HW4.R

tamtam

2021-02-13

library(tm)

## Loading required package: NLP

library(stringr)  
library(wordcloud)

## Loading required package: RColorBrewer

library(cluster)  
library(stringi)  
library(Matrix)  
library(tidytext) # convert DTM to DF  
library(plyr) ## for adply  
library(ggplot2)

##   
## Attaching package: 'ggplot2'

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

library(SnowballC)  
library(mclust) # for Mclust EM clustering

## Package 'mclust' version 5.4.7  
## Type 'citation("mclust")' for citing this R package in publications.

library(slam)  
library(factoextra)

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

library(philentropy)  
library(proxy)

##   
## Attaching package: 'proxy'

## The following object is masked from 'package:Matrix':  
##   
## as.matrix

## The following objects are masked from 'package:stats':  
##   
## as.dist, dist

## The following object is masked from 'package:base':  
##   
## as.matrix

library(sqldf)

## Loading required package: gsubfn

## Loading required package: proto

## Warning in system2("/usr/bin/otool", c("-L", shQuote(DSO)), stdout = TRUE):  
## running command ''/usr/bin/otool' -L '/usr/local/Cellar/r/4.0.3\_2/lib/R/library/  
## tcltk/libs//tcltk.so'' had status 1

## Loading required package: RSQLite

#library(readtext)  
#library(quanteda)  
# ONCE: install.packages("Snowball")  
## NOTE Snowball is not yet available for R v 3.5.x  
## So I cannot use it - yet...  
##library("Snowball")  
##set working directory  
## ONCE: install.packages("slam")  
#library(slam)  
  
## ONCE: install.packages("quanteda")  
## Note - this includes SnowballC  
  
#library(arules)  
## ONCE: install.packages("wordcloud")  
#library(wordcloud)  
##ONCE: install.packages('proxy')  
#library(proxy)  
  
#library(factoextra) # for fviz  
  
  
setwd("/Users/tamtam/Dropbox/Masters/s2 - Winter 2021/IST\_707\_Data\_Analytics/HW4")  
novelCorpus <- Corpus(DirSource("FedPapersCorpus"))  
#str(novelCorpus)  
names<- as.list(names(novelCorpus))  
fixed\_names <- str\_replace(names,"\_fed","")  
fixed\_names <- str\_replace(fixed\_names,".txt","")  
  
#meta(novelCorpus,tag="author",type="corpus") <- unlist(fixed\_names, use.names=FALSE)  
meta(novelCorpus[[1]])

## author : character(0)  
## datetimestamp: 2021-02-14 05:26:44  
## description : character(0)  
## heading : character(0)  
## id : dispt\_fed\_49.txt  
## language : en  
## origin : character(0)

meta(novelCorpus[[1]],5)

## [1] "dispt\_fed\_49.txt"

#novelsDF <- readtext("FedPapersCorpus/\*.txt",  
# docvarsfrom = "filenames",  
# dvsep = "\_",  
# encoding = "UTF-8"  
#)  
#novelCorpus <- corpus(novelsDF)  
#summary(novelCorpus)  
#str(novelCorpus)  
#(getTransformations())  
ndocs<-length(novelCorpus)  
  
  
  
summary(novelCorpus)

## Length Class Mode  
## dispt\_fed\_49.txt 2 PlainTextDocument list  
## dispt\_fed\_50.txt 2 PlainTextDocument list  
## dispt\_fed\_51.txt 2 PlainTextDocument list  
## dispt\_fed\_52.txt 2 PlainTextDocument list  
## dispt\_fed\_53.txt 2 PlainTextDocument list  
## dispt\_fed\_54.txt 2 PlainTextDocument list  
## dispt\_fed\_55.txt 2 PlainTextDocument list  
## dispt\_fed\_56.txt 2 PlainTextDocument list  
## dispt\_fed\_57.txt 2 PlainTextDocument list  
## dispt\_fed\_62.txt 2 PlainTextDocument list  
## dispt\_fed\_63.txt 2 PlainTextDocument list  
## Hamilton\_fed\_1.txt 2 PlainTextDocument list  
## Hamilton\_fed\_11.txt 2 PlainTextDocument list  
## Hamilton\_fed\_12.txt 2 PlainTextDocument list  
## Hamilton\_fed\_13.txt 2 PlainTextDocument list  
## Hamilton\_fed\_15.txt 2 PlainTextDocument list  
## Hamilton\_fed\_16.txt 2 PlainTextDocument list  
## Hamilton\_fed\_17.txt 2 PlainTextDocument list  
## Hamilton\_fed\_21.txt 2 PlainTextDocument list  
## Hamilton\_fed\_22.txt 2 PlainTextDocument list  
## Hamilton\_fed\_23.txt 2 PlainTextDocument list  
## Hamilton\_fed\_24.txt 2 PlainTextDocument list  
## Hamilton\_fed\_25.txt 2 PlainTextDocument list  
## Hamilton\_fed\_26.txt 2 PlainTextDocument list  
## Hamilton\_fed\_27.txt 2 PlainTextDocument list  
## Hamilton\_fed\_28.txt 2 PlainTextDocument list  
## Hamilton\_fed\_29.txt 2 PlainTextDocument list  
## Hamilton\_fed\_30.txt 2 PlainTextDocument list  
## Hamilton\_fed\_31.txt 2 PlainTextDocument list  
## Hamilton\_fed\_32.txt 2 PlainTextDocument list  
## Hamilton\_fed\_33.txt 2 PlainTextDocument list  
## Hamilton\_fed\_34.txt 2 PlainTextDocument list  
## Hamilton\_fed\_35.txt 2 PlainTextDocument list  
## Hamilton\_fed\_36.txt 2 PlainTextDocument list  
## Hamilton\_fed\_59.txt 2 PlainTextDocument list  
## Hamilton\_fed\_6.txt 2 PlainTextDocument list  
## Hamilton\_fed\_60.txt 2 PlainTextDocument list  
## Hamilton\_fed\_61.txt 2 PlainTextDocument list  
## Hamilton\_fed\_65.txt 2 PlainTextDocument list  
## Hamilton\_fed\_66.txt 2 PlainTextDocument list  
## Hamilton\_fed\_67.txt 2 PlainTextDocument list  
## Hamilton\_fed\_68.txt 2 PlainTextDocument list  
## Hamilton\_fed\_69.txt 2 PlainTextDocument list  
## Hamilton\_fed\_7.txt 2 PlainTextDocument list  
## Hamilton\_fed\_70.txt 2 PlainTextDocument list  
## Hamilton\_fed\_71.txt 2 PlainTextDocument list  
## Hamilton\_fed\_72.txt 2 PlainTextDocument list  
## Hamilton\_fed\_73.txt 2 PlainTextDocument list  
## Hamilton\_fed\_74.txt 2 PlainTextDocument list  
## Hamilton\_fed\_75.txt 2 PlainTextDocument list  
## Hamilton\_fed\_76.txt 2 PlainTextDocument list  
## Hamilton\_fed\_77.txt 2 PlainTextDocument list  
## Hamilton\_fed\_78.txt 2 PlainTextDocument list  
## Hamilton\_fed\_79.txt 2 PlainTextDocument list  
## Hamilton\_fed\_8.txt 2 PlainTextDocument list  
## Hamilton\_fed\_80.txt 2 PlainTextDocument list  
## Hamilton\_fed\_81.txt 2 PlainTextDocument list  
## Hamilton\_fed\_82.txt 2 PlainTextDocument list  
## Hamilton\_fed\_83.txt 2 PlainTextDocument list  
## Hamilton\_fed\_84.txt 2 PlainTextDocument list  
## Hamilton\_fed\_85.txt 2 PlainTextDocument list  
## Hamilton\_fed\_9.txt 2 PlainTextDocument list  
## HM\_fed\_18.txt 2 PlainTextDocument list  
## HM\_fed\_19.txt 2 PlainTextDocument list  
## HM\_fed\_20.txt 2 PlainTextDocument list  
## Jay\_fed\_2.txt 2 PlainTextDocument list  
## Jay\_fed\_3.txt 2 PlainTextDocument list  
## Jay\_fed\_4.txt 2 PlainTextDocument list  
## Jay\_fed\_5.txt 2 PlainTextDocument list  
## Jay\_fed\_64.txt 2 PlainTextDocument list  
## Madison\_fed\_10.txt 2 PlainTextDocument list  
## Madison\_fed\_14.txt 2 PlainTextDocument list  
## Madison\_fed\_37.txt 2 PlainTextDocument list  
## Madison\_fed\_38.txt 2 PlainTextDocument list  
## Madison\_fed\_39.txt 2 PlainTextDocument list  
## Madison\_fed\_40.txt 2 PlainTextDocument list  
## Madison\_fed\_41.txt 2 PlainTextDocument list  
## Madison\_fed\_42.txt 2 PlainTextDocument list  
## Madison\_fed\_43.txt 2 PlainTextDocument list  
## Madison\_fed\_44.txt 2 PlainTextDocument list  
## Madison\_fed\_45.txt 2 PlainTextDocument list  
## Madison\_fed\_46.txt 2 PlainTextDocument list  
## Madison\_fed\_47.txt 2 PlainTextDocument list  
## Madison\_fed\_48.txt 2 PlainTextDocument list  
## Madison\_fed\_58.txt 2 PlainTextDocument list

meta(novelCorpus[[1]])

## author : character(0)  
## datetimestamp: 2021-02-14 05:26:44  
## description : character(0)  
## heading : character(0)  
## id : dispt\_fed\_49.txt  
## language : en  
## origin : character(0)

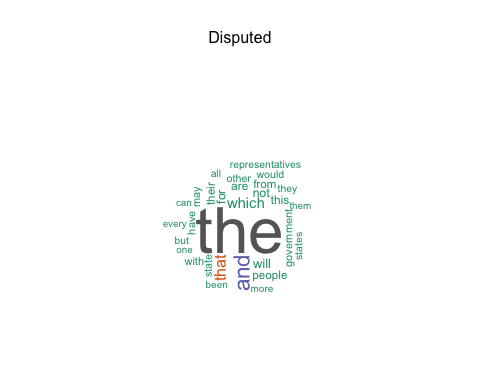
meta(novelCorpus[[1]],5)

## [1] "dispt\_fed\_49.txt"

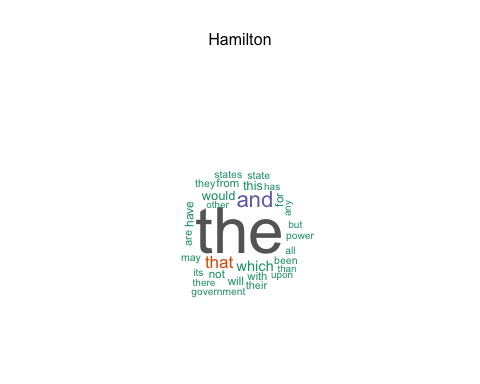
# ignore extremely rare words i.e. terms that appear in less then 1% of the documents  
minTermFreq <- ndocs \* 0.0001  
# ignore overly common words i.e. terms that appear in more than 50% of the documents  
maxTermFreq <- ndocs \* 1  
  
  
stopWords <-stopwords('english')  
dtm <- DocumentTermMatrix(novelCorpus,  
 control = list(  
 stopwords = FALSE,   
 wordLengths=c(3, 15),  
 removePunctuation = T,  
 removeNumbers = T,  
 tolower=T,  
 stemming = F,  
 remove\_separators = T  
 #stopwords = stopWords  
 #bounds = list(global = c(minTermFreq, maxTermFreq))  
 ))  
  
#Fix the document Label, so we get a nice graph label  
dtm$dimnames$Docs<-fixed\_names  
  
dtmMatrix <- as.matrix(dtm)  
dtmMatrix[1:13,1:5]

## Terms  
## Docs able above abuses accurate acquaintance  
## dispt\_49 2 1 1 1 1  
## dispt\_50 0 0 1 0 0  
## dispt\_51 1 0 1 0 0  
## dispt\_52 1 1 0 0 0  
## dispt\_53 0 0 0 1 1  
## dispt\_54 0 0 0 0 0  
## dispt\_55 0 1 0 0 0  
## dispt\_56 0 1 0 0 1  
## dispt\_57 0 1 0 0 0  
## dispt\_62 1 0 0 1 1  
## dispt\_63 4 0 2 0 0  
## Hamilton\_1 1 0 0 0 0  
## Hamilton\_11 4 0 0 0 1

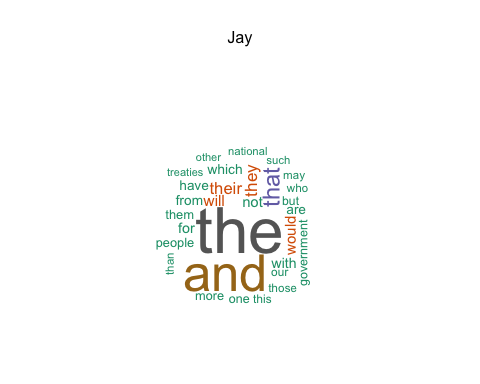
#tfidf<- weightTfIdf(dtm, normalize = TRUE)  
#print(tfidf)  
#tfidf2 <- weightTfIdf(dtm, normalize = FALSE)  
#tfidf2[1:13,1:5]  
  
normalizeDTM<-function(dtmMatrix){  
 dtmMatrix\_1 <- apply(dtmMatrix, 1, function(i) round(i/sum(i),3))  
 ## transpose  
 dtmMatrix\_norm <- t(dtmMatrix\_1)  
 return(dtmMatrix\_norm)  
}  
  
dtmMatrix\_norm<-normalizeDTM(dtm)  
  
#colors=rev(colorRampPalette(brewer.pal(9,"Blues"))(32)[seq(8,32,6)])  
wc<- function(matrix,title){  
 layout(matrix(c(1, 2), nrow=2), heights=c(1, 4))  
 par(mar=rep(0, 4))  
 plot.new()  
 text(x=0.5, y=0.5, title)  
 wordcloud(colnames(matrix), colSums(matrix), max.words = 30,colors=brewer.pal(8, "Dark2"), random.order=FALSE,rot.per=0.2,main="Title" )  
}  
#EDA  
disputedM<- subset(dtmMatrix, startsWith(rownames(dtmMatrix), 'dispt') )  
hamiltonM<- subset(dtmMatrix, startsWith(rownames(dtmMatrix), 'Hamilton') )  
jayM<- subset(dtmMatrix, startsWith(rownames(dtmMatrix), 'Jay') )  
madisonM<- subset(dtmMatrix, startsWith(rownames(dtmMatrix), 'Madison') )  
madison\_and\_Hamilton\_M<- subset(dtmMatrix, startsWith(rownames(dtmMatrix), 'HM') )  
  
wc(disputedM,"Disputed")



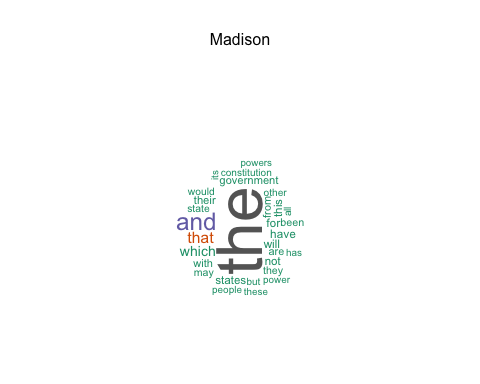
wc(hamiltonM,"Hamilton")



wc(jayM,"Jay")



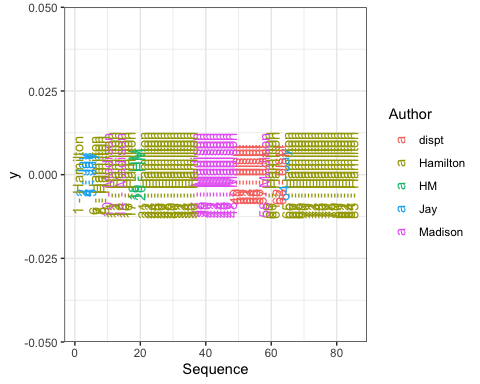
wc(madisonM,"Madison")



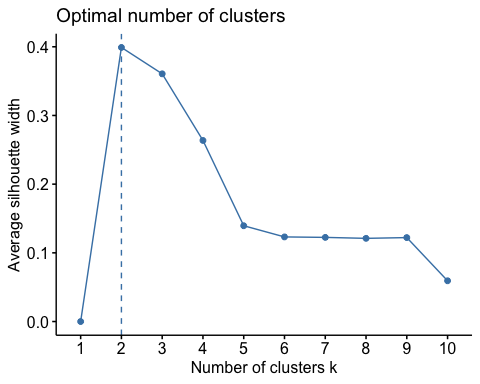
wc(madison\_and\_Hamilton\_M,"Hamilton & Madison")



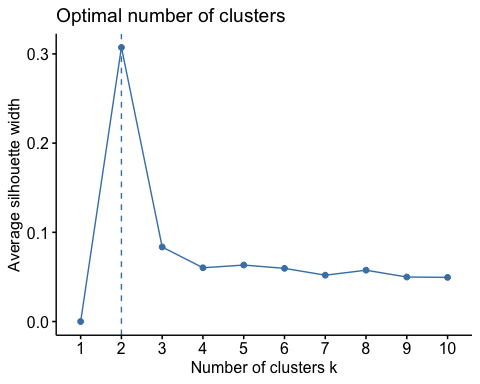
temp<-str\_split(fixed\_names,"\_",simplify = TRUE)  
  
dfSeq<-data.frame(temp)  
names(dfSeq)<-c("Author","Sequence")  
dfSeq$Sequence<- as.numeric(dfSeq$Sequence)  
dfSeq<- dfSeq[order(-dfSeq$Sequence),]  
dfSeq$label<-paste(dfSeq$Sequence,"-",dfSeq$Author)  
ggplot(dfSeq,aes(x=Sequence, label=label,y=0,color=Author))+   
 geom\_text(angle=90) + theme\_bw()



#(head(sort(as.matrix(dtm)[13,], decreasing = TRUE), n=20))  
  
set.seed(2780)  
  
  
m <- dtmMatrix  
m\_norm <- dtmMatrix\_norm  
  
  
#Determine number of clusters to divide  
fviz\_nbclust(m,FUN=hcut, method="silhouette")



fviz\_nbclust(m\_norm,FUN=hcut, method="silhouette")



#################### k means clustering -----------------------------  
X <- m\_norm  
## Remember that kmeans uses a matrix of ONLY NUMBERS  
## We have this so we are OK.  
## Manhattan gives the best vis results!  
#distance1 <- get\_dist(X,method = "manhattan")  
#fviz\_dist(distance1, gradient = list(low = "#00AFBB", mid = "white", high = "#FC4E07"))  
#distance2 <- get\_dist(X,method = "pearson")  
#fviz\_dist(distance2, gradient = list(low = "#00AFBB", mid = "white", high = "#FC4E07"))  
#distance3 <- get\_dist(X,method = "canberra")  
#fviz\_dist(distance3, gradient = list(low = "#00AFBB", mid = "white", high = "#FC4E07"))  
#distance4 <- get\_dist(X,method = "spearman")  
#fviz\_dist(distance4, gradient = list(low = "#00AFBB", mid = "white", high = "#FC4E07"))  
  
  
#X <- t(X)  
## Now scale the data  
#X <- scale(X)  
str(X)

## num [1:85, 1:8656] 0.002 0 0.001 0.001 0 0 0 0 0 0.001 ...  
## - attr(\*, "dimnames")=List of 2  
## ..$ Docs : chr [1:85] "dispt\_49" "dispt\_50" "dispt\_51" "dispt\_52" ...  
## ..$ Terms: chr [1:8656] "able" "above" "abuses" "accurate" ...

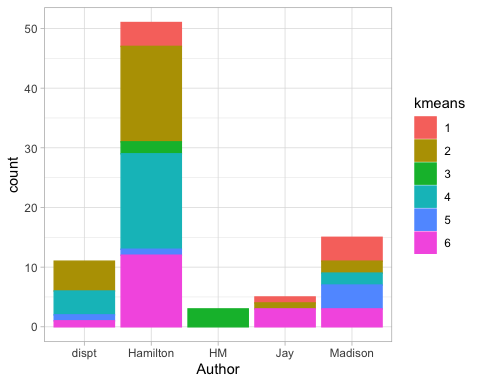
## k means  
kmeansFIT\_1 <- kmeans(X,centers=6)  
#(kmeansFIT1)  
summary(kmeansFIT\_1)

## Length Class Mode   
## cluster 85 -none- numeric  
## centers 51936 -none- numeric  
## totss 1 -none- numeric  
## withinss 6 -none- numeric  
## tot.withinss 1 -none- numeric  
## betweenss 1 -none- numeric  
## size 6 -none- numeric  
## iter 1 -none- numeric  
## ifault 1 -none- numeric

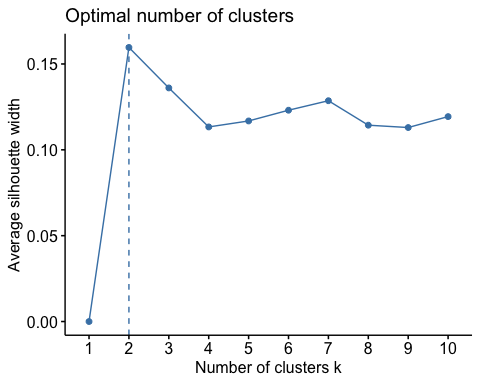
kmeansFIT\_1$cluster

## dispt\_49 dispt\_50 dispt\_51 dispt\_52 dispt\_53 dispt\_54   
## 6 1 6 4 2 4   
## dispt\_55 dispt\_56 dispt\_57 dispt\_62 dispt\_63 Hamilton\_1   
## 4 2 4 2 4 2   
## Hamilton\_11 Hamilton\_12 Hamilton\_13 Hamilton\_15 Hamilton\_16 Hamilton\_17   
## 2 2 2 2 4 6   
## Hamilton\_21 Hamilton\_22 Hamilton\_23 Hamilton\_24 Hamilton\_25 Hamilton\_26   
## 4 2 4 2 2 2   
## Hamilton\_27 Hamilton\_28 Hamilton\_29 Hamilton\_30 Hamilton\_31 Hamilton\_32   
## 4 4 6 2 4 6   
## Hamilton\_33 Hamilton\_34 Hamilton\_35 Hamilton\_36 Hamilton\_59 Hamilton\_6   
## 4 2 2 4 4 2   
## Hamilton\_60 Hamilton\_61 Hamilton\_65 Hamilton\_66 Hamilton\_67 Hamilton\_68   
## 5 5 5 5 6 1   
## Hamilton\_69 Hamilton\_7 Hamilton\_70 Hamilton\_71 Hamilton\_72 Hamilton\_73   
## 6 2 1 5 1 1   
## Hamilton\_74 Hamilton\_75 Hamilton\_76 Hamilton\_77 Hamilton\_78 Hamilton\_79   
## 6 5 1 1 4 1   
## Hamilton\_8 Hamilton\_80 Hamilton\_81 Hamilton\_82 Hamilton\_83 Hamilton\_84   
## 2 6 6 6 2 4   
## Hamilton\_85 Hamilton\_9 HM\_18 HM\_19 HM\_20 Jay\_2   
## 4 2 6 6 2 3   
## Jay\_3 Jay\_4 Jay\_5 Jay\_64 Madison\_10 Madison\_14   
## 3 3 3 3 2 4   
## Madison\_37 Madison\_38 Madison\_39 Madison\_40 Madison\_41 Madison\_42   
## 2 2 6 4 4 4   
## Madison\_43 Madison\_44 Madison\_45 Madison\_46 Madison\_47 Madison\_48   
## 6 6 6 6 6 1   
## Madison\_58   
## 4

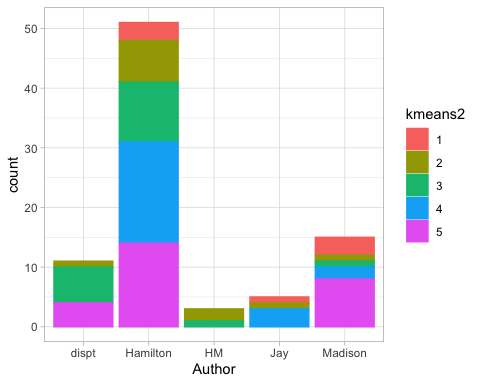
#fviz\_cluster(kmeansFIT\_1, data = X)  
#clusplot(X,as.factor(kmeansFIT\_1$cluster))  
  
dfSeq$kmeans<- as.factor(kmeansFIT\_1$cluster)  
  
ggplot(dfSeq, aes(x=Author,fill=kmeans,group=kmeans,color=kmeans))+geom\_bar(stat="count") + theme\_light()



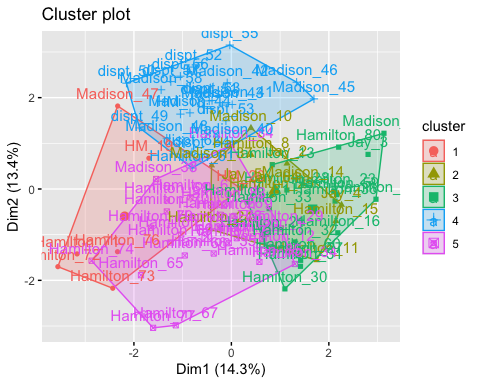
columSums<-data.frame(colSums(dtmMatrix))  
names(columSums)<-c("count")  
columSums$word<-rownames(columSums)  
columSums<-columSums[order(-columSums$count),]  
  
filteredColumns<- subset(columSums, count<400 & count>300)  
filteredColumns<- subset(filteredColumns,word!="author")  
  
#columns<-c("upon","all","may","also","even","from","shall","only")  
  
filteredDtmMatrix<- dtmMatrix[,filteredColumns$word]  
#filteredDtmMatrix<- dtmMatrix[,columns]  
  
filteredDtmMatrix\_norm<-normalizeDTM(filteredDtmMatrix)  
  
#filteredDtmMatrix\_norm<- sqldf("select \* from filteredDtmMatrix\_norm where row\_names not like 'HM%'",row.names=TRUE)  
#head(sort(columSums,decreasing=TRUE))  
#ggplot(columSums, aes(x=word,y=count))+geom\_point()+ theme\_light()  
  
  
X<- filteredDtmMatrix\_norm  
fviz\_nbclust(X,FUN=hcut, method="silhouette")



kmeansFIT\_2 <- kmeans(X,centers=5)  
dfSeq$kmeans2<- as.factor(kmeansFIT\_2$cluster)  
ggplot(dfSeq, aes(x=Author,fill=kmeans2,group=kmeans2,color=kmeans2))+geom\_bar(stat="count") + theme\_light()



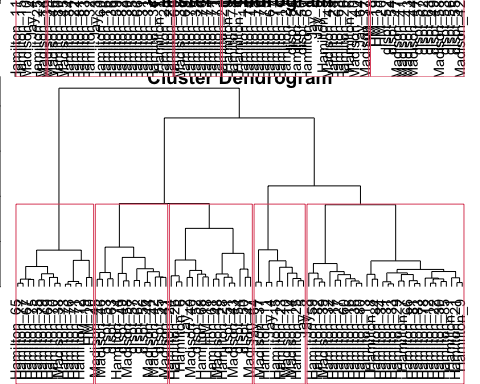
#clusplot(X,as.factor(kmeansFIT\_2$cluster))  
fviz\_cluster(kmeansFIT\_2, data = X)



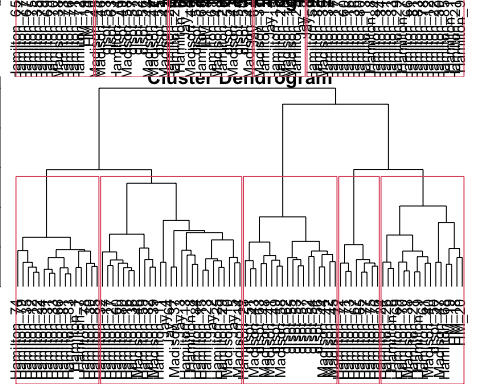
############## Distance Measures ######################  
getDistMethods()

## [1] "euclidean" "manhattan" "minkowski"   
## [4] "chebyshev" "sorensen" "gower"   
## [7] "soergel" "kulczynski\_d" "canberra"   
## [10] "lorentzian" "intersection" "non-intersection"   
## [13] "wavehedges" "czekanowski" "motyka"   
## [16] "kulczynski\_s" "tanimoto" "ruzicka"   
## [19] "inner\_product" "harmonic\_mean" "cosine"   
## [22] "hassebrook" "jaccard" "dice"   
## [25] "fidelity" "bhattacharyya" "hellinger"   
## [28] "matusita" "squared\_chord" "squared\_euclidean"  
## [31] "pearson" "neyman" "squared\_chi"   
## [34] "prob\_symm" "divergence" "clark"   
## [37] "additive\_symm" "kullback-leibler" "jeffreys"   
## [40] "k\_divergence" "topsoe" "jensen-shannon"   
## [43] "jensen\_difference" "taneja" "kumar-johnson"   
## [46] "avg"

# # # m <- m[1:2, 1:3]  
distMatrix\_E <- dist(filteredDtmMatrix, method="euclidean")  
#print(distMatrix\_E)  
distMatrix\_C <- dist(filteredDtmMatrix, method="cosine")  
#print(distMatrix\_C)  
distMatrix\_C\_norm <- dist(filteredDtmMatrix\_norm, method="cosine")  
#print(distMatrix\_C\_norm)  
  
distMatrix\_M\_norm <- dist(filteredDtmMatrix\_norm, method="manhattan")  
#print(distMatrix\_M\_norm)  
  
############# Clustering #############################  
## Hierarchical  
  
## Euclidean  
groups\_E <- hclust(distMatrix\_E,method="ward.D")  
plot(groups\_E, cex=0.9, hang=-1)  
rect.hclust(groups\_E, k=5)  
  
## Cosine Similarity  
groups\_C <- hclust(distMatrix\_C,method="ward.D")  
plot(groups\_C, cex=0.9, hang=-1)  
rect.hclust(groups\_C, k=5)



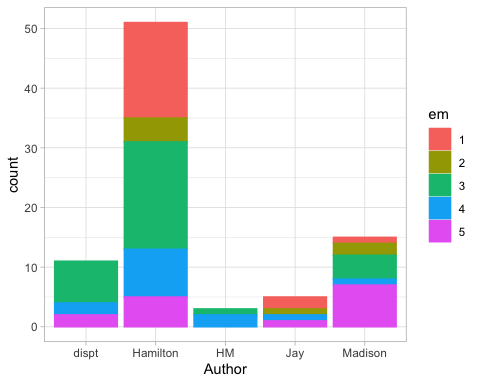
## Cosine Similarity for Normalized Matrix  
groups\_C\_n <- hclust(distMatrix\_C\_norm,method="ward.D")  
plot(groups\_C\_n, cex=0.9, hang=-1)  
rect.hclust(groups\_C\_n, k=5)  
  
  
## Manhattan for Normalized Matrix  
groups\_M\_n <- hclust(distMatrix\_M\_norm,method="ward.D")  
plot(groups\_M\_n, cex=0.9, hang=-1)  
rect.hclust(groups\_M\_n, k=5)



################# Expectation Maximization ---------  
  
#X<-m  
#X <- t(X)  
ClusFI <- Mclust(distMatrix\_M\_norm,G=5)  
summary(ClusFI)

## ----------------------------------------------------   
## Gaussian finite mixture model fitted by EM algorithm   
## ----------------------------------------------------   
##   
## Mclust EII (spherical, equal volume) model with 5 components:   
##   
## log-likelihood n df BIC ICL  
## 3197.849 85 430 4485.358 4485.357  
##   
## Clustering table:  
## 1 2 3 4 5   
## 19 7 30 14 15

dfSeq$em<-as.factor(ClusFI$classification)  
ggplot(dfSeq, aes(x=Author,fill=em,group=em,color=em))+geom\_bar(stat="count") + theme\_light()



#plot(ClusFI, what = "classification")