

k-means clustering and HAC

Load libraries and Set Knit

Load data

Today, we'll be working on the federalist papers data set. Papers written by Madison and Hamilton, and some papers where the author is unknown (assumed to be Madison or Hamilton). The text is “vectorized” – word counts are encoded as frequencies.

```
FederalistPapers <- read.csv("fedPapers85_fromClass.csv", na.strings = c(""))

# Create backup of FederalistPapers in case it's needed
FederalistPapers_Orig <- FederalistPapers

# Take a look at the data
View(FederalistPapers)

# Check for missing values
sum(is.na(FederalistPapers))
```

```
[1] 0
```

```
#-----
```

K means

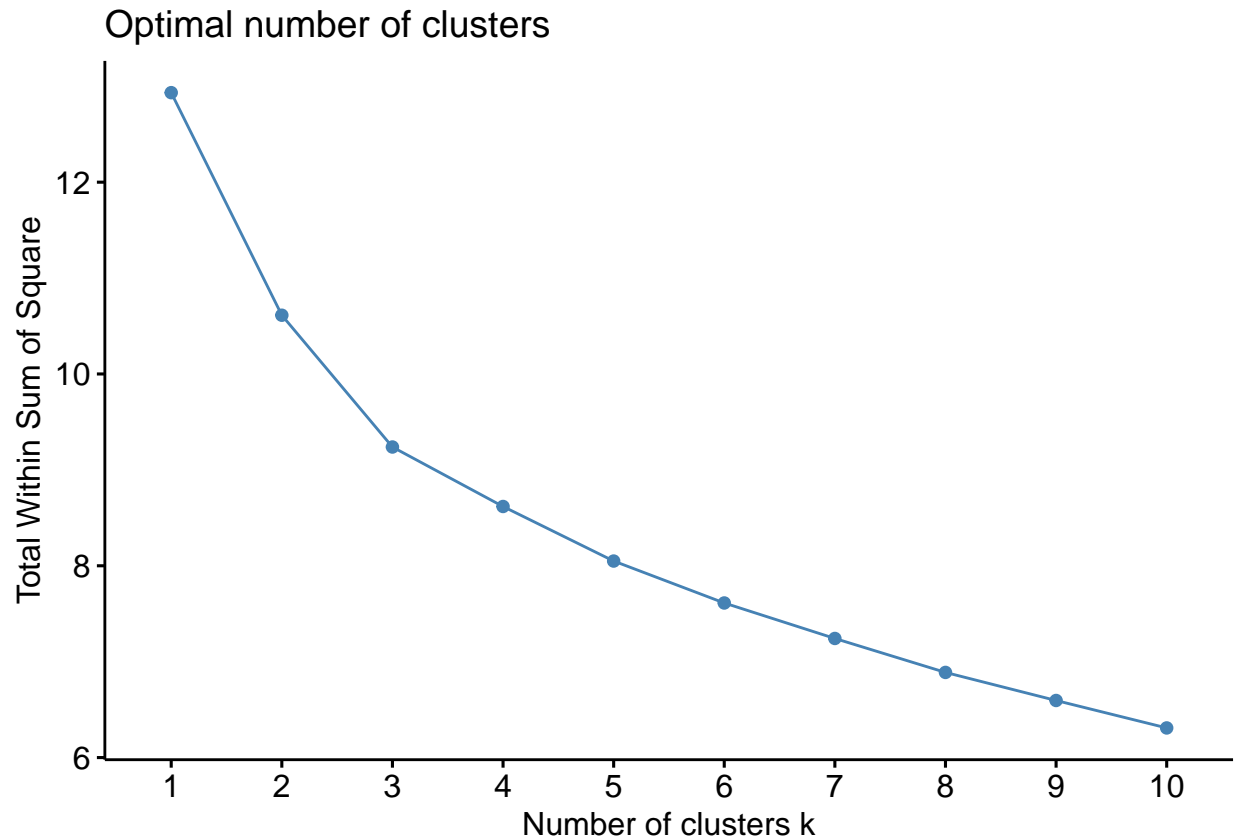
Once the data has been vectorized we can now apply analysis techniques such as clustering. First we will remove the labels and determine the “optimal” numbers of clusters for the clustering algorithm. Using total withinness we can estimate a “good” number of clusters.

```
# Remove author names from dataset for clustering purposes
FedPapers_km <- FederalistPapers[,2:72]

# Make the file names the row names. Need a dataframe of numerical values for k-means
rownames(FedPapers_km) <- FedPapers_km[,1]
FedPapers_km[,1] <- NULL

#View(FedPapers_km)

# Determine "Optimal" number of clusters
fviz_nbclust(FederalistPapers, FUN=hcut, method = "wss")
```



```
#fviz_nbclust(FederalistPapers, FUN =hcut, method = "silhouette")

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

# run k-means
Clusters <- kmeans(FedPapers_km, 4)
FedPapers_km$Clusters <- as.factor(Clusters$cluster)

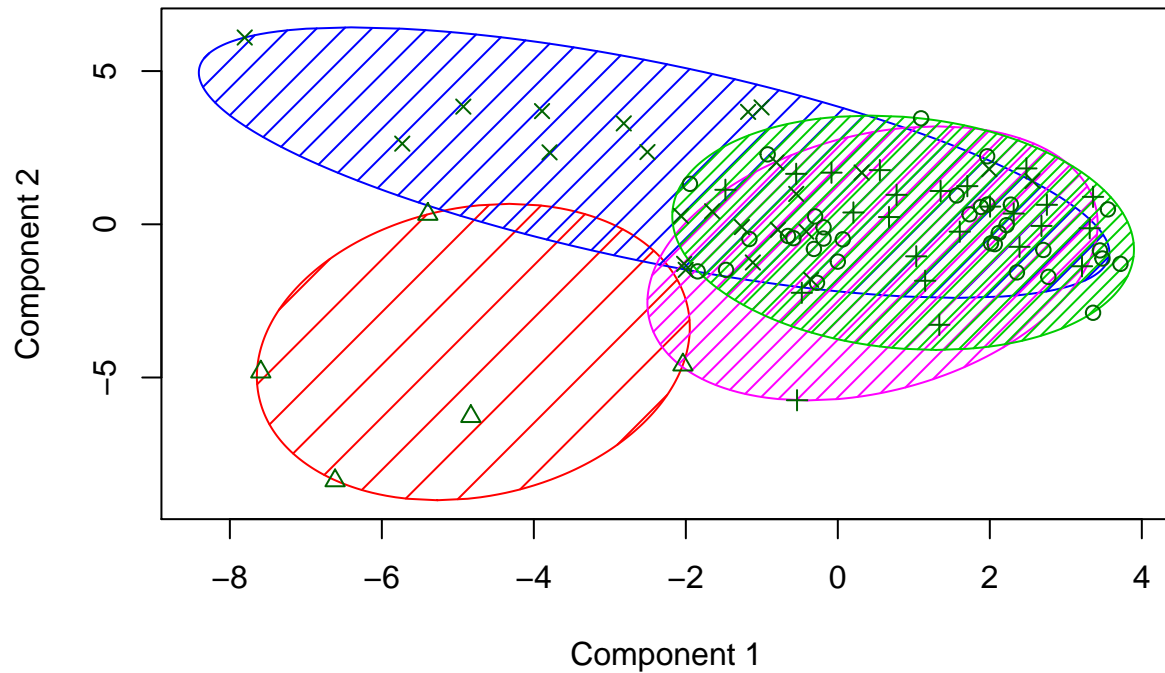
str(Clusters)
Clusters$centers
```

Next we will add the clustering results back to the dataframe and display the findings. We can attempt to identify if the clustering results intuitively group papers written by the same author into the same clusters.

```
# Add clusters to dataframe original dataframe with author name
FedPapers_km2 <- FederalistPapers
FedPapers_km2$Clusters <- as.factor(Clusters$cluster)

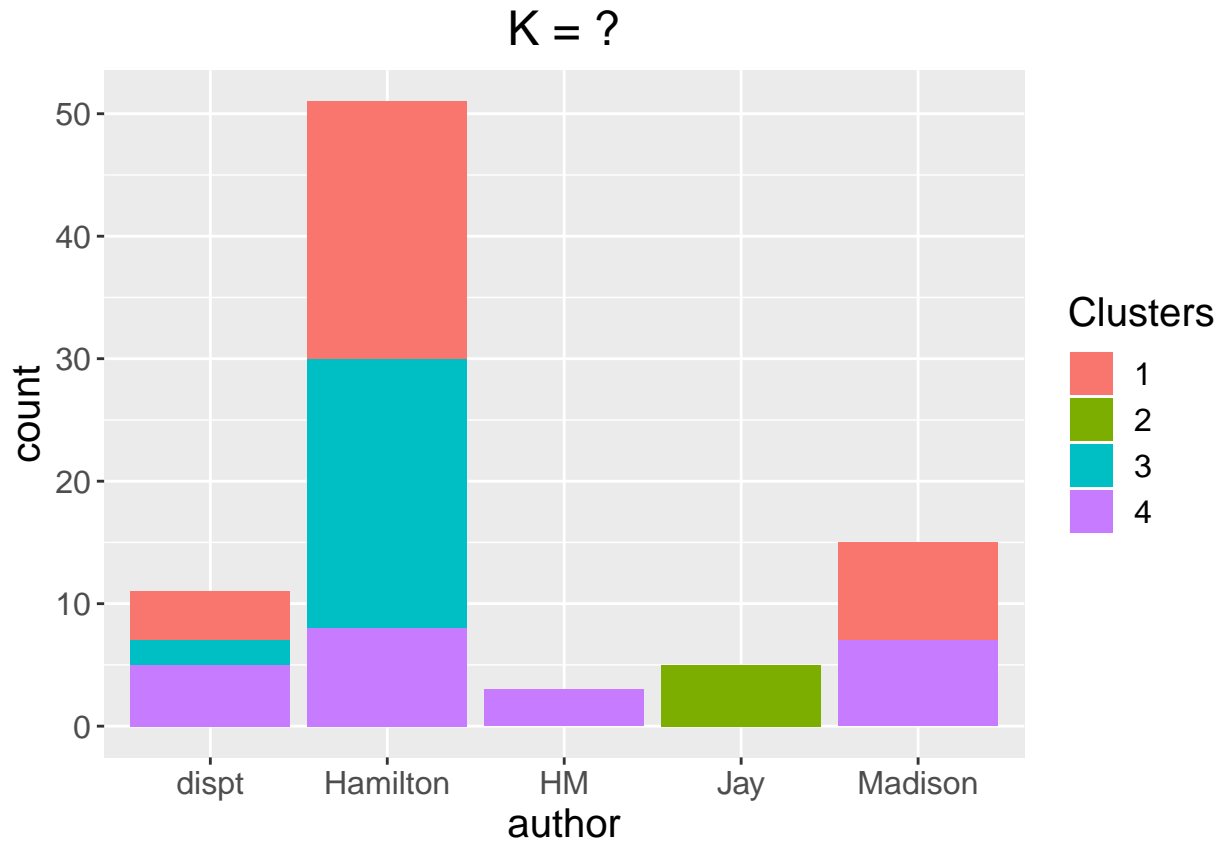
# Plot results
clusplot(FedPapers_km, FedPapers_km$Clusters, color=TRUE, shade=TRUE, labels=0, lines=0)
```

CLUSPLOT(FedPapers_km)



These two components explain 16.61 % of the point variability.

```
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))
```



#-----

Are these results good? Should we choose a different number of clusters?? Try to tweak / tune the model to improve results. What do you find??

Hierachical Clustering Algorithms (HAC)

Next we will apply hierarchical clustering methods using various distance metrics.

```
# Remove author names from dataset
FedPapers_HAC <- FederalistPapers[,c(2:72)]

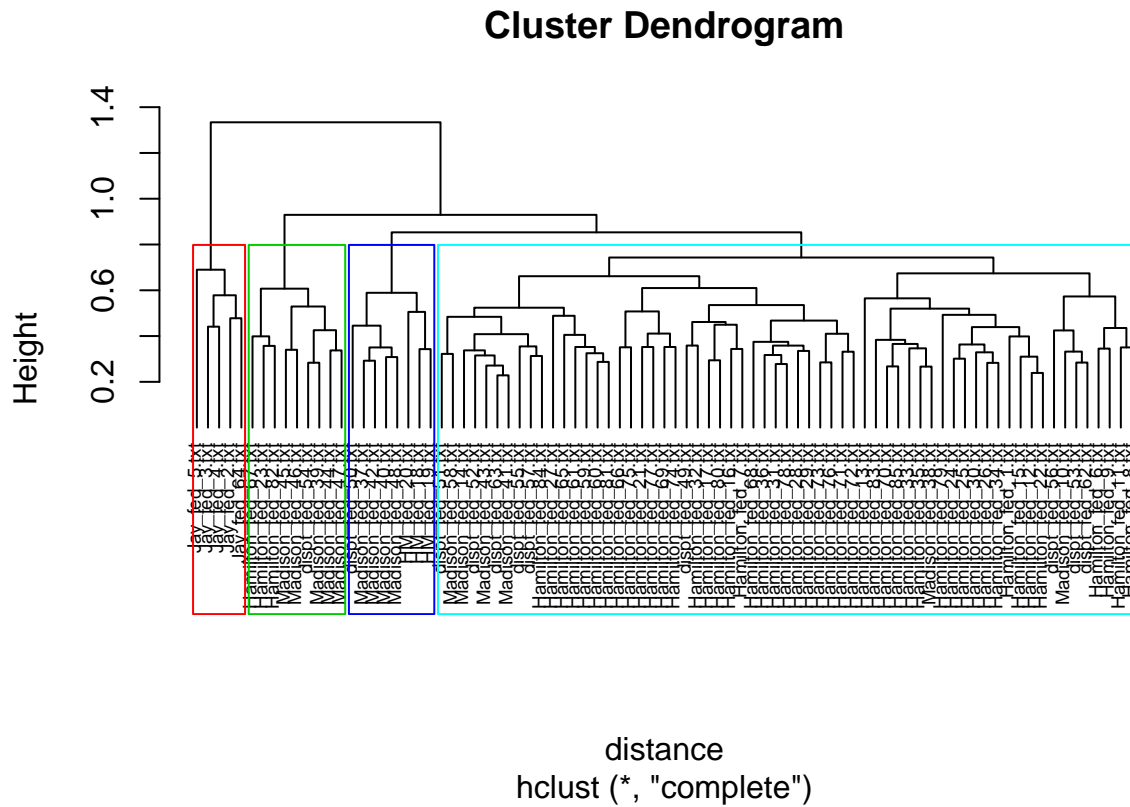
# Make the file names the row names. Need a dataframe of numerical values for HAC
rownames(FedPapers_HAC) <- FedPapers_HAC[,1]
FedPapers_HAC[,1] <- NULL

View(FedPapers_HAC)

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

Below we display the results of HAC. The boxes indicate clusters.

```
HAC <- hclust(distance, method="complete")
plot(HAC, cex=0.6, hang=-1)
rect.hclust(HAC, k =4, border=2:5)
```



```
HAC2 <- hclust(distance, method="single")
plot(HAC2, cex=0.6)
rect.hclust(HAC2, k =4, border=2:5)
```

Cluster Dendrogram

