History Mystery

## Introduction

Determining authorship of the Federalist Papers is a typical case used for teaching data mining, and as such this report intends to evaluate who is the author of the disputed Federalist Papers where authorship is not clear, analyze the Federalist Papers using the clustering algorithms k-Means, EM, and HAC.

The assignment instructions from Syracuse University Data Analytics course included the following background:

Quote from the Library of Congress <http://www.loc.gov/rr/program/bib/ourdocs/federalist.html>

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.” A bound edition of the essays was first published in 1788, but it was not until the 1818 edition published by the printer Jacob Gideon that the authors of each essay were identified by name. The Federalist Papers are considered one of the most important sources for interpreting and understanding the original intent of the Constitution.

The following background was also provided with the assignment:

These are the famous essays with disputed authorship. Hamilton wrote to claim the authorship before he was killed in a duel. Later, Madison also claimed authorship. Historians were trying to find out which one was the real author.

## Analysis and Models

### About the Data

Eighty-five (85) essays were provided for this report.

The set of 85 essay files included the following:

* 51 by Hamilton
* 15 by Madison
* 3 by Hamilton and Madison
* 5 by Jay
* 11 essays, labeled as being authored by “Hamilton or Madison,” are the disputed essays that are the goal of this report.

### Getting the Data

The files were provided in a folder and cvs (text) format. The files were uploaded to R in Corpus format to be conducive to text analysis.

### Data Cleaning

Next, the text was cleaned using the following steps:

Remove Numbers Remove Punctuations Remove <1% Words Remove Very Common Words (Stop Words)

### Cleaning Results

After cleaning the data, we can do some high-level data exploration.

First, we can confirm how many words are left to work with:

## [1] 4900

Next, we look at the first six words with the lowest frequencies:

## abhorr abject abraham abreg absenc absolv   
## 1 1 1 1 1 1

Next, we can look at the top six words with the highest frequencies:

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

### Normalization

After cleaning the data, the data was further prepared with additional normalization.

### Organization

Matrix and DF versions of the data were created in preparation for different types of analysis:

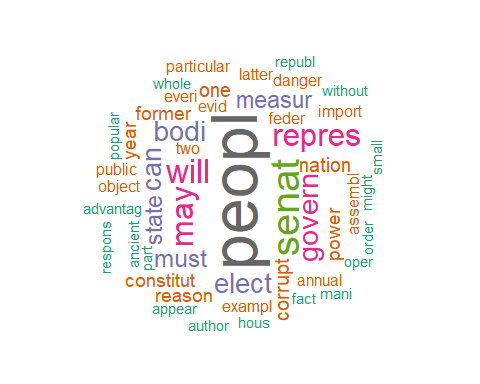
## Exploratory Data Analysis

WordClouds are an interesting word data exploration technique for text. Below are the word clouds and top 50 word counts for the following essay segments:

### Disputed Essays

WordCloud for the top fifty (50) words in the disputed essays:

## Loading required package: RColorBrewer



The top 50 words by count in the disputed essays:

## peopl senat will may repres govern bodi   
## 42 24 19 18 18 16 15   
## can elect must measur state corrupt nation   
## 14 14 12 11 11 9 9   
## one constitut former power reason year assembl   
## 9 8 8 8 8 8 7   
## exampl two annual danger everi evid feder   
## 7 7 6 6 6 6 6   
## import latter object particular public advantag ancient   
## 6 6 6 6 6 5 5   
## answer appear author charact fact first hous   
## 5 5 5 5 5 5 5   
## institut less mani member might oper order   
## 5 5 5 5 5 5 5   
## part   
## 5

### Hamilton vs. Madison

Word cloud comparison for the top fifty (50) words in the Hamilton and Madison essays:



*Madison*

*Hamilton*

## Results

Distance measurements of Manhattan, Euclidean, and Cosine were used.

Manhattan distance is named after counting number of blocks need to travel to go from point A to point B in Manhattan. It counts the distance you need to move in each dimension to get between points. It works well for binary data or data with limited number of dimensions.

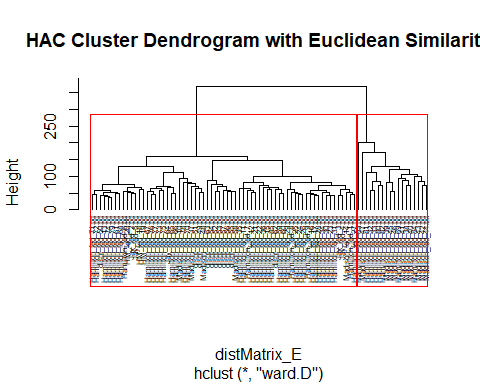
Euclidean measures the straight line between two datapoints and works well with data with a medium amount of dimensions. Between 9 and 10 dimensions issues begin to arise.

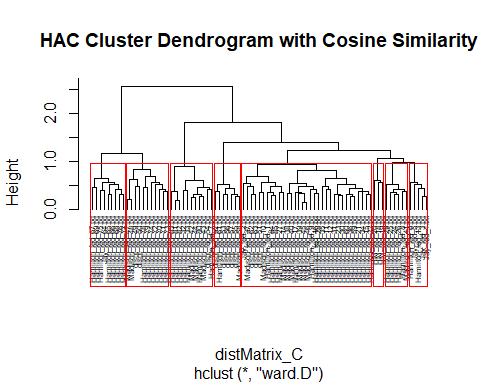
Cosine measures a little differently is measures the similarity between data points by measuring the angle between them. Cosine is well suited for data sets with many dimensions.

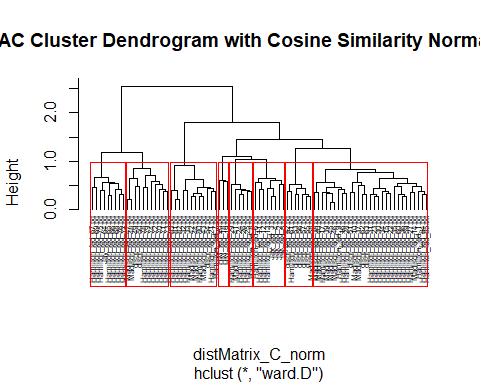
As noted in the provided code for this homework, “Cosine similarity works the best. Norm and not norm is about the same because the size of the Papers are not sig diff.”

### HAC Clustering

HAC Clustering uses a hieararchal approach.







### K-Means

K-means use an unsurprivised algorithm to group data into clusters based on distance between multiple variables.

##### K-Means Clustering with 2 Centers

## List of 9  
## $ cluster : Named int [1:85] 2 2 2 1 1 1 2 1 1 2 ...  
## ..- attr(\*, "names")= chr [1:85] "dispt\_fed\_49.txt" "dispt\_fed\_50.txt" "dispt\_fed\_51.txt" "dispt\_fed\_52.txt" ...  
## $ centers : num [1:2, 1:4900] 6.67e-05 1.09e-04 3.33e-05 1.82e-05 0.00 ...  
## ..- attr(\*, "dimnames")=List of 2  
## .. ..$ : chr [1:2] "1" "2"  
## .. ..$ : chr [1:4900] "abandon" "abat" "abb" "abet" ...  
## $ totss : num 0.216  
## $ withinss : num [1:2] 0.0794 0.1231  
## $ tot.withinss: num 0.203  
## $ betweenss : num 0.0137  
## $ size : int [1:2] 30 55  
## $ iter : int 1  
## $ ifault : int 0  
## - attr(\*, "class")= chr "kmeans"

##### K-Means Clusttering with 8 Centers

## List of 9  
## $ cluster : Named int [1:85] 2 2 2 8 8 8 8 8 8 2 ...  
## ..- attr(\*, "names")= chr [1:85] "dispt\_fed\_49.txt" "dispt\_fed\_50.txt" "dispt\_fed\_51.txt" "dispt\_fed\_52.txt" ...  
## $ centers : num [1:8, 1:4900] 0.00 5.26e-05 0.00 0.00 0.00 ...  
## ..- attr(\*, "dimnames")=List of 2  
## .. ..$ : chr [1:8] "1" "2" "3" "4" ...  
## .. ..$ : chr [1:4900] "abandon" "abat" "abb" "abet" ...  
## $ totss : num 0.216  
## $ withinss : num [1:8] 0.00201 0.03749 0.01862 0.01163 0.0068 ...  
## $ tot.withinss: num 0.158  
## $ betweenss : num 0.058  
## $ size : int [1:8] 2 19 10 6 4 25 11 8  
## $ iter : int 3  
## $ ifault : int 0  
## - attr(\*, "class")= chr "kmeans"

### K-means Visualization

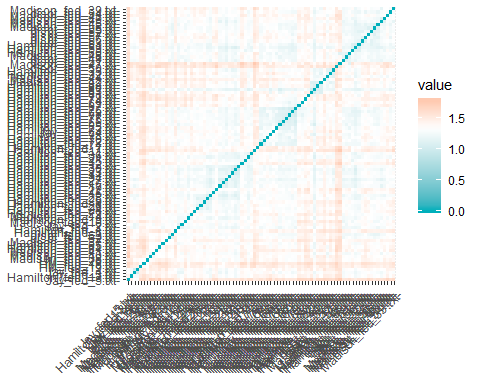
##### K-Means & Manahattan

## Loading required package: ggplot2

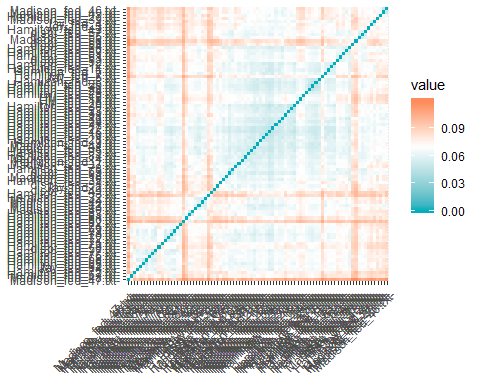
##   
## Attaching package: 'ggplot2'

## The following object is masked from 'package:NLP':  
##   
## annotate

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



##### K-Means & Euclidean



### EM (Expectation Maximization)

EM or Expectation Maximization clustering algorithm uses the following technique per wikibooks (<https://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Clustering/Expectation_Maximization_(EM)>):

’The EM algorithm is an unsupervised clustering method, that is, doesn’t require a training phase, based on mixture models. It follows an iterative approach, sub-optimal, which tries to find the parameters of the probability distribution that has the maximum likelihood of its attributes.

In general lines, the algorithm’s input are the data set (x), the total number of clusters (M), the accepted error to converge (e) and the maximum number of iterations. For each iteration, first is executed the E-Step (Expectation), that estimates the probability of each point belonging to each cluster, followed by the M-step (Maximization), that re-estimate the parameter vector of the probability distribution of each class. The algorithm finishes when the distribution parameters converges or reach the maximum number of iterations."

According to the R EMCluster package documentation (<https://cran.r-project.org/web/packages/EMCluster/EMCluster.pdf>), the title is ‘EM Algorithm for Model-Based Clustering of Finite Mixture’ and contains ‘EM algorithms and several efficient initialization methods for model-based clustering of finite mixture Gaussian distribution with unstructured dispersion in both of unsupervised and semi-supervised learning.’

## Conclusion

The HAC algorithm provided the most understandable results. The Cosine distance measurement provided the closest distance measurements.

Analyzing the results of the HAC clustering using cosine distance for 8 clusters and the higher hierarchical level of only two clusters shows the same results which should give an additional level of confidence to the analysis.

The results appear to be that Hamilton wrote more of the disputed essays based on numbers, but a granular look shows that it is likely that Madison also wrote some of the essays. Further analysis to determine the likely author at the singular level of each essay is warranted. Otherwise, one could say Hamilton likely wrote most of them, but not all of them.