : num [1:6, 1:8] 0.0333 0.0152 0.02 0.0647 0.018 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:6] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:8] "upon" "all" "may" "also" ...
##  $ totss       : num 0.313
##  $ withinss    : num [1:6] 0.0274 0.0245 0.0263 0.0386 0.0117 ...
##  $ tot.withinss: num 0.153
##  $ betweenss   : num 0.16
##  $ size        : int [1:6] 12 16 14 22 6 15
##  $ iter        : int 4
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

```
clusters$centers
```

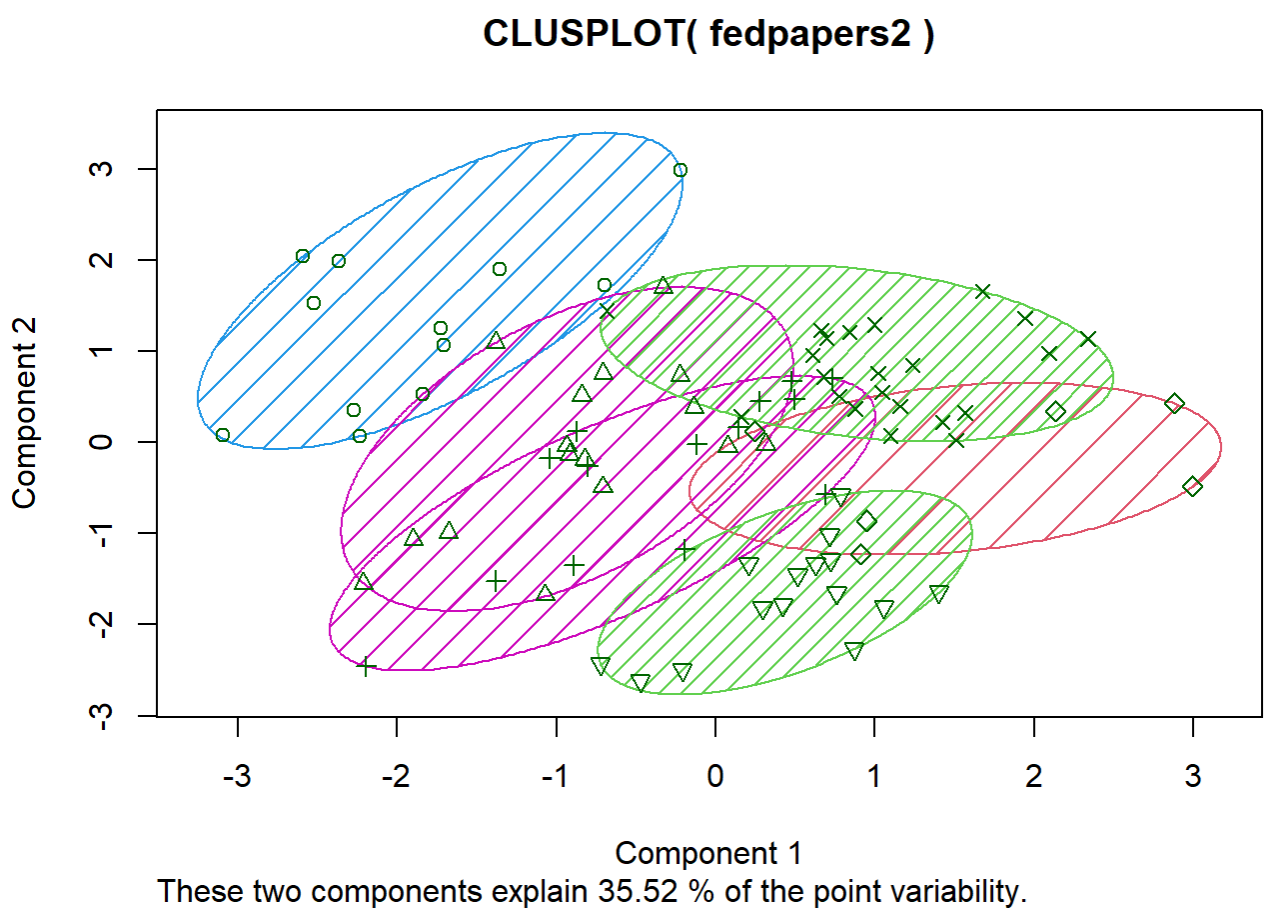
```
##          upon        all        may        also        even       from
## 1 0.033333333 0.03858333 0.10275000 0.007833333 0.006000000 0.08316667
## 2 0.015187500 0.04343750 0.03362500 0.011250000 0.011500000 0.06493750
## 3 0.020000000 0.04742857 0.04557143 0.009642857 0.013357143 0.11985714
## 4 0.064727273 0.05850000 0.05431818 0.003909091 0.014090909 0.07377273
## 5 0.018000000 0.10533333 0.05466667 0.005166667 0.009166667 0.05566667
## 6 0.001933333 0.05000000 0.08813333 0.008333333 0.010733333 0.07393333
##         shall       only
## 1 0.044916667 0.02533333
## 2 0.021000000 0.02731250
## 3 0.010214286 0.01828571
## 4 0.014000000 0.01963636
## 5 0.026500000 0.01450000
## 6 0.007266667 0.02860000
```

Doing some weird stuff Making a new data frame that categorizes the information that I harvested into a factoral value, and categorizes each paper by being in that factoral value. But I feel like I am being set up now because there are 6 values and I'm supposed to be figured out which of the 2 authors did what. I'm probably going to have to come back here.

```
fedpapers_km <- fedpapers
fedpapers_km$clusters <- as.factor(clusters$cluster)
fedpapers2$clusters <- as.factor(clusters$cluster)
```

Plotting the results

```
clusplot(fedpapers2, fedpapers2$clusters, color=TRUE, shade=TRUE, labels=0, lines=0)
```



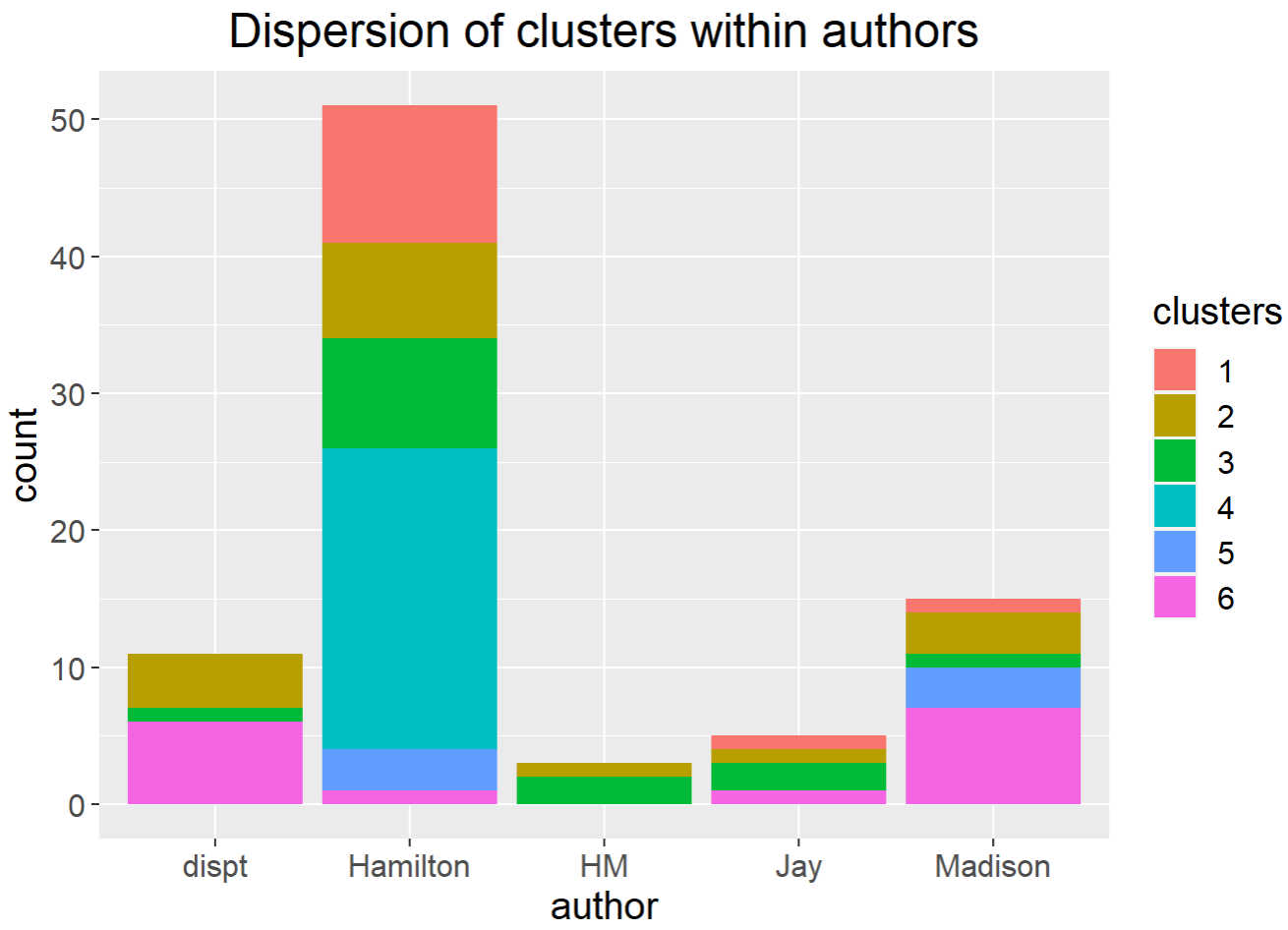

```
#str(MY_fedpapers_km)
```

Determine max distance from other points in signal words

```
colnames((fedpapers2[,1:8]))[apply((fedpapers2[,1:8]),1,which.max)]
```

```
##  [1] "all"  "from" "may"  "from" "from" "from" "may"  "may"  "from" "from"
## [11] "may"  "may"  "from" "from" "may"  "from" "from" "all"  "from" "from"
## [21] "all"  "from" "from" "may"  "from" "all"  "from" "upon" "upon" "from"
## [31] "may"  "from" "upon" "from" "may"  "from" "from" "may"  "from" "upon"
## [41] "upon" "from" "all"  "all"  "from" "from" "from" "upon" "may"  "from"
## [51] "from" "from" "from" "from" "all"  "all"  "may"  "may"  "all"  "may"
## [61] "may"  "all"  "from" "from" "from" "from" "from" "may"  "from" "may"
## [71] "may"  "may"  "from" "from" "from" "from" "may"  "may"  "may"  "all"
## [81] "may"  "may"  "from" "all"  "all"
```

```
ggplot(data=fedpapers_km, aes(x=author, fill=clusters))+
  geom_bar(stat="count")+
  labs(title="Dispersion of clusters within authors")+
  theme(plot.title = element_text(hjust=0.5),text=element_text(size=15) )
```




```
table(fedpapers_km[,1], fedpapers_km$clusters )
```

```
##
##             1  2  3  4  5  6
##   dispt     0  4  1  0  0  6
##   Hamilton 10  7  8 22  3  1
##   HM        0  1  2  0  0  0
##   Jay       1  1  2  0  0  1
##   Madison   1  3  1  0  3  7
```

Seems to me that we need to re-do this without two categories in here. Removing HM and JAY

```
my_fedpapers<- fedpapers %>%  filter(author=='Hamilton' | author == 'Madison'
                                     | author == 'dispt')
```

Doing this again

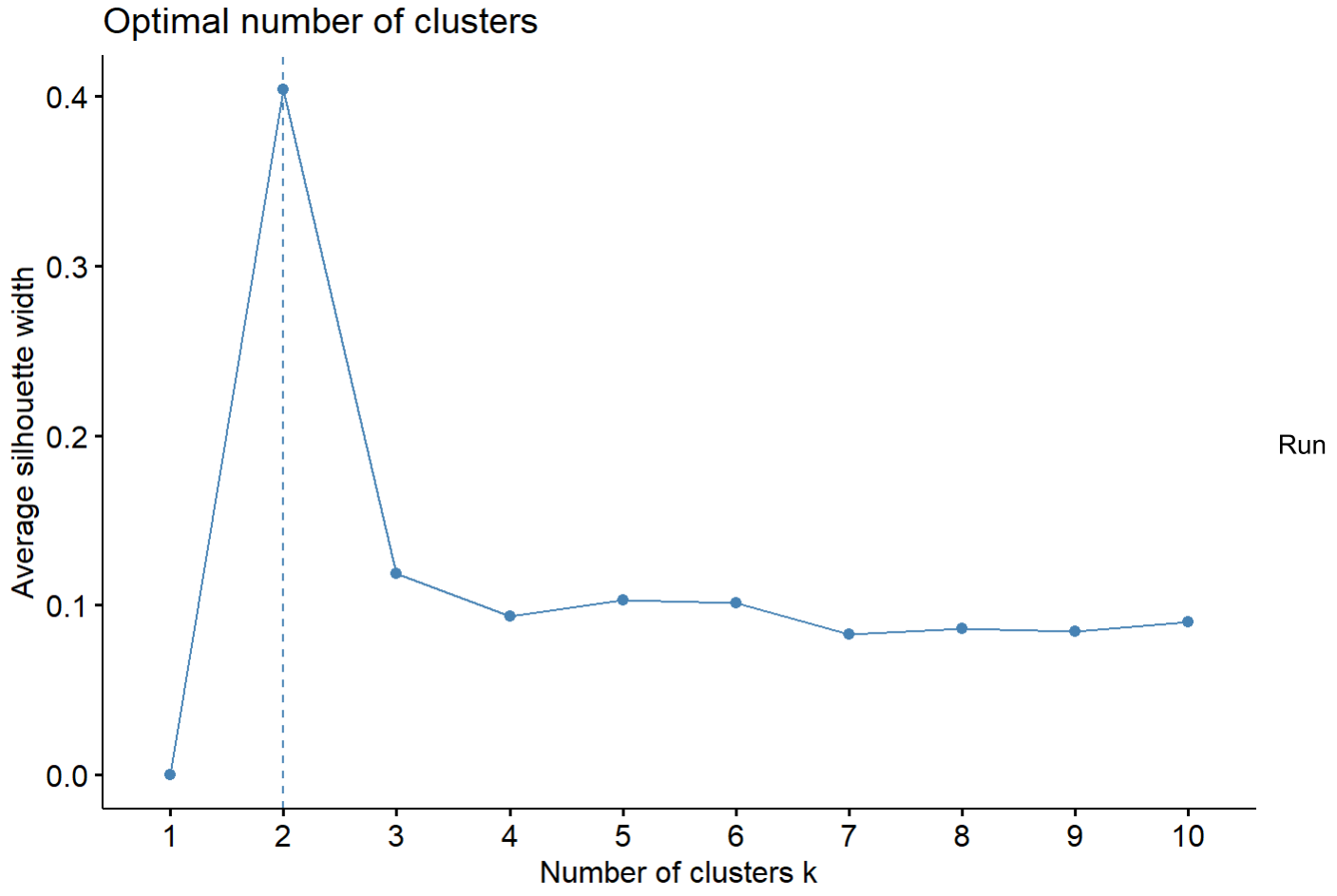
```
my_fedpapers2 <- my_fedpapers[,2:72]
my_fedpapers2 <-select(my_fedpapers2,filename, upon,all,may,also,even,from,shall,only)
rownames(my_fedpapers2)<-my_fedpapers2[,1]
my_fedpapers2[,1]<-NULL
```

The ideal clusters shouldn't have changed

```
fviz_nbclust(fedpapers, FUN = hcut, method="silhouette")
```

```
## Warning in stats::dist(x): NAs introduced by coercion
```

```
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion
```



K means for 2 clusters

```
set.seed(101)
#Running k-means
my_clusters <- kmeans(my_fedpapers2,4)
str(my_clusters)
```

```
## List of 9
##  $ cluster     : Named int [1:77] 1 1 3 3 1 3 3 3 1 3 ...
##   ..- attr(*, "names")= chr [1:77] "dispt_fed_49.txt" "dispt_fed_50.txt" "dispt_fed_51.txt"
"dispt_fed_52.txt" ...
##  $ centers     : num [1:4, 1:8] 0.01535 0.06052 0.00229 0.04 0.056 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:4] "1" "2" "3" "4"
##   .. ..$ : chr [1:8] "upon" "all" "may" "also" ...
##  $ totss       : num 0.271
##  $ withinss    : num [1:4] 0.041 0.0552 0.0303 0.0364
##  $ tot.withinss: num 0.163
##  $ betweenss   : num 0.109
##  $ size        : int [1:4] 20 25 17 15
##  $ iter        : int 4
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

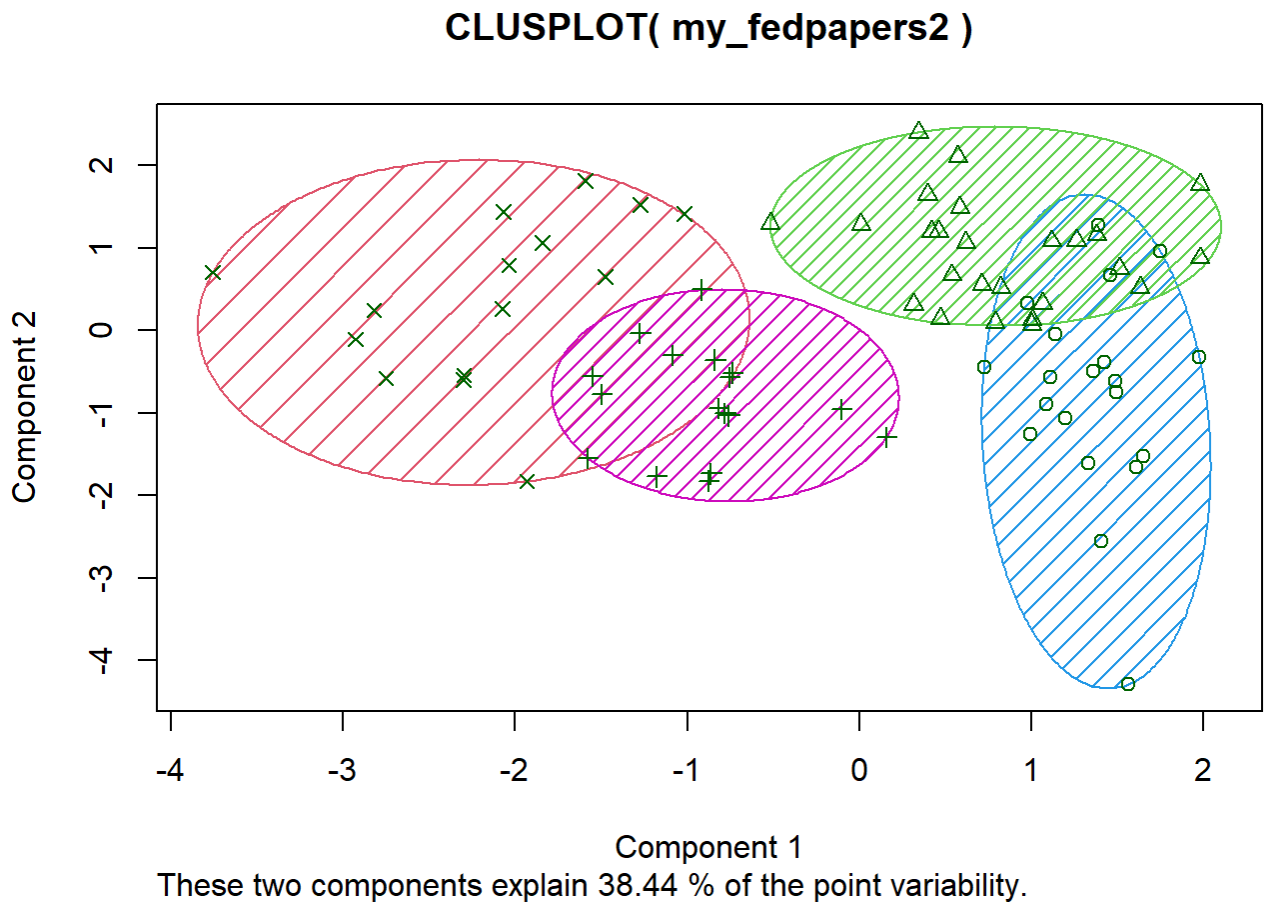
```
my_clusters$centers
```

```
##          upon     all        may        also       even       from      shall
## 1 0.015350000 0.05600 0.03585000 0.010850000 0.01165000 0.08360000 0.02225000
## 2 0.060520000 0.06528 0.05224000 0.003200000 0.01376000 0.06712000 0.01312000
## 3 0.002294118 0.05100 0.08576471 0.007941176 0.01158824 0.07241176 0.01058824
## 4 0.040000000 0.03760 0.09580000 0.006800000 0.01000000 0.09460000 0.03600000
##         only
## 1 0.01830000
## 2 0.02052000
## 3 0.02847059
## 4 0.02133333
```

```
MY_fedpapers_km <- my_fedpapers
MY_fedpapers_km$clusters <- as.factor(my_clusters$cluster)
my_fedpapers2$clusters <- as.factor(my_clusters$cluster)
```

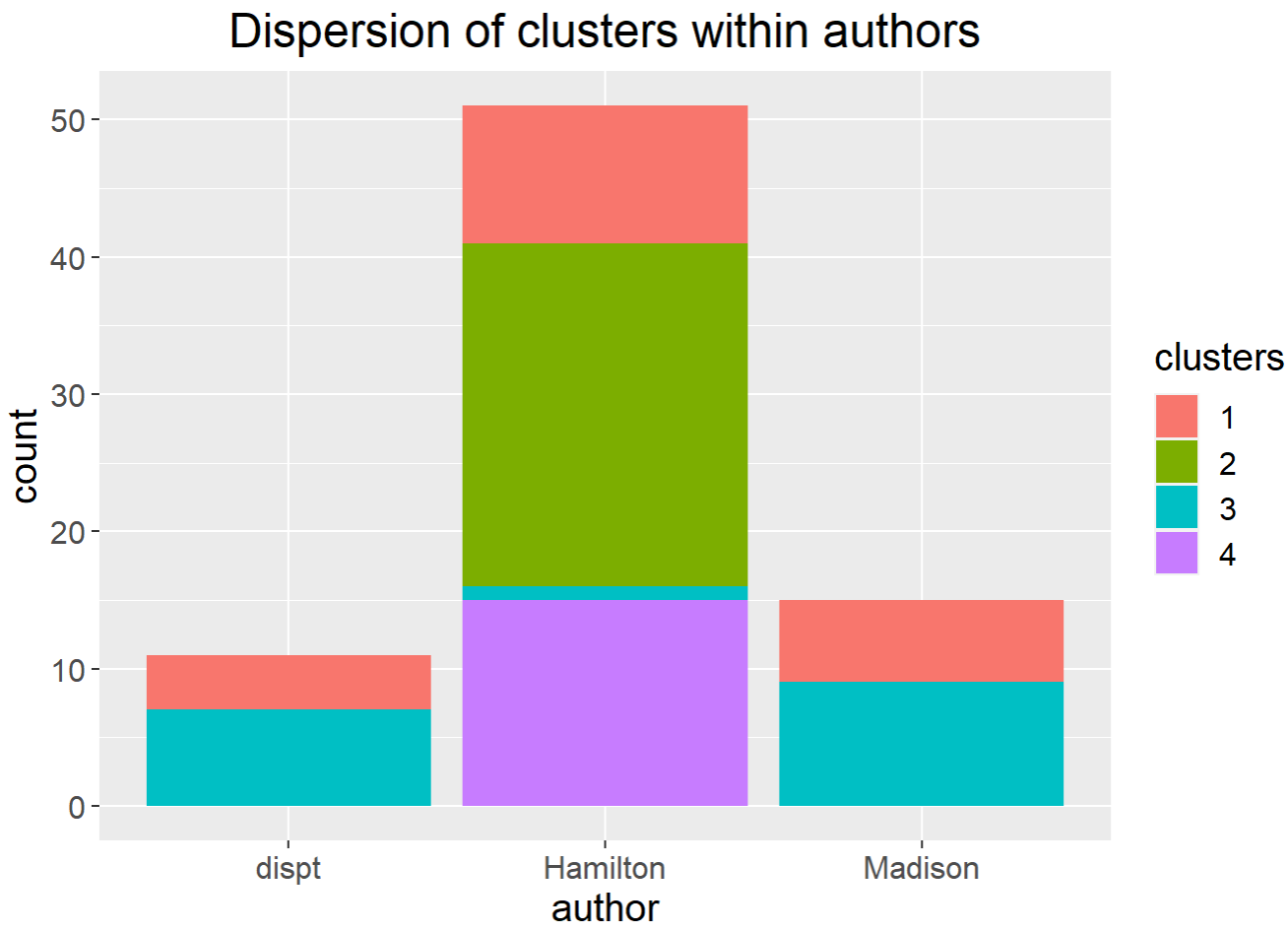
Now do the viz

```
clusplot(my_fedpapers2, my_fedpapers2$clusters, color=TRUE, shade=TRUE, labels=0, lines=0)
```



We now have clear demarcations between two groups of essays… And the winners are……..

```
ggplot(data=MY_fedpapers_km, aes(x=author, fill=clusters))+
  geom_bar(stat="count")+
  labs(title="Dispersion of clusters within authors")+
  theme(plot.title = element_text(hjust=0.5),text=element_text(size=15) )
```




```
table(MY_fedpapers_km[,1], MY_fedpapers_km$clusters )
```

```
##
##             1  2  3  4
##   dispt     4  0  7  0
##   Hamilton 10 25  1 15
##   Madison   6  0  9  0
```

Hierarchical algorithmic clustering

Establish different measures of distance to test for effect #Cleaning

```
# Remove author names
# We're going to use the dataset that I trimmed down to only include the disputed essays and the
authors they think authored them.
FedPapers_HAC <- my_fedpapers[,c(2:72)]
```

More Cleaning

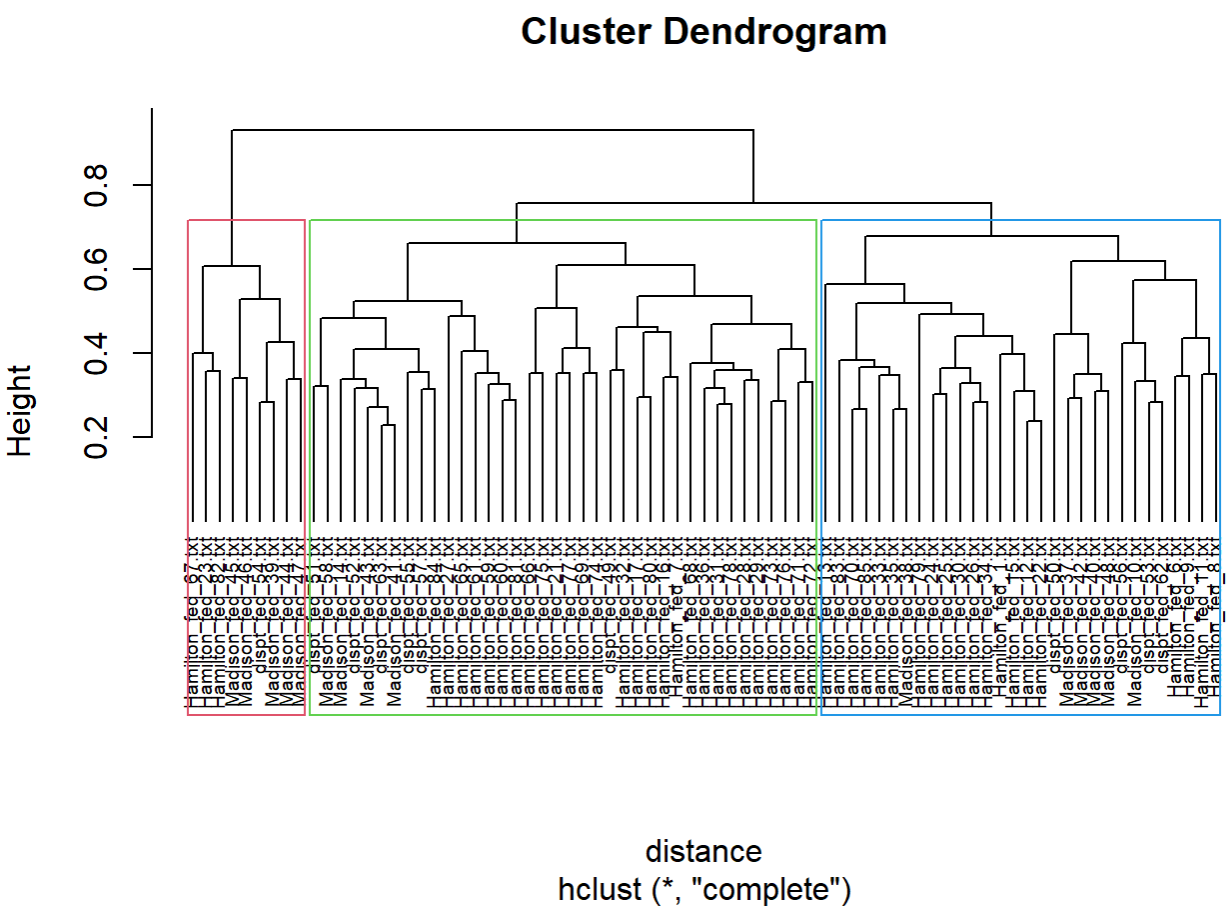
```
#The goal is to transform to numerical values for use in HAC
#Make the file names the row names so that the files are associated with results
rownames(FedPapers_HAC) <- FedPapers_HAC[,1]
FedPapers_HAC[,1] <- NULL
```

Establish measures of distance using different methods to swap out

```
# Calculate distance in a variety of ways
distance  <- 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")

#This one errored out

#distance6 <- dist(FedPapers_HAC, method = "minkowski")
```

Results

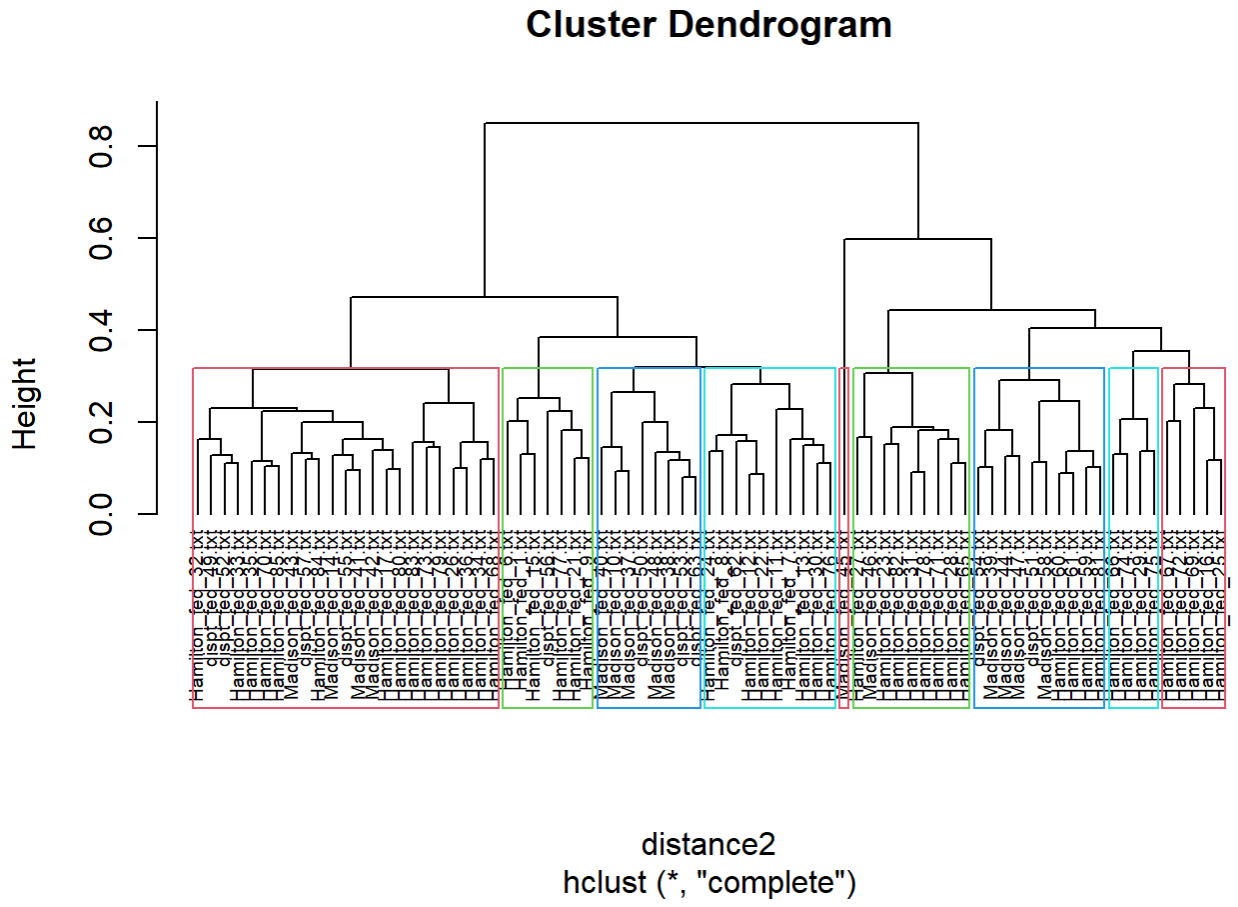
```
#Not doing anything for me.
HAC <- hclust(distance, method="complete")
plot(HAC, cex=0.6, hang=-1)
rect.hclust(HAC, k =3, border=2:5)
```



HAC

```
##
## Call:
## hclust(d = distance, method = "complete")
##
## Cluster method   : complete
## Distance         : euclidean
## Number of objects: 77
```

```
#Not doing anything for me.
HAC <- hclust(distance2, method="complete")
plot(HAC, cex=0.6, hang=-1)
rect.hclust(HAC, k =9, border=2:5)
```

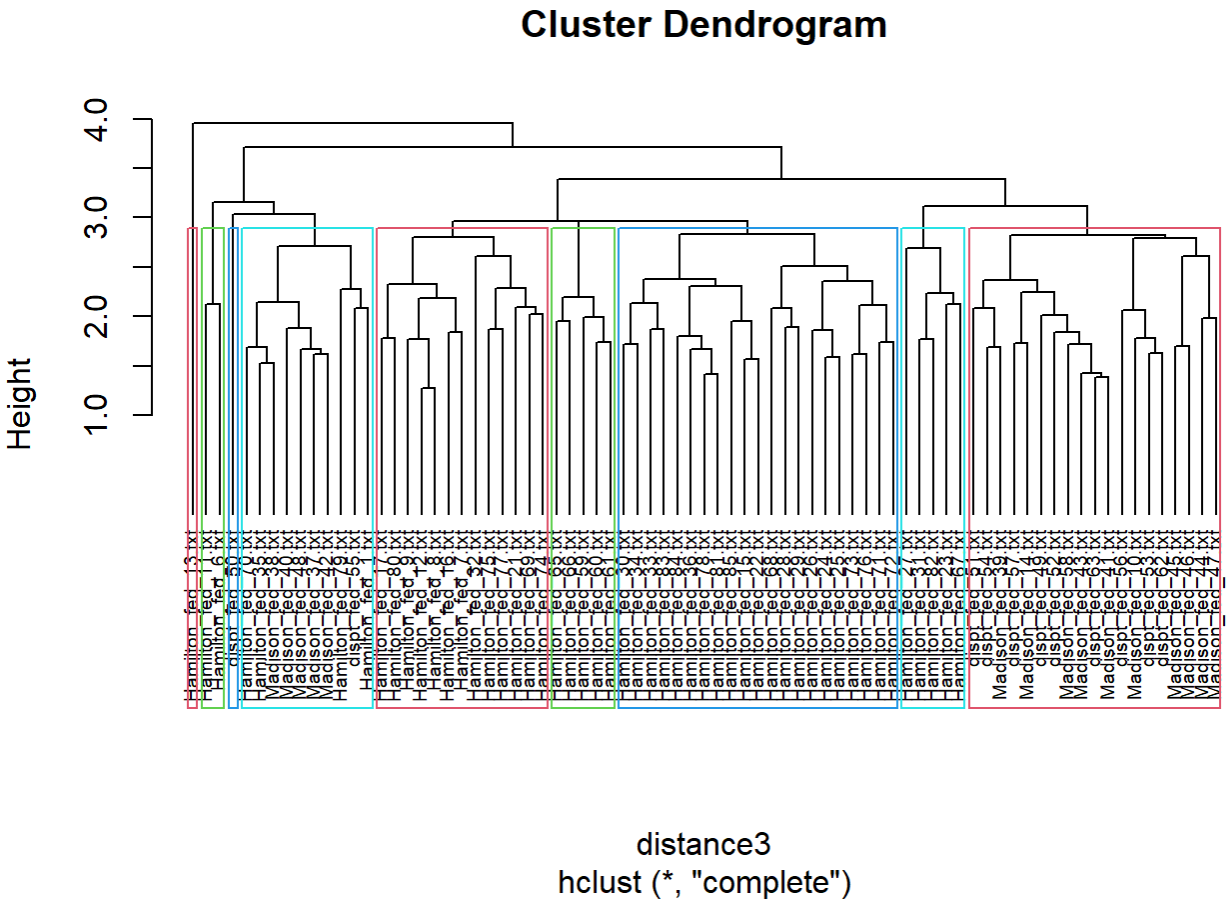


HAC

```
##
## Call:
## hclust(d = distance2, method = "complete")
##
## Cluster method   : complete
## Distance         : maximum
## Number of objects: 77
```

#This one is interesting

```
HAC <- hclust(distance3, method="complete")
plot(HAC, cex=0.6, hang=-1)
rect.hclust(HAC, k =9, border=2:5)
```

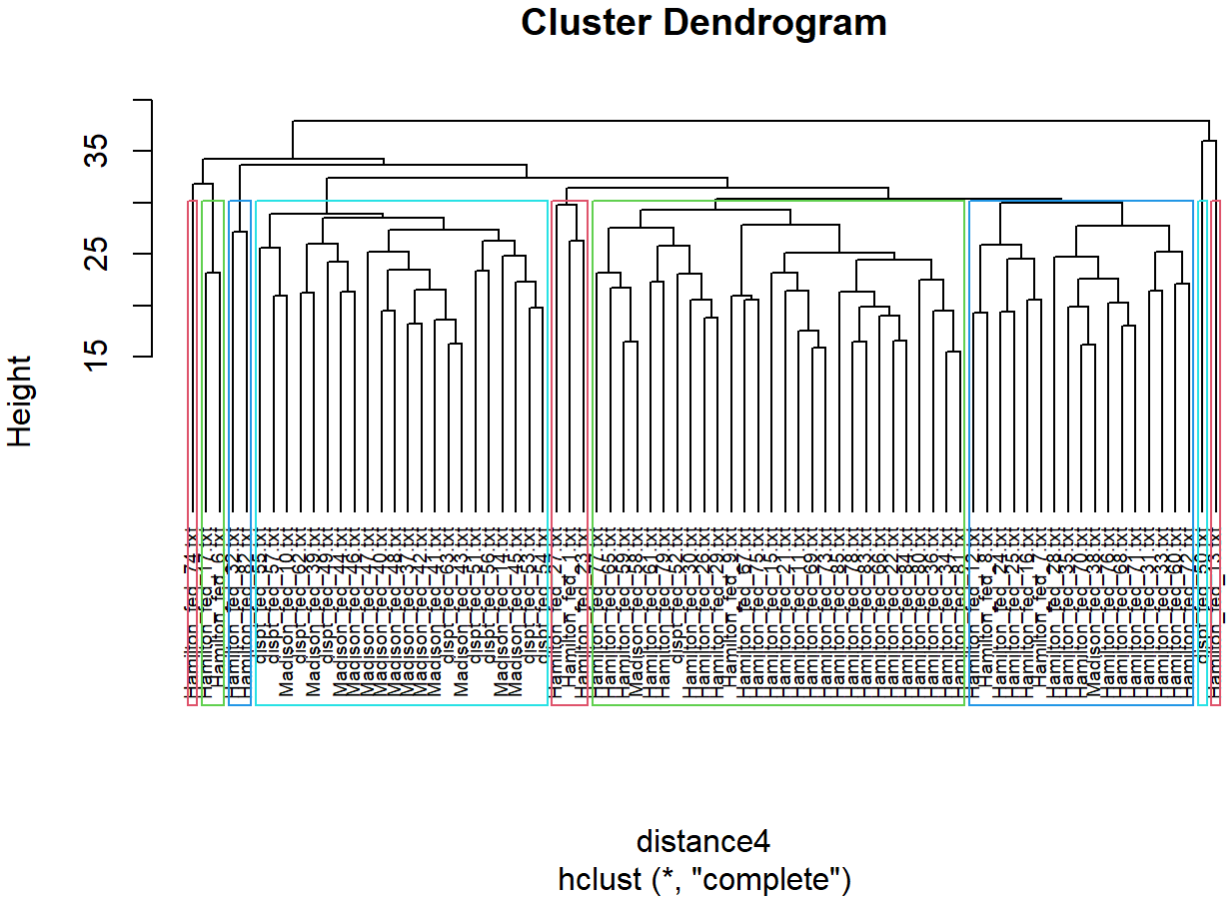



```
HAC
```

```
##
## Call:
## hclust(d = distance3, method = "complete")
##
## Cluster method   : complete
## Distance         : manhattan
## Number of objects: 77
```

#This one is really interesting

```
HAC <- hclust(distance4, method="complete")
plot(HAC, cex=0.6, hang=-1)
rect.hclust(HAC, k =9, border=2:5)
```

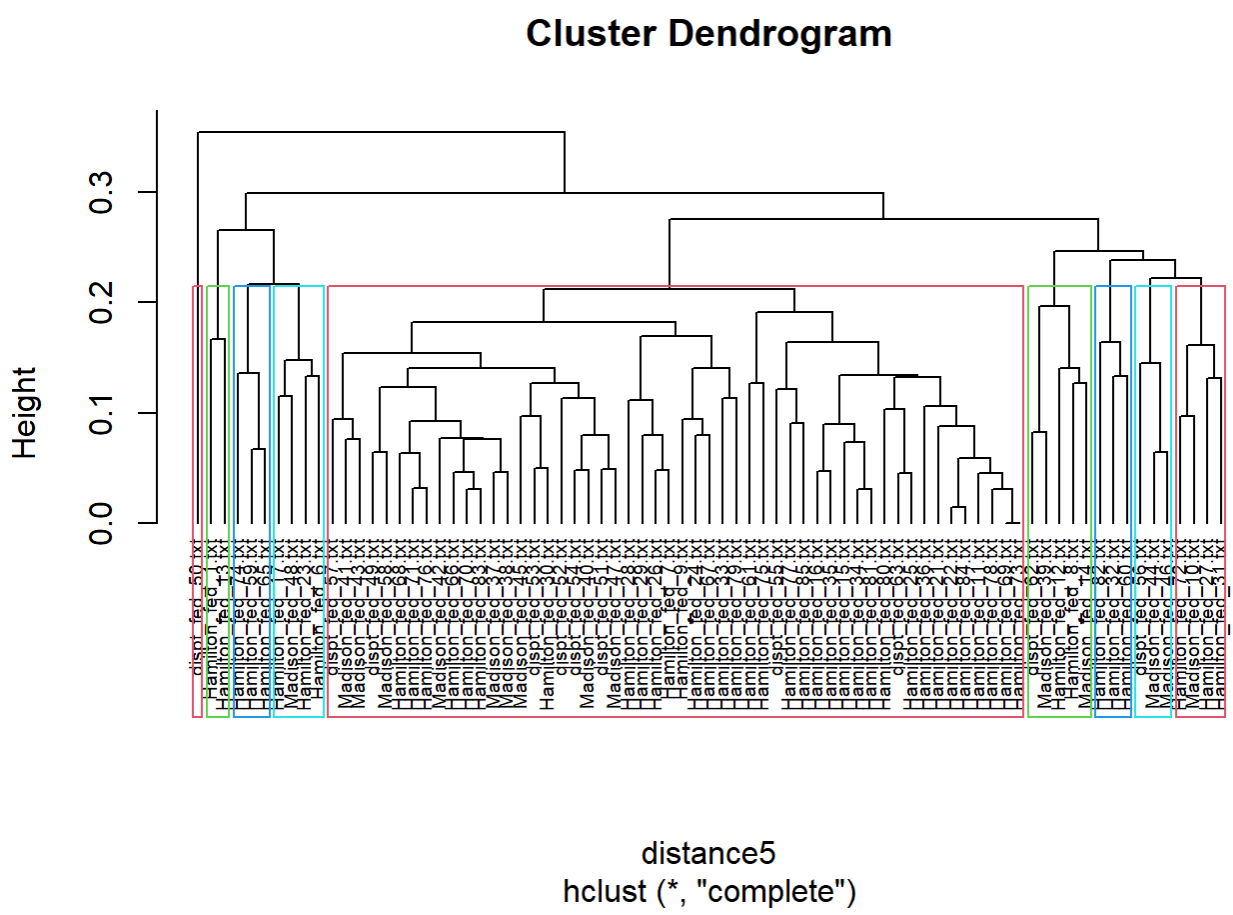



```
HAC
```

```
##
## Call:
## hclust(d = distance4, method = "complete")
##
## Cluster method   : complete
## Distance         : canberra
## Number of objects: 77
```

#This is good too

```
HAC <- hclust(distance5, method="complete")
plot(HAC, cex=0.6, hang=-1)
rect.hclust(HAC, k =9, border=2:5)
```



HAC

```
##
## Call:
## hclust(d = distance5, method = "complete")
##
## Cluster method   : complete
## Distance         : binary
## Number of objects: 77
```

Determine centroid values??? What are we talking about…